Peer-to-Peer Networking and Applications (2021) 14:2430–2442
https://doi.org/10.1007/s12083-020-00959-6

FAST: Fast Accessing Scheme for data Transmission in cloud computing

Suyel Namasudra[1] · Rupak Chakraborty[2] · Seifedine Kadry[3] · Gunasekaran Manogaran[4,5] · Bharat S. Rawal[6]

Received: 7 February 2020 / Accepted: 2 July 2020 / Published online: 28 August 2020

Abstract
With the rapid advancement of emerging technologies, a huge amount of data is created and processed in daily life. Nowadays, Cloud Computing (CC) technology is one of the frequently adopted technologies to access and store data over the internet. CC mainly provides on-demand and pay-per-use services to users. However, access control and security are two major issues that users face in a cloud environment. During data access from a cloud server, the searching time of a Data Owner (DO) and the data accessing time are high. Therefore, users utilize more cloud services and pay more. High system overhead is another issue of a cloud environment. In this paper, a novel access control model has been proposed to overcome all these concerns. Here, the Cloud Service Provider (CSP) maintains a temporary table based on the data type and popularity value of the DO for fast and efficient data accessing. The cloud service provider can easily search the data owner by using the table and the data accessing time is remarkably reduced. Experimental results and theoretical analysis of the proposed scheme prove its efficiency over the existing schemes.

Keywords Cloud computing · Access control · Cloud service provider · Data type · Popularity value · CloudSim

1 Introduction

Cloud computing is a radical paradigm for providing computational services and unlimited storage to the users on a pay-per-use basis. In a CC environment, numerous computers are combined by different technologies, namely utility computing, network system and many more [1]. CC is an advanced field in Information Technology (IT) industries because of its attractive features. There are mainly four types of cloud deployment models: (i) public cloud (ii) private cloud (iii) hybrid cloud, and (iv) community cloud. By definition, a cloud operated only by a single organization or by a third party is a "private cloud". The CSP in a "public cloud" provides resources to the users. Anyone can join in the public cloud. In a "community cloud", the cloud infrastructure is shared by a community of numerous organizations. All these organizations must have a common goal. A combination of community, private and pubic cloud is a "hybrid cloud". There are mainly three types of entities in a CC environment: (i) cloud service provider (ii) data owner, and (iii) user. The CSP plays the role of the central authority by controlling all the activities of a cloud environment. DOs are responsible for storing their data on the cloud server, and users access data or any service from the cloud server [2–4]. CC has several benefits, such as pay-per-use, on-demand service, unlimited storage capacity, flexibility and many more. However, it also has many limitations, such as access control, security, limited control, downtime and many more. Among all these limitations or issues, access control is the most critical one. Access control can be defined as a process of allowing only authorized parties to access any resource or data from a server or organization. There are three primitives of an access control model: (i) subject (ii) object, and (iii) operation. An entity that initiates an access request is considered as a subject and an object is an entity for which an access request is being initiated. An operation decides the actions that must apply to the object.

This article is part of the Topical Collection: *Special Issue on Network In Box, Architecture, Networking and Applications*
Guest Editor: Ching-Hsien Hsu

✉ Suyel Namasudra
suyelnamasudra@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Patna, Patna, Bihar, India

[2] Department of CSE, Guru Nanak Institute of Technology, Kolkata, West Bengal, India

[3] Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Beirut, Lebanon

[4] University of California, Davis, CA, USA

[5] Asia University, Taiwan, Taichung City, Taiwan

[6] Cyber Department, Gannon University, Erie, PA, USA

Implementing an efficient access control model mainly based on three concepts: (i) policy (ii) model, and (iii) mechanism. All the cloud environments use their own access control models for earning revenue. However, all of them face some common issues, such as high time to search a DO, high time to provide any data, high CPU utilization and high system overhead. In the cloud environment, there is no systematic process to save the details of the DOs. Due to this problem, when the user initiates a data access request to the CSP, the service provider may need to check the entire database to find the owner of the requested data. After finding the details of the DO, the CSP provides the Public Key of the DO (PCKDO) to the user, who initiates the request. Therefore, the CSP experiences high searching time of PCKDO, and as result, the user must wait for a long time to send the request to the respective DO for the secret key and Access Certificate (AC) of the requested data. The user is only allowed to send the request to the DO by using the corresponding PCKDO. Thus, the data accessing time is also increased. When the CSP takes more time to search the PCKDO from the cloud database, the users automatically utilize more cloud services and they need to pay more. High system overhead is another issue in the existing scheme [5] because the DO has to be always online during the complete data communication process.

Many access control models are already developed to solve these aforementioned problems [6–26]. Ferraiolo and Kuhn [27] have used the role of a job to propose the Role Based Access Control (RBAC) model. Job roles are used in RBAC to control data accessing of the users. However, this scheme consumes time to provide data to the users. Attributes of a ciphertext are utilized in Key Policy based Attribute-Based Encryption (KPABE) [28]. In KPABE, the DOs must depend on a key generator and they are not allowed to control the access policies. In [29], authors have designed a model that supports accessing any resource based on the user's attributes. There are many complex operations in their proposed model. In a particular organization, the AC of any user in the activity based access control model [30] for any data can be assigned by using the designation of the user. But, the system overhead of this model is high because of the presence of several components in its working architecture. Another model was introduced based on the substring index generation process [31]. Here, the searching time of a DO is high. Thus, the system overhead is automatically increased. In 2017, Zigzag Morse Code based Access Control Model (ZMCACM) was proposed by Murugan and Thilagavathy [32] for the CC environment. In ZMCACM, Morse code is used to support efficient data accessing. However, the CPU utilization is more in this scheme. A novel technique for a multitenant CC environment has been proposed by Almutairi et al. [33]. However, this scheme does not support fast data accessing. So, users must pay more for using the cloud services. All these existing schemes experience the high searching time of DO, high data accessing time and high system overhead. In addition, CPU utilizations are more in these existing schemes. These issues motivate to develop an efficient and fast access control model for the CC environment.

