A Secured Access Control Technique for Cloud Computing Environment Using Attribute Based Hierarchical Structure and Token Granting System

Balamurugan Balusamy[1], P. Venkata Krishna[2], G. S. Tamizh Arasi[3], and Victor Chang[4]
(Corresponding author: Balamurugan Balusamy)

School of Information Technology and Engineering & VIT University[1]
Vellore, Tamil Nadu, India.
(Email: balamuruganb@vit.ac.in)
Department of Computer Science & Sri Padmavathi Mahila Visvavidyalayam, University[2]
Tirupati, Andhra Pradesh, India
School of Computer Science and Engineering & VIT University[3]
Vellore, Tamil Nadu, India
Leeds Beckett University, UK[4]
(Received Mar. 01, 2016; revised and accepted June 10 & July 12, 2016)

Abstract

Cloud computing has drastically condensed the computational and storage costs of outsourced data. The existing access control techniques offer users access provisions centered on the common user attributes like Roles, which reduces the fine-grained access measure. The paper defines a Storage Correctness and Fine-grained Access Provision (SCFAP) scheme, that provides the user an exclusive access through the use of a hierarchical structure which is a combination of users unique and common attributes. Also, we deploy the concept of Token Granting system that allows the users to verify the correctness of outsourced data without the retrieval of the respective files. The tokens are derived from the metadata containing file location that helps in the process of storage correctness verification and improvises the storage efficiency. The experimental results show SCFAP has improved storage efficiency and error recovery measures than existing techniques.

Keywords: Access control, access structure, barrier limits, storage efficiency, token granting system

1 Introduction

Cloud computing is one of the widely used emerging technique that offers various methods to acquire and manage IT resources on a large-scale [19, 22]. Cloud computing, in turn, provides different types of services such as Infrastructure-as-a-service (IaaS) also sometimes called as hardware as a service (HaaS) [1, 7], Platform-as-a-service (PaaS) and Software-as-a-service (SaaS). Cloud computing planning promotes the resource sharing in a pure plug and provides a model that dramatically simplifies its infrastructure. The major advantage of cloud computing includes ease-of-use and cost-effectiveness in accessing the resources over the Internet. Employing the resources in the cloud provides greater expediency to the user because of its systematic manner. Cloud helps us to make use of the existing technologies such as virtualization, service-orientation and grid computing in large-scale distributed environment [4, 5]. To assure the cloud data integrity and availability, efficient approaches that enable storage correctness assurance on behalf of cloud users have to be premeditated. Hence, cloud operations should also imperatively support the dynamic features that make the system design even more challenging.

As Cloud computing is a new emergent technology despite having many beneficial factors, it faces many threats in various ways. It has spread very fast due to its flexibility over ease of access as it eliminates the need for extra hard drives and memory space allocation. As the cloud is a distributed system, the data stored in it is widespread in distinct locations, and it is accessed anywhere. The distributed nature of the data creates the requirement for high security over outsourced data as there exists a probability that anyone can exploit the outsourced data. The hackers [1, 2, 16], can also access the outsourced data by hacking any server virtually, and the statistical results

showed that one-third of the breaches happened from stolen or lost laptops exposing the data unintentionally from the users or the employee of the organization over the Internet. Further, nearly 16 percent of this data exposure is due to the insider theft. The cloud security providers were even trying to provide a solution to security problems such as security, privacy, reliability, legal issues, open standards, compliance, freedom and long-term viability.

Cloud affords three major types of deployment models, which comprises of Public, Private, and Hybrid Cloud. Most-common level people and some organizations make use of the public cloud model in a majority for data storage purposes because it consumes less cost and correspondingly provides utmost security over the outsourced data, but there is also a probability of data leakage in a public cloud environment. The private cloud model [9, 13], depends upon a particular firm but found to be comparatively costlier than the public cloud. The combination of either private- public or public-public or private-private infrastructure forms the Hybrid cloud environment [12, 15], providing the combined advantage of both the private and the public cloud. The significant benefit of the use of the hybrid cloud involves improvised security with lesser management costs.

The possession of fine-grained data access control and storage correctness verification remains to be a mandatory feature in any system, which shares the data contents among multiple users with different level of trust. To ensure the property of cloud data security, highly trusted cloud users might be allowed with full access rights while the other users were assigned partial access rights over the outsourced data. Efficient management of the fine-grained access provision in a system with users having different access privileges remains to be a challenging issue in cloud computing.

To provide better security features in cloud computing environment, a novel Storage Correctness and Fine-grained Access Provision Scheme (SCFAP) is given. It comprises of two parts, where the first part designates the access structures to the users and the second presents a storage correctness scheme through the use of the access structure defined at the preliminaries. A combination of public key, private key, and access structures is assigned to all the users of the system that is derived from the appropriate user attributes. Through the distributed keys and access structures, every single user of the system establishes the secure cloud connection and performs accesses to the cloud data. For every successful cloud data upload, the user is provided with a token, which is used to verify and validate the storage correctness associated with the outsourced data thereby improving the storage efficiency.

The paper is organized in the following manner. The section next to introduction details the literature survey, the next part, deals with the summary of limitations followed by preliminary concepts and algorithms, system design, the proposed SCFAP scheme, case study, Implementation Details, Results, and Discussion. The paper is ended with the conclusion and future work.

2 Related Works

This section describes and analyzes other approaches towards facing the challenge of fine-grained access provision to cloud users. Multiple solutions are examined, after which an overview of their works was given. This section also describes the comparison of two major approaches that is related to the fine-grained access provision techniques.

2.1 Overview

This section presents an overview of the works, which is related to the proposed SCFAP scheme.

2.1.1 Cloud based Access Control Techniques

[24] presents a data access control scheme called DAC-MAC for the multi authority cloud storage system. It provides a multi-authority CP-ABE scheme with efficient data decryption and user revocation functions. This work further offers an Extensive Data Access Control Scheme (EDAC-MACS) that provides secured user data access even at weaker security assumptions. The security analysis results of this scheme prove that this scheme is collusion resistance but lacks at the property of fine-grained access provision to the individual users of the system. In work done by [25, 10], integration of cryptographic techniques with RBAC techniques was made and it uses role keys for data decryption. Further this work presents a hybrid cloud architecture, where the public cloud contains the basic level details and most sensitive information over the private cloud. This work separates the property of user delegation to active and passive types and establishes effective role management through the use of delegation servers and protocols. The Cipher text-Policy Attribute-Based Encryption was given by; it realizes the complex access control mechanisms over the encrypted data [23, 14]. Here the attributes expressed solitarily the user credentials and the person who encrypts the data could fix the access limit to the users for data decryption. Through the use of this scheme, the data stored could be kept confidential even though it resides on the untrusted server. The ID-based cryptographic scheme [8], makes use of the user attributes such as user id for encryption and decryption process of the outsourced data. The development of ID-based cryptographic scheme provides the secured data storage over the public cloud and improved client authorization for other users to access the data content.

