JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 585-609 (2017)
DOI: 10.6688/JISE.2017.33.3.1

A New Table Based Protocol for Data Accessing in Cloud Computing

SUYEL NAMASUDRA AND PINKI ROY
Department of Computer Science and Engineering
National Institute of Technology, Silchar
Assam, 788010 India
E-mail: suyelnamasudra@gmail.com

Cloud computing is very lucrative technology because of its cost effectiveness, efficiency, flexibility, pay-per-use and scalability. With these advantages, cloud computing has many issues, and some existing issues have become very critical. Access control and security are two critical issues of cloud computing. Access control is a method, which allows a customer or user to access a data, file or any kind of resources from a system. In this paper, a new table based access control model has been proposed for cloud computing environment. The proposed scheme is very efficient because it can minimize many problems, such as high data accessing time, high searching time for providing the public key of the data owner, maintenance of the database, *etc.* Both the performance analysis and experimental results prove that the proposed scheme is more efficient and effective than the existing schemes.

Keywords: cloud computing, cloud service provider, data owner, encryption, decryption, CloudSim

1. INTRODUCTION

In Information Technology (IT) sector, uses of cloud computing are gradually increasing. Cloud computing offers flexibility, scalability, business continuity and unlimited storage. It offers users to use the cloud, when they demand it. Users need not to worry about the software or hardware in a cloud environment. In cloud computing, users do not know where the data are actually stored on the cloud server. Many business models have been developed by using cloud technology [1]. In future, it can be used as a model of business computing. Cloud computing can be defined as a way of delivering IT services in terms of hardware, software, resource, infrastructure, *etc.*

In 1996, the term "cloud computing" was first introduced for describing a model, where all the desktop applications were running on the cloud. In 2006, cloud computing gains much popularity. In 2007, after the collaboration between Google and IBM, cloud technology became very popular. Currently, many IT companies like Microsoft, Google, Yahoo, Amazon, *etc.* are providing cloud services [2].

In cloud computing, there are mainly three parties, namely Cloud Service Provider (CSP), Data Owner (DO) and users. The CSP provides cloud services to both DOs and users. The CSP controls all the tasks of a cloud server, and allows the DOs to store their data or files on the cloud server. The users can access these data or files from the cloud server according to their demand. Fig. 1 shows a simple scenario of cloud computing. There are many security issues in cloud computing [3-7], occur due to either manual

Received July 5, 2016; revised September 25, 2016; accepted October 10, 2016.
Communicated by Balamurugan Balusamy.

cause (hackers) or technical difficulty. Hackers are gradually creating very critical problems in the cloud server. Those hackers replace the original data by their fake data. Sometimes, when internet goes to shut down mode for a long period, all the tasks of a cloud environment used to stop.

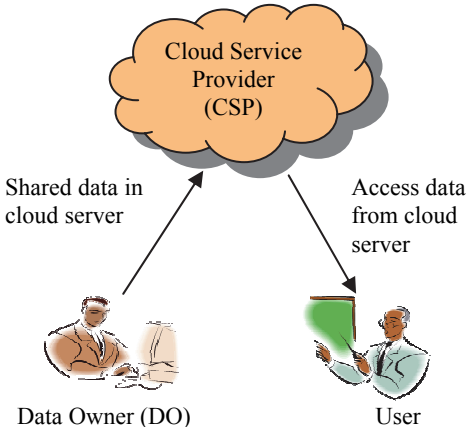

Fig. 1. Simple scenario of cloud computing.

Access control is a major issue out of all these issues. Many researchers proposed many Access Control Models (ACM) for efficient, high performance and secured data accessing in the cloud computing environment [8-18]. In the past years, users were only bounded in their own domain. But, recently, users want to work outside of their own domain. So, these traditional ACMs are not applicable in many cases. When users want to access a data from a cloud server, they must follow many access policies, which are assigned by the CSP or DO. Each CSP or DO has different types of access policies. Users face many problems at the time of accessing data from a cloud server, such as high accessing time, high searching cost, data security, *etc.* Zhu *et al.* [19] introduced Towards Temporal Access Control (TTAC) model, where the CSP records the current time *i.e.* when users make data requests. The user's access rights are restricted by the CSP in this scheme. Danwei *et al.* [20] proposed a scheme, namely Usage Control Based Access Control Model (UCON) for accessing the cloud services. In this model, there are several modules, and it takes much time to provide a data or file. Gao *et al.* [21] proposed a Novel Data Access Control (NDAC) scheme for secure data accessing and data confidentiality. Here, the DO must be always online through the entire data communication process. Wu *et al.* [22] proposed Gateway Based Access Control (GBAC) scheme for collaborative cloud environment. Yu *et al.* [23] introduced Attribute Based Access Control (ABAC) scheme, which is based on the attributes of users. In both ABAC and GBAC, the CSP may need to search the entire database to provide one data, and maintenance of database is very difficult. Therefore, data searching time as well as data accessing time may be very high.

To solve the above-mentioned problems of the existing ACMs, a new ACM has been proposed in this paper, namely Size Based Secure Access Control Model for Cloud Computing (SzSBAC). The proposed scheme is mainly based on the size of the data.

Here, the CSP maintains a temporary table, where data are stored based on the size that facilitates efficient and fast data accessing. When a user makes a request for data, the CSP searches the data based on the requested data size from the table, and provides the data. The CSP needs not to search the entire database for one data. So, the proposed model can minimize high data searching time. Since data searching time is decreased, data accessing time can be automatically decreased. In the proposed model, the DO can only come online for providing the data decryption key or secret key. Therefore, the DO needs not to be always online. The proposed scheme can also minimize the problem of database maintenance by monitoring the temporary table. The major contributions of this paper are listed below:

(A) In this paper, a new table based protocol for data accessing has been proposed for cloud computing environment based on the data size.
(B) The proposed scheme can minimize many problems, such as high data accessing time, high searching time for providing the public key of the data owner, database maintenance, *etc.*
(C) Detailed informal performance analyses as well as the experimental results have been presented in this paper. Both the performance analyses and experimental results prove that the proposed scheme is more efficient and effective than the existing schemes.

The rest of the paper is organized in different section. Section 2 reviews the related works. Section 3 highlights the problem statements. The proposed scheme has been discussed in section 4. In section 5, performance evaluation has been presented. Advantages of the proposed scheme are given in section 6. Finally, conclusions and future work are given in section 7.

2. RELATED WORKS

Access control is a very significant aspect of a cloud environment, where the CSP or the DO defines a set of user's rights or policies by an ACM.

In GBAC [22], there are several private clouds. If a user belongs to one private cloud, s/he can easily communicate with other private clouds. In GBAC, each organization must has a gateway that converts the user's or customer's data into Security Assertion Markup Language (SAML) format, and then, this SAML formatted data are transferred to the target organization. Here, the gateway has vital role for data communication. GBAC scheme required to search the whole database for providing one single data, which results in high time complexity and low maintenance of the database.

In ABAC [23], data access is provided to the users based on the attributes of the users. In this model, data or files are associated with meaningful attributes. A Dummy Attribute (Att_D) is maintained in this scheme for key management. In ABAC, the access structure of every user is represented by an access tree over the attributes, where the root node of the tree is an AND gate, and one child of the root node must be a leaf node associated with Att_D. Here, the CSP controls all the tasks, and the attributes of resources also play a vital role for data accessing [24]. This scheme does not provide confidentiality of

the data items and scalability simultaneously.