A novel Access Control Model (ACM) has been proposed in this paper, namely Fast Accessing Scheme for data Transmission (FAST) in the CC environment to overcome the problems of the existing schemes. In FAST, the CSP manages a temporary table based on the data type of the DO and the popularity value of the DO. Here, the CSP maintains a popularity value for every DO in the Table. A popularity value of any DO can be defined as the number of times that the DO provides data to the users. Whenever a user requests for a resource or data from the cloud server, the CSP initiates a query in the table based on the requested data type and the popularity value of the DO. Therefore, the CSP does not search the whole database to find one DO, and after finding the DO from the temporary table, the CSP easily can provide the PCKDO to the user in less time. Thus, the searching time of PCKDO can be decreased. The accessing time of data can be continually reduced in the proposed scheme since the searching of PCKDO is minimized. Therefore, the CPU utilization can also be less in FAST and the users can pay less for using the cloud service. The main contributions of the proposed work are mentioned below:

1 A novel data access control model has been proposed in this paper. In the proposed FAST, the CSP maintains a table based on the data type of the DO and the popularity value of the DO for fast and efficient data accessing.
2 The proposed scheme minimizes the searching time of PCKDO, data accessing time and CPU utilization. Moreover, the proposed ACM also reduces the system overhead.
3 Theoretical analysis and experimental results of FAST have been presented for proving its efficiency.

The rest of the paper consists of several parts. Section 2 discusses the literature reviews. Problem statements are presented in section 3. Section 4 and section 5 deal with the detailed discussion of the proposed scheme and complexity analysis of the proposed algorithms, respectively. Section 6 presents results and discussions of the proposed access control model. At last, section 7 concludes the entire paper.

2 Literature reviews

Many ACMs are already proposed for efficient data accessing from the cloud server.

In RBAC [27] , every user or customer gets a separate role during data access. Here, the role is used for assigning access rights, and users cannot give access right to other users. In

RBAC, users have to satisfy a few rules: (i) rule authorization (iii) rule assignment, and (iii) transaction authorization to access any data. Transaction authorization guarantees that only the valid users are able to execute a transaction by confirming both the rule assignment and rule authorization. RBAC takes much time to provide a resource or data to the users.

Yu et al. [29] have proposed an Attribute Based Access Control (ABAC) model in which access is granted based on the user's attributes. Here, the user must confirm the attributes at the time of accessing the data. In ABAC, to assign a set of attributes, the CSP faces problems due to the large number of users, especially, when the user wants to access data from outside his/her domain. ABAC provides fine-grained data accessing, and it does not reveal the original content of data. In ABAC, setting a group or set of attributes is quite difficult. In addition, it takes much time to provide data to the users, so users have to pay more.

In [34], authors have extended the proposed concept of Attribute Set Based Encryption (ASBE) with a hierarchical structure and proposed Hierarchical Attribute Set Based Encryption (HASBE) scheme . In this scheme, users' access rights are expired after a predefined time. This scheme increases overhead because the workload of the higher-level trusted authority is transferred to the lower-level domain authority.

Mutual Trust Based Access Control Model (MTBACM) [35] has been proposed to achieve mutual trust among different entities of a cloud environment. This scheme uses the credibility of the CSP and user. Here, the relationship of trust is achieved between the CSP and users by using mutual trust technique. There are five entities in the system model of MTBACM: (i) authentication and authorization centre (ii) CSP (iii) users (iv) CSP's trust database, and (v) user's behaviour trust database. However, MTBACM takes much time to search a DO from the cloud database, and it also increases the system overhead.

In 2017, Alam et al. [36] have proposed a scheme, namely Cross Tenant Access Control (CTAC) scheme, where the cloud service provider plays the activity of the trusted third party and supports the sharing of resources between two different tenants. There are four algorithms in this scheme for delegation mechanism and permission activation between tenants: (i) activation (ii) forward revocation (iii) delegation, and (iv) backward revocation. This scheme does not support fast data access.

In a CC environment, data leakage issue is increasing because of sharing resources. To solve this issue, a novel mechanism was proposed by Almutairi et al. [33] for the multitenant cloud environment, namely Notion Based ACM (NBACM). NBACM introduces the notion of sensitivity at the data centre's end. NBACM is based on RBAC, and it is designed to minimize the risk of assigning the Virtual Machine (VM) for each task. However, this scheme does not support to minimize the searching time of DO.

Modified Hierarchical Attribute-Based Encryption (MHABE) model has been proposed in [37] for focusing data processing, data accessing and data storing in the multi-user cloud environment. Nevertheless, the searching time of DO is high in MHABE and the CPU utilization is more in this scheme. Therefore, users need to pay more for using the cloud services.

3 Problem statements

This section presents the system model and design goals of the proposed scheme.

3.1 System model

The proposed FAST is composed of three entities:

1 *Cloud service provider:* CSP is the central authority of any cloud environment. It has the overall power, and uses numerous servers to provide various kinds of services to the users and DOs [38].
2 *Data owners:* DOs are the entities, who store their data on the CC environment in the encrypted form. However, they are completely dependent on the CSP to manage or control their data.
3 *User:* Users are basically the end customers, who access resources or data or services from a CSP. Only the authorized users are allowed to access data or service from the cloud server and they depend on the CSP for infrastructure.

3.2 Design goals

The main design goals of FAST are:

1 Achieving a fast and effective ACM for the cloud environment by reducing three aspects: (i) searching time of the DO to provide the corresponding PCKDO (ii) data accessing time, and (iii) CPU utilization.
2 To provide different data access rights or certificates for different users for the same data. It improves data security in the cloud environment.
3 To minimize the system overhead of the cloud environment by not keeping the DO always connected during the entire data communication process.

4 Proposed model

In the proposed FAST, the CSP maintains a temporary CSP Table (CSP-TAB). In the CSP-TAB, there are six attributes:

(i) Group Identity (ID) (GP_ID) (ii) Data Type (DT) (iii) Popularity Value (PV) (iv) Data Owner's ID (DO_ID) (v) DO's Time & Date (DO_T&D), and (vi) Group's Time & Date (GP_T&D). The identities of all the groups are maintained in the GP_ID. DT keeps the data types of the DOs. PV maintains the popularity value of each and every DO. When a DO provides data to the user, the PV of that DO is increased by 1. In FAST, threshold popularity value is considered as 10 i.e. the PV of any DO can be between 1 and 10. The data type and threshold PV of the DO are decided by the CSP as per convenience. There may be many DOs, who share the same type of data. So, they are kept in the same DO_ID. Respective date and time of the DO are kept in the DO_T&D i.e. when s/he has shared the requested data. GP_T&D holds the formation details of the groups. Whenever the DO shares data on the cloud server, the CSP adds the DO in the table based on the data type. At the time of data access, the CSP searches the DO from the CSP-TAB based on the user's requested data type and PV. Then, the CSP increases the PV of the DO by 1, if the DO already exists in the table and the PV of the particular DO has not reached to the threshold value. Otherwise, a new GP_ID is created with the details of the DO. Then, the CSP gives the PCKDO to the user. So, the DO having the threshold PV of a particular data type is placed at the top of the DO_ID, and the CSP doesn't need to look the entire database for providing the PCKDO. The user can send a request in lees time to the DO in order to get the credentials by using the PCKDO. Thus, the CSP-TAB can help to reduce the searching time of a DO as well as the data access time.

