De-Centralized Information Flow Control for Cloud Virtual Machines with Blowfish Encryption Algorithm

Yogesh B.Gurav[1] and Dr.Bankat M.Patil[2]
ybgurav1977@gmail.com patilbankat@gmail.com
Department of Computer Science and IT
Dr.Babasaheb Ambedkar Marathwada University
Aurangabad (MS), India

Summary

Today, the cloud computing has become a major demand of many organizations. The major reason behind this expansion is due to its cloud's sharing infrastructure with higher computing efficiency, lower cost and higher fle3xibility. But, still the security is being a hurdle that blocks the success of the cloud computing platform. Therefore, a novel Multi-tenant Decentralized Information Flow Control (MT-DIFC) model is introduced in this research work. The proposed system will encapsulate four types of entities: (1) The central authority (CA), (2) The encryption proxy (EP), (3) Cloud server CS and (4) Multi-tenant Cloud virtual machines. Our contribution resides within the encryption proxy (EP). Initially, the trust level of all the users within each of the cloud is computed using the proposed two-stage trust computational model, wherein the user is categorized bas primary and secondary users. The primary and secondary users vary based on the application and data owner's preference. Based on the computed trust level, the access privilege is provided to the cloud users. In EP, the cipher text information flow security strategy is implemented using the blowfish encryption model. For the data encryption as well as decryption, the key generation is the crucial as well as the challenging part. In this research work, a new optimal key generation is carried out within the blowfish encryption Algorithm. In the blowfish encryption Algorithm, both the data encryption as well as decryption is accomplishment using the newly proposed optimal key. The proposed optimal key has been selected using a new Self Improved Cat and Mouse Based Optimizer (SI-CMBO), which has been an advanced version of the standard Cat and Mouse Based Optimizer. The proposed model is validated in terms of encryption time, decryption time, KPA attacks as well.

Keywords[*Cloud Computing; Decentralized Information Flow Control; Multi-Tenant Architecture; Blowfish Algorithm;SI-CMBO.*

I. Introduction

Computer systems' primary aim is to protect the user's data. Nevertheless the large number of researchers who've been involved in safeguarding users' information have developed huge count of technologies [1] [2]. But, none of the approaches weren't satisfactory. Data sharing plays a key part in the cloud system, where users from many places and devices connect to the cloud, and security is a big concern. The majority of security breaches occur as a result of untrustworthy software that allows unauthorized access to consumers' sensitive data. There are currently no reliable platforms that allow programmers to establish secured strategies at the chosen abstraction and that enable end-to-end policy enforcement [1] [2] [3]. Effective security policy as well as the IFC both necessitate it immediately. In reality, IFC assures three key elements: database confidentiality, integrity, and availability. The DIFC approach has notably attracted considerable attention because of its potential to secure user information through connecting labels with information as well as restricting flow of information depending on these labels. DIFC rules control the flow of data between application components [4] [5] [6]. Users in the IFC can access and change the data of other users, but in the DIFC, powerful controls allow users to read files but will not broadcast information over an unprotected network channel [7] [8] [9] [10] [11]. On either a dynamically or statically implemented paradigm, the DIFC may enable end-to-end, user-defined security rules. It moreover allows users to follow the flow of information across the system [12][13][14] [15][16]. The flow of information offers clear-cut standards for lawful data dissemination as well as localization of cybersecurity policy decisions, which is one of DIFC's main advantages. The conventional DIFC approach, on the other hand, cannot oversee data kept outside the system because untrustworthy cloud storage providers or adversaries just outside of the system might expose the customer's data [17][18][19][20] [21] [22]. As a result, with in DIFC, the possibility to protect information inside an untrustworthy cloud environment should always be strengthened by combining particular encryption technologies [23][24][25]. Recent cryptography techniques, on the other hand, have a strong reliance on customers; as a response, unauthorized attacker or users manipulated by cybercriminals outside of the system potentially expose their secrets voluntarily.

The major contribution of this research work is

- Introduces a new user trust based multi-tenant DIFC model.
- Introduces a two-stage trust computational model that categorizes the users are primary and secondary users.

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.12.35

Based on this computed trust level, the access privilege has been provided to the cloud users by the data owners.

- In data encryption as well as decryption, the key generation is carried out in blowfish algorithm using the newly proposed Self Improved Cat and Mouse Based Optimizer (SI-CMBO).

The rest of this paper is organized as: Section II discusses the literature works accomplished in IFC. Section III tells about proposed decentralized cipher text information flow control framework based on multi-tenant sensitive information flow. The results acquired with the proposed work are manifested in Section IV. This paper is concluded in Section V.

II. Literature review

A. Related works

In 2019, Khurshid *et al.* [2] have proposed a unique technique termed Secure-CamFlow, with the goal of protecting customers' data during the process of migration (i.e. data transfer from devices to cloud). The recommended research used the Information Flow Strategy provided by the customer to supervise the in-cloud data flow. In addition, the researchers have described a model of the suggested work which has been tested in regards to energy usage as well as computational efficiency.

In 2019, Xi *et al.* [3] to examine the flow of information containing encrypted information, researchers initiated a unique approach known as encryption flow. The inter-relationships amongst encrypted information components and ancillary services have been demonstrated throughout this research design. Additionally, the researchers have developed a secure communication stream confirmation theorem depending on the encrypted flow to enhance the security requirements at all network elements. The recommended study clearly offered more effective centralized validation based on gathered outcomes.

In 2018, Bhushan *et al.* [4] have developed a mathematical model towards avoiding or minimizing FRC attacks on public cloud platforms. The recommended research towards identifying and mitigating FRC assaults has been reflected in the increasing cloud platform. The suggested work's effectiveness has been verified employing real-world benchmark functions, with the outcomes revealing that the approach provided improved consistency as well as reduced latency throughout the vulnerability intrusion detection and elimination process.

In 2020, Moussaid *et al.* [5] have investigated the behavior of entities in attempt to improve the security mechanisms of information dissemination throughout the cloud. In particular, the confidence degree as well as security classification were investigated, and a link between the 2 was discovered. Consequently, to assure the protection policy regarding of the CIA, a protection program was formulated. The suggested research does indeed have a high detection accuracy.

In 2017, Reddy *et al.* [6] have implemented the DIFC method to address SaaS security concerns. The suggested improvement has ensures the effectiveness and consistency of user information as it travels through the system. By appropriately managing the labels, confidentiality has indeed been maintained. As a consequence, the DIFC in SaaS cloud security has been shown to comply with the cloud providers' regulations and laws.

III. Proposed decentralized ciphertext information flowcontrol framework based onMulti-tenant sensitive information flow

A. Objective Function

The goal of this research development is to improve the key break time $T_{key\,break}$. The objective function is expressed numerically as per Eq. (1).

$$Obj = Min\left(\frac{1}{T_{key\,break}}\right) \tag{1}$$

With the intention of achieving the above specified objective, the keys for encryption as well as decryption utilized in blowfish algorithm are generated optimally using the SI-CMO.

