Light Weight Access Control Mechanism for Mobile-based Cloud Data Storage

Rajat Saxena
and
Somnath Dey
Cloud Computing Lab
Department of Computer Science and Engineering
Indian Institute of Technology Indore, Indore, India

Cloud computing is the fastest growing field of service provision in Information Technology (IT) industry. It provides on-demand and cost-effective services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In cloud environment, many security challenges have pinched out such as data security, malicious insider attack, cyber attack, and abuse of cloud services. In this paper, we have analyzed and identified the different major gaps between conventional access control schemes based on their demerits and requirements for the cloud access. We have proposed a Light-Weight Access Control (LWAC) model, which fulfills all cloud access control requirements. Our approach has ensured the secure and efficient sharing of resources among various non-trusted tenants and also has the capacity to support the different access permissions for the same user using multiple services securely. We have also implemented a prototype of our work which illustrates the efficient access control in the cloud environment. This prototype delivers the different cloud services within the distributed cloud environment. It also demonstrates the effective and secure access control to fetch multiple services for various resources within the capabilities assigned to the user. The comparative results show the genuine application of our approach within the latest distributed cloud environment.

Keywords: Cloud computing, Combinatorial Batch Codes(CBC), Role Based Access Control (RBAC), Access control models and Task-Role Based Access Control.

1. INTRODUCTION

Cloud computing (Saxena and Dey, 2014), (Saxena and Dey, 2015) is defined as on-demand services and applications that use simple networking standards and protocols to access virtual resources in a distributed network. These resources are limitless and virtual. In cloud (Saxena and Dey, 2017), the end-users have unaware of the implementation details of cyber-physical systems.

Storage of cloud data (Ruj and Saxena, 2015) requires multi-tenant systems which are allocated to the different servers within a large area. It is extremely difficult to maintain the security audit logs for a user who has limited resources. Thus, on behalf of a user, Cloud Service Provider (CSP) must devote for resources and security measures to maintain access control and privacy.

Users may have limited battery power, computation capacity, and communication resources. So, effective access control is one of the primary security issue in the cloud environment. Several access control models exist for the different communities, organizations, and environments. But each model has its own drawbacks and limitations.

Multi-tenancy, virtualization, sharing of resources, and credential transformation are crucial aspects in the cloud environment. Different access permissions to the same cloud user, and giving him/her ability to use multiple services are difficult in cloud environment. Some basic and conventional access control methods are described in paper (Saxena and Dey, 2016) for cloud computing. These methods are very error-prone, prohibitive, and time-consuming for cloud users. These conventional methods are suffered from lack of flexibility in scalability and attribute management in cloud environment. The evaluation of some methods are described in Table I

for cloud environment.