2.1.2 Hierarchical Based Access Control Schemes

In HASBE [17, 21], the user access rights were provided by the hierarchical access structure framed for each user of

the system. This scheme ensures the property of scalability through the extension of ASBE (Attribute-Set Based Encryption) technique [6]. It defines a hierarchical structure that delegates the operation of trusted authority and private key generation to the domain authorities of the lower level. Here the user attributes were converted into the stable structure of the recursive type that permits the users to define constraints dynamically by representing a different combination of attributes, which satisfies the user access policy. That ensures the property of flexibility and fine-grained access control over HASBE systems. The concept of Hierarchical Based Access Structure is extended to form the Hierarchical Structure used in this paper.

2.1.3 Token Based Access Verification Systems

[20] proposed a flexible distributed storage integrity auditing mechanism that consists of homomorphic tokens and erasure-coded data. Tokens are provided to the users from randomly chosen block indices from each data vector space analogous to the memory location of the user requested file in the cloud. The use of erasure coded data technique protects the user data and eliminates the system errors such as data redundancy, fault tolerance and server crashes. In Privacy-Preserving Public Auditing for Secure Cloud Storage by [18, 11] comprises a third-party auditor (TPA) for auditing the integrity of outsourced data; this eradicates the new threats and realizes the data privacy. This scheme uses random masking technique integrated with a homomorphic authenticator that ensures the privacy of public auditing. Flexible distributed storage integrity checking mechanism is proposed by [3] using homomorphic tokens and it avoids security problems like identifying unknown users. Through the use of homomorphic tokens and distributed erasure coded data, users were permitted to audit the outsourced data. This auditing allows the users to identify both the improper data access and cloud server misbehaviors. This scheme even ensures the cloud data security, which allows the users to perform dynamic operations efficiently over the outsourced data. Experimental analysis of their proposed scheme proves that it provides high efficiency against Byzantine failure, unknown user attacks and attacks on cloud data modification. Access control schemes based on the token system were developed to provide greater security over the cloud storage systems.

2.2 Comparison of Related Works

This section presents a brief summary about two major approaches relating to the proposed SCFAP scheme. The HASBE scheme given by [17], and the flexible integrity auditing mechanism provided by Wang Cong et al, were taken into comparison, and it is described as follows:

2.2.1 Work by Wan Zhiguo et al.

To ensure the property of scalability and flexibility over outsourced data, a solution is presented in work done by [17]. This work shows a Hierarchical Attribute-Set-Based Encryption (HASBE) scheme to cloud users, which extends the property of Cipher-text attribute-set-based encryption technique. This scheme not only aims in the achievement of scalability, it even inherits the property of flexibility and fine-grained access provision through the management of compound attributes. The HASBE scheme makes use of multiple value access expiration time to deal with user revocation problems. The first part of this work describes the extension of HASBE from ASBE technique using the hierarchical structure. Whereas the second part provides a clear demonstration of the implementation of access control scheme based on HASBE for cloud computing.

The cloud computing system considered in this work consists of five major entities. The cloud service provider provides services to users. The data owners share their data contents through the cloud in an encrypted manner. Data consumers decrypt the shared contents to perform their respective access operations. Each data owner and data consumer was assigned with a domain authority, where each domain authorities could be managed through parent domain authorities or trusted domain authorities. The major responsibility of every domain authority is to administer the domain authorities at next level or the data owner or consumer in its domain. In HASBE scheme the data users were only assumed to possess read access. All the entities associated with this scheme were organized in a hierarchical manner to accomplish their tasks.

A recursive set based key structure is formed for every user, where each element of the set is either a set or an element corresponding to a user attribute. The depth of the key structure is found using the level of recursions in the recursive set, which is similar to the definition of depth tree. For a key structure of depth 2, members of the set can either be sets or attribute elements at depth1. At depth 2 it is mandatory that all the members of the set should be of attribute elements. A unique label for the user attributes was formed using key structure. The access structure to the users in HASBE was formed in a similar way to the ASBE scheme given by [3]. In access tree structures the leaf nodes were considered to be the attributes, and non-leaf nodes represent the threshold gates. The non-leaf nodes were defined using its children and threshold values.

This work provides user access provision with the help of the hierarchical access structure, and it is formed using appropriate user key structure and access structures. It means that the user with private key corresponding to attributes in key structure would be able to access the data, only when their attributes satisfies the access policies defined by the access structure. System Setup, Top-Level Domain Authority Grant, New Domain Authority/User Grant, New File Creation, User Revocation, File Access,

and File Deletion are the seven major operations associated with Wan Zhiguo et al HASBE scheme. Each major system operations related to the HASBE scheme invokes the appropriate algorithms associated with it to accomplish their tasks, and it works by bilinear mapping concepts. Through the use of this operations, every user of the system shares and uses their data contents using HASBE scheme.

Though Wan Zhiguo et al system provides a better solution to scalability and flexibility issues, the complete support for compound values and multiple value assignments are measured and found to be lagging in efficiency. Which reduces the level of fine-grained data access. The proposed SCFAP scheme defines users with their role-based classification. Provides efficient support for compound attributes and multiple value assignments. The hierarchical structure described in SCFAP scheme improves the level of fine-grained access provision associated with individual users of the system. The HASBE scheme further does not allow write access to the data users of the system. This makes its application inappropriate to critical systems like financial sectors, where several users require write operations to be performed. SCFAP scheme allows the users to perform write operations in an effective manner, and it is achieved through the use of token granting system, which preserves the storage correctness of the outsourced data.

2.2.2 Work by Wang Cong et al.

An approach to form solution for security risks accompanying the correctness of physical possession over outsourced data were done by [17]. This work presents a flexible distributed storage integrity auditing mechanism, which ensures the correctness of outsourced data through the use of homomorphic token and distributed erasure-coded data. This scheme provides efficient user auditing of cloud data with very lightweight communication and computation cost. The auditing result provides both storage correctness guarantee as well as fast data error localization (identification of server misbehaviors). It even allows user access operations over outsourced data including block deletion, modification and appends functionalities. The overall contributions of this work is summarized as follows:

1) In comparison to many of its predecessors, this scheme achieves both the storage correctness insurance and data error localizations.

2) This scheme further supports secure and dynamic operation over data blocks including update, delete and append.

3) The work further makes an extensive security analysis that shows its resistance towards Byzantine failures and malicious data modification attack and server colluding attacks.

The flexible integrity auditing mechanism discussed in this section consists of four major entities, which includes User, Cloud Service Provider (CSP), Cloud Server (CS) and Third Party Auditor (TPA). Users share their data through cloud storage services, and a user can be either enterprise or an individual customer. Cloud Server (CS) is managed by the CSP to provide better computation and storage facilities to the users of the system. TPA is an optional entity with expertise qualities that user does not possess. TPA assesses and describes the risk of cloud storage services on behalf of users upon request. This work provides more focus towards file oriented data rather than non-file oriented applications like social networking systems. Block level operations over user data were considered as block update, block delete, block insert and append operations. The major focus of this work is to identify the key integrity issues like unauthorized data modifications and corruptions, caused due to server compromises and random Byzantine failures.