B. Systewm Model

Based on the multi-tenant sensitive information flow, this research study develops a unique decentralized cipher text information flow control system. Three key components make up the proposed cloud computing environment: (a) CSP, (b) DS, and (c) User. Our contribution is included in CSP, which comprises the following components: (1) the CA, (2) the encryption proxy (EP), (3) the Cloud server CS, and (4) the Cloud tenant virtual machines. The CA, as the key management centre as well as label administration unit, is in charge of overseeing the labels and disseminating the constructed keys. Furthermore, the EP within that virtual machine hypervisor is now in charge of executing security measures as well as performing storage encryption proxy procedures for authorized cloud tenants. The Cloud server is indeed a cloud storage server with such a substantial quantity of computation and storage potential. All virtual machines throughout the Cloud tenancy seamlessly access data (i.e. read, update, and modify) through communicating with CS through the use of the EP. The blowfish algorithm is being used to implement security procedures in this EP. Every tenant submits an application towards CA for the formation of a new secret-domain label and also the ability to give trustworthy tenants to virtual machines. Because the CA and EP are both regarded to just be legitimate in this study, the protection labels and generated keys would be dutifully preserved by the CA for any and all multi-tenants. The term "multi-tenant architecture" refers to a design model in which multiple users share the same software or system components. While exchanging physical resources in a multi-tenant design, tenants should preserve mutual information separation. Therefore, a two-level trust model has been introduced in this research work. In particular, the data files are stored in metadata in the Data Store containing their applicable authenticity as well as privacy tags. The consumers are those who use the cloud. Patients, doctors and specialists, scholars, chemists, family members and friends of patient populations, and agents of healthcare companies are all examples of cloud users in the case of medical databases.

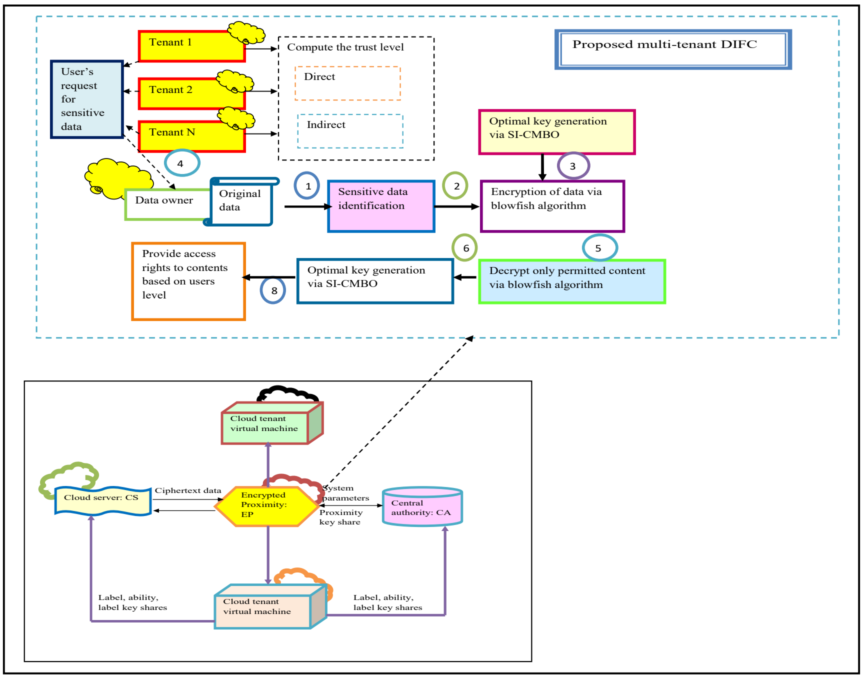

Tenant 1
Tenant 2
Tenant N
User's request for sensitive data
Compute the trust level
Direct
Indirect
Proposed multi-tenant DIFC
Optimal key generation via SI-CMBO
4
Data owner
Original data
1
Sensitive data identification
2
3
Encryption of data via blowfish algorithm
6
5
Provide access rights to contents based on users level
8
Optimal key generation via SI-CMBO
Decrypt only permitted content via blowfish algorithm
Cloud tenant virtual machine
Cloud server: CS
Ciphertext data
Encrypted Proximity: EP
System parameters
Proximity key share
Central authority: CA
Label, ability, label key shares
Cloud tenant virtual machine
Label, ability, label key shares

Fig. 1. Architecture of Proposed Work

C. System Elements

(a) Entity: The entity contains all operational items as well as privileged information associated with the system. Both the cloud tenant virtual machine and the tenants' private information files are maintained within that entity.

(b) Personal information protection is indeed a term that describes the practice of securing sensitive data. The secret domain has been represented by the symbol W_J, that has been used to designate the private information category. Secret-domain set alludes towards the secret-domain labeling symbol combination whereby the subjects correspond (I.e. $Q_O \in Set$). In this example, Q_J pertains to O's personal information, and lowest tag Q_J refers to the person who finds the data. In order to appropriately secure private information, this research project offers a blowfish paradigm. The secrete domain contains the subject's confidential level, and indeed the secret-level label set Set is added. Based on the secret level from minimum to maximum, the confidential level has already been classified as open, secret, confidential, and top secret.

D. Overall Methodology

The proposed work includes two major phases: (a) user trust level computation and (b) optimal key based cipher-text information flow security. The overall architecture of the proposed work is shown in Fig.2

The steps followed in the proposed work are depicted below:

1. The data owner O stores his/her data D (inclusive of sensitive information) onto the cloud by means of encrypting it using the blowfish algorithm.

2. The trust level is computed for all the users in the cloud environment. In this research work, a two-level trust computation has been carried out: primary and secondary trust. In case of medical chain, the primary users are the first level users like the doctors and the friends of the patients, while the secondary users are the pharmacist and insurance agent.

3. On the basis of the already computed trust level for read, modify and write operation, the access privilege is given to the concern user.
4. When an another cloud user (say doctor or pharmacist) wants to access the sensitive information of O, they request the cloud owner O. Since, the trust level of the users has already been computed, the data owner doesn't want to provide access rights every time and at the same time the users doesn't want to wait for a long time to acquire the access privilege.
5. Based on the level of accessibility permitted to the users, the decryption of D 's segments takes place. Here, the decryption is carried out using the blowfish algorithm.
6. The cipher text information flow security strategy based on the proposed Trust evaluation is implemented faithfully by EP. The user's data is encrypted using the proposed Improved Signcryption algorithm, wherein the optimal key are generated using the newly proposed Self-Improved Cat Mouse based Optimizer (SI-CMBO).

E. System Initialization

- ✓ CA establishes both the public and private keys P_{key} and S_{key}.
- ✓ Based on the original mechanism, CA allocates an identification tid to every one of the labels.
- ✓ Development of the secrete-domain label key: the label t has so far been considered, and CA constructs the tenant key share $k_{tid-user}$ and customer proximity share k_{tid-ep} using the key generation technique $keygenerate(P_{key}, tid, S_{key})$. The proxy key is indeed obtained over the encrypted channel through the EA. Furthermore, every one of the secret-domain tags is generated independently.
- ✓ For every personal data secret domain, CA establishes a secret-domain secret-level label key. K_{key1}, K_{key2}, K_{key3} and K_{key4} Secret-domain Key highly confidential key, secret-domain Key private key, secret-domain Key secret key, and secret-domain Key public key are the secret-domain Key top secret key, secret-domain Key confidential key, secret-domain Key secret key, and secret-domain Key public key, respectively.
- ✓ CA receives a security label as well as virtual machine capabilities whenever a virtual machine has been established. As a result, the tenant key shares $k_{tid-user}$ that correspond towards the label have been delegated.