Table I: Evaluation of different Access Control Methods for Cloud environment

Method	Advantages	Disadvantages
1. Mandatory Access Control (MAC)(Ausanka-Crues, 2001)	1. Access decision made by central authority. 2. It assigns secure assignment to each object. 3. MAC policies reduce security errors	1. Secure information flow between subject and object is required for relationship. 2. MAC enforced operating systems (OS) delineate and label incoming application data, which creates a specialized external application access control policy.
2. Discretionary Access Control (DAC) (Lampson, 1974)	1.User may transfer object ownership to another user(s). 2.User may determine the access type of other users. 3. Unauthorized users are blind to object characteristics, such as file size, file name, and directory path. 4. After several attempts, authorization failures restrict user access.	1. Inherent vulnerabilities (Trojan horse) 2. Limited negative authorization power 3. ACL maintenance or capability 4. Grant and revoke permissions maintenance
3. Role-based Access Control (RBAC) (Laurie, 2009)	1. It reduces administrative work and IT support. 2. It maximizes operational efficiency. 3. It improve compliance. 4. Frequency delegation of a large number of users, different classification, and mobility features.	1. It limits what objects the role group is allowed to manage. 2. It does not provide any sensitivity to the information. 3. It does not support delegation principle which is applicable in case of absences of employees. 4. This model does not support dynamic activation of access rights for certain tasks assigned to the staff.
4. Task-Role Based Access Control (T-RBAC) (Sun, Wang, Yong, and Wu, 2012)	1. It is based on Role Based Access Control model and assigns permissions to the tasks instead of roles. 2. The user is assigned roles, and it assigns tasks that have permissions.	1. It uses a workflow authorization model for synchronizing workflow with authorization flow. 2. This model uses tasks to support active access control and roles to support passive access control.
5. Attribute-Based Access Control (ABAC)(Al-Kahtani, Sandhu, et al., 2002)	1. It is based on a set of attributes associated with a requester or resource to accessed and makes decisions. 2. After defining attributes, each attribute can consider as a discrete value and values of all attributes are compared against a set of values by a policy decision point to deny or grant access.	1. It requires effort to define policies. 2. It requires user training and technical efforts for implementation.
6. Adaptive Access Control (Wang, Han, Song, and Wang, 2011)	1. It is based on contextual information such as time and security information. 2. In it, authors build a trust relationship between Cloud Service Providers (CSP's) and its consumers with the role-based access control system. 3. A trust management system has maintained, which update and change trust level after each transaction.	1. Access decision requires Authority Authorization Centre (AAC). 2. This model has suffered from potential single point of attack and policy information failure.
7. Cloud Optimized Risk-Based Access Control (co-RBAC) (Tianyi, Weidong, and Jiaxing, 2011)	1. It inherits the features of the distributed environment, merge distributed authentication services, and has the ability to issue certificate same as Certificate Authorities (CA). 2. In this model, hierarchical cache has been embedded to improve the overall efficiency of access control system.	1. Dependency on CA for issuing certificate might cause efficiency and scalability problems because for each access time a new certificate is needed. 2. It is suffering from lack of flexibility in scalability and attribute management.
8. Ontology using Role Based Access Control (O-RBAC) (Tsai and Shao, 2011)	1. It provides the appropriate policy with an exact role for every tenant. Every subject can have multiple roles in multiple sessions. 2. A role hierarchy is based on domain ontology and can transfer between various ontological domains.	1. This model has to ensure granting access decisions in a reasonable time and according to system requirements. 2. Diversity in access control policies and interfaces may cause improper interoperability. The model implementation is difficult for cloud computing.

To address these limitations, we propose improvements in existing access control models based on the user attributes. We propose a novel Light-Weight Access Control (LWAC) model, which has many levels of security depending upon the trust hierarchy. It supports many sensitive levels of information to implement restriction on reading and modification of information on the cloud. Our approach verifies and guarantees that the CSP could not learn about any data content stored in the cloud server during the efficient access control. Specifically, our contributions in this work are summarized in the following three aspects:

(1) We motivate the Light-Weight Access Control (LWAC) of data and provide a new access control scheme with Combinatorial Batch Codes (CBC).
(2) To the best of our knowledge, the proposed scheme is the first to support scalable and efficient Light-Weight access control.
(3) We have analyzed our scheme with current state-of-the-art methods with respect to the different performance parameters.

Organization. The rest of the paper is organized as follows. In section 2, we describe our Light-Weight Access Control (LWAC) model based on requirement observation. In section 3, we describe implementation phases of our approach. In section 4, we present comparative performance analysis of our scheme with other current state-of-the-art methods. Finally, we conclude in section 5.

2. THE PROPOSED SCHEME

In this section, we present our access control scheme for cloud services. First, we explain the details about CBC. After that, we describe our scheme with CBC and discuss algorithms which subsequently represent our scheme.

2.1 Combinatorial Batch Codes

Combinatorial Batch Code $\mathcal{C}$ (n, N, k, m, t) (Stinson, Wei, and Paterson, 2009) is a set system ($\mathcal{F}$, $\mathcal{S}$), where $\mathcal{F}$ is a set of n elements (called items), $\mathcal{S}$ (called servers) is a collection of m subsets of $\mathcal{F}$ and $N = \sum_{s\varepsilon\mathcal{S}} |s|$, such that for each k-subset $\{f_{i_1}, f_{i_2}, ..., f_{i_k}\} \subset \mathcal{F}$ there exists a subset $\mathcal{C}_i \subseteq \mathcal{S}_i$, where $|\mathcal{C}_i| \leqslant t$, i= 1,.....,m, such that

$$\{f_{i_1}, f_{i_2}, ..., f_{i_k}\} \subset \bigcup_{i=1}^{m} \mathcal{C}_i \tag{1}$$

If we are fixing t=1; it means CBC permits only one item to be retrieved from each server. This CBC denotes as an (n, N, k, m)-CBC. We use CBC to distribute the different blocks of a file in cloud servers that belong to separate Cloud Service Provider (CSP).