Hota *et al.* [25] proposed a model, namely Capability Based Access Control (CBAC) model. CBAC is composed of three entities, namely CSP, DO and user. Here, DOs encrypt a data by the symmetric keys, and give these keys to only authorized users to decrypt the data or file. Modified D-H key exchange procedure is used in this model for data transferring between the CSP and user. CBAC is secured against the man-in-the-middle attack. However, the data searching time can be more as CBAC required searching the entire database, even to provide a single data.

Crues [26] proposed an access control model, which is controlled by the CSP. Here, the CSP decides that whether a user is able to access a data. In this scheme, the CSP provides the access policies to only authorized users. In this scheme, all the tasks are centrally controlled by the CSP.

Towards temporal access control model gives emphasis on the security at the time of data accessing [19]. In TTAC, the DO uses an access Policy (P) to encrypt a data. When the CSP receives a data access request, s/he verifies whether the temporal constraint is satisfied in policy within the current time t_c. The major benefits of TTAC are flexibility, supervisory and security. TTAC scheme restricts access rights of user's by assigning time, according to which only the user can access data from the cloud server.

Usage control based access control model is a conceptual model, which gives decision making ability to the users [20]. UCON has all the advantages of traditional access control models. Many access control models can be established by the UCON. UCON has three entities, namely cloud user, security assertion markup language server and cloud service. It consists of six parts: subject, objects, rights, obligation, authorization and conditions. Here, a negotiation module is used to increase the flexibility in such a way that when user's data access request does not match with the access rules, the user's request does not directly terminate, rather UCON model permits the user to get a subsequent choice by negotiating in some other condition. So, the user can get one more chance to access the data or file.

3. PROBLEM STATEMENTS

In this section, system model, adversary model, system requirements and design goals have been discussed.

3.1 System Model

The proposed model consists of three entities: cloud service provider, data owners and users.

(A) Cloud service provider

CSP is the central administrator of any organization, who provides cloud services to both DO and user. The CSP provides many services by using a number of servers having sufficient space for data storing and computational power. The CSP issues the system parameters, and generates all the certificates, which are necessary during a transaction. The CSP controls accessing of the data or cloud services, and monitors all the tasks of the cloud server.

(B) Data owner

DOs are the entities who wish to store their data on the cloud server, and depend on the CSP for data maintenance. Data owner can be any organization or any individual user. A DO may also be any entity even from outside the organization. At first, all the DOs must be registered at the CSP for storing any data. The DO can restrict users to access any file or data.

(C) User

Users are those parties who want to access data from the cloud server. Only authorized users have the decryption key to decrypt the encrypted file or data. Fig. 2 shows the system model of the proposed scheme.

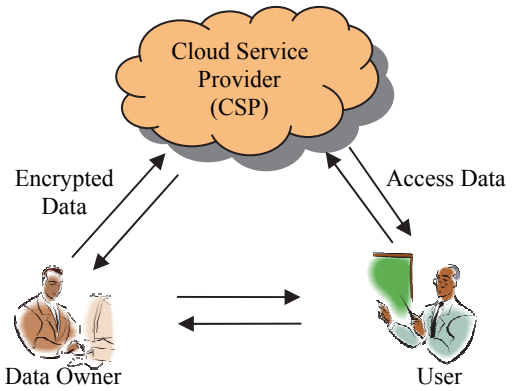

Fig. 2. System model.

3.2 Adversary Model

In the proposed system model, a point to point communication is assumed among the CSP, DO and user. Since there are many attackers, communications are not fully trusted.

(A) CSP is semi-honest and curious

Semi-honest implies that the CSP may follow the protocol. The CSP may be curious as s/he attempts to get user's personal information, such as the content of data, reason for data accessing, *etc.* [27].

(B) User is sensitive and rational

Users are sensitive that implies they do not want to disclose their confidential data, but they want to get the information from others. Rational means that user's behaviour does not depend on emotion or feeling.

(C) DO is curious and sensitive

DOs may be curious to get the user's personal information. DOs may also want to get the information about the CSP.

There are mainly two attacks in the cloud server, namely internal and external at-

tacks [28]. System model covers these security threats and other vulnerabilities. External attacks are the threats from an outsider adversary, who can manipulate the data storage, data modification or data deletion. Internal attacks are mainly triggered by internal entities, such as DO and user. Sometimes, an employee of an organization leaves the job, and then, s/he reveals all the sensitive information of that organization to malicious users. This kind of attack is known as internal attack, which is more malicious than an external attack.

3.3 System Requirements

Nowadays, the time required to access a data from the cloud server is very high. Therefore, users have to pay much money. Since there are many hackers, every user wants that his/her data must be secured on the cloud server. Data should not reveal to any unauthorized users, and the CSP should not use the data for earning revenue. The CSP should provide data security against both internal and external attackers. The requirements of the proposed scheme are mentioned below:

(A) Fine-grained access control

The CSP should ensure that only authorized users are able to access a data, for which they have access rights. Users are not allowed to access the data as they wish. Data owner assigns access policies for each and every data, which must be satisfied by a user to access the data from the cloud server.

(B) Reducing data accessing time

On the cloud servers, there are huge amounts of data. DOs share their data from all over the world. Since users pay money on the basis of pay-per-use, reducing data accessing time is always required for users, so that they can pay less money for using the cloud services.

(C) Maintaining the database

Though in cloud computing, storage space is not limited, the CSP always needs to easily maintain the cloud database, and wants to minimize the problem of data redundancy.

3.4 Design Goals

The main goals of the proposed scheme are as follows. To achieve a scalable and efficient access control model for the cloud environment. Data accessing time should be less, so that users can pay less money for using the cloud services. Data searching time should be less, so that the CSP can take less time to provide the public key of the DO. The CSP should monitor and maintain the database in an easy and efficient way. In traditional schemes, the DO must be always online during the entire communication, which results in an increase in the system overhead. In the proposed scheme, an attempt is made to reduce the overhead of the system.

4. PROPOSED SCHEME

In cloud computing, DOs can store any size of file or data. The proposed scheme uses the information about the sizes of the data, which users want to access from the

cloud server. There are many units of data size, such as Bit, Nibble, Word, Byte, Kilobyte, Megabyte, Terrabyte, Gigabyte, Petabyte, Zettabyte, *etc.* [29]. Bit is the smallest unit of data size.

The proposed scheme is composed of three entities, namely CSP, DOs and users, Here, in the proposed scheme, the CSP maintains a temporary table, namely CSP Table (CSPTAB), and it consists of four attributes: Group No (GPNO), Size of data (Sz), Data Owner's ID (DOID) and DO's Date & Time (DOD&T). The detail procedures of message passing in the proposed model are presented in this section.

Table 1. Example of CSPTAB.

Group Number (GPNO)	Data Owner's ID (DOID)	Data Size (Sz) (In GB)	Data Owner's Date & Time (DOD&T)	
1	033	$2^{-33} < Sz \leq 2^{-30}$	29-05-15	11.32
	001		21-08-14	01.19
	612		03-07-14	17.46
	342		17-06-13	07.36
	561		22-12-11	05.59
2	210	$2^{-30} < Sz \leq 2^{-20}$	16-09-15	06.08
	003		13-11-14	03.36
	103		09-07-12	23.46
	198		12-06-12	10.42
	712		13-12-07	23.49
3	178	$2^{-20} < Sz \leq 2^{-10}$	12-03-14	21.23
	564		15-07-13	12.32
	725		28-11-12	20.16
	069		19-11-12	12.23
	149		22-07-10	09.30
.	.	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
	.	.	.	
50	613	$2^{-10} < Sz \leq 1$	18-03-13	16.23
	071		03-02-13	17.41
	687		27-10-12	14.09
	738		21-09-11	17.48
	068		25-11-10	11.57

In the CSPTAB, the GPNO contains the total number of groups. The Sz contains a range of data size of any particular group. Each and every group contains a different range of data size, which is defined by the CSP for efficient data accessing. In a cloud environment, there may be many DOs who share same size of data. Whenever the DOs share a data on the cloud server, the CSP records all the information about the data and add the DO's ID in the CSPTAB at respective positions. The CSP groups IDs in DOID field of those DOs who store the same size of data. The DOD&T contains corresponding latest time and date of DOs that when they have provided the data to the users. In the CSPTAB, DO's IDs are placed based on Least Recently Used (LRU). After providing the data to the users, most recent DO's IDs are placed at the top of the group. The

CSPTAB is mainly used for fast and efficient data accessing. Whenever a user makes a request for accessing data from the cloud server, at first, the CSP checks his/her authorization, and then, the CSP searches the data on the basis of the requested data size. Therefore, the CSP does not need to search the entire database to provide one data. Thus, data searching time as well as data accessing time can be decreased by using the CSPTAB.