F. Trust Evaluation

Two types of trust has been computed: (a) direct trust and (b) indirect trust.

In this research work, we consider 5 users: doctors $User1$ researchers $User2$ pharmacists $User3$, friends or family members of data owner $User4$ and representatives of health insurance companies $User5$. The data is denoted as O, who has two files $File(1)_1$ $File(2)_1$. Among these $File(2)_1$ higher count of sensitive fields.

(a) Primary Trust: the users who've already have the access privileges for the owners documents are said to be the primary users. The tenant (can be $User1$, $User2$, $User3$, $User4$ and $User5$) already have access right to the file of $User1$. In this case, the past transmission and data access are stored in trust table.

For illustration: doctors $User1$ and friends or family members of patients $User4$ alone has been given the access right in earlier days, so $User1$ and $User4$ can only access the file of O. Furthermore, $User1$ and $User4$ can only read a certain section of the file, not the entire document. Only that particular part of the content has been encrypted.

(b) Secondary Trust: When a new user, who do not have access right to the file request, then his/her interaction history is computed. For illustration, when researchers $User2$ request for the access rights to O, his/her previous interaction is validated, if $User2$ do not posses any previous interaction, then the access right is denied. On the other hand, when pharmacists $User3$, who has already interacted with O (but don't possess all time access right alike $User2$ and $User4$) request for access rights to O, the access is provided by O. Moreover, $User3$ can only read the particular part of the file, and not the whole document. That particular part of the content only has been encrypted.

G. Management of Labels for $Tenant_1$ and $Tenant_2$

i. The data owner tenant virtual machine would demand the CA to construct the confidential data secret-domain label Q, and the CA would agree to carry out Q 's key initialization procedure. Following that, the tenant virtual machine transmits the tenant key private key share $k_{tid-user}$ as well as proxy key share k_{tid-ep}. The renters' ability set generally includes the $Q^{+}, Q^{-}, Q^{\pm}$ .Tenant O 's virtual machine demands that the CA furnish tenant $User2$'s virtual machine with ability W^{+}_{key-p} (i.e. $User1$'s virtual machine can append labels $(Q_k.p)$ to its own privacy label). The security label and ability set of O is $Label_O = [(Q_O, p), (Q_c, s)]$ and $Ability_O = \{Q^{\pm}_O, Q^{-}_{c-s}\}$, respectively. The security label and ability set of

$User2$ is $Label_{user2} = [(Q_{User2}, p)]$ and $\{Q^{+}_{User2-p}, Q^{\pm}_{user2}\}$, respectively. On detecting the secrete- domain label C the transmission of $File(2)_1$ to $User2$ is accomplished.

ii. CA additionally investigated at whether O virtual machine's ability set seemed to have the label $Q^{\pm}_{k}$. Whereas if answer to such a question is yes, the CA of the 'B' tenant virtual machine applies ability set t^{+}_{key-p} towards the tenant virtual machine.

iii. In responding to requests from CA, the ability label $(t_k.SC_k)$ has indeed been added to $User2$'s virtual machine. CA additionally examines whether the tenant virtual machine has label $Q^{+}_{key-SC_k}$ ability. The tenant vm has now been given the tenant key share $k_{tid-user}$ depending on one's own connectivity, and the label $(Q_k.SC_k)$ has indeed been provided to the tenant vm independently depending upon $Q^{+}_{key-SC_k}$.

iv. $Tenant_1$ demands that the CA withdraw the $Tenant_2$ tenant's label $(Q_k.SC_k)$ or Q^{+}_{k} or Q^{-}_{k} ability. The ability set of a vm is validated via CA. The $Tenant_1$ tenant virtual machine's labeling $(Q_k.SC_k)$ or Q^{+}_{k} or Q^{-}_{k} ability gets dropped whenever the ability set $Q^{\pm}_{k}$ does become available inside the $Tenant_2$ virtual machine.

v. The tenant vm O has requested that the CA provide the ability $Q^{\pm}_{k}$ towards the tenant vm $User2$, as well as the CA has reported the presence of label $Q^{\pm}_{k}$ within the O vm. If label $Q^{\pm}_{k}$ is active in O vm, the ability set is transferred to $User2$.

vi. The $File(1)_1$ is written onto the cloud storage server by O, and in the hypervisor virtual machine, the current label $[(Q_O, p), (Q_c, s)]$ is used for encryption proxy of O to call the blowfish algorithm (encryption algorithm) for performing the encryption operation. The cipher text of $File(1)_1$ is acquired at the end of encryption process. In the cloud storage system, when $User2$ has been denied from reading $File(2)_1$, as it haven't met the file reading rules. On the other hand, if $User2$ is controlled by a malicious user outside the system, and if has acquired $File(1)_1$ illegally. But still $File(1)_1$ stored in the cloud server is protected by the cipher text information flow control system. So, the $User2$ cannot download or write over it. On the other hand, $File(2)_1$ is send from O to $User2$ by the information flow control module residing within the virtual machine hypervisor, only when the information flow security rules are met. Then the file $File(2)_1$ can be received by $User2$. Further, when $Tenant_2$ wants to write over $File(2)_1$, the written file is decrypted by EP using the blowfish algorithm.

H. Communication between $Tenant_1$ and $Tenant_2$

File $File(1)_1$ is sent from O virtual computer to $User2$ vm. A validation of the data-flow policy rule has been performed in order to execute the DIFC module within the hypervisor virtual machines of O and $User2$. While the regulation is in effect, the tenant virtual machine $User2$ s privacy label becomes tainted, and $File(1)_1$ may now be viewable by $User2$.

I. Information flow between $Tenant_1$ and $Tenant_2$

The data file $File(1)_1$ has been transferred from the O vm to the $User2$ vm. The entity of O with the flow of information to entity $User2$ commences whenever the constraints are satisfied. This can be seen in Eq. (2).

$$Tenant_1 \rightarrow bif Label_{Tenant_1} - Ability^{-}_{Tenant_1} \subseteq Label_{Tenant_2} + Ability^{-}_{Tenant_2} \quad (2)$$

When information is received by vm $User2'$, $User2$'s security mark gets corrupted, leading mostly in modifications. This is demonstrated in Eq (3).

$$Label'_{User_2} \leftarrow Label'_{User_2} \cup (Label'_{User_1} - Ability^{-}_{User_1}) \quad (2)$$

The information outflow entity O's transmitting ability and the information inflow entity $User2$'s receiving ability, as well as the implementation of label taint propagation, must all be considered when securing low-security information flows to the secret domain's high-security level.

J. Cloud network file read and write encryption and decryption processes for $Tenant_1$ and $Tenant_2$

Tenant Virtual machine $Tenant_1$ writes File $File(1)_1$ towards the cloud storage server: The network file writing rules that correlate to the data - flow strategy have been implemented by the vm hypervisor within the security secret proxy module. The privacy encryption module has been used by algorithm $Encrypt(S_{key}, tids, P_{key})$ to encrypt $File(1)_1$ based on the confidentiality labeling inside the user's security labeling. The document privacy label is adjusted and afterwards transmitted towards the cloud storage server as per the writing policy.