2.2 Light-Weight Access Control (LWAC) Model

Light-Weight Access Control (LWAC) model has based on CBC for distribution of file blocks into different CSP servers. If we generalize our strategy to use CBC, then abstract formulation of our strategy remains follows:

A file of n blocks is to be stored among m servers in such a way that any k of the n data blocks can be retrieved by at most t blocks from each server, and the total number of blocks stored in m servers is N. Our aim is to find out the optimal CBC for which minimal total storage N is required for the given values of n, m, k, and t.

In this model, the private key generation is very computationally intensive task because it depends on the identity of an entity. Thus, this approach requires careful selection of the identity of an entity before issuing a private key.

LWAC model delegates the private key generation to the Private Key Generator (PKG). The PKG is also responsible for the selection of identity and secure transmission of the private key to its lowest levels. The PKG also have the information about the set of public parameters P

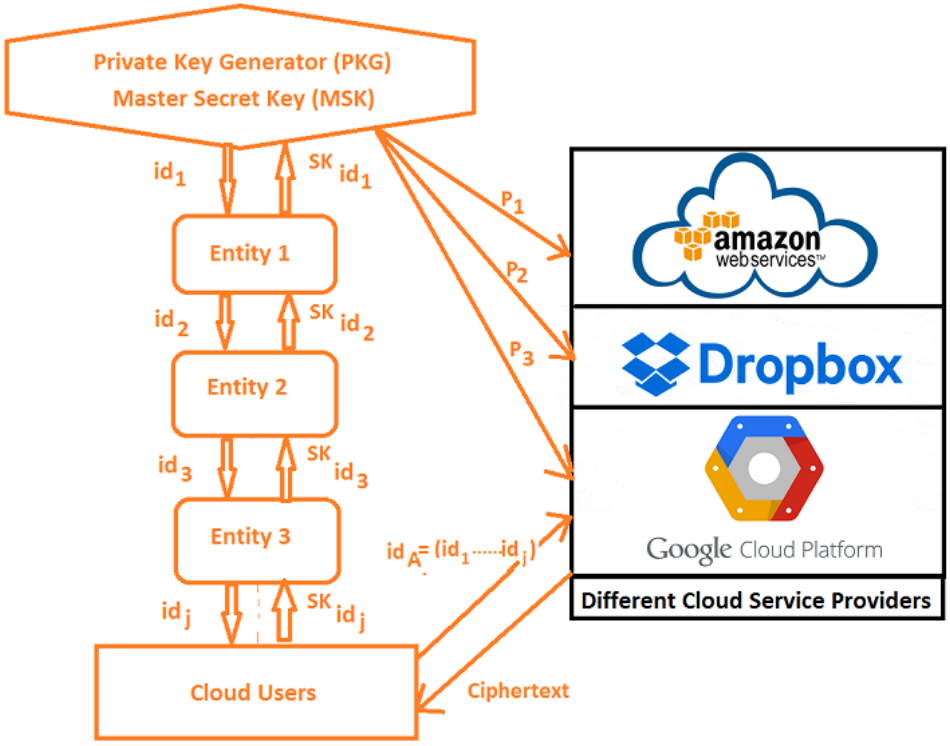

Figure 1 : LWAC Model for Cloud Environment

and master key MSK. $P = \{P_1, P_2, P_3\}$, Where $P_1, P_2, and\ P_3$ are the public parameters for the different CSPs. The identities at the different levels do not have any public parameters associated with them.

