A Secure, Scalable, Flexible and Fine-Grained Access Control Using Hierarchical Attribute-Set-Based Encryption (HASBE) in Cloud Computing

Prashant A. Kadam
Department of Computer Engineering
JSPM Narhe Technical Campus
Pune, India

Avinash S. Devare
Department of Computer Engineering
JSPM Narhe Technical Campus
Pune, India

Abstract: Cloud Computing is going to be very popular technology in IT enterprises. For any enterprise the data stored is very huge and invaluable. Since all tasks are performed through network it has become vital to have the secured use of legitimate data. In cloud computing the most important matter of concern are data security and privacy along with flexibility, scalability and fine grained access control of data being the other requirements to be maintained by cloud systems Access control is one of the prominent research topics and hence various schemes have been proposed and implemented. But most of them do not provide flexibility, scalability and fine grained access control of the data on the cloud. In order to address the issues of flexibility, scalability and fine grained access control of remotely stored data on cloud we have proposed the hierarchical attribute set-based encryption (HASBE) which is the extension of attribute- set-based encryption(ASBE) with a hierarchical structure of users. The proposed scheme achieves scalability by handling the authority to appropriate entity in the hierarchical structure, inherits flexibility by allowing easy transfer and access to the data in case of location switch. It provides fine grained access control of data by showing only the requested and authorized details to the user thus improving the performance of the system. In addition, it provides efficient user revocation within expiration time, request to view extra-attributes and privacy in the intra-level hierarchy is achieved. Thus the scheme is implemented to show that is efficient in access control of data as well as security of data stored on cloud with comprehensive experiments.

Keywords: Fine-grained access control, attribute-set-based encryption, hierarchical attribute-set-based encryption.

1. INTRODUCTION

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS).The datacenter hardware and software is what we will call a Cloud.

Cloud computing is a web-based application that provides computation, software, infrastructure, platform, devices and other resources to users on the basis of pay as you use. Clients can use cloud services without any installation and the data uploaded on cloud is accessible from anywhere in the world, the only requirement is the computer with active internet connection. As a customizable computing resources and a huge amount of storage space are provided by internet based online services, the shift to online storage has contributed greatly in eliminating the overhead of local machines in storage and maintenance of data. Cloud provides a number of benefits like flexibility, disaster management and recovery, pay-per-use and easy to access and use model which contribute to the reason of switching to cloud. Cloud gives the provision for storage of important data of users. Thus cloud helps to free up the space on the local disk.

Cloud computing has emerged as one of the most influential paradigms in the IT industry. Almost all companies, organizations store their valuable data on the cloud and access it. Due to this security to the cloud is a major concern. Also flexibility, scalability of data stored on cloud which are the system performance parameters and which degrade the system response time required to be handled by cloud systems. They should provide a secure environment and maintenance of data in hierarchy.

The prominent security concern is data storage security and privacy in cloud computing due to its Internet-based data storage and management. The data security issue becomes vital when the data is a confidential data. In cloud computing, users have to give up their data to the cloud service provider for storage and business operations, while the cloud service provider is usually a commercial enterprise which cannot be totally trusted. So the data integrity and privacy of data is at risk.

Flexible and fine-grained access control is strongly desired in the service-oriented cloud computing model. Various schemes which provide access control models have been proposed. But the problem related with these schemes is that they are limited to data owners and service providers which exist in the same trusted domain.

2. EXISTING SYSTEM

2.1 Vipul et al.“ (Abe)Attribute based encryption”. [1]

As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE).

2.2 Rakesh et al. “Attribute-Sets: A Practically Motivated Enhancement to Attribute-Based Encryption”, University of Illinois at Urbana-Champaign, July 27, 2009. [2]

In distributed systems users need to share sensitive objects with others base on the recipients' ability to satisfy a policy. Attribute-Based Encryption (ABE) is a new paradigm where such policies are specified and cryptographically enforced in the encryption algorithm itself. Cipher text-Policy ABE (CP-ABE) is a form of ABE where policies are associated with encrypted data and attributes are associated with keys. In this work we focus on improving the flexibility of representing user attributes in keys. Specifically, we propose Cipher text Policy Attribute Set Based Encryption (CP-ASBE) - a new form of CP-ABE - which, unlike existing CP-ABE schemes that represent user attributes as a monolithic set in keys, organizes user attributes into a recursive set based structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy. We show that the proposed scheme is more versatile and supports many practical scenarios more naturally and efficiently. We provide a prototype implementation of our scheme and evaluate its performance overhead