The virtual machine hypervisor inside the DIFC component verifies the file reading rule based on the tenant vm security marking and file marking. Tenant Vm $User2$ reads file $File(1)_1$ from the cloud storage server. The $Decrypt(S_{key},\{sk_{tid-ep}\}tid \in tids, C)$ algorithm is being used to decode the $\{m_{tid-ep}\}tid \in tids$ encrypted segment employing the proxy key sharing if indeed the file reading criteria are fulfilled.

The cipher text for the file would then be distributed, and the decrypted pieces would be sent to the tenant vm. The tenant vm $User2$ exceeds the authority if the file writing rule isn't really met. To retrieve the $\{m_{tid-user}\}tid \in tids$ decrypted fragment, the tenant vm $User2$ must decrypt the document with the security label using the tenant key sharing execution method $Decrypt(S_{key},\{sk_{tid-eA}\}tid \in tids, C)$. The combination algorithm $combine(S_{key}\{m_{tid-w}\}tid \in tids, w \in [user, eA], C)$ with encrypted fragments is being used to obtain the plain text of the algorithm. The read process is now completed.

K. Blowfish Algorithm

Bruce envisioned blowfish as a low-cost, quick alternative to traditional encryption methods. It is increasingly gaining acceptance as a more secure encryption method. There are several advantages to using the Blowfish method. It is fast and suitable for hardware execution, and it does not require a licence. The Blowfish approach's core operators are XOR, addition, and database lookup. The following are some of the conditions of the Blowfish approach:

A 64-bit block cipher with an unequal key length is included.

There are four 32-bit -boxes and a -array in this file. The -array has 18 32-bit sub-keys, whereas each -box contains 256 items.

"A key-expansion portion and a data-encryption part" are the two components of the technique.

The input is a 64-bit data element.

The F operation deploys the substitution boxes, where there are 4, every one comprising 256 32-bit entries. If block XL is split to 8-bit blocks a, b, c , d then function $F(XL)$ is specified by Eq. (3).

$$F(XL)=((P_{1,a}+P_{2,b}\,\text{mod}2\text{\^{}}32)\oplus P_{3,c})+P_{4,d}\,\text{mod}2\text{\^{}}32$$
(3)

The keys utilized in blow-fish algorithm are optimally selected using the SI-CMBO model. The solution fed as input to SI-CMBO model is depicted in Fig.2, Fig.3 and Fig.4 for varying key sizes.

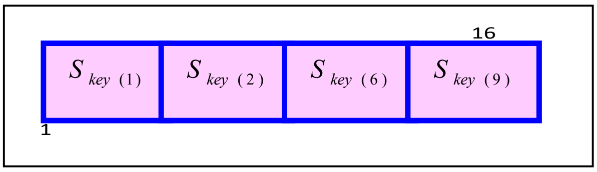


Fig. 2. Solution Encoding for Key size=16

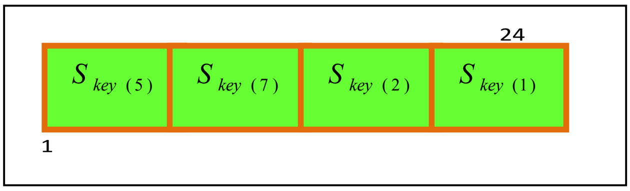


Fig. 3. Solution Encoding for Key size=24

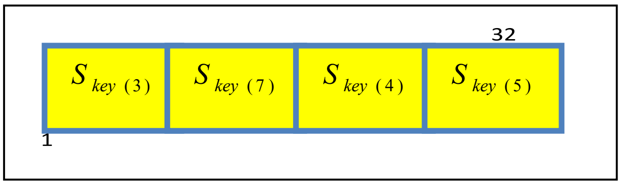


Fig. 4. Solution Encoding for Key size=32

The steps followed in the SI-CMBO model is furnished below:

In fact, SI-CMBO is the extended version of standard CMBO model.

Step 1: The initial population of N search agents is initialized.

Step 2: The parameters of N, N_c, N_m, T are initialized. Here, N is the count of members in population matrix Y .

Step 3: The initial population is created as per Eq. (4).

$$Y=\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}_{N*m}=\begin{bmatrix} y_{1,1} & \cdots & y_{1,d} & \cdots & y_{1,m} \\ \vdots & \ddots & \vdots & \cdot^{\cdot^{\cdot}} & \vdots \\ y_{i,1} & \cdots & y_{i,d} & \cdots & y_{i,m} \\ \vdots & \cdot^{\cdot^{\cdot}} & \vdots & \ddots & \vdots \\ y_{N,1} & \cdots & y_{N,d} & \cdots & y_{N,m} \end{bmatrix}_{N*m} \quad (4)$$

Here, $y_{i,d}$ is the d^{th} problem variable.

Step 4: The fitness of the search agents are computed as per Eq. (1).

Step 5: Using Eq. (5) to Eq. (6), update the sorted population matrix Y^S . Here, i^{th} population of sorted population matrix is denoted as $y_{i,d}^{S}$. In addition, Obj^{S} is the sorted objective function based vector.

$$Y^{S}=\begin{bmatrix} Y_1^S \\ Y_2^S \\ \vdots \\ Y_N^S \end{bmatrix}_{N*m}=\begin{bmatrix} y_{1,1}^S & \cdots & y_{1,d}^S & \cdots & y_{1,m}^S \\ \vdots & \ddots & \vdots & \cdot^{\cdot^{\cdot}} & \vdots \\ y_{i,1}^S & \cdots & y_{i,d}^S & \cdots & y_{i,m}^S \\ \vdots & \cdot^{\cdot^{\cdot}} & \vdots & \ddots & \vdots \\ y_{N,1}^S & \cdots & y_{N,d}^S & \cdots & y_{N,m}^S \end{bmatrix}_{N*m} \quad (5).$$

$$Obj^{S}=\begin{bmatrix} Obj_1^S & \min(Obj) \\ Obj_2^S & \min(Obj) \\ \vdots & \vdots \\ Obj_N^S & \min(Obj) \end{bmatrix}_{N*1} \quad (6).$$

Step 6: Using Eq. (7), the mice population is chosen

$$M=\begin{bmatrix} M_1=X_1^S \\ \vdots \\ M_i=X_i^S \\ \vdots \\ M_{N_m}=X_{N_m}^S \end{bmatrix}_{N_m*m}=\begin{bmatrix} y_{1,1}^S & \cdots & y_{1,d}^S & \cdots & y_{1,m}^S \\ \vdots & \ddots & \vdots & \cdot^{\cdot^{\cdot}} & \vdots \\ y_{i,1}^S & \cdots & y_{i,d}^S & \cdots & y_{i,m}^S \\ \vdots & \cdot^{\cdot^{\cdot}} & \vdots & \ddots & \vdots \\ y_{N_m,1}^S & \cdots & y_{N_m,d}^S & \cdots & y_{N_m,m}^S \end{bmatrix}_{N_m*m} \quad (7)$$