Users store their valid credential to cloud servers through CSPs. The problem of data redundancy could be employed through the technique of erasure correcting code. This scheme further tolerates faults and server crashes that happen due to increasing data users. The users interact with cloud servers for processing file retrieval request through CSPs. As it is not feasible for the users to possess their data locally, it is necessary to verify the correctness and maintenance of the cloud data. The users were provided with the pre-computed tokens that provide correctness assurance to the users of the system. Tokens are derived from the subset of file blocks in a random manner. The verification token helps the users to ensure correctness of data operation request processed by the CSP. Tokens are issued to the user based on randomly chosen block indices from each data vector space corresponding to memory position of the requested file in the cloud and erasure-coded data. In cases of inappropriate situations like insufficient resources and time the users can delegate their responsibilities to TPA. The system is designed in such a way that leakages of user? Outsourced data towards auditing protocol were prohibited. This work achieves secure data storage through five major steps, which includes file distribution preparation, challenge token pre-computation, correctness verification and error localization, file retrieving and auditing and finally, towards third party auditing. The algorithms associated with each stage helps in the management of activities accompanying data storage management and correctness verification processes. This scheme provides an approach methodology that prevents CSP to process data dynamics without knowing user secret key materials and ensures users that dynamic data operation request done by CSP were processed faithfully. In this manner the property of integrity assurance and storage correctness where done in [17] scheme.

Tokens were provided to the users, based upon randomly generated block indexes and memory position of the file. This makes the property of storage correctness

associated with integrity auditing scheme to be a probabilistic feature. The proposed SCFAP scheme solves this issue by granting tokens to the users in a deterministic manner. In SCFAP scheme tokens were derived from Metadata containing file locations and distributed to all the users of the system during appropriate phases. Here the major advantage is that as a result of the write operation done by the authorized system an updated token would be provided to all the users of the system, through which the property of storage correctness is achieved.

3 Construction of Storage Correctness and Fine-grained Access Provision Technique

3.1 System Design

This section presents a conceptual design of the novel scheme called Storage Correctness and Fine-grained Access Provision (SCFAP) scheme, which is described in Figure 1. The proposed SCFAP scheme consists of two parts. The first part deals with the construction of hierarchical based user access structures and the second part depicts the algorithmic phases associated with SCFAP scheme that helps in the achievement of fine-grained data access and improved storage efficiency across the outsourced cloud data storage. A set of appropriate cryptographic keys and access structures derived from the exact user attributes were distributed to all the users of the system. Through the use of the access structures and cryptographic keys every user of the system performs the cloud data access in a secure way. As a result of the encryption process, both the data owners and users were provided with a token, which assists in the process of integrity and security verification over the outsourced data.

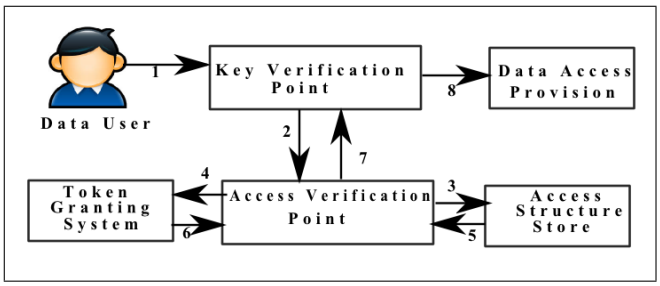

Figure 1: System design of SCFAP

The proposed system consists of five major entities and the description to the entities were given as follows, Attribute Authority (AA): The major responsibility of the Attribute Authority (AA) is to manage all the attribute related activities in specialization with the activities confining to the management of user roles. This includes maintenance of role revocation, delegation, key allocation to users and authentication of the user given credentials like the public key, private key, etc. Cloud server (CS): Cloud server performs all the computation related activities. This includes the computation of user given inputs and producing corresponding computational results and acknowledgments to the users. Cloud Service Provider (CSP): The Cloud Service Provider (CSP) provides services to the users of the system and performs the validation of the user given inputs and outputs during the process of encryption and decryption. Service Consumers (Users): A Service consumer is also called as the user of the system, consumes the services provided by the cloud computing environment. Data Owner: Shares valid data contents over cloud computing environment and fixes data access limits across data users.

3.2 Assumptions

This work assumes an existing data access control model to build upon, and the proposed design makes use of the access control properties defined previously at related works. The hierarchical structure described in this paper is assumed to provide many-to-many data sharing in a secured manner through which the property of fine-grained access control, confidentiality, and non-repudiation of the outsourced data was achieved.

3.3 Key Terminologies

3.3.1 Access Assignment Structure

A summary on Access assignment structure is depicted in Figure 2.

3.3.2 Hierarchical Structure

The hierarchical structure defines the access policy associated with the individual users of the system. A hierarchy is framed from the combination of the user unique and common attributes. Each hierarchy represents the one to one relationship between the user and their access policies. The access policy defines the set of operations (read or write access) the user could perform over the outsourced data.

3.3.3 Key Structure

Key structures were designed to preserve the security of the outsourced data. Key structures are derivatives from the user common attributes like roles. The formation of key structure assigns the access privileges to the set of the common users over the outsourced data. This states that users beneath a particular role were assigned with a key structure such that they could gain access to a particular set of files.

3.3.4 Access Structure

Access structures were designed to achieve the property of fine-grained user access and it is derived from the user unique attributes like user id. It defines the extent to which an individual user could access the data.

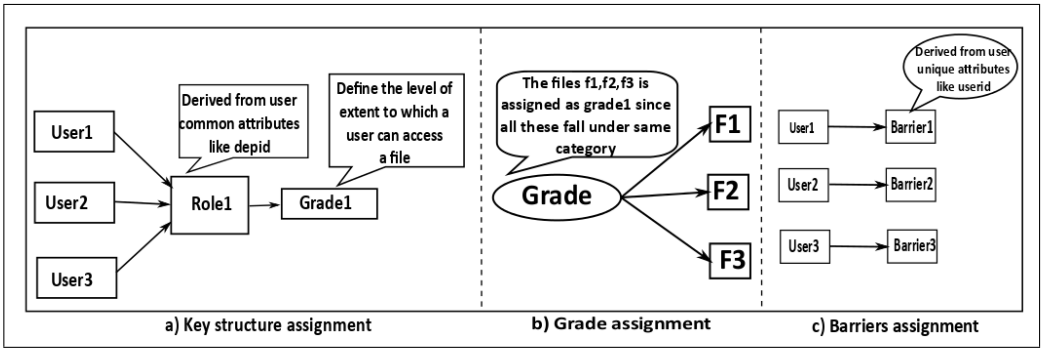

Figure 2: Access assignment structure

3.3.5 Grade

Grade denotes the level of the extent to which the set of common users could gain access to a particular set of files. Each grade formally represents a key structure, such that a user with certain grade could gain access to all the files that comes with the scope of a particular grade. Grades were derived from the user common attributes like dep id, such that it represents a set of files that belongs to the particular department. An example of the measure of a grade is described in Figure 3.

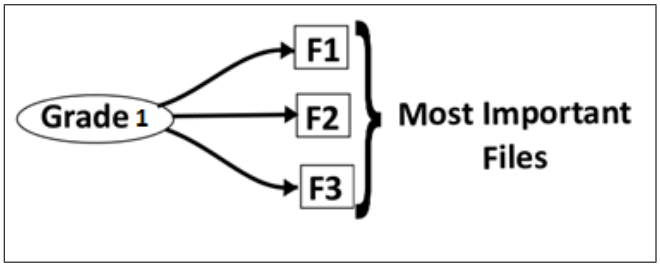

Figure 3: Access limits associated with a grade

3.3.6 Barriers

Barriers are restrictions imposed over the grades to achieve the fine-grained user access level. It has been found that it is not necessary for a user with a particular grade to access all the files that come under a particular grade. To solve this issue, barriers were designed and imposed over the user grades.