We use any identity of cloud user as their public key. Cloud user is allowed to do any communication with others only when user encrypts their file using their public key. If PKG selects a set of maximum h identities as a public key, the maximum height of hierarchy will be h for communication.

In LWAC model, identities are represented as vector. So for a maximum height h of hierarchy (which is denoted as h-LWAC) any identity id is a tuple $(id_1, ..., id_\tau)$, where $1 \leqslant \tau \leqslant h$. Let, $id^{'} = id_1^{'}, ..., id_j^{'}, j \leqslant \tau$ be another identity tuple. We say $id^{'}$ is a prefix of id if $id_i^{'} = id_i$ for all $1 \leqslant i \leqslant j$.

For all identities at the first level, the PKG generates the private key by using master key MSK. For identities at the second level onwards, the private key is generated by the PKG or any of the ancestors of that identity. Figure 1 shows the LWAC model for the different CSP servers. In this scheme, the private key sk_{id} of id is generated by an entity whose identity is a prefix of the id and who has obtained the corresponding private key.

Our LWAC model $\mathcal{H}$ is specified by following four probabilistic polynomial time (in the security parameter) algorithms:

1. Set-up: This operation generates the initial security parameters. Here, we use a string of 1 or 0 of length k as input and derive the P and MSK by randomizing the input. The generated master key is known only to the PKG. The PKG also contains the message space M, the ciphertext space$C$and the identity space$I$. Algorithm 1 presents the steps of setup operation.

Algorithm 1 : Setup operation

Input: $\{0,1\}^k$.
Output: Set of Public Parameters P, Initial Master Secret Key (MSK).

1: Initial Master Secret Key (MSK) generates by $k \xleftarrow{R} \{0,1\}^k$.
2: Set of Public Parameters P where $P = \{P_1, P_2, P_3\}$.
3: $P_1 \xleftarrow{R} \{0,1\}^k$.
4: $P_2 \xleftarrow{R} \{0,1\}^k$.
5: $P_3 \xleftarrow{R} \{0,1\}^k$.

2. Key-Generation: This operation generates the private key $sk(id^j)$ corresponding to the j^{th} identity. This method uses bilinear pairing (Saxena and Dey, 2016) between identity tuple $id = (id_1, ..., id_j)$, $j \geqslant 1$ and the private keys $sk(d_{id}|(j-1))$ for the identities $((id_1, ..., id_{(j-1)}))$. Bilinear pairing (Saxena and Dey, 2016) defines a map between two cyclic groups of some prime order and satisfies bi-linearity, non-degeneracy and efficient computability properties.

In this algorithm, we define bilinear pairing as BilinearPair(.,.) function. Initially, for j = 1, MSK and id_1 are used to generate $sk(id_1)$. By invoking Key-generation algorithm, PKG or identity at any level can produce the decryption key. Key generation algorithm is given in Algorithm 2.

Algorithm 2 : Key Generation

Input: identity tuple id = $(id_1, ..., id_j)$, where $j \geqslant 1$ and the private key $sk_{id|j-1}$ for the identity $(id_1, ..., id_{j-1})$.
Output: private key sk_{id}.

1: **if** j=1 **then**
2: $sk_{id_1} \leftarrow BilnearPair(MSK, id_1)$.
3: **else**
4: $sk_{id_j} \leftarrow BilnearPair(sk_{id_{j-1}}, id_j)$.
5: **end if**
6: **return** sk_{id_j}.

3. Encryption:

This process encrypts a message M by set of public parameters P of an identity id and produces a cipher-text C. We use a standard encryption algorithm E. The steps of encryption operation have specified in Algorithm 3.

Algorithm 3 : Encryption

Input: Set of Public parameters P, id_j, and $\mathcal{M}$.
Output: Cipher-text $\mathcal{C}$.

1: $\mathcal{C} \leftarrow E_{id_j}(\mathcal{M})$

4. Decryption:

This process takes the public parameter P, an identity id, a Cipher-text $\mathcal{C}$ and a private key sk_{id} as input and compute the original message $\mathcal{M}$. If the cipher-text is not valid, this algorithm produces $\perp$. We use standard decryption algorithm D corresponding to E in the decryption process. Decryption (Algorithm 4) is presented as follow:

Algorithm 4 : Decryption

Input: public parameters P, id_j, $\mathcal{C}$, and sk_{id}.
Output: Message $\mathcal{M}$, $\perp$.