2.3 Pankaj et al. “Cloud Computing Security Issues in Infrastructure as a Service”, 2012. [3]

Cloud computing is current buzzword in the market. It is paradigm in which the resources can be leveraged on peruse basis thus reducing the cost and complexity of service providers. Cloud computing promises to cut operational and capital costs and more importantly it let IT departments focus on strategic projects instead of keeping datacenters running. It is much more than simple internet. It is a construct that allows user to access applications that actually reside at location other than user's own computer or other Internet-connected devices. There are numerous benefits of this construct. For instance other company hosts user application. This implies that they handle cost of servers, they manage software updates and depending on the contract user pays less i.e. for the service only. Confidentiality, Integrity, Availability, Authenticity, and Privacy are essential concerns for both Cloud providers and consumers as well. Infrastructure as a Service (IaaS) serves as the foundation layer for the other delivery models, and a lack of security in this layer will certainly affect the other delivery models, i.e., PaaS, and SaaS that are built upon IaaS layer. This paper presents an elaborated study of IaaS components' security and determines vulnerabilities and countermeasures. Service Level Agreement should be considered very much importance.

2.4 John et al. “(CP-ABE) Cipher text-Policy Attribute-Based Encryption” John et al. [4]

In several distributed systems a user should only be able to access data if a user posses a certain set of cre- dentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call Cipher text-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is not trusted; moreover, our methods are secure against collusion attacks. Previous Attribute- Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data deter-mines a policy for who can decrypt. Thus, our meth-ods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our sys- tem and give performance measurements.

2.5 Ayad et al. "Enabling Data Dynamic and Indirect Mutual Trust for Cloud Computing Storage System", 2012. [6]

In this paper, we propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features: (i) it allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append, (ii) it ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data, (iii) it enables indirect mutual trust between the owner and the CSP, and (iv) it allows the owner to grant or revoke access to the outsourced data. We discuss the security issues of the proposed scheme. Besides, we justify its performance through theoretical analysis and experimental evaluation of storage, communication, and computation overheads.

2.6 Guojun et al. "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers", 2011. [8]

With rapid development of cloud computing, more and more enterprises will outsource their sensitive data for sharing in a cloud. To keep the shared data confidential against untrusted cloud service providers (CSPs), a natural way is to store only the encrypted data in a cloud. The key problems of this approach include establishing access control for the encrypted data, and revoking the access rights from users when they are no longer authorized to access the encrypted data. This paper aims to solve both problems. First, we propose a hierarchical attribute-based encryption scheme (HABE) by combining a hierarchical identity-based encryption (HIBE) system and a ciphertext-policy attribute-based encryption (CP-ABE) system, so as to provide not only fine-grained access control, but also full delegation and high performance. Then, we propose a scalable revocation scheme by applying proxy re-encryption (PRE) and lazy re-encryption (LRE) to the HABE scheme, so as to efficiently revoke access rights from users.

2.7 Qin et al. "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services". [9]

Cloud computing, as an emerging computing paradigm, enables users to remotely store their data into a cloud so as to enjoy scalable services on-demand. Especially for small and medium-sized enterprises with limited budgets, they can achieve cost savings and productivity enhancements by using cloud-based services to manage projects, to make collaborations, and the like. However, allowing cloud service providers (CSPs), which are not in the same trusted domains as enterprise users, to take care of confidential data, may raise potential security and privacy issues. To keep the sensitive user data confidential against untrusted CSPs, a natural way is to apply cryptographic approaches, by disclosing decryption keys only to authorized users. However, when enterprise users outsource confidential data for sharing on cloud servers, the adopted encryption system should not only support fine-grained access control, but also provide high performance, full delegation, and scalability, so as to best serve the needs of accessing data anytime and anywhere, delegating within enterprises, and achieving a dynamic set of users. In this paper, we propose a scheme to help enterprises to efficiently share confidential data on cloud servers. We achieve this goal by first combining the hierarchical identity-based encryption (HIBE) system and the cipher text-policy attribute-based encryption (CP-ABE) system, and then making a performance-expressivity tradeoff, finally applying proxy re-encryption and lazy re-encryption to our scheme.