Table 2. Notation and description.

Notation	Description
PUSP	Public key of cloud service provider
EK	Encryption
PRSP	Private key of cloud service provider
DK	Decryption
PUUSR	Public key of user
Certi	Certificate
PRUSR	Private key of user
Sec	Secret key or data decryption key
PUOWN	Public key of data owner
AR	Access rights
PROWN	Private key of data owner
$GPNO_i$	*i*th user group
$PUOWN_j$	Public key of *j*th data owner
$DOID_j$	*j*th data owner's ID
Y_{Sz}	Data size of Y
DO_{jSz}	Data size of *j*th data owner
$GPNO_{iSz}$	Data size corresponding to *i*th GPNO
$GPNO_{iDOID}$	DOID field of corresponding *i*th GPNO
$DOID_{NEW}$	DOID field after processing
$DOID_{PREV}$	DOID field before processing
X_{Sz}	X's requested data size
EK_{Sec}	Encryption with secret key
DK_{Sec}	Decryption with secret key
EK_{PUSP}	Encryption with public key of CSP
EK_{PRSP}	Encryption with private key of CSP
DK_{PRSP}	Decryption with private key of CSP
DK_{PUSP}	Decryption with public key of CSP
EK_{PROWN}	Encryption with private key of DO
DK_{PUOWN}	Decryption with public key of DO
EK_{PUUSR}	Encryption with public key of user
EK_{PRUSR}	Encryption with private key of user
DK_{PRUSR}	Decryption with private key of user
DK_{PUUSR}	Decryption with public key of user

In Table 1, an example of CSPTAB is shown. In the CSPTAB, 50 GPNOs have been considered, and each group contains one data size, 5 DO's ID, who provide the same size of data or the same range of data and corresponding latest date and time of

each DO that when s/he has provided the data. Table 2 represents description of notations, which are used in this paper.

In the proposed technique, there are mainly four processes, namely user authorization, data storage, processes on the CSPTAB and data access.

4.1 User Authorization

On the cloud server, authorization is the first step for communication among the CSP, DOs and users. At first, for data accessing from a cloud server, users must be registered at the cloud server.

In user's registration phase, a registration request is sent by a user to the CSP. The CSP authenticates the user by the digital signature, and collects all the significant information of the user. After collecting all the information of the user, the CSP sends a registration acknowledgment to the user [30]. Then, the user sends a request to the CSP for providing the data. The CSP sends the DO's public key to the user for getting the certificate or AR and the secret key *i.e.* data decryption key from the DO. Using this public key, the user sends a request to the DO for getting the secret key and certificate. When the DO gets a request from the user, after checking the user's authorization, s/he provides the corresponding secret key and AR of the data. The user shows this certificate to the CSP, and after verifying the certificate with respect to the requested data, the CSP provides the requested data to the user. Otherwise, the CSP terminates the data access request.

4.2 Data Storage

In this sub-section, the data storage processes have been discussed. At first, DOs must be registered at the cloud server for storing any data. When DOs want to share a data on the cloud server, they have to create a secret key or password for each data [23]. DOs also share a certificate or AR along with the data.

There are three encryption processes for storing a data on the cloud server. At first, DOs encrypt their data by the secret key. Secondly, the data are encrypted by the DO's private key, and then, DOs encrypt the data and certificate by the CSP's public key. DOs make a bundle of the encrypted data, and send it to the CSP along with the certificate. The CSP uses his/her own private key, and then, again uses the DO's public key to decrypt the bundle [21]. There are few processes for data storing between DOs and CSP, which are given below:

(A) Data owner

- DOs encrypt the data by using the secret key, which is created by the DO.
  $Data \rightarrow EK_{Sec}(Data)$
- DOs encrypt $EK_{Sec}(Data)$ by the DO's private key.
  $EK_{Sec}(Data) \rightarrow EK_{PROWN}\{EK_{Sec}(Data)\}$
- At last, DOs encrypt $EK_{PROWN}\{EK_{Sec}(Data)\}$ and certificate by using the CSP's public key. Then, make a bundle, and send it to the CSP.
  $EK_{PROWN}\{EK_{Sec}(Data)\}$ and $Certi \rightarrow EK_{PUSP}[EK_{PROWN}\{EK_{Sec}(Data)\}$ and $Certi]$
  $EK_{PUSP}[EK_{PROWN}\{EK_{Sec}(Data)\}$ and $Certi] \rightarrow CSP$

(B) Cloud service provider

- At first, the CSP decrypts the bundle by using his/her private key.
  $EK_{PROWN}\{EK_{Sec}(Data)\}$ and $Certi$
  $\leftarrow DK_{PRSP}[EK_{PUSP}[EK_{PROWN}\{EK_{Sec}(Data)\}$ and $Certi]]$
- Again the CSP decrypts the message by using the DO's public key.
  $EK_{Sec}(Data)$ and $Certi \leftarrow DK_{PUOWN}[EK_{PROWN}\{EK_{Sec}(Data)\}]$ and $Certi$

The CSP stores $EK_{Sec}(data)$ and certificate on his/her own database. The CSP cannot be able to read the original content of the bundle because the secret key is not shared with the CSP. DOs share their secret key to only the authorized users. Fig. 3 shows message transfer sequences among the CSP, DO and user.

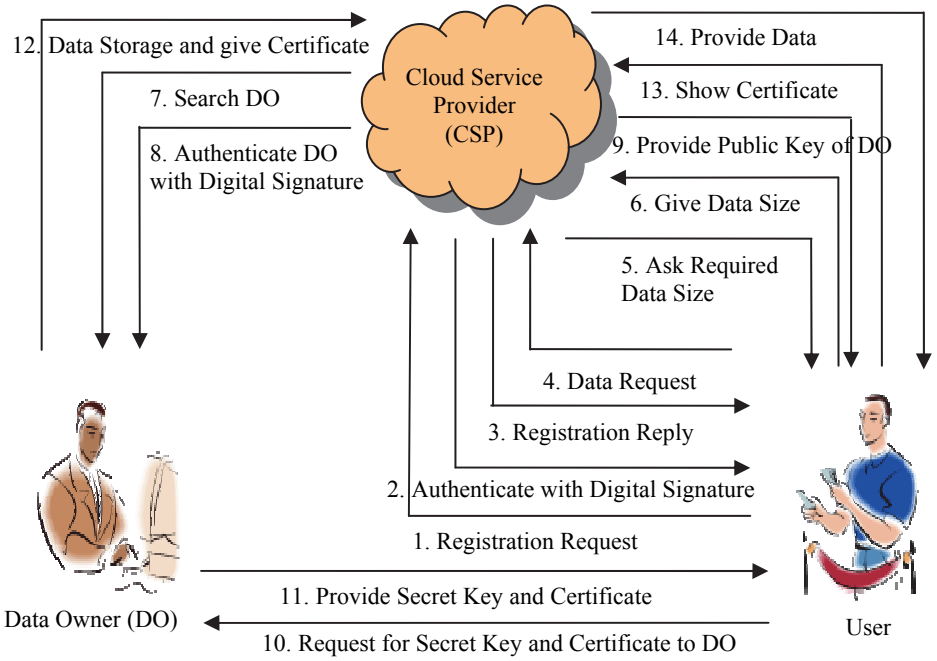

Fig. 3. System architecture of proposed scheme.