1: **if** Cipher-text is not valid **then**
2: **return** $\perp$.
3: **else**
4: **return** $\mathcal{M} \leftarrow D_{id}(\mathcal{C})$
5: **end if**

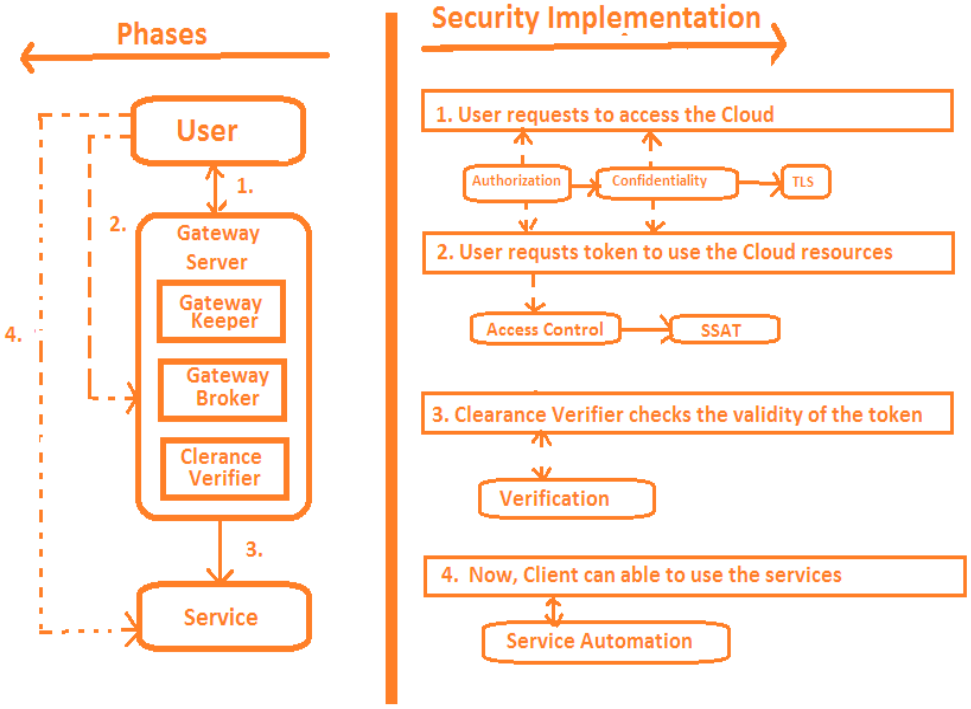

Figure 2:Phases of Access Control for Cloud Environment

3. IMPLEMENTATION

To demonstrate our approach, we have implemented an application based on Hadoop and MapReduce framework. The experiment has run on two PCs configured with Intel Core i7-2600S 2.80 GHz and 16 GB RAM. We have configured Citrix Xen Server 6.2.0 (XenServer, 2014) on one machine that was used for file storage. On this server, we have configured three Virtual Machines (VMs)that access Google Cloud platform, Dropbox, and Amazon Web Service (AWS), respectively. This VM's have used three public parameters $\{P_1, P_2, P_3\}$ for secure access to the file.

The second PC configured with Cloudera CDH 5.3.0-0 (Cloudera, 2014). It has used as a cloud user and provides access control to the stored files. The working of LWAC model has divided into four phases of access control. Figure 2 describes these phases.

(1) **Phase 1:**

A client program is installed on or downloaded to every endpoint (laptop, cell-phone, etc.)

when the user accesses the client end. A server or gateway hosts the centralized security program, which verifies logins and sends updates and patches when needed. In this phase, the user contacts the Gate Keeper (GK) service in the Gateway Server(GS), where the communication with the GK (or any other service) uses Transport Layer Security to protect against eavesdropping attacks.