Step 7: Using Eq. (8), the cat population is selected

$$C=\begin{bmatrix} C_1=X_{N_m+1}^S \\ \vdots \\ C_i=X_{N_m+j}^S \\ \vdots \\ C_{N_c}=X_{N_m+N_c}^S \end{bmatrix}_{N_c*m}=\begin{bmatrix} y_{N_m+1,1}^S & \cdots & y_{N_m+1,d}^S & \cdots & y_{N_m+1,m}^S \\ \vdots & \ddots & \vdots & \cdot^{\cdot^{\cdot}} & \vdots \\ y_{N_m+j,1}^S & \cdots & y_{N_m+j,d}^S & \cdots & y_{N_m+j,m}^S \\ \vdots & \cdot^{\cdot^{\cdot}} & \vdots & \ddots & \vdots \\ y_{N_m+N_c,1}^S & \cdots & y_{N_m+N_c,d}^S & \cdots & y_{N_m+N_c,m}^S \end{bmatrix}_{N_c*m}$$

(8)

Step 8: Here, M, N_m, M_i, C, N_c, C_j points to the mice population, count of mice, j^{th} mice, cat population, count of cats and i^{th} cat, respectively.

Step 9: Our contribution resides in this phase. Here, C_j is updated using the newly proposed expression given in Eq. (9) to Eq. (10)

$$C_j^{new}=W*C_{j,d}+r*(M_{k,d}-I*C_{j,d}) \quad (9)$$

Here, $I=round(1+rand)$ (10)

C_j^{new} points to the new position of the j^{th} cat and $C_{j,d}$ is the new value for d^{th} problem. In addition, the random value r is estimated using chaotic map, rather than generating it randomly within the limit [0,1].

W is the inertial weight that is computed using the newly proposed expression given in Eq. (11).

$$W=W_{\max}-(W_{\max}-W_{\min})+\left(\frac{itr}{\max^{itr}}\right) (11)$$

Here, itr points to the current iteration and $\max^{itr}$ is the maximal iteration. In addition, $W_{\max}$ and $W_{\min}$ are the maximal and minimal inertial weight

Step 10: If $j=N_c$

(a)If the above condition is satisfied then H_i is created using Eq. (12).

$$H_i=h_{i,d}=y_{l,d}\ \&i=1:N_m, d=1:m, l\in 1:N \ (12)$$

Step 11: Then, M_i is updated using the expression given in Eq. (13) to Eq. (14), respectively.

$$M_i^{new}:m_{i,d}^{new}=m_{i,d}+r*(h_{i,d}-I*m_{i,d})*Sign(F_i^m-F_i^H)\&i=1:N_m,d=1:m \quad (13)$$

$$M_i=\begin{cases} M_i^{new} & \mid F_i^{m,new}<F_i^m \\ M_i & \mid else \end{cases} \quad (14)$$

Step 12: (b) In case, if the above condition is not satisfied, then increase j by 1, and again update C_j using the newly proposed expression given in Eq. (9) to Eq. (10).

(c) Terminate the if condition

Step 13: If $i=N_m$ then

(a) if the above condition is satisfied, then check if $t=T$.

(b) if the above condition is not satisfied, then increase i by 1.

(c) Endif

Step 14: If $t=T$, then best solution acquired so far is returned,

Step 15: If $t\neq T$, then increase i by 1 and move back to step 8.

Step 16: Terminate

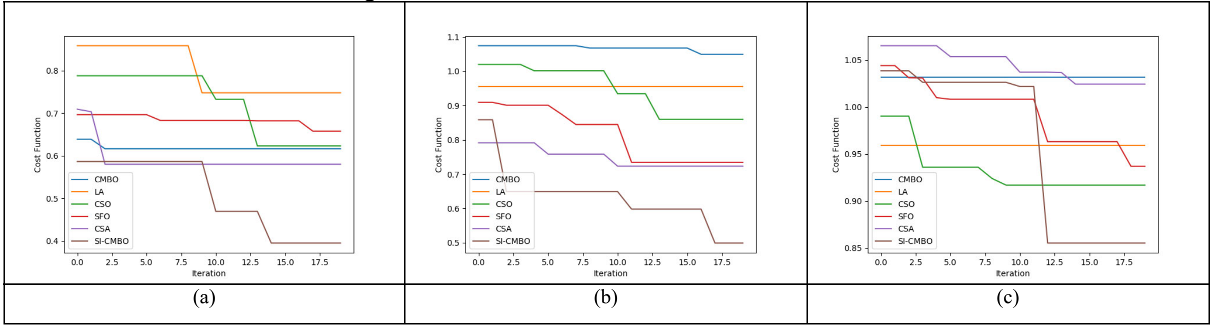
IV. Result and Disussion

A. Experimental Setup

The proposed work (DIFC+Blowfish+SI-CMBO) has been implemented in Python, and the corresponding results acquired are noted. To validate the efficiency of the projected work, three different database has been used: “heart disease dataset :https://www.kaggle.com/ronitf/heart-disease-uci; lung cancer dataset:https://www.kaggle.com/yusufdede/lung-cancer-dataset; breast cancer dataset :https://www.kaggle.com/uciml/breast-cancer-wisconsin-data”, respectively. The evaluation has been carried out in terms of convergence analysis, encryption time, decryption time, turnaround time, KPA analysis as well.

B. Convergence Analysis

The convergence analysis is undergone to test to efficiency of the introduced SI-CMBO over the existing CMBO, LA,CSO, SFO and CSA. This evaluation has been undergone by varying the count of iterations from 0, 2.5, 5, 7.5, 10, 12.5,15 and 17.5, respectively. The major objective of this research work is to minimization function, therefore the approach that records the least cost function is said to be highly convergent. The results acquired by the proposed work under heart disease dataset, lung cancer dataset and breast cancer dataset are depicted in Fig.5. On observing the outcomes, the proposed work seems to have achieved the least cost function for every variation in the iteration count, and hence the proposed SI-CMBO is said to achieve the highest convergence speed. Therefore, the desired objective of key break time improvement can be achieved successfully by the proposed work.



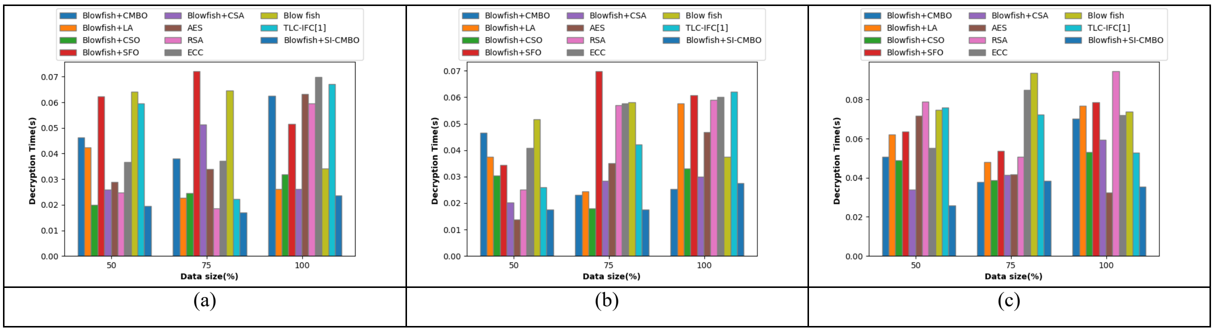
(a) (b) (c)