2.8. Patrick et al. "Methods and Limitations of Security Policy Reconciliation". [10]

A security policy is a means by which participant session requirements are specified. However, existing frameworks provide limited facilities for the automated reconciliation of participant policies. This paper considers the limits and methods of reconciliation in a general-purpose policy model. We identify an algorithm for efficient two-policy reconciliation, and show that, in the worst-case, reconciliation of three or more policies is intractable. Further, we suggest Efficient heuristics for the detection and resolution of intractable reconciliation. Based upon the policy model, we describe the design and implementation of the Ismene policy language. The expressiveness of Ismene, and indirectly of our model, is demonstrated through the representation and exposition of policies supported by existing policy languages. We conclude with brief notes on the integration and enforcement of Ismene policy within the Antigone.

3. PROPOSED SYSTEM

In our propose system instead of showing complete data from cloud we are fetching only those data which is essential for that user. We are not fetching all data so it takes less time for fetching data so system response time is very less due to which system performance increases. We are performing encryption before storing data so even if data get hack by hacker data cannot be easily understand by hacker. We are performing hierarchical structure so even if lower authority is absent for particular days at that time higher authority handle all work of lower authority so work of company will not be stopped. The HASBE scheme for realizing scalable, flexible and fine-grained access control in cloud computing. The HASBE scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments of attributes. We formally proved the security of HASBE based on the security of CP-ABE. Finally, we completed the detailed analysis of proposed scheme, and conducted comprehensive performance analysis and evaluation, which showed its efficiency and advantages over existing schemes.

3.1 Project Scope

1. This system is designed to provide security to data stored on cloud and improve performance of system by showing only the required details requested by an employee.
2. Security is provided by generating a secret key from the various attributes stated in the form which is filled by the employee at the time of registration.
3. This system is designed to provide flexibility of the data where in case of transfer of employee, his data could be transferred to respective location with ease.
4. It also provides scalability in case when an employee is absent his work could be handled by the senior employee securely.

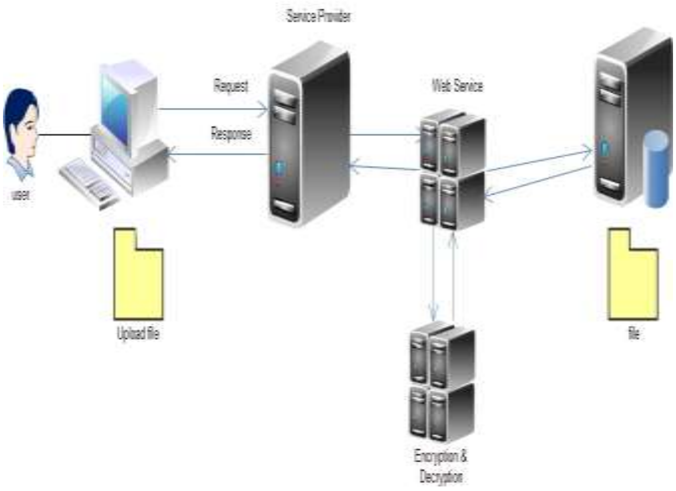


Figure 1. General Architecture of the System

3.1 Methodology

1. Registration and login by user:
In this user fill his/her own complete data. Request is sent to the CEO for confirmation. CEO confirms his/her request and assigns attribute and time period for that user. Once Account get confirm password and key is sent to that user by email so he/she can access his/her account**.**

2. Approve User and Assign attributes:
Out of the selected attributes according the roles defined in hierarchy of the system the attribute visibility access is decided. Each attribute is encrypted.

3. Key Generation and Verification

Key is generated based on the attributes filled by the user in registration form. In attribute key verification, when a key is used for login ,it is first checked with the key stored in the database. If a match is found then user is allowed for further process else the user is rejected for further process.