The proposed scheme consists of mainly five phases: (i) system setup (ii) registration (iii) login (iv) processes on the CSP-TAB, and (v) data access. The entire workflow of FAST has been shown in Fig. 1.

4.1 System setup phase

In this phase, the CSP chooses a big prime number (p) to calculate a multiplicative group Z_p^*. This multiplicative group is used to choose the Public-Private Key Pair (PPKP) of the CSP. The CSP also chooses PPKP for the DO and user from the Z_p^* that are sent to them during their registration. In FAST, the Public Key of the User (PCKUSR) is accessible to all the registered entities of the cloud server, and only the registered user knows the respective PCKDO after sending a data access request. All the registered users and DOs know the Public Key of the CSP (PCKCSP), and the private key of the CSP and other entities are always kept secret.

4.2 Registration phase

Every user must be registered in the cloud server by sending a registration request to the CSP. Then, the CSP uses some confidential information of the user, namely date of birth, address, full name and first school name to generate a personal profile of the user. Then, the Secure Socket Layer (SSL) is developed to provide the PPKP of the user. The DOs are also registered in the cloud server by the same process.

4.3 Login phase

When the users complete the registration phase, the CSP allows them to login to the cloud server. To login to the cloud server, the user must send a request to the CSP. The CSP verifies, if the user is authorized or not. If the user is authorized, the CSP accepts the login request and sends a reply to

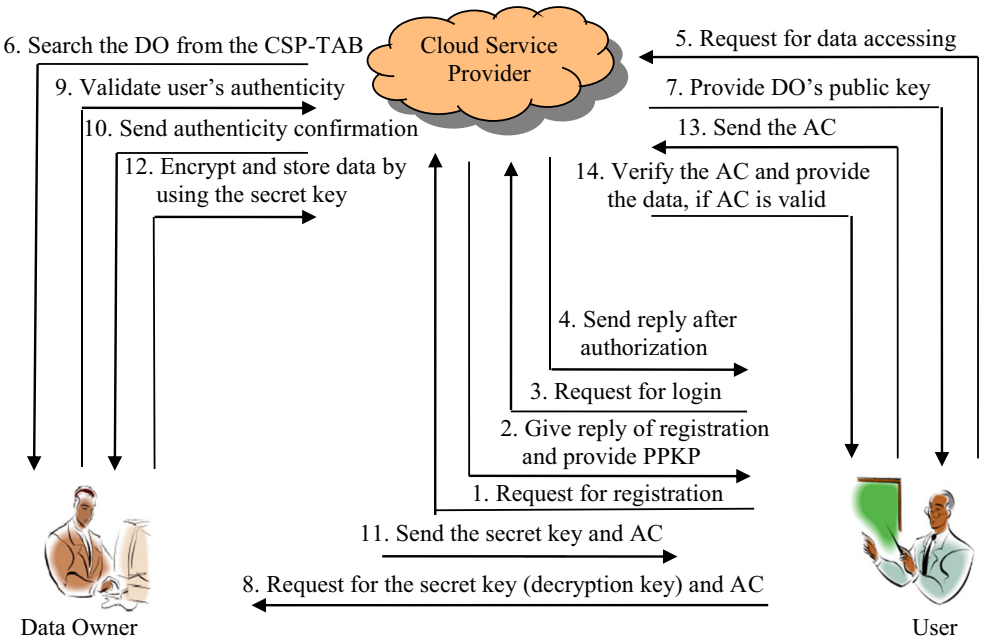

Fig. 1 Workflow of the proposed FAST

the user. Then, the user can send a data access request to the CSP.

4.4 Processes on the CSP-TAB phase

In the CSP-TAB, DOs' IDs are placed based on the popularity values. The DO, who has the threshold PV is kept at the top position of the respective DO_ID. Then, in the next position, the DO is placed, who has the next PV and so on. There are 3 main operations on the CSP-TAB: (i) insertion (ii) deletion, and (iii) searching a PCKDO. Let us assume that GP_ID is defined by $S = \{GID_1, GID_2, GID_3, \ldots, GID_P\}$ and DO_ID is defined by $S' = \{DID_1, DID_2, DID_3, \ldots, DID_Q\}$. In the CSP-TAB, P and Q denote the maximum number of GP_ID and the maximum number of DOs' IDs per DO_ID, respectively. Here, GID_x represents x^{th} GP_ID and DID_x represents x^{th} DO's ID. Table 1 demonstrates an example of a CSP-TAB.

4.4.1 Insertion algorithm

The insertion algorithm is used to insert the ID of a DO in the CSP-TAB. At first, the DO must be registered in the cloud server. Then, the DO needs to login to the cloud server to share any data. When the DO shares data, to inset the DO in the CSP-TAB, at first, the CSP checks whether the data type of the DO exists in the CSP-TAB or not. If the type of data is already existed in the CSP-TAB, the CSP checks whether the respective DO_ID field is full or not. If it is not full, the CSP adds the DO in the corresponding position of the CSP-TAB. Otherwise, a new group is added to the CSP-TAB with the details of the DO, if the CSP-TAB is not full. When DO_ID and CSP-TAB is full, the CSP deletes a DO's ID from DO_ID and a group from CSP-TAB, respectively based on the condition to add the details of the DO. After adding the DO in the corresponding position, the CSP sets the PV of the DO as 1. Table 2 presents the notations and their descriptions, which are used throughout this paper.

Algorithm to Insert the ID of a DO in the CSP-TAB

Input: ID of an authorized DO.
Output: Updated CSP-TAB.

```
1  if D == Authorized DO's ID
2     c0 ← P,  c1 ← Q, m[ ]
3        if c0 ≠ FULL
4              for all GID_x ∈ S
5                  if D_DT == GID_x_DT
6                       for all DID_x ∈ S' in GID_x
7                           if c1 ≠ FULL
8                                GID_x_DO_ID ← GID_x_DO_ID ∪ D
9                                D_PV ← 1
10                          else
11                               DEL_DO_ID (GP_ID)
12                               GID_x_DO_ID ← GID_x_DO_ID ∪ D
13                               D_PV ← 1
14                      end for
15             end for
16       else
17             r ← DEL_GP_ID()
18             GID_r_DO_ID ← D
19             GID_r_DT ← D_DT
20             GID_r_PV ← 1
21             GID_r_DO_T&D ← D_T&D
22             GID_r_GP_T&D ← D_T&D
23             UPDATE CSP-TAB
24 else
25    STOP
```