(2) **Phase 2:**
The GS needs to identify their users securely through authentication. After that, a user must gain authorization for doing certain tasks. With the Single Sign-On Token(SSAT), a user logs in once and gain access to all systems without being prompted to log in again in each of them. The Clearance Verifier (CV) checks the validity of the token. If there is no verification in the SSAT, that service should contact the CV.

(3) **Phase 3:**
This step is a precaution against SSAT forging. If the CV reports back that the Gateway Server does not generate the SSAT, the request will be blocked. If the SSAT is examined and proved valid, the CV attaches a verification token to the SSAT.

(4) **Phase 4:**
Now, user can able to use the services.

4. PERFORMANCE ANALYSIS

We evaluate our scheme based on different performance parameters. This parameter classified into two categories (1) User Behavior Parameters, and (2) Service Level Agreement (SLA) Parameters.

4.1 User Behaviour Parameters

This type of parameters are based on the behavior of the cloud users. These parameters are used for evaluating the ethical trust value of the user for accessing any resource from the CSP.

(1) **Fake Request Rate (FRR)**
Fake requests are the heavy load of the dummy and illogical requests that are sent to cloud servers for consuming the Cloud resources. Fake requests is represented intensionally used for implementing the Denial of Service(DOS) attack by achieving the bandwidth starvation. Thus, this parameter affects the availability of the Cloud server.
If N_{Fake} is the number of fake requests and N_{Total} is the total number of requests sent by the user in a unit time interval, then FRR is represented by the following equation:

$$FRR = \frac{N_{Fake}}{N_{Total}}$$

(2) **Unauthorized Request Rate (URR)**
Unauthorized requests are the illegal requests send by the customer for stealing or modifying the contents of a data file stored on Cloud server. This parameter affects the confidentiality and authorization of cloud data.
If $N_{Unauthorized}$ is the number of unauthorized requests and N_{Total} is the total number of requests sent by the user in a unit time interval, then URR is represented by the following equation:

$$URR = \frac{N_{Unauthorized}}{N_{Total}}$$

(3) **Resource Affection Rate (RAR)**
RAR is the measurement of the affected resources with respect to the total resources accessed by a user in any unit time interval. This parameter affects the reliability of Cloud data.
If $N_{Affected}$ is the number of affected resources and N_{Total} is the total number of resources accessed in a unit time interval by the user, then RAR is represented by the following equation:

$$RAR = \frac{N_{Affected}}{N_{Total}}$$

(4) **User's Performance**
The W_1, W_2, and W_3 are the weighted values of FRR, URR, and RAR, respectively. Then the User Performance (UP) is evaluated by the following equation:

$$UP = [1 - ((w_1 * FRR) + (w_2 * URR) + (w_3 * RAR))] * 100$$

Thus, we calculate the current UP value.
To evaluate the Average User Performance (AUP), we use the current UP value and previous AUP value.
Let t_n and t_{n-1} are the time interval for the current time window and previous time window respectively. The Average User Performance (AUP) is evaluated by the following equation:

$$AUP = \gamma * (UP)_{t_n} + (1 - \gamma) * (AUP)_{t_{n-1}}$$

We have chosen the value of W_1, W_2, and W_3 as follows: $W_1 = 0.1$, $W_2 = 0.2$ and $W_3 = 0.3$. The assessment of UP given in Table II.

Table II: Assessment of User's Performance

N_{Total}	N_{Fake}	FRR	$N_{Unauthorized}$	URR	$N_{Affected}$	RAR	UP
100	21	0.21	19	0.19	9	0.09	91.4
200	38	0.19	22	0.11	20	0.10	92.9
300	66	0.22	51	0.17	24	0.08	92
400	92	0.23	64	0.16	44	0.11	91.2
500	120	0.24	65	0.13	75	0.15	90.5
600	126	0.21	72	0.12	72	0.12	91.9
700	175	0.25	98	0.14	91	0.13	90.8
800	176	0.22	144	0.18	88	0.11	90.9
900	171	0.19	135	0.15	108	0.12	91.5
1000	230	0.23	165	0.165	111	0.111	91.37

The comparative analysis of Average User Performance for the different methods have shown in Figure 3.