From Figure 4. It is understood that though the user belongs to grade1 which provides access rights to three files F1, F2, F3. The imposition of the barrier B1 over the particular user grade G denies the user file access request to the file F2; such that the user could only be able to access the files F1 and F2. It provides the appropriate access rights to the users through which the property of fine-grained access provision is achieved.

3.3.7 Tokens

Tokens were derived from the metadata containing the file location and it acts as a user authentication entity. To-

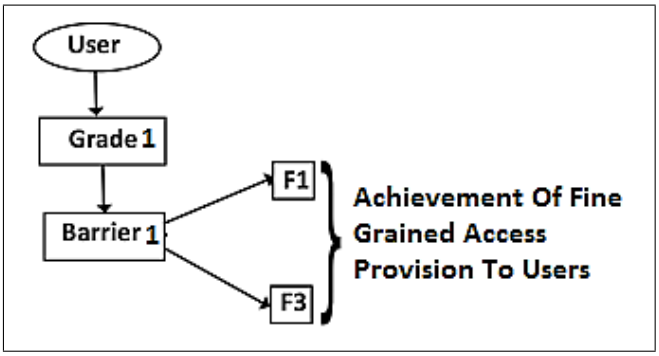

Figure 4: Barrier generations

kens were issued to the data users as a result of the data encryption process. Through the use of the tokens, the user could easily verify the existence of their corresponding files in a convenient manner. Since the token represents the Metadata about the file location, it assists in the process of easier file retrieval. This improves the storage efficiency of the proposed system. Further, clear descriptions about the working of the tokens were described in phase 6, 7 and 8 of the SCFAP scheme. A new method of mathematical modelling was used to identify the functions and variables of the proposed scheme.

4 Preliminary Concepts and Algorithms

4.1 KeyGen()

KeyGen() is a basic algorithm for key generation through which all the other keys associated with the data users were derived. This algorithm is invoked automatically whenever the process of key generation is required.

Let us consider the set of keys k such that it contains a set of integers up to n. Such that $K_n = \{k_1, k_2, \cdots, k_n\}$.

$$k_n : \sum_{Z=0}^{n} K_Z \approx \sum_{m=0}^{n} d_m.$$

Where d is the set of derived keys. It could be of any other keylike master key, public key and private key. In

a simpler way the KeyGen algorithm generates random number of keys in accordance to the user given input parameters.

4.2 Formation of Hierarchical Access Structure

The proposed SCFAP scheme makes use of the hierarchical access structure to define the user access rights. The basic concept behind the hierarchical access structure was described in the previous section of the paper. In SCFAP scheme each user is assigned with a hierarchical structure, which is derived from their respective key and access structures.

Key structure is premeditated to preserve the security of the outsourced data and it represents the access rights to the group of users with a common identity. The basic concepts behind the formation of the key structure were given in the previous section. It is formed using the user common attributes like dep id. In an organization the most important or most secured files could be accessed only by the personals at the top-most designation order, least important files by the low-level personals and ordinary files could be accessed by the mid-level individuals. In correspondence to the user designation order grades for a group of members with common identity (users under a particular role) is calculated from "grade1.grade n". For every user with a role, grades were allocated with respect to their access privilege that defines the level of extent to which the users could access the data.

4.2.1 Access Structure

The access structure represents the access rights to the individual user of the system. Even though a particular user is assigned with a grade representing the key structure, it is not mandatory that the user could access all the files that come under a particular grade. The access structures associated with the SCFAP scheme were designed in such a way that solves the problem of above mentioned issue. The access structure was framed from the user barrier limits, which are derived from the user unique attributes like user id. Barriers are restrictions that were imposed over the user access grades to achieve the fine-grained access control. The assignment of the access structure defines the individual access limits over the set of files. In addition to this, phase 3 of the storage correctness scheme provides a brief summary about the algorithmic implementation of the user access structure assignment.

Through the use of the key and access structure discussed above, a hierarchical access structure is formed in the proposed SCFAP scheme, and it is illustrated in Figure 2.

4.2.2 Token Granting System

The proposed SCFAP scheme makes use of the token granting system through which the property of storage correctness is achieved. As it is described at the previous section tokens were derived from the Meta data containing the file location that assist in both ways, through which the process of storage correctness as well as the easier retrieval of the outsourced files could be made. The prime idea behind the use of token granting system in SCFAP scheme is that at the end of every successful data encryption process the data users were provided with the tokens, through which the data users verifies the existence of the outsourced data. The users could also be able to perform the decryption process only when the Meta data of the user given token points to the user requested file.

4.3 SCFAP Phases

The storage correctness phases and fine-grained access provision scheme consists of nine phases through which the property of fine-grained access provision and storage correctness verification is achieved. The SCFAP phases apply the concept of hierarchical access structure and token granting system described in preliminaries part.

4.3.1 Phase 1: SetUp()

It takes the user security parameter λ as an input and generates master key m_k as an output. This step is done by the cloud server through automatically invoking the KeyGen algorithm.

$$K : m_k = \lambda \bowtie k_n. \tag{1}$$

Equation 1 joins the user security parameter with the unique key generated by KeyGen() algorithm and distributes the master key to the corresponding users of the system.

4.3.2 Phase 2: GradeGen(m_k, R_{id})

This phase is performed by the Attribute Authority and it takes the master key m_k and Role id R_{id} as an input, produces public key p_k and grade g as an output. Public key is derived from the master key m_k by manually invoking the KeyGen() algorithm. Let us consider two sets, $R = \{R_1, R_2, R_3, \cdots, R_n\}$ and $G = \{g_1, g_2, g_3, \cdots, g_n\}$ be the set of roles and grades. Such that $R \approx G$ (means that the role R is isomorphic to grade G).

$$Z : \forall R_{id} \in R | R_{id} \subseteq R.$$

Any R_{id} that belongs to R is the subset of R.

$$Z : \forall R_{id} : P(R_{id}) | R_{id} < \bullet R. \tag{2}$$

At least for one value of R_id the value of R_id in R is true. Such that R_id is covered by r where $r \in R$.

$$\therefore Z : \exists R_{id} \rightarrow r | r \Leftrightarrow G.$$

There exists G and R_{id} that implies a role such that the role corresponds to a grade G.

4.3.3 Phase 3: BarrierGen(U_{id}, r_k, p_k)

It takes (U_{id}, r_k, p_k) use rid, role key, and public key as an input and as a result of computation the barrier limit b_l and the private key p_{rk} is returned to the users.

The private key is manually generated by role admin through the invocation of KeyGen() algorithm.

Let U be the universal set that contains all the users of the system and can be written as $U = \{U_1, U_2, \cdots, U_n\}$ and B be the set of barriers such that can be written as $B = \{b_1, b_2, \cdots, b_n\}$.

$$\begin{aligned} Z &: \forall U_{id} \in U_{id} \subseteq U. \\ \forall U_{id} &: P(U_{id}) | U_{id} \prec \bullet U. \end{aligned}$$

Similarly from Equation 2 this step is derived.

$$\sum_{k=1}^{n} U_k \exists B | \sum_{k=1}^{n} U_k \in B | \sum_{k=1}^{n} U_k \subseteq B.$$

Means that for all the users there exists a barrier limit such that all the users in U belong to barriers in B. So that U is the subset of B. Such that there exists an $U_{id} \in B$. where

$$Z : b = \coprod_{u \in U} bu : \coprod_{u \in U} Ru \rightarrow G.$$

Since, all the users U is the subset of B there exists a b corresponding to the user u, where the barriers can be calculated as a co-product of user and barrier sets.