4. Encryption and decryption of data

User fills his/her data during registration. Once it is click on submit button data is send to encryption algorithm that are RSA and AES. After performing encryption data is stored in encrypted format in database.

5. Access Right:
The user can view the selected attributes of the same level as well as other levels according to the access authority using attribute key.

6. Fine Grained Access

In our propose system instead of showing complete data, the fetching of necessary data is allowed. Due to this system provides a quick response time.

7. Request for extra attribute:

The user can access attributes of same level as inter level counterparts. He can request for extra attributes in case of emergency as well as ease of work.

8. Flexibility

In this module flexibility can be done by suppose user is transfer from one location to another location and for that new location that user's data is not accessible then authority request for accessing data of that user from old location. Once authority got request that data should be access from new location and it is not visible for old location

9. Scalability:

We are performing hierarchical structure so even if lower authority is absent for particular days at that time higher authority handle all work of lower authority so work of company will not be stopped.

10. Efficient User Revocation:

It can be done by two steps request to the admin and response to the user from admin within expiration time.

9. Privacy:

Default it is public but a user can set intra-level privacy by restricting access to attributes.

3.2 Process Summary

Following Processes will be involved in Project:

1. Encrypt data before Insert

After user click on submit button data encrypted using RSA and AESs algorithm. Once data get encrypted it is stored into database and when user wants to retrieve data it again decrypted and shown in original form.

2. Request for New Attributes

In this phase one of the lower authority may absence then at that time higher authority may handle both attributes, one is its own attributes and another is attributes of the lower authority who is absent for particular time period. User can also request for new attribute if needed in any case.

3. Getting information of other user

In this when user transfer from one location to another location at that time new location does not having rights to access data of that user at that time getting grant for accessing data of that user. When user's data is accessible from new location then it can -not access from old location.

3.3 System Functional Features

The cloud server provides the six main functions to the user.

1. Fine-Grained Access

In our propose system instead of showing complete data, the fetching of necessary data is allowed. Due to this system provides a quick response time.

2. Scalability

We are performing hierarchical structure so even if lower authority is absent for particular days at that time higher authority handle all work of lower authority so work of company will not be stopped.

3. Flexibility

When an employee gets transferred, his data could be accessible to the branch where he will be transferred only not to the older branch. So data will be transferred on request of CEO safely. Hence data can be transferred easily between branches.

4. Encryption

Encryption is a process in which data is hidden in a way that is accessible to the authorized user only. In this system we are providing encryption (converting into unreadable) so that data is not accessible by any illegal user like a hacker.

5. Decryption

Decryption is a process in which encrypted data i.e unreadable format is converted into readable format.

6. Key Generation and Verification

Key is generated based on the attributes filled by the user in registration form. In attribute key verification, when a key is used for login, it is first checked with the key stored in the database. If a match is found then user is allowed for further process else the user is rejected for further process.

4. ALGORITHM AND MATHEMATICAL MODEL

4.1 Algorithm

4.1.1RSA (Rivest Shamir Adleman)

Key generation

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.

Encryption

User1 transmits her public key *(n, e)* to User2 and keeps the private key secret. User1 then wishes to send message *M* to User2.He first turns *M* into an integer *m*, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a scheme. He then computes the cipher text *c* corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done quickly using the method of exponentiation by squaring. User1 then transmits *c* to User2.Note that at least nine values of *m* will yield a cipher text *c* equal to *m*, but this is very unlikely to occur in practice.

Decryption

User can recover m from c by using her private key exponent.

$$m \equiv c^d \pmod{n}$$

Given *m*, user can recover the original message *M* by reversing the padding scheme.

4.1.2 Advanced Encryption Standard Algorithm

The AES algorithm is also used for improving the searching and access mechanism.

4.2 Mathematical Model

We are using **NP-Complete** because it gives output within fix interval of time.