4.3 Processes on the CSPTAB

All the main works of the proposed scheme are executed on the CSPTAB. When the DOs share their data on the cloud server, the CSP checks their data size. If DO's data size is presented in the CSPTAB, the CSP adds that DO in the CSPTAB at the appropriate position. Otherwise, the CSP adds a new data size field in the CSPTAB, and then, add that DO in the new field. The CSP also can delete a whole group number or DO's ID from the CSPTAB. There are mainly four algorithms on the CSPTAB, namely insert a DO's ID in the CSPTAB, delete a DO's ID from the CSPTAB, delete a whole GPNO from the CSPTAB and search a DO's ID from the CSPTAB for providing the public key

of the DO. Let us consider that N is the maximum number of GPNO in the CSPTAB, and the maximum number of DOID per GPNO is M.

4.3.1 Algorithm for inserting a DO's ID in the CSPTAB

Step 1: If Y = ID of an authorized DO
Goto step 2
Else
Goto step 4
Step 2: Flag = 0,
For i = 1 to N
If $Y_{Sz} = GPNO_{iSz}$
If $GPNO_{iDOID} \neq$ Full
$DOID_{NEW} = DOID_{PREV} \cup Y$
Flag = 1
Break
Else
DELETE an ID from ith DOID
$DOID_{NEW} = DOID_{PREV} \cup Y$
Flag = 1
Break
End For
If Flag = 0 and CSPTAB $\neq$ Full
Add a new GPNO with Y_{Sz}
Add details of Y in the new GPNO
Else if Flag = 0 and CSPTAB = Full
DELETE a whole GPNO
Add a new GPNO with Y_{Sz}
Add details of Y in the new GPNO
Step 3: UPDATE CSPTAB
Step 4: STOP

4.3.2 Deletion

There are two cases in deletion. First one is to delete a DO's ID from the CSPTAB, and the second one is to delete a whole GPNO from the CSPTAB. To delete a DO's ID from the CSPTAB, the CSP searches the GPNO and its corresponding DOID field on the basis of LRU policy. If the CSP finds that any GPNO has not been used for a long time, the CSP can delete that full GPNO after searching it on the basis of LRU policy.

(A) Condition no 1
Algorithm for deleting a DO's ID from the CSPTAB:

Step 1: For i = 1 to N
For j = 1 to M
$k = SEARCH_{LRU}(DOID_j)$

$DOID_{NEW} = DOID_{PREV} - k$
End For
End For
Step 2: UPDATE CSPTAB
Step 3: STOP

(B) Condition no 2
Algorithm for deleting a whole GPNO from the CSPTAB:

Step 1: SEARCH LRU GPNO
Step 2: DELETE entries of *i*th GPNO
Step 3: UPDATE CSPTAB
Step 4: STOP

4.3.3 Algorithm for searching a DO's ID from the CSPTAB for providing PUOWN

Step 1: If $X = ID$ of an authorized user
Goto step 2
Else
Goto step 3
Step 2: For $i = 1$ to N
If $X_{Sz} = GPNO_{iSz}$
SEARCH DO in $GPNO_i$
For $j = 1$ to M
If $X_{Sz} = DO_{jSz}$
PROVIDE $PUOWN_j$ to X
End For
Else
Display INVALID data size
End For
Step 3: STOP

4.4 Data Access

After the user's authorization, users have to send a data request to the server for accessing data or file. If users want to access a data for the first time, the CSP asks users to get the data decryption key or secret key and certificate from the DO for decrypting the data. When users make a data request for the second time, after the user's authorization, the CSP directly provides the data. In this case, the CSP does not ask the users to get the secret key and certificate from the DO because users have already got the secret key and certificate from the DO, when they have accessed the data for the first time [23]. For data accessing, following processes are executed:

Step 1: Users send a request to the CSP for data accessing. The CSP searches the DO from the CSPTAB, and authenticates the DO by the digital signature. The CSP encrypts the DO's public key by using his/her own private key and user's public key, and then, sends it to the users. After decrypting the message, users get the DO's public key.

Step 2: In the second step, users send a query to the DO for providing the secret key and certificate. Users send this message after encrypting by using their own private key, and then, again encrypting by using the DO's public key. The DO decrypts the message by using his/her private key, and then, again decrypts it by using the user's public key.

Step 3: In the third step, the DO checks the user's authorization from the CSP. The CSP sends the authorization confirmation to the DO.

Step 4: After confirmation about the user's authorization, the DO sends the secret key and certificate to the user after encrypting by using his/her own private key and user's public key. The whole encryption processes are shown below:

Sec and $Certi \rightarrow EK_{PROWN}(Sec$ and $Certi)$
$EK_{PROWN}(Sec$ and $Certi) \rightarrow EK_{PUUSR}\{EK_{PROWN}(Sec$ and $Certi)\}$
$EK_{PUUSR}\{EK_{PROWN}(Sec$ and $Certi)\} \rightarrow User$

Step 5: Users decrypt the message by using their own private key and DO's public key.

$EK_{PROWN}(Sec$ and $Certi) \leftarrow DK_{PRUSR}[EK_{PUUSR}\{EK_{PROWN}(Sec$ and $Certi)\}]$
Sec and $Certi \leftarrow DK_{PUOWN}\{EK_{PROWN}(Sec$ and $Certi)\}$

Step 6: Users encrypt the certificate by using their own private key, and then, again by using the CSP's public key. Then, they send it to the CSP.

$Certi \rightarrow EK_{PRUSR}(Certi)$
$EK_{PRUSR}(Certi) \rightarrow EK_{PUSP}\{EK_{PRUSR}(Certi)\}$
$EK_{PUSP}\{EK_{PRUSR}(Certi) \rightarrow CSP$

Step 7: The CSP decrypts the message.

$EK_{PRUSR}(Certi) \leftarrow DK_{PRSP}[EK_{PUSP}\{EK_{PRUSR}(Certi)\}]$
$Certi \leftarrow DK_{PUUSR}\{EK_{PRUSR}(Certi)\}$

Step 8: The CSP checks whether the certificate is matched with the requested data. If it is matched, the CSP provides the requested data to the users after encrypted by using his/her own private key and user's public key.

$EK_{Sec}(Data) \rightarrow EK_{PRSP}\{EK_{Sec}(Data)\}$
$EK_{PRSP}\{EK_{Sec}(Data)\} \rightarrow EK_{PUUSR}[EK_{PRSP}\{EK_{Sec}(Data)\}]$
$EK_{PUUSR}[EK_{PRSP}\{EK_{Sec}(Data)\}] \rightarrow User$

Step 9: Users decrypt the encrypted data by using their own private key, and then, decrypt by using the CSP's public key. At last, users again decrypt the message by using the secret key for getting the original data.

$EK_{PRSP}\{EK_{Sec}(Data)\} \leftarrow DK_{PRUSR}[EK_{PUUSR}[EK_{PRSP}\{EK_{Sec}(Data)\}]]$
$EK_{Sec}(Data) \leftarrow DK_{PUSP}[EK_{PRSP}\{EK_{Sec}(Data)\}]$
$Data \leftarrow DK_{Sec}\{EK_{Sec}(Data)\}$

Step 10: Users get the original data.