4.3.4 Phase 4: Encrypt(f, r_k, p_k)

This phase is done by the CSP and it takes the file f, role key r_k and Public key p_k as an input and the outputs cipher text c_p to the users of the system. Data encryption is done as a part of file upload.

$$\begin{aligned} Z &: f \times r_k \times p_k \Leftrightarrow f^{(r_k, P_k)} \\ Z &: f \times r_k \times p_k \Leftrightarrow c.t | c.t \approx f^{(r_k, P_k)}. \end{aligned}$$

Encryption is done as a combination of input parameters.

4.3.5 Phase 5: TokGen(f, r_k, p_k)

It takes file f, role key r_k and Public key p_k as an input. It is the most important part of the encryption process and it is done during the process of file upload. Since tokens were derived from the data containing file locations, here we use the concept of reduction to reduce a file to tokens. Let $F = \{f_1, f_2, \cdots, f_n\}$ be the set of files such that by property of reduction

$$\begin{gathered} \text{if} \\ \exists f \in d_b \bullet \forall R \in F \Leftrightarrow f(R) \in t_i \\ \text{then} \\ F \leq_{db} t_i \end{gathered}$$

$F \leq_{db} t$ Denotes that a file set F can be reduced to token t_i and it is achieved through the data blocks. At the end of this phase tokens generated were distributed to the data users to verify the correctness of the outsourced data.

4.3.6 Phase 6: Token Computation

It is done by the CSP and cloud server as a part of the data decryption process. Let $F = \{f_1, f_2, \cdots, f_n\}$ be the set of files and $T_i = \{t_{i1}, t_{i2}, \cdots, t_{in}\}$ be the set of tokens associated with the files. Then,

$$\coprod [t_{i1}, t_{i2}, \cdots, t_{in}](F_i) = \{d_b[t_{i1}, t_{i2}, \cdots, t_{in}]; d_b \in F\}.$$

Where, $F_i = \{1, 2, \cdots, n\}$ (Only the tokens between $[t_{i1}, t_{i2}, \cdots, t_{in}]$ can access the files in data blocks). Tokens out of this scope would be computed as corrupted and cannot be accessed. It is based on the projection property. Where,

$$d_b[t_{i1}, t_{i2}, \cdots, t_{in}] = \{(t_i, v) \in d, t \in [t_{i1}, t_{i2}, \cdots, t_{in}]\}.$$

Means the remaining set of data blocks corresponds to some other tokens.

The result of proportion $\coprod [t_{i1}, t_{i2}, \cdots, t_{in}](F)$ can be found only if $[t_{i1}, t_{i2}, \cdots, t_{in}]$ is $\subseteq (F)$. It means a file would be accessed only when token matches with it.

4.3.7 Phase 7: Token Update

Whenever the data user performs the write operation the tokens associated with the users were updated and distributed to all the associated system entities. This is due to the reason that the process of write operation may extend or delete some part of the file that leads to the change of the Meta data containing the file location. The process of token update is described as follows:

$$newt_i = t_i \bowtie w_c.$$

Where w_c is the newly written content.

4.3.8 Step 8: Token Correctness

It is done as a part of data decryption during the process of file download. It takes (t_i, c_p) as an input. Let t_i, c_p be an algebraic function over F then $Z : t_i \in T_i; c_p \in C_p$; let us take an element $t_i \in f$. Such that,

$$\begin{aligned} Z &: (t_i, c_{p1}) + (t_i, c_{p1}) \sim (t_{i1} + t_{i2} + c_p) \\ Z &: (t_i, c_{p1}) + (t_i, c_{p1}) \sim (t_{i1}, c_{p1} + c_{p2}) \\ Z &: f(t_i, c_p) \sim (ft_i, c_p) \sim (t_i, fc_p) \Leftrightarrow t_i \otimes c_p. \end{aligned}$$

It matches the values in the token and cipher text and returns the mismatch thus the token correctness is verified.

4.3.9 Phase 9: Decryption($c_p, r_k, b_l, p_{rk}, t_i$)

Data decryption is done as a part of the file download process. It takes cipher text, role key, barrier limits, private key and token as input and returns the plain text to the users based on their respective access structures.

$$\begin{aligned} Z &: c_p \bowtie t_i = \coprod b_1(c_p \bowtie t_i) \\ Z &: c_p \bowtie t_i = \coprod b_1(c_p \bowtie t_i) | (c_p \bowtie t_i) = \coprod b_1(c_p \bowtie t_i) \Leftrightarrow P_t. \end{aligned}$$

It combines the cipher text and token depending upon the user barrier levels and provides the plain text.

Table 1: Summary of SCFAP phases

Phase No	Phase Name	Input	Output	Doneby
1	SetUP( )	λ	M_k	CS
2	GradeGen( )	M_k, R_{id}	P_k, G	AA
3	BarrierGen( )	U_{id}, R_k, P_k	B, P_rk	RA
4	Encrypt( )	F, R_k, P_k	C_t	CSP
5	TokGen( )	F, R_k, P_k	T_i	CS,CSP
6	TokenComp( )	F, T_i	C_t	CS,CSP
7	TokenUpdate( )	T_i	$newT_i$	CS,CSP
8	TokenCorrectness( )	T_i, C_t	File Validity	CS,CSP
9	Decrypt( )	T_i, C_t, R_k, B, P_{rk}	Plaintext	CSP

5 Advantages of Proposed SCFAP Scheme

1) Through the use of the barrier limits in user hierarchical access structure helps in the achievement of fine-grained access rights to the users of the system.

2) The algorithmic deployment of the token granting system helps in the achievement of the storage correctness verification of the outsourced data with improved storage efficiency.

3) The tokens were derived from the Meta data containing the file location. This reduces the file retrieval time associated with the user data access request.

6 Case Study

It has been found that government agencies adopt cloud services to meet their scaling industrial needs. Though social networking sites were found to be the major users of cloud usage, currently banking contributes to the most activities in the cloud, utilizing 64% of the overall cloud services. Factors like modernization, innovation in information technology, financial products, liberalization and consolidation of financial markets enforces the transformation of the ordinary banking system to the cloud-based one.

In cloud banking systems, the clients could access different types of banking services like balance enquiry, fund transformations, etc. The banking system taken in this scenario falls under the category of fully electronic based transaction systems, where the banking sites exhibit all the information like bank locations, bank products, loan enquiry, loan eligibility details, transaction details, statement of accounts and money transfer facilities.