Fig. 5. Convergence Analysis of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset

C. Analysis on Decryption Time

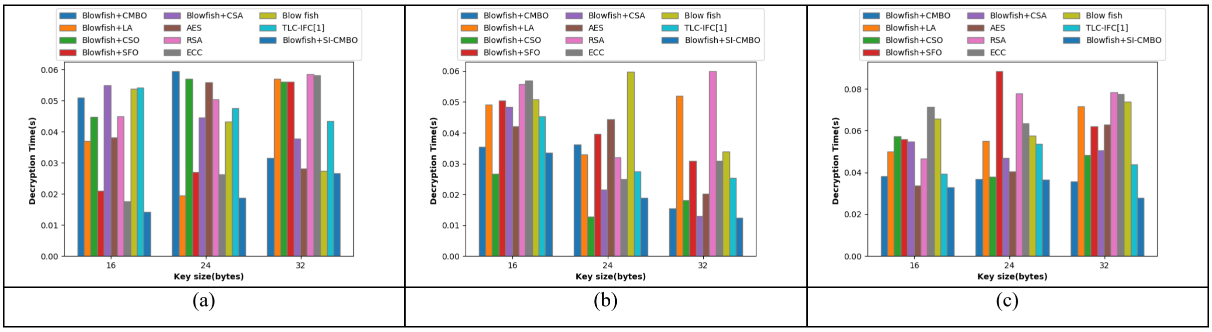
owing to the optimally selected keys using SI-CMBO model.

The suggested work's decryption time for the heart disease dataset, lung cancer dataset, and breast cancer dataset is presented in Figure 4. The encryption time was measured in two ways: (A) by adjusting the data size from 50% to 100%; and (B) by altering the key size from 16 bytes to 24 bytes to 32 bytes. According to the results, the suggested approach has the shortest decryption time in both areas. The properly chosen keys utilizing the SI-CMBO model are the main cause for this improvement.



(a) (b) (c)

Fig. 6. Analysis on decryption time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the data size

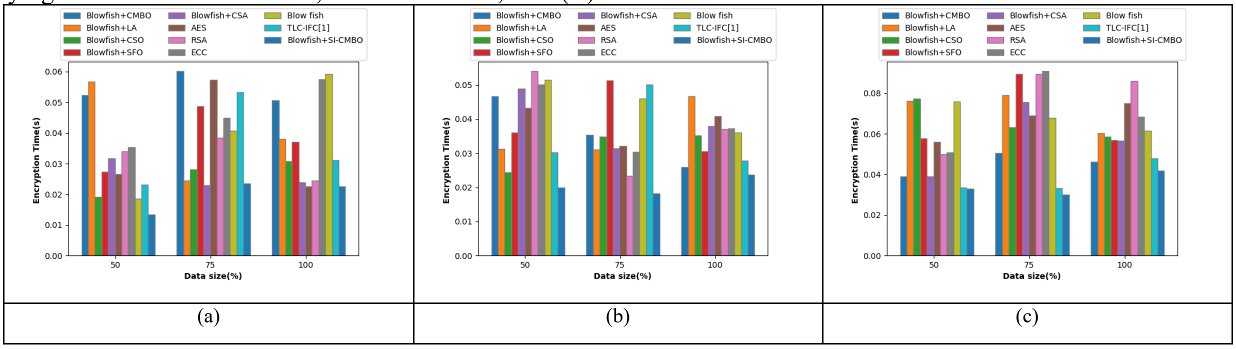


(a) (b) (c)

Fig. 7. Analysis on decryption time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the key size

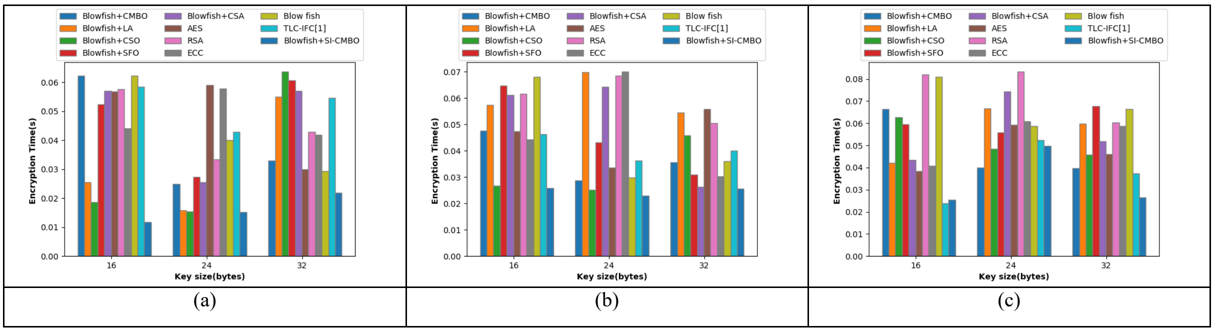
D. Decryption time

The encryption time recorded by the proposed work for heart disease dataset, lung cancer dataset and breast cancer dataset is manifested in Fig.8 and Fig.9. This evaluation has been carried out under two different aspects: (A) by varying the data size from 50%, 75% and 100%; and (B) by varying the key size from 16 bytes, 24 bytes and 32 bytes. The proposed work has recorded the least encryption time while varying the data size as well as key size. The major reason behind this improvement is owing to the optimally selected key using the SI-CMBO model.



(a) (b) (c)

Fig. 8. Analysis on encryption time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the data size

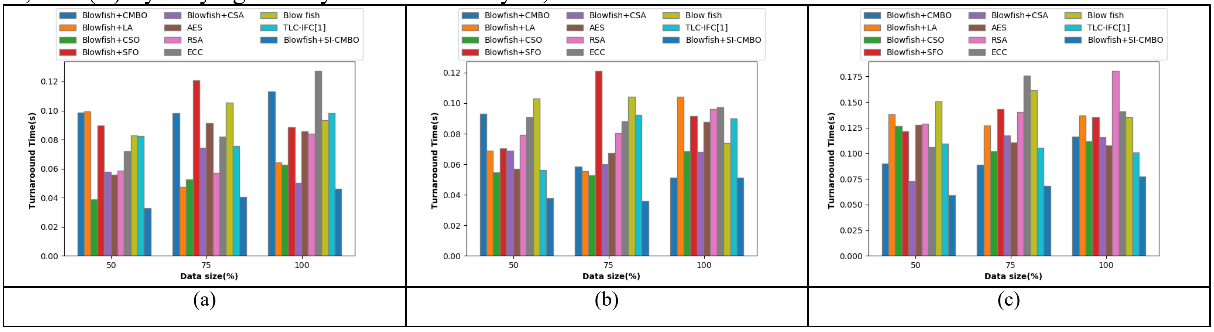


(a) (b) (c)

Fig. 9. Analysis on encryption time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the key size