5. PERFORMANCE EVALUATION

This section verifies the reliability and accuracy of the proposed scheme through simulation and comparison of the performance with several existing well-known schemes.

5.1 Simulation Environment

In cloud computing, users usually access data from the cloud server. But, to build a real cloud environment is not always possible. To evaluate the validity of the proposed scheme, a cloud simulation platform has been set up. In this paper, CloudSim 3.0.3 has been used to evaluate the performance of the proposed scheme [31]. CloudSim was developed by a research group at university of Melbourne.

CloudSim toolkit consists of mainly four layers, namely cloud resources, cloud services, virtual machine services and user interface structures. In bottom most cloud resource layer, there are mainly two entities, namely host and datacenter. In cloud services layer, many services like Virtual Machine (VM) provisioning, CPU allocation, storage allocation, memory allocation and bandwidth allocation are provided. Virtual machine services are responsible for mainly VM management. The top most user's interface structure layer deals with the physical host specification. This layer also deals that how the user's request is processed by the system. The CloudSim provides a work environment, where a host can concurrently use by many VMs on the basis of user's QoS specification [31]. Load distribution process also improves the performance of the CloudSim. There are several components in CloudSim toolkit, such as Cloud Information Service (CIS), datacenter, host, virtual machine, broker, cloudlet and VM Manager (VMM).

In CloudSim, for executing a task, at first, the datacenter sends a request to the CIS to register his/her resources. Each datacenter has a number of hosts, and each host has several VMs having some essential characteristics, such as processing elements, RAM, bandwidth, *etc.* The broker has many cloudlets, which are waiting for executions. The broker requests to the CIS to provide the available lists of resources to execute the cloudlets. Then, the characteristics of the datacenter are sent to the broker. Now, VMs are created for user or broker after getting the request from the user or broker. If VMs are created, the datacenter sends an acknowledgment to the broker. After receiving the acknowledgment, the broker submits the cloudlet to the datacenter broker module for executing the task. The datacenter assigns VMs for cloudlet. When a cloudlet is executed, an acknowledgment is sent to the broker.

CloudSim package is installed on DELL OPTIPLEX 9020 desktop with 3.40 GHz Intel corei7 processor, 8 GB RAM and 1 TB storage capacity with Windows 7 Operating System. Java version 7 is installed on the system, and Apache Common-math 3.5 is also configured with CloudSim for execution purpose [32, 33]. A heterogeneous cloud environment has been created using the CloudSim toolkit. 10 datacenters have been considered that consist of 1000 physical nodes, where 500 nodes are HP ProLiant ML110 G4 servers, and other 500 nodes are HP ProLiant ML 110 G5 servers. Each 500 nodes are assigned with 1860 MIPS and 2660 MIPS for each and every core of G4 and G5 servers, respectively with 4 GB of storage capacity, and network bandwidth is considered as 1 GB/s. The RAM is divided into four types of VMs: Type 1 (500 MIPS, 512MB), Type 2 (1000 MIPS, 1GB), Type 3 (1500 MIPS, 1.5 GB) and Type 4 (2000 MIPS, 2 GB). All

VMs are 2 GB of size, and bandwidth is 1000 MB/s.

For the simulation purpose, few modifications have been made in the CloudSim toolkit. A new class, namely *dynamicmemory* is added in the CloudSim toolkit. This class helps to guess the required resource, and after a cloudlet execution, it performs updating the resource allocation of VMs. Few classes are also modified, namely *VM, powerhost, VMScheduler* and *CloudletScheduler* to achieve the dynamic resource distribution. To maintain the CSPTAB, a new class, namely *csptab* has been added in the CloudSim toolkit. We have assumed that there are 50 groups, and each group contains ID of 5 DOs. To implement the existing attribute based access control model, three new classes have been added, namely *attribute, attributehistorylist* and *accessright*. The *attribute* class is used to set a number of attributes. The *attributehistorylist* class is used to maintain the evolution history of all attributes. We have assumed that the *attributehistorylist* class can contain the history of maximum 200 attributes. The *accessright* class is used to assign a set of actions for each user or broker. Three new classes have been added in the CloudSim, namely *temporalconstraint*, *policy* and *currenttime* to implement the towards temporal access control model. The *temporalconstraint* class is used to assign constraints. The *policy* class is used by the CIS to set policies for each data item. The *currenttime* class is used to record the time, when users send data access request. To implement capability based access control model, the *capabilitylist* class has been added in the CloudSim toolkit. This class keeps information of the user ID, file ID and access rights.

In the simulation process, the main emphasis is given on the proposed scheme to depict how it helps the CIS for efficient data accessing from the cloud server. By using the data size, the CIS can easily search the data, and provide it to the user. In the simulation process, data searching time, data accessing time and CPU utilization are considered to evaluate the performance analysis. The key generation time, data encryption and decryption times are not considered in the simulation process; as in the proposed scheme, the data security issue and the data access control issue are treated separately for secure and efficient data access control. Few parameters have been considered for the experimental purposes, namely number of users, size of data or file, data searching time, data accessing time and CPU utilization. There are three assumptions, which are necessary for the experimentations. These assumptions include:

(A) All the transactions are executed in the mutual exclusion manner. Users are not allowed to suddenly stop the transaction until it gets completed.
(B) Every user is allowed to send the access request to only one cloud server. Users are not allowed to send request outside the cloud server.
(C) Each user can send at most 20 requests per minute to the cloud server. The time interval between two requests is random.

5.2 Results and Discussions

To evaluate the performance analysis, data searching time for providing the PUOWN, data accessing time and CPU utilization have been calculated for the proposed scheme and other existing schemes, namely ABAC, TTAC and CBAC.

Fig. 4 shows the results of the first experiment for calculating the searching time for providing the public key of the DO. From Fig. 4, it is clear that the proposed technique

gives much better result than the existing schemes for providing the public key of the data owner. Experiments have been executed with different size of files to get the accurate results. In the proposed scheme, the CSP does not search the whole database for providing the public key of the data owner. Here, searching is based on the requested data size. In the simulation process, as discussed in section 4, for finding the DO's ID, the CIS matches the DO's data size with respect to the requested data size from the CSPTAB. In the proposed scheme, the CIS can directly find where the DO is actually placed in the database. Thus, after getting the details of the DO, the CIS can easily provide the public key of the DO. If the same size of data is requested by another user, the CIS can easily locate the position of the data owner because the CIS records the recent data accessing details of same size of data. Thus, data searching time is less in the proposed scheme, and this produces a straight line curve with a slight increase at initial. In ABAC, when a user requests to access any data, the CIS matches the attributes of the users with the requested data. So, time complexity is increased, which gives a zigzag curve. In ABAC, TTAC and CBAC, when the CIS receives a data request, the CIS searches the entire database for a single data. So, data searching time is increased in ABAC, TTAC and CBAC.

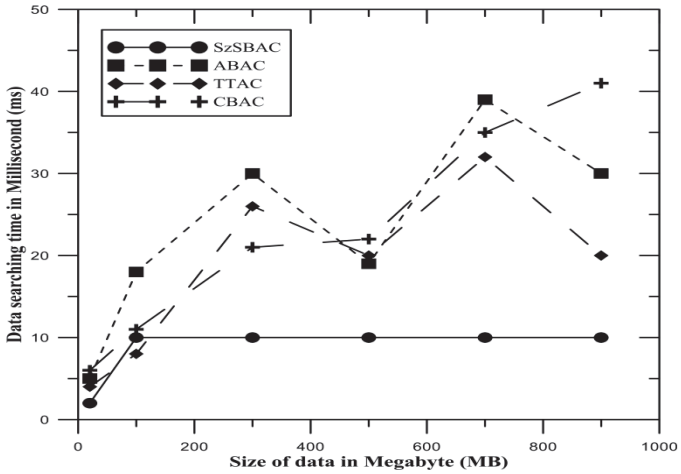

Fig. 4. Data searching time vs. size of data.