Set Theory Analysis

A] Identify the Employees

E= {e1, e2, e3….}

Where ‘E’ is main set of Employees like e1, e2, e3…

B] Identify the Attribute

AT= {at1, at2, at3….}

Where ‘AT’ is main set of registered Attribute like at1, at2, at3…

C] Identify the employee requested For another Attribute

RAA= {raa1, raa2, raa3}

Where ‘RAA’ is main set of Request for another Attribute raa1, raa2, raa3

D] Identify the employee requested for another employee Information

REI= {rei1, rei2, rei3}

Where ‘REI’ is main set of Request for another Attribute rei1, rei2, rei3

E] Identify Attribute Key of New employee

AK= {ak1, ak2, ak3….}

Where ‘AK’ is main set of attribute key of users ak1, ak2, ak3…

F] Identify the processes as P.

P= {Set of processes}
P = {P1, P2, P3,P4……}
P1 = {e1, e2, e3}
Where
{e1= upload data on server}

{e2= make the entry in database using different encryption algorithm}

{e3= get new attribute after request}

{e4= get new employee information when employee get transfer.}

G] Identify failure cases as FL

Failure occurs when –
FL= {F1, F2, F3…}
a) F1= = {f| ’f’ if error in uploading due to interrupted Internet connection}
H] Identify success case SS:-
Success is defined as-
SS= {S1, S2, S3, S4}

a) S1= {s|'s' if fast and no interrupted Internet connection}

b) S2= {s|'s' if data is added into database}

c) S2= {s|'s' if data is retrieve from database}

I] Initial conditions as I_0

a) User has good internet connection

b) Admin has good internet connection

H is universal set i.e cloud.

H= {E, B, U, R}

E=employee set

B=attribute set

U=user set

R=registered

A] Identify the Employees

E= {e1, e2, e3….}

Where 'E' is main set of Employees like e1, e2, e3…

B] Identify the Attribute

B= {at1, at2, at3….}

Where 'B' is main set of registered Attribute like at1, at2, at3…

C] Identify the employee requested For another Attribute

A= {raa1, raa2, raa3}

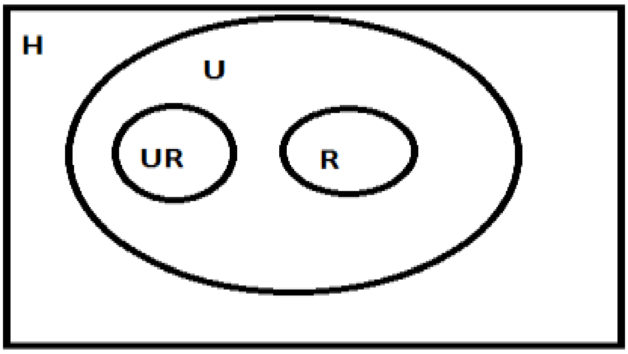
Where 'A' is main set of Request for another Attribute raa1, raa2, raa3

INITIAL STATE:

U={R, UR}

R=registered user

UR=unregistered user



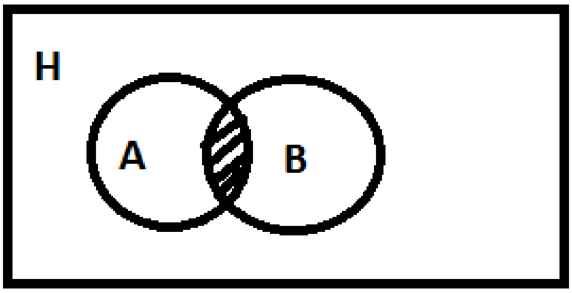
INTERMEDIATE STATE:

Request for new attribute

A=request for new attribute

B=contain all the attribute

R=provide requested attribute



R = ▨

S1=A $\cap$ B

Hierarchy

H = {H1, H2, H3, H4}

Where,

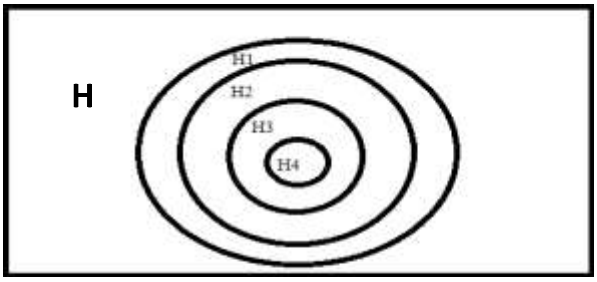
H is cloud

H1 is CEO.

H2 is general manager.

H3 is the list of managers.

H4 is the list of employees.