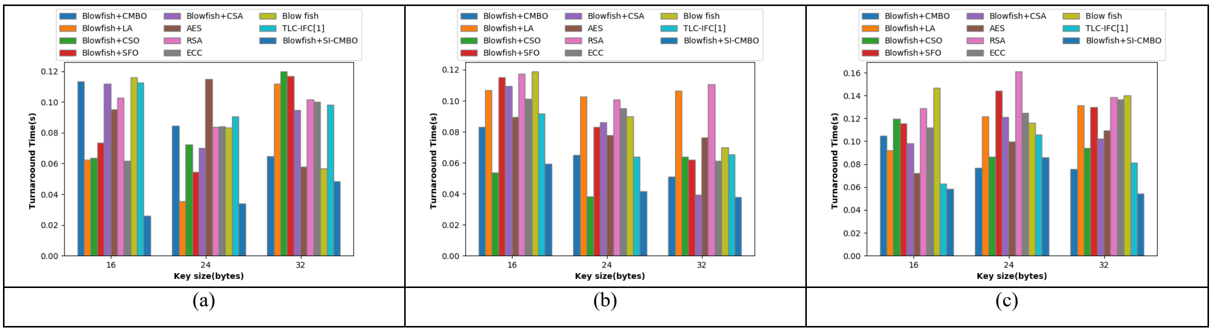
E. Analysis on Turnaround time

The turnaround time is the summation of the encryption as well as decryption time. The turnaround time has been evaluated (A) by varying the data size from 50%, 75% and 100%; and (B) by varying the key size from 16 bytes, 24 bytes and 32 bytes. Since, the encryption as well as decryption time is lower, the turnaround time automatically has been reduced. The results acquired are shown in Fig.10 and Fig.11, respectively.



(a) (b) (c)

Fig. 10. Analysis on turn around time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the data size



(a) (b) (c)

Fig. 11. Analysis on turn around time of SI-CMBO for (a) heart disease dataset (b) breast cancer dataset and (c) lung cancer dataset by varying the key size

F. KPA Analysis

The KPA analysis has been undergone for the heart disease dataset, lung cancer dataset and breast cancer dataset. The results acquitted in term of heart disease dataset, lung cancer dataset and breast cancer dataset are manifested in Table I, Table II and Table III, respectively. This evaluation has been accomplished by varying the key size from 16 bytes, 24 bytes and 32 bytes. On observing the outcomes, the proposed work has attained the least KPA value, and hence portrays that the information is highly secured during its flow.

TABLE I. KPA ANALYSIS OF DIFC+BLOWFISH+ SI-CMBO FOR HEART DISEASE DATASET

Key size	DIFC+AES	DIFC+RSA	DIFC+ECC	DIFC+Blowfish	TLC-IFC[1]	DIFC+Blowfish+ SI-CMBO
16	0.581322	0.529304	0.611854	0.523902	0.593185	0.321262
24	0.557818	0.620705	0.561321	0.624296	0.639228	0.52029
32	0.568836	0.609439	0.584103	0.644738	0.59506	0.423376

TABLE II. KPA ANALYSIS OF DIFC+BLOWFISH+ SI-CMBO FOR LUNG CANCER DATASET

Key size	DIFC+AES	DIFC+RSA	DIFC+ECC	DIFC+Blowfish	TLC-IFC[1]	DIFC+Blowfish+ SI-CMBO
16	0.683691	0.591169	0.586892	0.614421	0.656764	0.500755
24	0.580324	0.641816	0.698902	0.669993	0.676819	0.545091
32	0.617713	0.663102	0.700103	0.643223	0.675152	0.46973

TABLE III. KPA ANALYSIS OF DIFC+BLOWFISH+ SI-CMBO FOR BREAST CANCER DATASET

Key size	DIFC+AES	DIFC+RSA	DIFC+ECC	DIFC+Blowfish	TLC-IFC[1]	DIFC+Blowfish+ SI-CMBO
16	0.550158	0.617276	0.557851	0.610733	0.516406	0.279081
24	0.567806	0.604296	0.558095	0.629957	0.618801	0.400238
32	0.623874	0.599474	0.577731	0.541797	0.592039	0.440799

G. Parametric analysis-Encryption time(s)

The parametric analysis on encryption time has been undergone by varying the I value from 1, 2, 3 and 4, respectively. The results acquired are shown in Table IV. In case of Heart Disease, the least encryption time has been recorded as 0.010266 at I =4. In addition, under breast cancer and Lung Cancer dataset, the proposed SI-CMBO has recorded the least encryption time as 0.005356 and 0.006679 at I =1 and I =3, respectively.

TABLE IV. PARAMETRIC ANALYSIS-ENCRYPTION TIME(S) BY VARYING I VALUE

Dataset	I=1	I=2	I=3	I=4
Heart Disease	0.020308	0.020307	0.006332	0.010266
Breast Cancer	0.005356	0.018215	0.007166	0.020308
Lung Cancer	0.020308	0.007631	0.006679	0.013886

H. Parametric analysis-Decryption time(s)

The parametric analysis on decryption time has been undergone by varying the I value from 1, 2, 3 and 4, respectively. The results acquired are shown in Table V. the proposed work has recorded the least decryption time as 0.007124 at I =2, 0.007335 at I =3 and 0.006149 at I =3 for heart disease, breast cancer and Lung Cancer dataset, respectively.

TABLE V. PARAMETRIC ANALYSIS-DECRYPTION TIME(S) BY VARYING I VALUE

Dataset	I=1	I=2	I=3	I=4
Heart Disease	0.007124	0.008714	0.008358	0.020307
Breast Cancer	0.020309	0.011788	0.007335	0.009984
Lung Cancer	0.021986	0.008983	0.006149	0.007684

V. CONCLUSION

In this research work**,** a novel Multi-tenant Decentralized Information Flow Control (MT-DIFC) model is introduced in this research work. The proposed system will encapsulate four types of entities: (1) The central authority (CA), (2) The encryption proxy (EP), (3) Cloud server CS and (4) Multi-tenant Cloud virtual machines. Our contribution resides within the encryption proxy (EP). Initially, the trust level of all the users within each of the cloud is computed using the proposed two-stage trust computational model, wherein the user is categorized bas primary and secondary users. The primary and secondary users vary based on the application and data owners preference. Based on the computed trust level, the access privilege is provided to the cloud users. In EP, the cipher text information flow security strategy is implemented using the blowfish encryption model. For the data encryption as well as decryption, the key generation is the crucial as well as the challenging part. In this research work, a new optimal key generation is carried out within the blowfish encryption Algorithm. In the blowfish encryption Algorithm, both the data encryption as well as decryption is accomplishment using the newly proposed optimal key. The proposed optimal key has been selected using a new Self Improved Cat and Mouse Based Optimizer (SI-CMBO), which has been an advanced version of the standard Cat and Mouse Based Optimizer. The proposed model is validated in terms of encryption time, decryption time, KPA attacks as well.