Fig. 5 highlights the experimental results for data accessing time. Since the searching time for providing the public key of the DO is decreased by the proposed scheme, data or file accessing time is automatically decreased. But, for the other existing schemes, namely ABAC, TTAC and CBAC, the data accessing time is increased in the same way as the data searching time. So, the curves we got for these schemes are same as Fig. 4. In case of the proposed scheme, we have obtained an almost linearly increasing curve for the increasing number of users with respect to the data accessing time in second. Experimental results show that if there are many users, data accessing time is increased. For data accessing from the cloud server, several factors have been considered, namely exact matching, aggregation and response time. Response time denotes how long users have to wait for getting the response from the server. There may be some requests, which are

timed out. Time out may occur for several reasons, such as too many requests are in the waiting queue and network is not responding for a long time period. When time out is crossed a threshold value it means that a request is waiting for a long time. We have assumed that time out is 60 seconds, and the numbers of requests are same for all users. Initially, one request per minute is tested, and it is gradually increased up to 20.

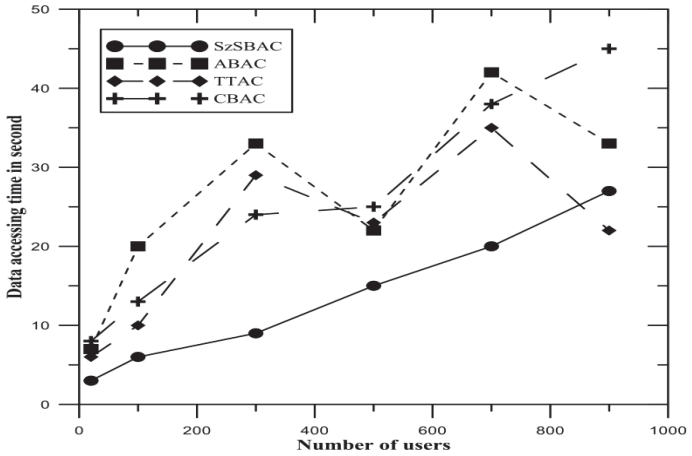

Fig. 5. Data accessing time vs. number of users.

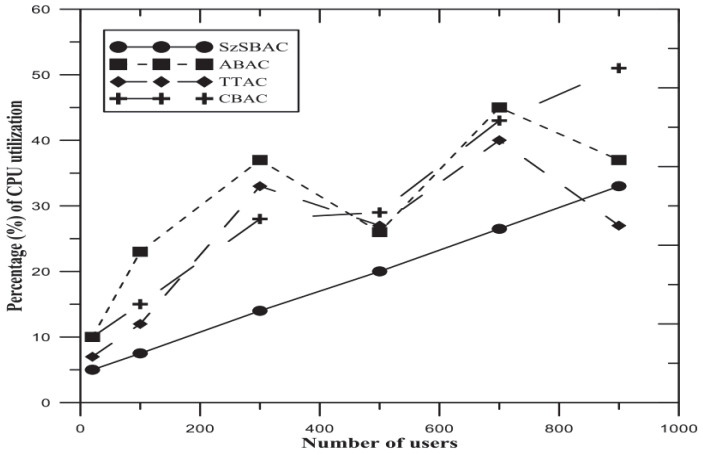

Fig. 6. Percentage of CPU utilization vs. number of users.

In the last simulation process, utilizations of CPU have been calculated. At the beginning, 5 VMs are allocated on each host. Virtual machines are sorted in descending order of CPU utilizations and assigned them to hosts in a first fit method. VMs are migrated from one host to another if CPU utilizations are optimized. Fig. 6 illustrates the experimental results of CPU utilization. From Fig. 6, it is clear that in the proposed scheme, the CPU utilization is increased linearly with the number of users, resulted in yielding a linearly increasing curve. When there are many users in the cloud server, the

numbers of requests for accessing the cloud services are increased. As discussed earlier, the proposed scheme takes less time to search the data, and thus, data accessing time also decreased. Therefore, in the proposed scheme, the CPU does not take much time to respond to the users in comparison to the other existing schemes. Thus, the CPU utilization is less. Each experiment runs for 60 seconds, and it is repeated if the server replies with an error. During the experiment, CPU utilizations have been calculated per 3 seconds, which means 20 values per experiment. Then, average CPU utilizations have been calculated out of all 20 values for each experiment. CPU utilizations have been calculated by Eq. (1).

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

where $\sum_{i=1}^{n} U_i(n)$ = Sum of all CPU usage of all CPU cores, and n = Total number of CPU used for an experiment. So, $n \in 1, 2, 3, \ldots$

5.3 Performance Analysis

The proposed scheme consists of mainly four operations, namely creation of the CSPTAB, insertion of a DO's ID in the CSPTAB, deletion of a DO's ID or GPNO from the CSPTAB and searching PUOWN from the CSPTAB. Complexities have been calculated for these four operations.

(A) Creation of the CSPTAB

In the proposed scheme, the CSPTAB is formed by the CSP. For the creation of the CSPTAB, the CSP has to examine and monitor the data size of each and every data owner. The computational overhead of creation of the CSPTAB occurs, when the CSP monitors the sizes of all data or files.

(B) Insertion of a data owner's ID in the CSPTAB

For inserting a DO's ID in the CSPTAB, at first, the CSP records the data size, and then, the CSP matches the DO's data size with respect to the existing data size of the CSPTAB. If DO's data size is matched with the CSPTAB's data size, add the DO's ID in that particular GPNO. Otherwise, add a new GPNO in the CSPTAB with the DO's data size, and add the DO's ID in the new GPNO. We have assumed that the maximum group number in the CSPTAB is N, and M is the maximum number of DO's ID in the DOID field. In the worst case, to insert a DO's ID, the CSP has to search all GPNO fields to find the exact data size, and in that matched GPNO, the CSP again has to search the corresponding M positions. So, in the worst case, the complexity is $\mathcal{O}(NM)$ to insert a DO's ID in the CSPTAB.

(C) Deletion of a data owner's ID or group number from the CSPTAB

Deletion operations are fully handled by the CSP. The CSP can delete a DO's ID from the CSPTAB or the CSP can also delete a whole GPNO field from the CSPTAB. In section 4, it has already been discussed that the deletion operation is mainly based on the LRU concept. The CSP deletes a DO from the CSPTAB, who has shared his/her data before a long time ago. At the time of deleting a DO's ID, the CSP searches the GPNO,

and then, again the CSP searches the whole DOID field corresponding to the GPNO field. So, in the worst case, the complexity is $\mathcal{O}(NM)$ to delete a DO's ID from the CSPTAB. If the CSP wants to delete a whole GPNO field, in the worst case, the CSP has to search N number of GPNO. So, the complexity is $\mathcal{O}(N)$.

(D) Search public key of the DO from the CSPTAB

This is one of the main operations of the proposed scheme. For providing the public key of the DO, the CSP initiates a query in the CSPTAB based on the requested data size. In the worst case, the CSP searches in all GPNO field for providing the PUOWN. At the mean time, the CSP also searches the DOID field for searching the DO's ID. The CSP does not search the entire database. So, the complexity is $\mathcal{O}(NM)$. Table 3 summarizes the complexity of the proposed scheme in the worst case scenario.