FLEXIBILITY:

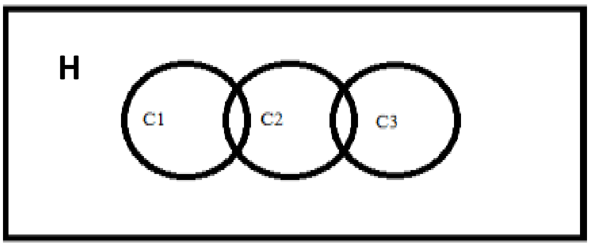
H= {C1, C2, C3}

Where,

C1 is the old branch of the company where employee worked before transfer.

C2 is the employee being transferred.

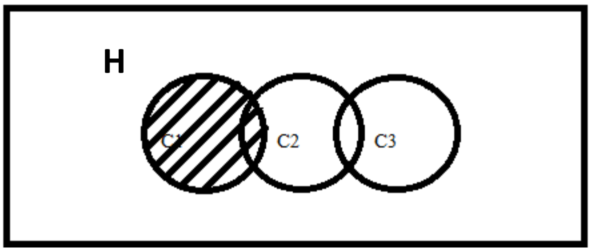
C3 is the new branch where employee got transferred to.



H= {C1, C2, C3}

Where,

S2 is employee data should be accessed to new branch only not old branch.



S2= (C1-C2) U C3

Scalability

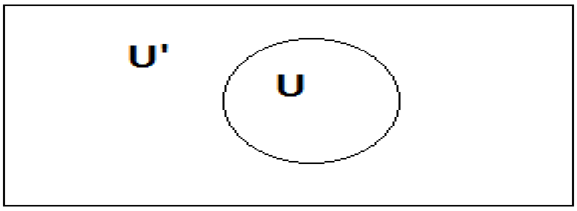
H = {H1, H2, H3, H4}

U= {H1, H2}

U'= {H3, H4}

U=present user

U'=absent user



S3

FINAL STATE:

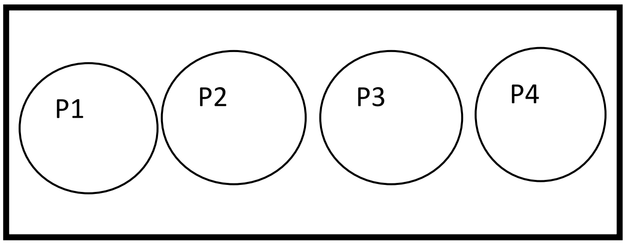
Identify the processes as P.

P= {Set of processes}

P = {P1, P2, P3, P4……}

Where

P1 = {S1, S2, S3}

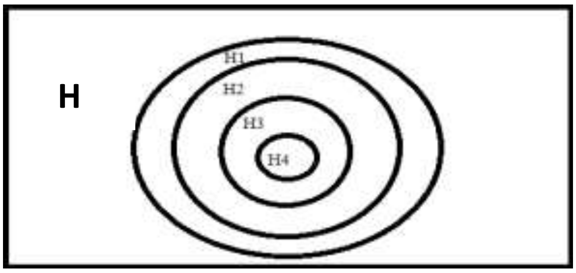


Where,

{S1= get new attribute after request}

{S2= get new employee information when employee get transfer.}

{S3= get access of lower authority}



FLEXIBILITY:

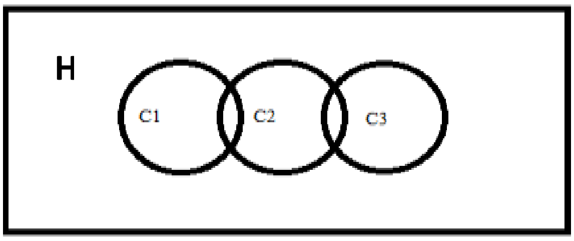
H= {C1, C2, C3}

Where,

C1 is the old branch of the company where employee worked before transfer.

C2 is the employee being transferred.