Table 1 CSP-TAB

Group ID (GP_ID)	Data Owner's ID (DO_ID)	Popularity Value (PV)	Data Type (DT)	DO's Time & Date (DO_T&D)		Group's Time & Date (GP_T&D)	
1	198	10	MP3	06:08	16-09-2015	04:59	22-12-2011
	033	9		04:59	22-12-2011		
	612	8		17:46	03-07-2014		
	738	2		20:16	28-11-2012		
	071	2		10:32	29-05-2015		
2	687	9	DOCX	03:36	13-11-2014	22:49	13-12-2007
	003	9		13:09	27-10-2012		
	712	8		01:19	21-08-2014		
	001	6		23:46	09-07-2012		
	342	3		22:49	13-12-2007		
.	.		.	.		.	
	.	.		.			
	.	.		.			
	.	.		.			
	.	.		.			
50	546	6	JPG	09:30	22-07-2010	09:30	22-07-2010
	613	5		17:41	03-02-2013		
	561	4		12:42	12-06-2012		
	069	3		11:57	25-11-2010		
	149	1		15:23	18-03-2013		

4.4.2 Deletion

The deletion operation is controlled by the CSP based on the requirement. An entire GP_ID or a particular DO's ID can be deleted from the DO_ID to manage the entries of the CSP-TAB. Here, both the deletions are actually parts of the aforementioned insertion algorithm. Therefore, an authorized DO's ID is the input of both the deletion algorithms.

1 *DO's ID deletion algorithm (**DEL _ DO _ ID** (**GP _ ID**)):* The CSP can delete the ID of a DO from the DO_ID after traversing GP_ID and the respective DO_ID. After finding the DO having the lowest PV, the DO's ID is assigned to an array *m*[ ], and finally, DO's ID is deleted from DO_ID. If the CSP finds that there are two or more DOs, who have the same lowest PV, the CSP follows the Least Recently Used (LRU) principle to delete the DO's ID from DO_ID. The CSP gets the information of LRU from DO_T&D. Then, the CSP updates the CSP-TAB.

2 *Entire GP_ID deletion algorithm* (***DEL _ GP _ ID***()): The CSP also uses the LRU principle to delete a long time unused GP_ID from the CSP-TAB. At first, the details of the LRU GP_ID have been searched from GP_T&D, then, the CSP deletes the entire entries of GID_x and updates the table.

Algorithm (*DEL_DO_ID* (*GP_ID*)) to Delete the ID of a DO from the CSP-TAB

Input: ID of an authorized DO.
Output: Updated CSP-TAB.

1 $m[\,] \leftarrow SEARCH_{MIN}\ (DID_{x_{PV}})$
2 if $count(m) > 1$
3 $\quad SEARCH_{LRU}\ (DID_x)$
4 $\quad DO\_ID_{NEW} \leftarrow DO\_ID_{PREV} - DID_x$
5 else
6 $\quad DO\_ID_{NEW} \leftarrow DO\_ID_{PREV} - DID_x$
7 UPDATE CSP-TAB
8 STOP

Algorithm ($DEL\_GP\_ID()$) to Delete a GP_ID from the CSP-TAB	
Input: ID of an authorized DO. **Output:** Updated CSP-TAB.	
1	$SEARCH_{LRU}\ (GID_x)$
2	DELETE entries of GID_x
3	UPDATE CSP-TAB
4	STOP

4.4.3 Searching the public key of a DO

After login to the cloud server, when a user requests to access any resource or data, the first task of the CSP is to search the PCKDO from the cloud server and send it to the user. In the proposed FAST, to search a PCKDO, the CSP matches the U_{DT} against the $GID_{x_{DT}}$. Then, again the CSP searches each entry of the corresponding DO_ID. As the DOs' IDs are systematically organized by using the popularity value, the CSP can simply find the PCKDO from the corresponding position of the CSP-TAB. Then, the DO provides it to the user in less time. Otherwise, the CSP shows an invalid data type.

Algorithm to search a PCKDO from the CSP-TAB	
Input: ID of an authorized user. **Output:** Search PCKDO and provide it to the user.	
1	if $U ==$ Authorized user's ID
2	for $i = 1$ to P
3	if $U_{DT} == GID_{x_{DT}}$
4	for $j = 1$ to Q
5	SEARCH DO from $GID_{x_{DO\_ID}}$
6	PROVIDE PCKDO to U
7	if $D_{PV} \neq Threshold$
8	$D_{PV} = D_{PV} + 1$
9	end for
10	else
11	Show INVALID data type
12	end for
13	else
14	STOP

4.5 Data access phase

There are the following steps to access any resource or data from the cloud server:

Step 1: A registration request is sent from the user to the CSP.
Step 2: The CSP uses the user's information to generate a corresponding profile. Then, the CSP sends a registration reply along with PPKP.
Step 3: The user can login after a successful registration process i.e. after getting the registration reply.
Step 4: In the fourth step, if the user is authorized to login, the CSP sends an acknowledgment.
Step 5: The user sends a data access request to the CSP in the encrypted form by using the Private Key of the User (PRKUSR) and PCKCSP.
Step 6: The CSP uses the CSP-TAB to search the corresponding PCKDO in the sixth step.
Step 7: The CSP provides the PCKDO to the requested user after encrypting by the Private Key of the CSP (PRKCSP) and PCKUSR. The user decrypts the message by using the PRKUSR and PCKCSP.

Table 2 Notations and their descriptions

Notation	Description	Notation	Description
EK_{SKD}	Encryption by using SKD	$c0, c1, r, i, j$	Variables
DK_{SKD}	Decryption by using SKD	$GID_{x_{DT}}$	Data type of x^{th} GP_ID
EK_{PRKDO}	Encryption by using PRKDO	D_{DT}	Data type of authorized DO
DK_{PCKDO}	Decryption by using PCKDO	D_{PV}	PV of authorized DO
EK_{PCKCSP}	Encryption by using PCKCSP	$GID_{x_{DO\_ID}}$	DO_ID of x^{th} GP_ID
EK_{PRKCSP}	Encryption by using PRKCSP	$GID_{r_{DO\_ID}}$	DO_ID of r^{th} GP_ID
DK_{PRKCSP}	Decryption by using PRKCSP	$GID_{r_{DT}}$	Data type of r^{th} GP_ID
DK_{PCKCSP}	Decryption by using PCKCSP	$GID_{r_{PV}}$	PV of r^{th} GP_ID
EK_{PRKUSR}	Encryption by using PRKUSR	U_{DT}	User's requested data type
EK_{PCKUSR}	Encryption by using PCKUSR	$D_{T\&D}$	Time and date of authorized DO
DK_{PRKUSR}	Decryption by using PRKUSR	$GID_{r_{DO\_T\&D}}$	DO_T&D of r^{th} GP_ID
DK_{PCKUSR}	Decryption by using PCKUSR	$GID_{r_{GP\_T\&D}}$	GP_T&D of r^{th} GP_ID
DO_ID_{NEW}	DO_ID after processing	$DID_{x_{PV}}$	PV of x^{th} DO_ID
DO_ID_{PREV}	DO_ID before processing	$SEARCH_{MIN}\ (DID_{x_{PV}})$	Search DID_x having minimum PV
$count(m)$	A function to calculate the number of DOs' IDs in m [ ]	$SEARCH_{LRU}\ (DID_x)$	Search DID_x based on LRU
		$SEARCH_{LRU}\ (GID_x)$	Search GID_x based on LRU