Table 3. Complexity of the proposed scheme in the worst case.

Operation	Complexity
Insertion	$\mathcal{O}(NM)$
DOID Deletion	$\mathcal{O}(NM)$
GPNO Deletion	$\mathcal{O}(N)$
Search Public Key of the Data Owner	$\mathcal{O}(NM)$

(E) File access

There are mainly two conditions for file accessing. The first condition is when the users are new in the cloud server. When the users are new, the CSP asks them to put the required data size. The CSP also asks users to collect the secret key and certificate from the DO. All the users must put the required data size, and they have to collect the secret key and certificate. The old users need not collect the secret key and certificate from the DO. After the user's authorization, the CSP provides the requested data to the old users. Computational overhead for file accessing is minimized because for the old users, after the user's authorization, the CSP can easily provide the requested data to the users.

5.4 Comparisons with Existing Schemes

In ABAC [23], the CSP has to assign attributes for each and every user, and keeps an Attributes History List (AHL) for maintaining evolution history of all attributes. In a cloud environment, maintaining a large number of attributes, and set them along with the AHL is very difficult. Thus, complexity of the system may be increased. In ABAC, users loose AR if they log out from the cloud server. Therefore, the overhead of the system may be also increased because after log out, when users want to access a data, then, again they have to get the AR from the CSP. Thus, data accessing time may be automatically increased in ABAC. In SzSBAC, if the user had already accessed data from a cloud server, after getting a data access request from the user, the CSP just checks the authorization, and provides the requested data to the user. Thus, the proposed scheme can be considered simple, and the overhead of the system may be reduced.

In TTAC [19], current time is introduced with respect to proxy re-encryption. TTAC specifies some predefined time that when the user can access a data or file from the cloud server. The users have to access the data within the specified time range only.

This feature often dimmed the user-friendliness of the TTAC scheme. In TTAC, data searching time may also increase because for providing the requested data, the CSP may need to search the entire database. Thus, data accessing time may be automatically increased in TTAC. In the proposed model, no specific time is mentioned by the CSP for data accessing. So, users can access data at any time they want. Thus, the proposed scheme can be considered as user-friendly. As discussed earlier, the proposed scheme maintains a table for fast data accessing. The CSPTAB facilitates the CSP to provide the requested data without searching the entire database. Thus, the CSPTAB of the proposed scheme can minimize data searching time as well as data accessing time.

In CBAC [25], DOs maintain a capability list for data accessing. Here, when users want to register at the CSP, they have to send user ID and file ID to the DO. Then, the DO updates the capability list, and sends the users requested encrypted data along with the capability list to the CSP. The maintenance of the capability list, in turn, caused to increase the overhead at the DO's end. Another major problem of CBAC is that when users make data request, the CSP may need to search the entire database for a single data. So, data searching time as well as data accessing time may be increased in CBAC. In the proposed scheme, DOs do not contain any list for data accessing. Therefore, the overhead at the DOs end is reduced as compared to CBAC. In the proposed scheme, when users request for data accessing, the CSP searches the data based on the requested data size from the CSPTAB, and provides the data to the user. The CSP needs not search the entire database. Thus, in the proposed scheme, data searching time may be decreased, which in turn facilitates to have less data accessing time.

The main issue of usage control based access control model is that it takes much time for providing a data because there are many modules in UCON, such as cloud service, policy enforcement point, policy decision point, policy information point, policy administration point, XACML policy database and negotiation module [20]. For providing one data, each user's request must go through each and every module. So, complexity can be increased in UCON, and time can be also consumed for data accessing. The proposed scheme does not contain many modules. The CSP can directly provide the requested data to the respective user without using of several modules. Therefore, complexity is not high as UCON, and data accessing time can be also decreased in the proposed scheme.

In GBAC [22], the CSP may have to search the entire database for one data. So, data searching time as well as the overhead of the system can be increased. The user has to wait a long time for getting the data access from the cloud server. In the proposed model, the CSP records all the details about a data in the CSPTAB, and searches the requested data by using the requested data size from the CSPTAB. Here, for providing one data, the CSP does not need to search the entire database. Therefore, data searching time can be reduced, and the overhead of the system can be also decreased.

NDAC scheme provides trusted communication among CSP, DO and user [21]. NDAC is secured against the man-in-the-middle attack and replay attack. In NDAC, DOs must be always online for the entire data communication process. So, the load on the system can be increased. In the proposed scheme, DOs may come online just for providing the secret key to the users. Therefore, DOs need not to be always online, and thus, the load on the system can be decreased. Table 4 shows comparisons among the existing schemes and the proposed scheme (SzSBAC).

Table 4. Comparisons among the existing schemes and the proposed scheme.

Existing schemes	SzSBAC	Advantages of SzSBAC over existing schemes
In the existing schemes [19, 22, 23, 25], searching cost as well as data accessing time can be high because the CSP may need to search the entire database for providing one data.	In the proposed scheme, the CSP searches data based on the requested data size.	Searching cost can be reduced. Data accessing time can be less.
In NDAC [21], data owners should be online during the entire data accessing process, and they cannot go offline during a transaction.	In SzSBAC, data owners can go offline after providing the secret key and certificate.	DO's are not required to be always online. The load on the system can be decreased.
In TTAC [19], a specific time period for data accessing is defined by the CSP for the users. Thus, this scheme cannot be considered as user-friendly.	In the proposed scheme, specific time period is not mentioned for data accessing from the cloud server.	Users can access data at any time they want. Thus, the proposed scheme can be considered as user-friendly.
In UCON [20], user's data request has to go through many modules, which increases the complexity of the system.	SzSBAC does not contain many modules as compared to UCON.	Complexity can be less as compared to UCON.

6. ADVANTAGES OF THE PROPOSED SCHEME

This section describes the advantages of the proposed scheme:

(A) Flexibility

The proposed scheme is very user-friendly and flexible since users can access data at any time they want. There is no time specification in the proposed scheme for accessing data from the cloud server.

(B) Security

In the proposed scheme, for storing a data, at first, the DO encrypts the data by the secret key. Then, the DO uses his/her own private key and public key of the CSP to encrypt the data. The DO shares the secret key to only authorized users. Neither the CSP nor any malicious user or hacker gets the secret key. So, the proposed scheme is secured.

(C) Maintenance of the database

In the proposed scheme, those DOs who have the same size of data or the same range of data make a group. Each Sz field can contain only one range of data size. Same size of data cannot be presented in several Sz fields. So, the CSP can easily monitor and maintain the database by the CSPTAB.

(D) Reduced searching time

In SzSBAC, for finding a PUOWN, the CSP must send a query to the CSPTAB

based on the data size. So, searching time is reduced since the CSP does not need to search all the DOs.

(E) Reduced data accessing time

The proposed scheme is very useful for fast data accessing. By using the CSPTAB, the CSP can easily provide the DO's public key. So, searching time is decreased for providing the PUOWN. Therefore, users do not have to wait a long time for getting the PUOWN. After getting the PUOWN, users can send a request for the secret key. Because of minimization of the searching time of the PUOWN, data accessing time is automatically decreased. So, users can pay less money for using the cloud services.

(F) Reduced overhead

In the proposed scheme, DOs need not to be always online. They can be online just for providing the secret key and certificate to the users. After providing the secret key and certificate, they can go offline. Therefore, the overhead of the system is reduced.