The management of activities associated with this type of cloud banking system includes several entities like bank clients, cloud service providers, and entities related to the payment gateway. In such a case it is not necessary for the system management entities with common attributes like similar roles to access all the end user provided valid credentials. This creates the need for the application of fine-grained access control scheme over the cloud banking system. To solve these issues, the proposed SCFAP scheme were applied to the Cloud banking system that solved the problem of fine-grained access level associated with the individual system entities. Let us consider the situation, where the bank client performs the money transaction across the cloud banking system. The process of money transfer requires the fulfillment of several essential details including the client's valid password. Once the client completes the form filling activity, the client details were transformed to their respective financial institution cloud servers. The client transaction request could be processed through various system management entities like bank account manager, fund transfer manager, cloud service provider, etc. Though all the associated entities possess the same authority of power, it is not necessary for them to process or access all the user given credentials. During fund transfer, it is necessary that the fund transfer manager could access only the required client details like user account balance and eligibility to perform the transaction. But the fund transfer manager is no way related to the user personal and account credentials. It is appropriate for the account manager to verify and validate client account detail apart from the other user given inputs.In order to implement this restriction, a hierarchical access structure is formed in a similar way to the SCFAP scheme. In this case, a hierarchy could be formed through the system entities, common attributes like organization ID along with their unique user ID. Further, the storage correctness phases depicted in this paper could be applied to the cases resembling the need for data integrity assurance. In cloud banking, during the process of user registration the user uses their valid credentials as an input and as a result of the user registration, a login id and password were given to them that act as a token. It could be used by the clients to verify the validity of the user registered details. Thus the storage correctness of the user given inputs such as bank details were verified.

The above scenario clearly explains the application of SCFAP scheme to cloud banking system and it is illustrated in Figure 5. This scheme can also be applied to similar scenarios ranging from the medium-sized to large-sized enterprises. Our system is highly efficient in cases, where a significant number of users with similar roles needs the fine-grained access provisions and frequent ver-

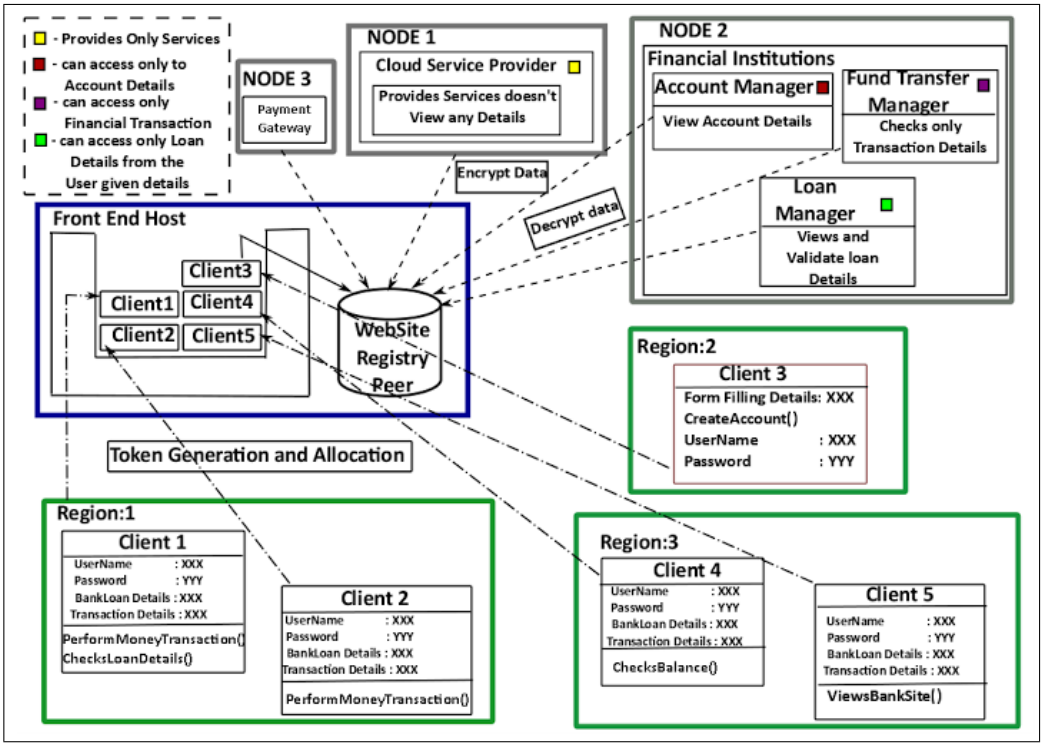

Figure 5: Cloud based financial system-A case study

ifications to the outsourced data.

7 Experimental Study

7.1 Deployment of SCFAP Over Eucalyptus Cloud

An application is created in Eucalyptus an open source cloud platform, which deploys the proposed SCFAP scheme. An interface is developed using JSP to enable users to authenticate and view the cloud storage. Eucalyptus consists of several other interfaces like cloud42, tAWSTanacasino, EC2 Dream, but it is not feasible for the proposed implementation. MySQL community server5.7 is used for storage purposes. The proposed SCFAP scheme runs at the infrastructure layer of the eucalyptus, and it works on a layered basis to accomplish its tasks. The SCFAP scheme consists of four layers such as user registration layer, authentication layer, access security management layer and instance management layer as it is described in Figure 6. The registration layer performs the user registration process. Authentication layer validates the cloudlet credentials, and the access security management layer allows or denies the user file access requests to the virtual machines with the aid of the functionalities present in the instance management layer. The application consists of the Command Line User Interface (CUI) using Euca2ools, which allows the users to interact with the system. Here the users of the system were considered to be the subjects and the files uploaded to the cloud were assumed to be the objects. Every subject creates the newer objects and requests their corresponding object access through the proposed SCFAP scheme that preserves the storage correctness and fine-grained access provision of the user data. The security management layer controls and directs the access control schemes. The implementation consists of the web interface that possesses the property of ease of use through which the Cloud Service Providers (CSP) or Attribute Authorities (AA) creates the restriction over the cloud instances. The implemented SCFAP scheme allows the AA to manage access to cloud resources, instances, virtual machines and common user groups associated with the cloud computing environment.

The first step associated with the implementation of SCFAP scheme over eucalyptus cloud consists of the setup phase. Once the subject is registered with the system, the AA adds the subject to the common user groups. This states that all the subjects under the common user group contain the common access privileges with particular individual restrictions. These restrictions were imposed on the subjects through the assignment key structure and access structure to the subjects. Every subject can create new objects and gain access to existing objects based upon the following access restrictions:

1) Grades Defines the level of the extent to which a common user group can access.

2) Barriers Defines the level of extent through which an individual user can access.

3) Tokens Derived from Meta data containing file location.