$$PCKDO \rightarrow EK_{PRKCSP}(PCKDO) \rightarrow EK_{PCKUSR}\{EK_{PRKCSP}(PCKDO)\} \rightarrow User$$

$$PCKDO \leftarrow DK_{PCKCSP}\{EK_{PRKCSP}(PCKDO)\} \leftarrow DK_{PRKUSR}[EK_{PCKUSR}\{EK_{PRKCSP}(PCKDO)\}]$$

Step 8: In this step, the user sends a request to the corresponding DO by using the PCKDO. The request is sent in the encrypted form by the PRKUSR and PCKDO to provide the Secret Key of the Data (SKD) and corresponding AC of the data.

Step 9: The DO checks the authorization of the user from the cloud server's end.

Step 10: The CSP confirms the user's authorization in the tenth step.

Step 11: In this step, the DO sends the SKD along with its corresponding AC to the user as an encrypted message by the Private Key of the DO (PRKDO) and PCKUSR. Here, the DO generates different ACs for different users for the same data to improve data security. Thus, two ACs cannot be the same in any condition. The user uses the PRKUSR and PCKDO to get the SKD and AC.

$$SKD\ and\ AC \rightarrow EK_{PRKDO}(SKD\ and\ AC) \rightarrow EK_{PCKUSR}\{EK_{PRKDO}(SKD\ and\ AC)\} \rightarrow User$$

$$SKD\ and\ AC \leftarrow DK_{PCKDO}\{EK_{PRKDO}(SKD\ and\ AC)\} \leftarrow DK_{PRKUSR}[EK_{PCKUSR}\{EK_{PRKDO}(SKD\ and\ AC)\}]$$

Step 12: Here, the DO encrypts the requested resource or data as a message by the SKD. The DO then uses the PRKDO and PCKCSP to encrypt the message and the AC of the corresponding data before storing it on the

Table 3 Complexities of the proposed algorithms in the worst case

Sl. No.	Algorithm	Complexity
1	Insertion	$\mathcal{O}(PQ)$
2	DO_ID deletion	$\mathcal{O}(PQ)$
	GP_ID deletion	$\mathcal{O}(P)$
3	Search a PCKDO	$\mathcal{O}(PQ)$

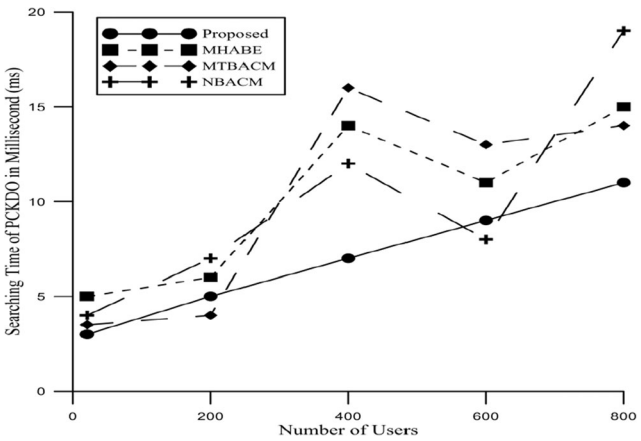

Fig. 2 Searching time of the public key of a DO vs. number of users

cloud database. The CSP is only able to moderately decrypt the encrypted message by the PRKCSP and PCKDO.

$$Data \rightarrow EK_{SKD}(Data) \rightarrow EK_{PRKDO}\{EK_{SKD}(Data)\ and\ AC\} \rightarrow EK_{PCKCSP}[EK_{PRKDO}\{EK_{SKD}(Data)\ and\ AC\}] \rightarrow CSP$$

$$EK_{SKD}(Data)\ and\ AC \leftarrow DK_{PCKDO}[EK_{PRKDO}\{EK_{SKD}(Data)\ and\ AC\}] \leftarrow DK_{PRKCSP}[EK_{PCKCSP}[EK_{PRKDO}\{EK_{SKD}(Data)\ and\ AC\}]]$$

Step 13: In the thirteenth step, the user sends the AC to the CSP by using the PRKUSR and PCKCSP. The CSP decrypts the encrypted message by the PRKCSP and PCKUSR.

$$AC \rightarrow EK_{PRKUSR}(AC) \rightarrow EK_{PCKCSP}EK_{PRKUSR}(AC) \rightarrow CSP$$

$$AC \leftarrow DK_{PCKUSR}\{EK_{PRKUSR}(AC)\} \leftarrow DK_{PRKCSP}[EK_{PCKCSP}\{EK_{PRKUSR}(AC)\}]$$

Step 14: In this fourteenth step, the CSP verifies the AC of the user against the AC provided by the DO. If the AC is authentic, the CSP provides the ciphertext of the resource or data in the encrypting form by using the PRKCSP and PCKUSR. Otherwise, the user's data access request is rejected by the CSP.

$$EK_{SKD}(Data) \rightarrow EK_{PRKCSP}\{EK_{SKD}(Data)\} \rightarrow EK_{PCKUSR}[EK_{PRKCSP}\{EK_{SKD}(Data)\}] \rightarrow User$$

The user executes three decryption processes by using the PRKUSR, PCKCSP and SKD to retrieve the original resource or data.

$$Data \leftarrow DK_{SKD}\{EK_{SKD}(Data)\} \leftarrow DK_{PCKCSP}[EK_{PRKCSP}\{EK_{SKD}(Data)\}] \leftarrow DK_{PRKUSR}[EK_{PCKUSR}[EK_{PRKCSP}\{EK_{SKD}(Data)\}]]$$

5 Complexity analysis

There are mainly three algorithms in the proposed scheme, which are discussed in the previous section. Therefore, complexity has been calculated for these three algorithms:

1 *Insertion:* To insert the ID of a DO in DO_ID of the CSP-TAB, the CSP needs to consider two aspects: (i) data type of the DO, and (ii) popularity value of the DO. Here, the CSP matches the data type of the DO against the data type of the CSP-TAB. If the matched data type is not found after searching P number of groups, the CSP adds the DO in a newly created group. Otherwise, in the worst case, the CSP searches all P number of groups, and in the matched GP_ID, the CSP again needs to search Q number of DO IDs to insert a DO's ID. When the CSP adds the DO in the CSP-TAB, the PV of the DO is set as 1. Thus, the worst-case complexity of insertion algorithm is $\mathcal{O}(PQ)$.
2 *Deletion:* When the CSP deletes a group, it needs to find the LRU GP_ID. In the worst case, the CSP needs to search P number of groups to find the LRU GP_ID. So, the worst-case complexity is $\mathcal{O}(P)$. To delete a DO's ID, the CSP needs to search all the Q entries of the DO_ID in the worst case. As the deletion algorithms are actually parts of the insertion algorithm, the CSP may need to delete the DO's ID of the P^{th} GP_ID in the worst case. Thus, the worst-case complexity to delete a DO's ID is $\mathcal{O}(PQ)$.
3 *Searching the public key of a DO:* Searching a PCKDO from the CSP-TAB is the main algorithm of the proposed access control model. To search a PCKDO from the CSP-TAB, the CSP uses the requested data type of the user. If it is matched with the DT of the CSP-TAB, the CSP again requires searching the entries of the corresponding DO_ID. Therefore, the CSP searches P number of groups and its corresponding Q number of DOs' IDs to find a PCKDO in the worst case. After finding the PCKDO, it is sent to the user. Then, the CSP increases the PV of the DO by 1, if the PV is not reached the threshold value. Thus, the worst-case complexity of searching a PCKDO is $\mathcal{O}(PQ)$. Complexities of the proposed algorithms in the worst case are shown in Table 3.

6 Results and discussions

Results and discussions of the proposed FAST have been presented in this section in detail.

6.1 Simulation setting

The proposed FAST has been evaluated by using a cloud simulation environment. Here, CloudSim 3.0.3 is used to develop a simulation environment to evaluate efficiency [39]. CloudSim 3.0.3 has been setup on a Dell Optiplex 7050 MT desktop with the following configurations:

1 Storage capacity: 2 TB
2 RAM: 16 GB
3 Operating system: Windows 10
4 Processor: Corei5

CloudSim is used to configure Apache Commons Math 3.6.1 [40], and Java version 8 [41] is also installed on the

above mentioned Dell desktop. Forty data centres have been used to develop a heterogeneous CC environment. The memory capacity of the heterogeneous cloud environment is 4 GB, and there are 4000 physical nodes. Four types of VMs are considered in the simulation with 1 GB/s bandwidth:

1 1500MIPS, 2.5 GB
2 2000MIPS, 3 GB
3 2500MIPS, 3.5 GB
4 3000MIPS, 4 GB

6.2 Results and discussions

The proposed scheme i.e. FAST is evaluated against three existing schemes: (i) notion based access control model [33] (ii) mutual trust based access control model [35], and (iii) modified hierarchical attribute-based encryption scheme [37]. 200 experiments are performed with different criterion to get the searching time of PCKDO, data accessing time and percentage of CPU utilization. At last, average values are calculated from 200 experiments. CityPulse Dataset Collection [42] is used to collect pollution dataset for the experiments.

The first experiment is for calculating the searching time of the public key of a DO from the cloud database. Figure 2 shows that the proposed FAST gives better results to search the PCKDO than three other existing schemes: MTBACM, NBACM and MHABE. In FAST, when the CSP gets an access request for data, the time is recorded. The CSP searches the DO from the CSP-TAB by using the user's requested data type and popularity value. The CSP matches the user's requested data type with the entries of the CSP-TAB i.e. data type of the CSP-TAB. As the CSP organizes the DOs based on their popularity values, the respective DO is searched based on the top-down approach in the particular DO_ID field. Then, the CSP easily gets the corresponding public key of the DO, and after encrypting it by the CSP's private key and the public key of the user, it is sent to the user in less time. Then, again the time is recorded. Thus, the CSP does not require finding the entire cloud database for a single DO. However, if the number of users increases, the load on the CSP also increases and it consumes a little bit of time to search the PCKDO, which produces a linearly increasing curve for the proposed FAST. In the existing schemes, namely MHABE, MTBACM and NBACM, the CSP stores the details of the DOs in a scattered matter. These schemes do not maintain any technique or CSP-TAB to store the details of the DOs. Sometimes, the CSP searches the entire database to find a DO, and sometimes, the CSP does not require searching the entire database to find a DO, which consumes less time. Thus, zig-zag curves are generated for the existing schemes.

The next result is to validate the data accessing time. The data accessing time is the time interval between the data request time and the data decryption time. In FAST, the CSP easily searches the PCKDO from the CSP-TAB in less time by using the popularity value of the DO and the requested data type. Here, the users do not wait to get the corresponding PCKDO, and by using the PCKDO, the users initiate requests to the data owners for the respective secret keys and ACs. As the searching time of the PCKDO is less in the proposed access control model, the data accessing time is automatically reduced. Thus, in Fig. 3, a linearly increasing curve is generated for the proposed ACM. In all the existing schemes, the CSP sometimes searches the entire cloud database to find the PCKDO, and as result, the time to search the PCKDO is increased and the users need to wait to send the requests to the corresponding data owners for the secret keys and access certificates. Thus, the data accessing time is much high for MHABE, MTBACM and NBACM.

The last experiment validates the efficiency of CPU utilization. The proposed ACM does not take more time to search the PCKDO and to provide the data to the users. Here, the CSP instantly responses to the users without consuming time after receiving the data access requests. As both the searching time of the PCKDO and the data accessing time are minimized in the proposed data access control model, the CPU utilization is also less. As the number of data accessing requests is proportional to the number of users, the data access requests are automatically increased, when there are numerous users in the cloud server. Thus, the CPU utilization is linearly increased in FAST, which can be seen in Fig. 4. Attribute-based encryption and hierarchical identity-based encryption are combined in MHABE to achieve efficient data access control. MHABE consists of many complex operations because of combining both the schemes, which result high CPU utilization. In MTBACM, many communications are required among the five entities to complete a data