7. CONCLUSIONS AND FUTURE WORK

Cloud computing is one of the advanced fields in IT. In this paper, a new access control scheme has been proposed. The proposed scheme is mainly based on the data size stored by the data owners, and it can minimize several problems for data accessing. In the proposed scheme, the CSP does not check the entire database for providing the public key of the data owner. So, searching time for providing the public key of the data owner can be reduced. Data accessing time can be automatically reduced because of the minimization of the searching time, so users can pay less money for using the cloud services. In the proposed scheme, the overhead of the system is reduced, and the CSP can easily monitor and maintain the database. Both experimental results and theoretical analysis show that the proposed scheme can be more efficient than the existing schemes. In future, the proposed scheme can be improved by developing a new data security scheme for user's sensitive data.

REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stocia, and M. Zaharia, "Above the clouds: a Berkeley view of cloud computing," Technical Report No. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
2. Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Internet Services and Applications*, Vol. 1, 2010, pp. 7-18.
3. L. M. Vaquero, L. R. Merino, and D. Moran, "Locking the sky: a survey on IaaS cloud security," *Computing*, Vol. 91, 2011, pp. 93-118.
4. Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Communications Surveys and Tutorials*, Vol. 15, 2013, pp. 843-859.
5. S. Pearson, "Privacy, security and trust in cloud computing," in S. Pearson and G. Yee, eds., *Privacy and Security for Cloud Computing*, Springer, London, 2013, pp.

3-42.
6. S. Namasudra, P. Roy, P. Vijayakumar, S. Audithan, and B. Balamurugan, "Time efficient secure DNA based access control model for cloud computing environment," *Future Generation Computer Systems*, 2017. http://dx.doi.org/10.1016/j.future.2017.01.017.
7. E. Aguiar, Y. Zhang, and M. Blanton, "An overview of issues and recent developments in cloud computing and storage security," in J. H. Keesook, Y. C. Baek and S. Sejun, eds., *High Performance Cloud Auditing and Applications*, Springer, NY, 2014, pp. 3-33.
8. A. Majumder, S. Namasudra, and S. Nath, "Taxonomy and classification of access control models for cloud environments," in Z. Mahmood, ed., *Continued Rise of the Cloud*, Springer, London, 2014, pp. 23-53.
9. S. Namasudra and P. Roy, "A new secure authentication scheme for cloud computing environment," *Concurrency and Computation: Practice and Exercise*, 2016. DOI: 10.1002/cpe.3864
10. B. Balamurugan and P. V. Krishna, "Extensive survey on usage of attribute based encryption in cloud," *Journal of Emerging Technologies in Web Intelligence*, Vol. 6, 2014, pp. 263-272.
11. S. Namasudra and P. Roy, "Secure and efficient data access control in cloud computing environment: a survey," *Multiagent and Grid Systems*, Vol. 12, 2016, pp. 69-90.
12. S. W. Huang, C. K. Shieh, C. C. Liao, C. M. Chiu, M. F. Tsai, and L. W. Chen, "A cloud-based efficient on-line analytical processing system with inverted data model," in *Proceedings of the 11th EAI International Conference on Heterogeneous Networking for Quality*, *Reliability*, *Security and Robustness*, 2015, pp. 341-345.
13. S. Namasudra and P. Roy, "Time saving protocol for data accessing in cloud computing," *IET Communications*, DOI: 10.1049/iet-com.2016.0777, 2017.
14. S. Namasudra, S. Nath, and A. Majumder, "Profile based access control model in cloud computing environment," in *Proceedings of International Conference on Green Computing*, *Communication and Electrical Engineering*, 2014, pp. 1-5.
15. S. W. Huang, T. C. Huang, S. R. Lyu, C. K. Shieh, and Y. S. Chou, "Improving speculative execution performance with coworker for cloud computing," in *Proceedings of the 17th IEEE International Conference on Parallel and Distributed Systems*, 2011, pp. 1004-1009.
16. B. Balamurugan, P. V. Krishna, G. V. R. Lakshmi, and N. S. Kumar, "Cloud cluster communication for critical applications accessing C-MPICH," in *Proceedings of International Conference on Embedded Systems*, 2014, pp. 145-150.
17. S. Namasudra and P. Roy, "Size based access control model in cloud computing," in *Proceedings of International Conference on Electrical*, *Electronics*, *Signals*, *Communication and Optimization*, 2015, pp. 1-4.
18. B. Balamurugan, P. V. Krishna, N. S. Kumar, and G. V. R. Lakshmi, "An efficient framework for health system based on hybrid cloud with ABE-outsourced decryption," in L. P. Suresh, S. S. Dash and B. K. Panigrahi, eds., *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, Springer, India, 2014, pp. 41-49.
19. Y. Zhu, H. Hu, G. J. Ahn, D. Huang, and S. Wang, "Towards temporal access control in cloud computing," in *Proceedings of IEEE INFOCOM*, 2012, pp. 2576-2580.
20. C. Danwei, H. Xiuli, and R. Xunyi, "Access control of cloud service based on

UCON," in *Proceedings of the 1st International Conference on Cloud Computing*, 2009, pp. 559-564.
21. X. Gao, Z. Jiang, and R. Jiang, "A novel data access scheme in cloud computing," in *Proceedings of the 2nd International Conference on Computer and Information Applications*, 2012, pp. 124-127.
22. Y. Wu, V. Suhendra, and H. Guo, "A gateway-based access control scheme for collaborative clouds," in *Proceedings of the 7th International Conference on Internet Monitoring and Protection*, 2012, pp. 54-60.
23. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of IEEE INFOCOM*, 2010, pp. 1-9.
24. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, Vol. 9, 2006, pp. 1-30.
25. C. Hota, S. Sanka, M. Rajarajan, and S. K. Nair, "Capability-based cryptographic data access control in cloud computing," *Advanced Networking and Applications*, Vol. 1, 2011, pp. 1152-1161.
26. R. A. Crues, "Methods for access control: advances and limitations," http:// www.cs.hmc.edu/~mike/public_html/courses/security/s06/projects/ryan.pdf, 2015.
27. S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati "Over-encryption: management of access control evolution on outsourced data," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, 2007, pp. 123-134.
28. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Towards secure and dependable storage services in cloud computing," *IEEE Transactions on Cloud Computing*, Vol. 5, 2012, pp. 220-232.
29. The HJO3 project, http://www.hjo3.net/bytes.html, 2015.
30. M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, Vol. 13, 1999, pp. 393-408.
31. R. N. Calheiros, R. Ranjan, and A. Beloglazov, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software-Practice and Experience*, Vol. 41, 2011, pp. 23-50.
32. Java, http://java.com/en/download/index.jsp, 2016.
33. Apache Commons Math, http://commons.apache.org/proper/commons-math/download_math.cgi, 2016.

Suyel Namasudra received his B. Tech. degree in Computer Science and Engineering from Institute of Engineering and Technology, Uttar Pradesh, India in 2012. He received his M. Tech. degree in Computer Science and Engineering from Tripura University, Tripura, India in 2014. Currently, he is pursuing Ph.D. degree in Computer Science and Engineering from National Institute of Technology Silchar, Assam, India. His research interests include cloud computing, information security and distributed computing.

Pinki Roy received her B. Tech as well as M. Tech. degree in Computer Science and Engineering from Dr. Babasaheb Ambedkar Technological University, Maharashtra, India in 2002 and 2004, respectively. She received her Ph.D. degree in Computer Science and Engineering from National Institute of Technology Silchar, India under IITG-NITS MOU scheme. She is currently working as an Assistant Professor in Department of Computer Science and Engineering at National Institute of Technology Silchar, India.

She is having several publications in International Conferences and Journals. She is the recipient of Rastriya Gaurav award, Young scientist award, Bharat excellence award and Best Indian golden personalities award. She has also received distinguished alumnus award from Dr. Babasaheb Ambedkar Technological University, Maharashtra, India. Her research interests include cloud computing, distributed computing, information security, speech processing and machine intelligence.