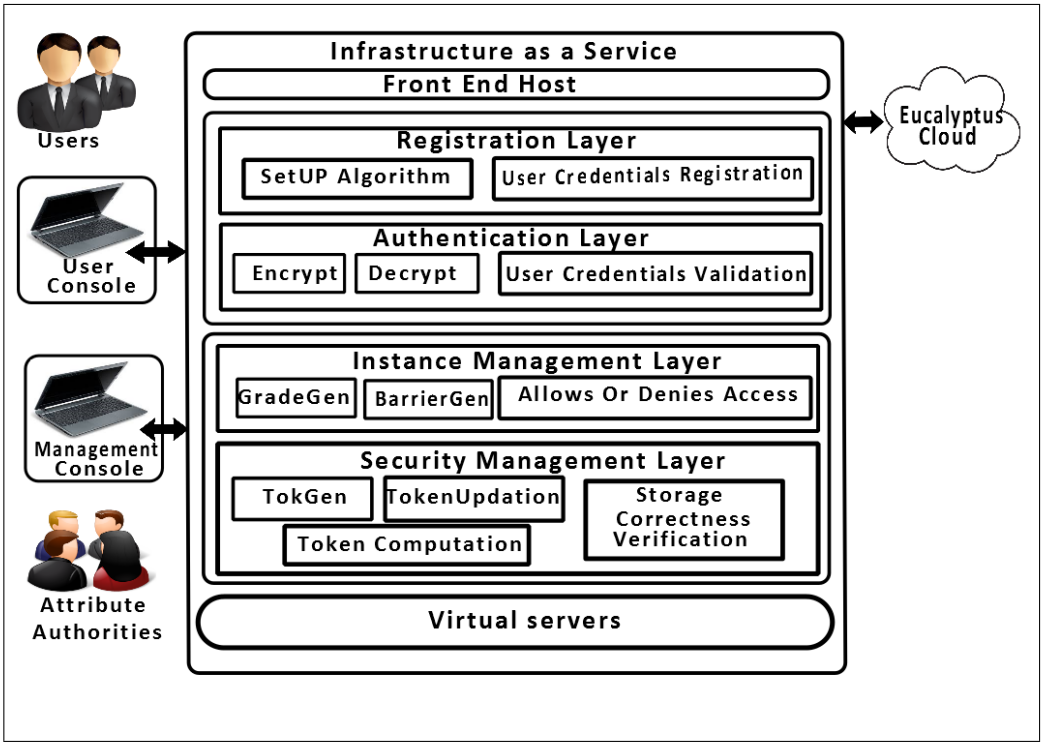

Figure 6: Architecture of SCFAP over eucalyptus cloud

Each subject shares their newly created objects to the other subjects of common user groups. The subject can also share their objects to other Common User groups which they were not a member. To gain access to the shared objects and instances the subject accessing the object could not violate the SCFAP access policy. Each subject could ensure the property of integrity over their respective subjects through the use of token granting systems. An updated token would be distributed to every subject associated with the shared object in case of modifications to the shared data objects. In this manner, the subject could gain access to the outsourced objects.

The implemented SCFAP consists of two distinct types of interfaces that include subject management and object management interfaces. The subject management interface assists in the management of all the subject related activities that include subject authentication, user common group assignments, subject key allocation, and management. The object management interface is responsible for the process of management of all the object-related activities that include object storage, object retrieval, token generation, token computation, and management.

7.2 Validation of Premises

Built on the open source cloud platform Eucalyptus the application of SCFAP scheme to the developed prototype could be validated through the verification of premises that happens in four steps, which are described as follows:

1) Each authenticated subject should be assigned to the common user group.
2) Each subject should be assigned with an appropriate hierarchical access structure.
3) Token computation results should match with the user given token and the user given file access request.
4) Encryption and decryption processes could be performed when the subject given inputs are valid.

7.3 Access Verification Tests

The first test comprises the access request to the object from the subject who is not the member of the any of the common user group associated with the developed system. The request would be denied by the setup algorithm present at the authentication layer. The next test comprises the file access request from the subject with inappropriate hierarchical access structure. This request is blocked by GradeGen and AccessGen algorithms functioning at the access security management layer. A subjects access request for an object with invalid user credentials like invalid tokens and the secret key is denied, and the subject is blocked from accessing the services if he repeats the same for three times. The request for encryption or decryption processes with inappropriate cryptographic keys or inputs by the subjects is blocked through the several algorithms like Setup, GradeGen, BarrierGen, Token computation and Encrypt or Decrypt algorithm present at the user registration layer, Authentication Layer Security management layer and instance management layer of the developed prototype. In this

manner the exception handling capabilities of the developed prototype where clearly described using the system access verification tests.

8 Results and Discussions

The major objective of the experimental implementation is to validate the level of an extent to which the proposed SCFAP scheme provides the property of fine-grained data access to the cloud users. To validate this objective prototypes using traditional access control, techniques like ABE and RBAC were implemented, and it is compared with the proposed SCFAP scheme. From the results of the implementation, a comparison is made regarding both system performance and fine-grained access provision, which is described in Figure 7 and 8. First, a comparison is made between the amount of data to be retrieved and file retrieval time to find the system performance. At client side, n numbers of client nodes were created, and a large number of files from different client node was uploaded to the cloud storage. Client file access requests were given from various nodes and the number of client requests per minute was calculated regarding size of the data files accompanying the client requests and it is kept as X limits. The time taken by the cloud server to respond to user file access requests was calculated in seconds, and this forms the Y limits. It has been observed that the time taken for file retrieval on the size of the data file for our SCFAP scheme remains constant up to a particular threshold. Even though there is a tremendous increase in file size that happens after a particular threshold, the time taken for file retrieval increases in a consistent manner. But the observation of traditional access control schemes like ABE and RBAC deviates highly and takes more time for file retrieval after a particular threshold. This is due to the inconsistent nature of their underlying access policies. The comparison between SCFAP with traditional ABE and RBAC techniques regarding file retrieval time proves that the proposed SCFAP scheme takes reduced file retrieval time than the existing schemes. This is due to the use of the token granting systems. Since the tokens were derived from the Meta data containing the file location, the time taken for file retrieval and storage correctness verification has been comparatively improved. The overall simulation results depict that the file retrieval has been reduced by 0.5 seconds in comparative to the existing access control techniques. This improves the overall performance measure of the system.

A measure to the level of fine-graininess associated with the SCFAP scheme in comparison to the traditional access control methods like ABE and RBAC were made, which is depicted in Figure 7.

The level of fine-grained access control is measured by the extent to which the appropriate access rights were provided to the users of the system. User access policies based upon SCFAP, ABE and RBAC models were designed for each client nodes associated with the system.

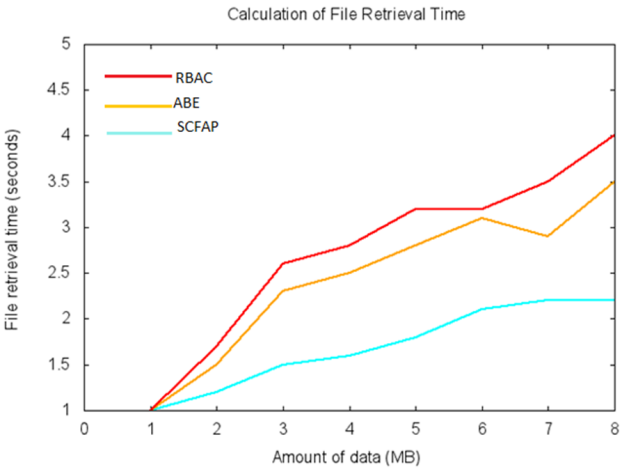

Figure 7: Comparison of file retrieval time in SCFAP scheme

Through the implementation of SCFAP scheme over the eucalyptus-cloud and the use of a vast number of the client nodes, a hierarchy is formed for every user accompanying the client node. A comparative measure of fine-grained access level has been made in association with the depth of access structure. The depth of the access structures was kept in X limits and fine-grained access level in percentage were fixed at Y limits. It has been found that our proposed SCFAP scheme provides better fine-grained access level to the data users even at lesser access structure depth. The other access control techniques taken into comparison were found to be lagging inefficiency at the lower level of access structure depth. Achieved through the derivation of appropriate hierarchical structures associated with SCFAP scheme, which provides better access provision even at the lower access structure depth. The existing technique lags at fine-grained access provision through the complex access structure formation. The tests were conducted using banking research dataset of the Federal Bank of New York.