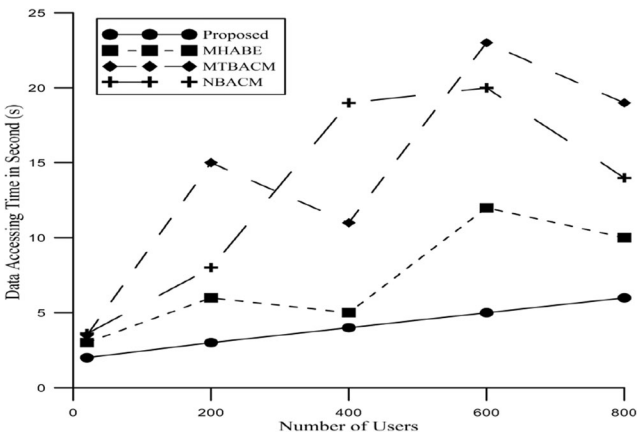

Fig. 3 Data accessing time vs. number of users

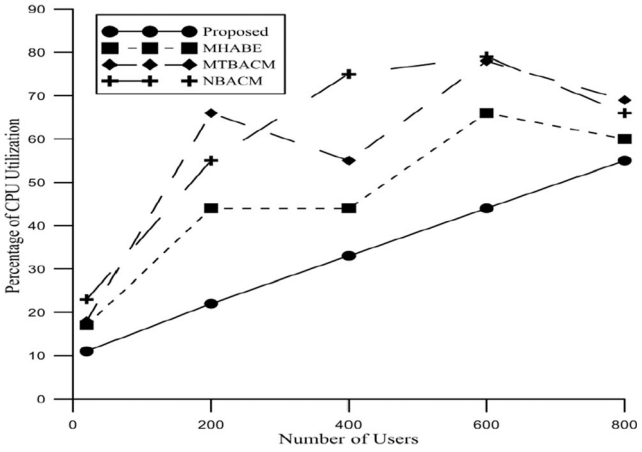

Fig. 4 Percentage of CPU utilization vs. number of users

communication process. In NBACM, the notion of sensitivity is considered for data accessing i.e. degree of data sharing among many tenants that consumes much time. In addition, as all these existing schemes take more time to deliver the requested data to the users, the CPU utilization is automatically high. The CPU utilization has been considered per 5 s, which involves 12 values/min. At last, the average value is calculated from all the values. To calculate the CPU utilization, the following Eq. (1) has been used:

$$U_n = \frac{\sum_{i=1}^{n} U_i(n)}{n} \tag{1}$$

where, n is the full utilization of CPU for each experiment. So, $n \in 1, 2, 3, \ldots$

$\sum_{i=1}^{n} U_i(n)$ is the total CPU usage.

7 Conclusions and future works

In a cloud environment, access control is one of the major problems, where users need to pay more for using cloud services. In this paper, a novel fast and efficient access control model has been introduced. In the proposed access control model, the CSP maintains a temporary table based on the data types of the data owners and the popularity values of the data owners. Here, the DO must be online to provide the secret key and access certificate to the user. Then, the DO needs not to be online and can go offline, which reduces the system overhead. The results of many experiments demonstrate that the proposed access control model is much better than the other existing schemes in the cloud computing environment. The authentication and identity management technique of the cloud environment can be improved in the future for both the data owner and user.

References

1. Huth A, Chebula J (2011) The basics of cloud computing,. Carnegie Mellon University
2. Namasudra S, Nath S, Majumder A (2014) Profile based access control model in cloud computing environment. In: *Proceedings of the International Conference on Green Computing,* Communication and Electrical Engineering, IEEE, Coimbatore, India, pp. 1–5
3. Li S, Wang G, Yang J (2019) Survey on cloud model based similarity measure of uncertain concepts. CAAI Transactions on Intelligence Technology 4(4):223–230
4. Namasudra S, Roy P (2017) A new table based protocol for data accessing in cloud computing. J Inf Sci Eng 33(3):585–609
5. Gao X, Jiang Z, Jiang R. A novel data access scheme in cloud computing. In *Proceedings of the 2nd International Conference on Computer and Information Applications*, 2012, pp. 124–127
6. Namasudra S (2019) An improved attribute-based encryption technique towards the data security in cloud computing. Concurrency and Computation: Practice and Exercise. https://doi.org/10.1002/cpe.4364
7. Sarkar M, Saha K, Namasudra S, Roy P (2015) An efficient and time saving web service based android application. SSRG Int J Comp Sci Eng 2(8):18–21
8. Namasudra S, Roy P, Balamurugan B (2017) Cloud computing: fundamentals and research issues. In: *Proceedings of the 2nd International Conference on Recent Trends and Challenges in Computational Models*, IEEE, Tindivanam, India
9. Namasudra S, Deka GC, Johri P, Hosseinpour M, Gandomi AH (2020) The revolution of blockchain: state-of-the-art and research challenges. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-020-09426-0
10. Namasudra S (2018) Taxonomy of DNA-based security models. In: Namasudra S, Deka GC (Eds) Advances of DNA Computing in Cryptography. Taylor & Francis, pp. 53–68
11. Namasudra S, Roy P, Balamurugan B, Vijayakumar P (2017) Data accessing based on the popularity value for cloud computing. In *Proceedings of the International Conference on Innovations in Information,* Embedded and Communications Systems (ICIIECS), IEEE, Coimbatore, India
12. Alguliyev RM, Aliguliyev RM, Sukhostat LV (2020) Efficient algorithm for big data clustering on single machine. CAAI Trans Intelligence Tech 5(1):9–14
13. Namasudra S, Roy P, Vijayakumar P, Audithan S, Balamurugan B (2017) Time efficient secure DNA based access control model for cloud computing environment. Futur Gener Comput Syst 73:90–105
14. Namasudra S, Devi D, Kadry S, Sundarasekar R, Shanthini A (2020) Towards DNA based data security in the cloud computing environment. Comput Commun 151:539–547
15. Namasudra S (2020) Data access control in the cloud computing environment for bioinformatics. International Journal of Applied Research in Bioinformatics (IJARB) (in press)
16. Namasudra S, Deka GC (2018) Advances of DNA computing in cryptography. Taylor & Francis
17. Namasudra S, Devi D, Choudhary S, Patan R, Kallam S (2018) Security, privacy, trust, and anonymity. In Advances of DNA Computing in Cryptography, S. Namasudra and G. C. Deka, Eds., Taylor & Francis, pp. 153–166
18. Namasudra S, Chakraborty R, Majumder A, Moparthi NR (2020) Securing multimedia by using DNA based encryption in the cloud computing environment. ACM Transactions on Multimedia Computing, Communications, and Application (in press)
19. Namasudra S, Deka GC (2018) Introduction of DNA computing in cryptography. In Advances of DNA Computing in Cryptography, S. Namasudra and G. C. Deka, Eds., Taylor & Francis, pp. 27–34