References

[1] Z. Zhang, Z. Yang, X. Du, W. Li, X. Chen and L. Sun, "Tenant-Led Ciphertext Information Flow Control for Cloud Virtual Machines," IEEE *Access*, vol. 9, pp. 15156-15169, 2021.
doi: 10.1109/ACCESS.2021.3051061

[2] Anum Khurshid1 Abdul Nasir Khan1 Fiaz Gul Khan1 Mazhar Ali1 Junaid Shuja1 Atta ur Rehman Khan, "Secure-CamFlow: A device-oriented security model to assist information flow control systems in cloud environments for IoTs, Wiley, 2019

[3] Ning Xi, Jianfeng Ma, Cong Sun, Di Lu, Yulong Shen, "Information flow control on encrypted data for service composition among multiple clouds", Distrib Parallel Databases, 2019

[4] Kriti Bhushan & Brij B. Gupta, "Network flow analysis for detection and mitigation of Fraudulent Resource Consumption (FRC) attacks in multimedia cloud computing", Multimed Tools Appl, 2018

[5] Nadya El Moussaid* and Maryam El Azhari , "Enhance the security properties and information flow control", Int. J. Electronic Business, Vol. 15, No. 3, 2020

[6] K.Sravya Reddy, N.Vijay Kumar, "INFORMATION FLOW CONTROL FOR SECURE CLOUD COMPUTING", International Journal For Technological Research In Engineering, VOl.4, No.3, 2017

[7] Charilaos Skandylas, Narges Khakpour, Jesper Andersson, "Adaptive Trust-Aware Decentralized Information Flow Control", IEEE, 2020

[8] Maxwell Krohn, Alexander Yip, Micah Brodsky ,Natan Cliffer ,M. Frans Kaashoek ,Eddie Kohler ,Robert Morris,"Information Flow Control for Standard OS Abstractions", SOSP, 2017

[9] Kalev Alpernas,Cormac Flanagan, Sadjad Fouladi, Leonid Ryzhyk, Mooly Sagiv, Thomas Schmitz, Keith Winstein, "Secure serverless computing using dynamic information flow control", Proceedings of the ACM on Programming Languages, Vol.2, 2018

[10] Z. Su, Y. Peng, F. Ge, C. Song, F. Ma and F. Biennier, "Collaboration-oriented information flow analysis and control for Mobile Cloud," *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, India, 2017, pp. 1-6.
doi: 10.1109/CIACT.2017.7977319

[11] Z Z. Zhang, Z. Yang, X. Du, W. Li, X. Chen and L. Sun, "Tenant-Led Ciphertext Information Flow Control for Cloud Virtual Machines," IEEE *Access*, vol. 9, pp. 15156-15169, 2021.
doi: 10.1109/ACCESS.2021.3051061

[12] Z. Su, Y. Peng, F. Ge, C. Song, F. Ma and F. Biennier, "Collaboration-oriented information flow analysis and control for Mobile Cloud," *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, India, 2017, pp. 1-6.
doi: 10.1109/CIACT.2017.7977319

[13] T. F. J. -. Pasquier, J. Singh, J. Bacon and D. Eyers, "Information Flow Audit for PaaS Clouds," *2016 IEEE International Conference on Cloud Engineering (IC2E)*, Berlin, Germany, 2016, pp. 42-51.
doi: 10.1109/IC2E.2016.19

[14] Z. Ruifeng, L. Shiming, L. Yang, W. Bin, G. Wenxin and L. Jiangang, "Application Analysis and Prospect of Cloud Platform in Operation Control of New Energy Power System," *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Suzhou, China, 2019, pp. 980-985.
doi: 10.1109/CYBER46603.2019.9066672

[15] G. Kozhevnikov, O. Pihnastyi and M. Glavchev, "Input Flow Control Algorithms of the Trasnsport System," *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2020, pp. 301-305.
doi: 10.1109/PICST51311.2020.9468108

[16] X. Lu, L. Cao and X. Du, "Dynamic Control Method for Tenants' Sensitive Information Flow Based on Virtual Boundary Recognition," IEEE *Access*, vol. 8, pp. 162548-162568, 2020.
doi: 10.1109/ACCESS.2020.3021415

[17] I. Yen, F. Bastani, N. Solanki, Y. Huang and S. Hwang, "Trustworthy Computing in the Dynamic IoT Cloud," *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, UT, USA, 2018, pp. 411-418.
doi: 10.1109/IRI.2018.00067

[18] M. Elsayed and M. Zulkernine, "IFCaaS: Information Flow Control as a Service for Cloud Security," *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Salzburg, Austria, 2016, pp. 211-216.
doi: 10.1109/ARES.2016.27

[19] T. Jia, L. Yang, P. Chen, Y. Li, F. Meng and J. Xu, "LogSed: Anomaly Diagnosis through Mining Time-Weighted Control Flow Graph in Logs," *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, Honololu, HI,

USA, 2017, pp. 447-455. doi: 10.1109/CLOUD.2017.64

[20] *C. Lai, A. N. Tantawi and C. Pu, "Coarse-Grained Information Flow Control on Hybrid Clouds,"* 2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, 2016, pp. 319-326. doi: 10.1109/CLOUD.2016.0050*

[21] R. K. Shyamasundar, N. V. N. Kumar and M. Rajarajan, "Information-Flow Control for Building Security and Privacy Preserving Hybrid Clouds," *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Sydney, NSW, Australia, 2016, pp. 1410-1417. doi: 10.1109/HPCC-SmartCity-DSS.2016.0201

[22] Priyanka S. Mane, Yogesh B. Gurav,"Secure Cloud Computing Using Decentralized Information Flow Control", International Advanced Research Journal in Science, Engineering and Technology, Vol.3, No.6, 2016

[23] Neeta R. Somavanshi,Y. B. Gurav,"Security in Cloud Computing Environment by Decentralized Information Flow Control", International Journal of Innovative Research in Computer and Communication Engineering, Vol.4, No.6, 2016

[24] Neeta R. Somavanshi, Y. B. Gurav, "SURVEY OF DECENTRALIZED INFORMATION FLOWCONTROLFOR RELATIONAL DATABASE-IFDB", International J. of Engg. Research & Indu. Appls. (IJERIA), Vol.8, No.8, 2015

[25] Salve Bhagyashri, Prof. Y.B.Gurav, "Privacy-Preserving Public Auditing For Secure Cloud Storage",IOSR Journal of Computer Engineering (IOSR-JCE), Vil.16, No.4, 2014

Yogesh B. Gurav His research activities are currently twofold : while the first research activity is set to explore the Protection and Authentication of Issues in Wireless and ADHOC Networks in Epidemic Conditions ; the second major research theme that he is pursuing is focused on De-Centralized Information Flow Control for Cloud Virtual Machines with Hybrid AES- ECC and Improved Meta-Heuristic Optimization based optimal Key generation.He has also presented various academic as well as research-based papers in several national and international conferences and journals including the ” Proceeding of First Doctoral Symposium on Natural Computing Research,2020 Lecture Notes in Networks and Systems 169, under exclusive license to Springer Nature Singapore. His areas of research interest are Cloud Computing an Data Security.

Dr.Bankat M. Patil received his Ph.D. degree in Computer Science (Data Mining) from Indian Institute of Technology , Roorkee (India) in Nov 2011. He is currently working as Professor in Computer Science and Engineering in MBES’s College of Engineering, Ambajogai. His research interests are generally in the areas of Data Mining, Soft Computing, Decision Support System in Medicine, Computer Networking, Cloud Computing. He has also presented various academic as well as research-based papers in several national and international conferences and journals.