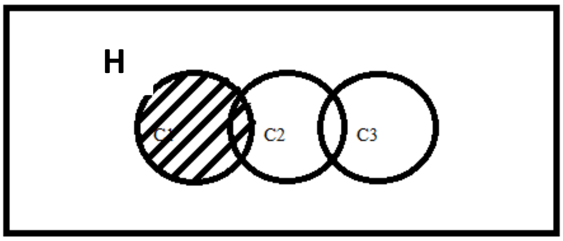
C3 is the new branch where employee got transferred to.



H= {C1, C2, C3}

Where,

S2 is employee data should be accessed to new branch only not old branch.



S2= (C1-C2) U C3

Scalability

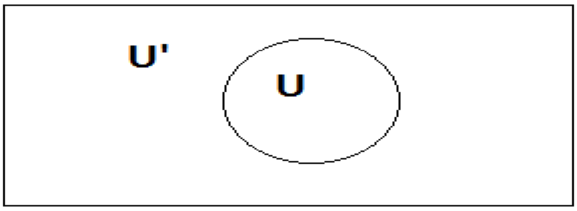
H = {H1, H2, H3, H4}

U= {H1, H2}

U'= {H3, H4}

U=present user

U'=absent user



S3

FINAL STATE:

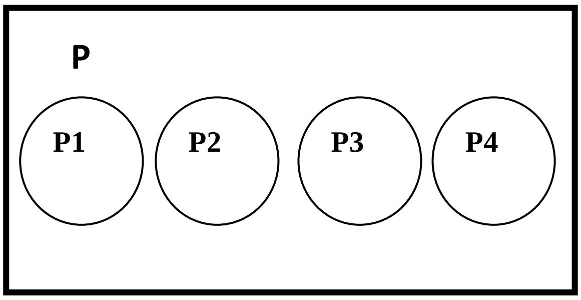
Identify the processes as P.

P= {Set of processes}

P = {P1, P2, P3, P4……}

Where

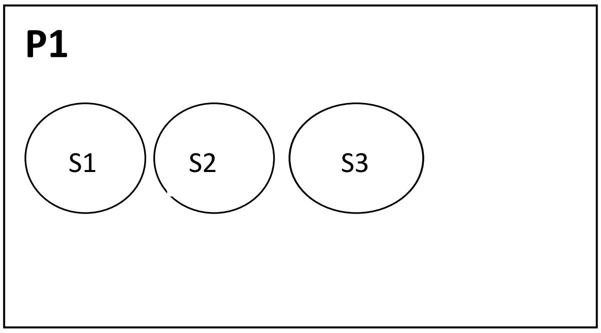
P1 = {S1, S2, S3}



{S1= get new attribute after request}

{S2= get new employee information when employee get transfer.}

{S3= get access of lower authority}



5. CONCLUSION

Thus, our system efficiently provides a fine grained access control with flexibility and scalability with a hierarchical structure in our HASBE system**.** Our paper will be providing security to the users from outsiders or intruders by implementing session hijacking and session fixation security in our system with sql injection attack prevention. The core is for sure, a cloud-base thus giving us a choice of multi-user access including security from intruder attacks. Hence we benefit the users with attack handling and many advantages over the existing systems.

6. REFERENCES

[1] Vipul et al.“ (Abe)Attribute based encryption”.

[2] Rakesh et al. “Attribute-Sets: A Practically Motivated Enhancement to Attribute-Based Encryption”, University of Illinois at Urbana-Champaign, July 27, 2009

[3] Pankaj et al. “Cloud Computing Security Issues in Infrastructure as a Service”,2012.

[4] John et al. “(cp-abe) Ciphertext-Policy Attribute-Based Encryption”John et al..

[5] Suhair et al. “Designing a Secure Cloud-Based EHR System using Ciphertext-Policy Attribute-Based Encryption”, 2011

[6] Ayad et al. ”Enabling Data Dynamic and Indirect Mutual Trust for Cloud Computing Storage System”, 2012.

[7]Chandana et al. “Gasbe: A Graded Attribute-Based Solution For Access Control In Cloud Computing”, 2011.

[8] Guojun et al. “Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers”, 2011.

[9] Qin et al. “Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services”.

[10] Patrick et al. “Methods and Limitations of Security Policy Reconciliation”.

[11]http://searchwindowsserver.techtarget.com/definition/IIS.

[12]http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

[13] http://en.wikipedia.org/wiki/.NET_Framework.