20. Namasudra S, Roy P (2018) PpBAC: popularity based access control model for cloud computing. Journal of Organizational and End User Computing 30(4):14–31
21. Zhao X, Li R, Zuo X (2019) Advances on QoS-aware web service selection and composition with nature-inspired computing. CAAI Transactions on Intelligence Technology 4(3):159–174
22. Namasudra S, Deka GC, Bali R (2018) Applications and future trends of DNA computing. In: Namasudra S, Deka GC (Eds) Advances of DNA Computing in Cryptography. Taylor & Francis
23. Devi D, Namasudra S, Kadry S (2020) A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. Int J Data Warehouse Min 16(3). https://doi.org/10.4018/IJDWM.2020070104
24. Namasudra S, Roy P (2017) Time saving protocol for data accessing in cloud computing. IET Commun 11(10)
25. Ferraiolo DF, Kuhn DR (1992) Role-based access controls. In Proceedings of the 15th National Computer Security Conference, Baltimore, USA, pp. 554–563
26. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, New York, USA, pp. 89–98
27. Yu S, Wang C, Ren K, Lou W (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE INFOCOM, San Diego, CA, pp. 1–9
28. Ajgaonkar S, Indalkar H, Jeswani J (2015) Activity based access control model for cloud computing. Int J Curr Eng Techn 5(2):708–713
29. Raghavendra S, Meghana K, Doddabasappa PA, Geeta CM, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM (2016) Index generation and secure multi-user access control over an encrypted cloud data. Procedia Comp Sci 89:293–300
30. Murugan A, Thilagavathy R (2018) Cloud storage security scheme using DNA computing with Morse code and zigzag pattern. In Proceedings of the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017), IEEE, pp. 2263–2268
31. Almutairi A, Sarfraz MI, Ghafoo A (2018) Risk-aware management of virtual resources in access controlled service-oriented cloud datacenters. IEEE Trans Cloud Computing 6(1):168–181
32. Wan Z, Liu J, Deng RH (2012) HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans Inf Forensics Sec 7(2):743–754
33. Lin G, Wang D, Bie Y, Lei M (2014) MTBAC: a mutual trust based access control model in cloud computing. China Comm 11(4):154–162
34. Alam Q, Malik SUR, Akhunzada A, Choo KKR, Tabbasum S, Alam M (2017) A cross tenant access control (CTAC) model for cloud computing: formal specification and verification. IEEE Trans Inf Forensics Sec 12(6):1259–1268
35. Xie Y, Wen H, Wu B, Jiang Y, Meng J (2019) A modified hierarchical attribute-based encryption access control method for mobile cloud computing. IEEE Trans Cloud Computing 7(2):383–391
36. Namasudra S (2018) Cloud computing: a new era. J Fund Appl Sci 10(2):113–135
37. Calheiros RN, Ranjan R, Beloglazov A (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software-Pract Exp (SPE) 41(1):23–50
38. (2020) Apache commons math. Available: http://commons.apache.org/proper/commons-math/download_math.cgi. Accessed on 13 Jan 2020
39. (2020) Java. Available: http://java.com/en/download/index.jsp. Accessed on 13 Jan 2020
40. (2020) CityPulse dataset collection. Available: http://iot.ee.surrey.ac.uk:8080/datasets.html. Accessed 13 Jan 2020
41. Tripura J, Roy P (2017) Flow forecasting in multiple sections of a river system. KSCE J Civ Eng 21(2):512–522
42. Tripura J, Roy P, Barbhuiya AK (2018) Application of RBFNNs incorporating MIMO processes for simultaneous river flow forecasting. J Eng Technol 50(3):434–449

Dr. Suyel Namasudra is an Assistant Professor in the Department of Computer Science and Engineering at the National Institute of Technology Patna, Bihar, India. Prior to joining the National Institute of Technology Patna, Dr. Namasudra was an Assistant Professor in the Department of Computer Science Engineering at the Bennett University, India. He has received PhD in Computer Science and Engineering from National Institute of Technology Silchar, Assam, India. His research interests include Cloud Computing, Information Security, DNA Computing and Blockchain. Dr. Namasudra has edited 1 book and 25 publications in refereed journals, book chapters and conference proceedings. He has participated in many international conferences as an Organizer and Session Chair. Dr. Namasudra is a member of the Editorial Board and Reviewer of many journals.

Dr. Rupak Chakraborty is an Assistant Professor of Department of Computer Science and Engineering at Guru Nanak Institute of Technology, Kolkata, India. He has obtained Ph.D. degree in Computer Science and Engineering from DIT University, Dehradun, India in 2020. Dr. Chakraborty has total research and teaching experience of about 8 years in the field of Computer Science and Engineering. His research interests include image processing, evolutionary computing, machine learning and cloud computing. He has published many research papers in refereed journals and renowned conferences. Dr. Chakraborty also serves as a reviewer for many journals.

Dr. Seifedine Kadry is a Professor of Data Science at Beirut ArabUniversity, Lebanon. He has a Bachelor degree in applied mathematics in1999 from Lebanese University, MS degree in computation in 2002 fromReims University (France) and EPFL (Lausanne), PhD in 2007 from BlaisePascal University (France), HDR degree in engineering science in 2017from Rouen University. At present his research focuses on education usingtechnology, system simulation, operation research, system prognostics,stochastic systems, and probability and reliability analysis. He is an ABET program evaluator,ACBSP program reviewer, NTCM program reviewer. He is an IET Fellow.

Dr. Gunasekaran Manogaran is currently working as a Big DataScientist in University of California, Davis, USA. He is also an AdjunctAssistant Professor, Department of Computer Science & InformationEngineering, Asia University, Taiwan and Adjunct Faculty, in School ofComputing, SRM Institute of Science and Technology, Kattankulathur,India. He is a visiting researcher/scientist in University of La Frontera, Colombia andInternational University of La Rioja, Spain. He received his Ph.D. from the Vellore Institute ofTechnology University, India. He received his Bachelor of Engineering and Master ofTechnology from Anna University, India and Vellore Institute of Technology University, Indiarespectively. He is the author/co-author of more than 100 papers in conferences, book chaptersand journals including IEEE Transactions on Industrial Informatics, IEEE Transactions onComputational Social Systems, IEEE Internet of Things, IEEE Intelligent System, IEEE Access,ACM Transactions on Multimedia Computing, Communications, and Applications. He iscurrently serving as an Associate Editor in Ambient Intelligence & Humanized Computing(Springer), International Journal of Automation and Computing (Springer), Data in Brief(Elsevier) and International Journal of Interactive Multimedia and Artificial Intelligence. He isone of the Advisory Board Member of Information System (Elsevier) and an Editorial BoardMember of International Journal of Computer Applications in Technology. He also appointed asInternet of Things Section Editor in Sensors (MDPI).