9 Conclusion

The paper defines an SCFAP scheme that solves the problem of fine-grained access provision and storage correctness associated with the existing access control techniques. The first part of the SCFAP scheme involves the formation of hierarchical structures that fixes the appropriate access policies to the users; this improves the fine-grained ness associated with the access policy. The next part deals with the achievement of storage correctness related to the files, and it is made through the usage of the token granting system. In addition to this, the use of token granting system improves the storage efficiency, security, and performance of the proposed system. As this paper explains only on the key structure and Access Structure associated with the plain text but not about the Cipher Text Access Structure. In future, this work could

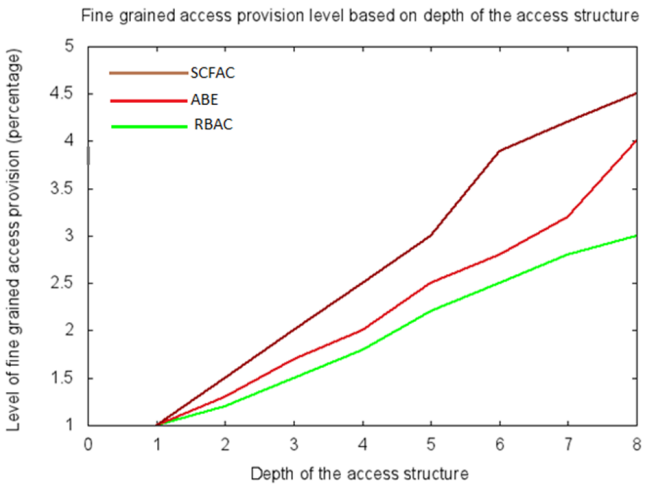

Figure 8: Comparison to fine-grained access provision measure in SCFAP scheme

be extended for outsourced data decryption techniques.

Acknowledgments

The author greatfuly acknowledge the VIT Univerity for providing us an wonderful work infrastructure.

References

[1] P. Arora, R. C. Wadhawan, and E. S. P. Ahuja, "Cloud computing security issues in infrastructure as a service," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 1, pp. 1–7, 2012.

[2] V. Bhangotra and A. Puri, "Enhancing cloud security by using hybrid encryption scheme," *International Journal of Advanced Engineering Technology*, vol. 6, no. 4, pp. 34–40, 2015.

[3] R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in *Computer Security (ESORICS'09)*, pp. 587–604, Springer, 2009.

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[5] Z. Cao, C. Mao, L. Liu, "Analysis of one secure anti-collusion data sharing scheme for dynamic groups in the cloud," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 68–72, 2016.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[7] W. Huang, A. Ganjali, B. H. Kim, S. Oh, and D. Lie, "The state of public infrastructure-as-a-service cloud security," *ACM Computing Surveys*, vol. 47, no. 4, pp. 68, 2015.

[8] N. Kaaniche, A. Boudguiga, and M. Laurent, "ID-based cryptography for secure cloud data storage," in *2013 IEEE Sixth International Conference on Cloud Computing*, 2013.

[9] R. Ko and R. Choo, *The Cloud Security Ecosystem: Technical, Legal, Business and Management Issues*, Syngress, 2015.

[10] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 425–437, 2015.

[11] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[12] O. Mazhelis and P. Tyrväinen, "Role of data communications in hybrid cloud costs," in *2011 37th IEEE EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 138–145, 2011.

[13] S. Ramgovind, M. M. Eloff, and E. Smith, "The management of security in cloud computing," in *Proceeding of IEEE Information Security for South Africa (ISSA'10)*, pp. 1–7, 2010.

[14] B. D. Revathy, M. P. Ravishankar, and C. I. T. Ponnampet, "Enabling secure and efficient keyword ranked search over encrypted data in the cloud," 2015.

[15] P. Samarati and S. De C. di Vimercati, *Cloud security: Issues and concerns*, Wiley, New York, 2016.

[16] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[17] Z. Wan, J. Liu, and R. H. Deng, "Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[18] C. Wang, S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[19] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[20] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[21] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE*

Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1265–1277, 2016.

[22] H. Witti, C. Ghedira, E. Disson, and K. Boukadi, "Security governance in multi-cloud environment: A systematic mapping study," in *12th World Congress on Services (SERVICES'16)*, 2016.

[23] Y. Wu, Z. Wei, and R. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing networks," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 778–788, 2013.

[24] K. Yang and X. Jia, "Dac-macs: Effective data access control for multi-authority cloud storage systems," in *Security for Cloud Storage Systems*, pp. 59–83, Springer, 2014.

[25] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1947–1960, 2013.

Biography

Balamurugan Balusamy is with VIT University as Associate Professor in School of Information Technology and Engineering.His research interests has evolved from cloud computing,cloud security to Big data and Ph.D thesis is on Cloud Access Control.

P. Venkata Krishna is a Professor at Sri Padmavati Mahila Visvavidyalayam, Tirupati, India. He received his BTech in Electronics and Communication Engineering from Sri Venkateswara University, Tirupathi, India, MTech in Computer Science and Engineering from REC, Calicut, India, and he received his PhD from VIT University, Vellore, India. Dr. Krishna has several years of experience working in academia, research, consultancy, academic administration, and project management roles. His current research interests include mobile and wireless systems, QoS, and Cloud computing. He has been the recipient of several academic and research awards, such as the Cognizant Best Faculty Award for the year 2009-2010 and VIT Researcher Award for the year 2009-2010. He has authored over 150 research papers in various reputed journals and conferences. He has delivered several keynote addresses in reputed conferences. He is currently serving as Editor-in-Chief for IJSGGC, Inderscience Publishers.

G. S. Tamizh Arasi is a PG scholar in VIT University and her research interests are Cloud Computing security and Cloud access control.

Victor Chang is a Senior Lecturer in Computing at Leeds Beckett University. He has been a technical lead in web applications, web services, database, grid, cloud, storage/backup, bioinformatics, financial computing which subsequently have become his research interests. Victor has also successfully delivered many IT projects in Taiwan, Singapore, Australia, and the UK since 1998. Victor is experienced in a number of different IT subjects and has 27 certifications with 97% on average. He completed PGCert (Higher Education, University Greenwich, 2012) and PhD (C.S, University of Southampton, 2013) within four years while working full-time, whereby the distance between his work and research is about hundreds of miles away. He has over 70 published peer-reviewed papers, including several high-quality journals up-to-date. Victor won G20,000 funding in 2001 (Singapore-Cambridge Trust) and G81,000 funding in 2009 (Department of Health). He was involved in part of the G6.5 million project in 2004, part of the G5.6 million project in 2006 and part of a G300,000 project in 2013. He is a PI and Co-PI in some projects. Victor is a winner in 2011 European Identity Award in "On Premise to Cloud Migration". He was selected to present his research in the House of Commons, UK, in 2011. He won the best student paper in CLOSER 2012. Dr Victor Chang has taught numerous undergraduate and postgraduate modules. In some modules he taught, students like his teaching and enjoy his labs and lectures.