4.2 SLA Parameters

SLA parameters are evaluated the successful agreement between the cloud user and CSP's at the time of proceeding to the services. It also judges the requirements of cloud user that are fulfilled by the CSP's. It observes the ability of CSP for providing services to the user.

(1) **Turn Around Efficiency (TAE)**
Turn Around Time (TAT) is the exact measured time between submission of a job by a user and delivery of successfully completed job to the user. The estimated TAT is different from actual TAT. If TAT_{Est} is the Estimated TAT and TAT_{Actual} is the actual TAT for a job allocated to resource R_k, then Turn Around Efficiency (TAE) of resource R_k is represented by the following equation:

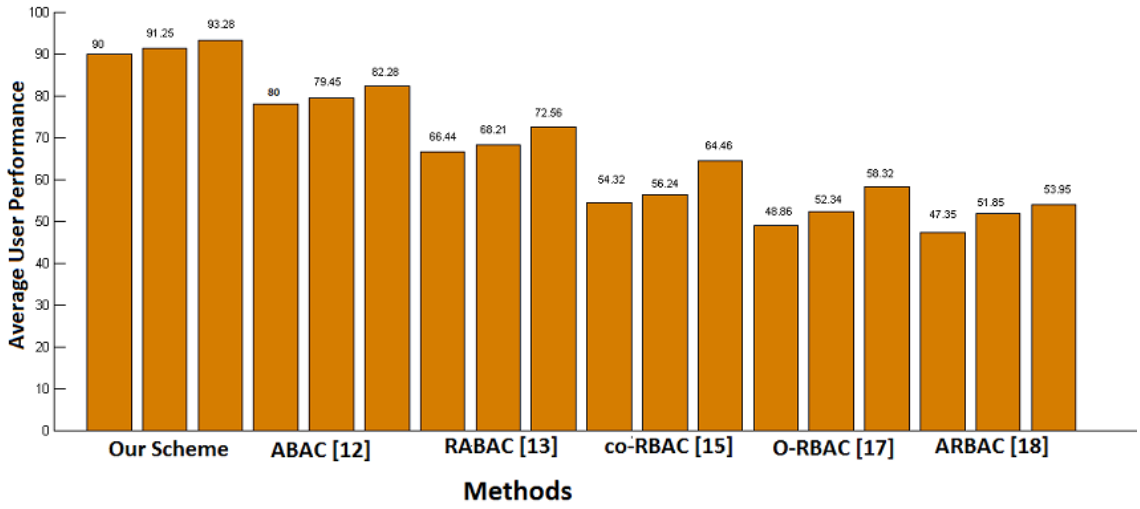

Figure 3: Comparative Analysis of Average User Performance

$$TE(R_k) = \frac{TAT_{Est}}{TAT_{Actual}}$$

Average Turn Around Efficiency (ATAE) of any resource R_k is the average of the turn around efficiency over all the jobs submitted during the period T.

$$ATAE(R_k) = \frac{\sum_{i=1}^{n}}{n}$$

(2) **Resource Availability (RA)**
If the cloud resources are affected by malicious attackers then they are not available for executing the jobs. We assume that $R_1, R_2 ..... R_k$ are the cloud resources. N_k denotes the number of jobs assigned to each R_k in the duration of time T. If A_k represents the number of jobs accepted by R_k, then RA is represented by the following equation:

$$RA(R_k) = \frac{A_k}{N_k}$$

(3) **Successful Transaction Rate (STR)**
It defines the rate of successfully completed jobs assigned to a resource R_k. We assume that A_k is the number of jobs accepted for execution by R_k. S_k is the number of jobs successfully completed by the R_k over the period T. The STR for R_k is represented by following equation.

$$STR(R_k) = \frac{S_k}{A_k}$$

(4) **Integrity Preservation (IP)**
It defines Integrity Preservation (IP) of jobs assigned to a resource R_k. We assume that C_k is the number of jobs preserved the integrity of a total number of jobs D_k assigned to a resource R_k over the period T. The IP for R_k is represented by following equation.

$$IP(R_k) = \frac{C_k}{D_k}$$

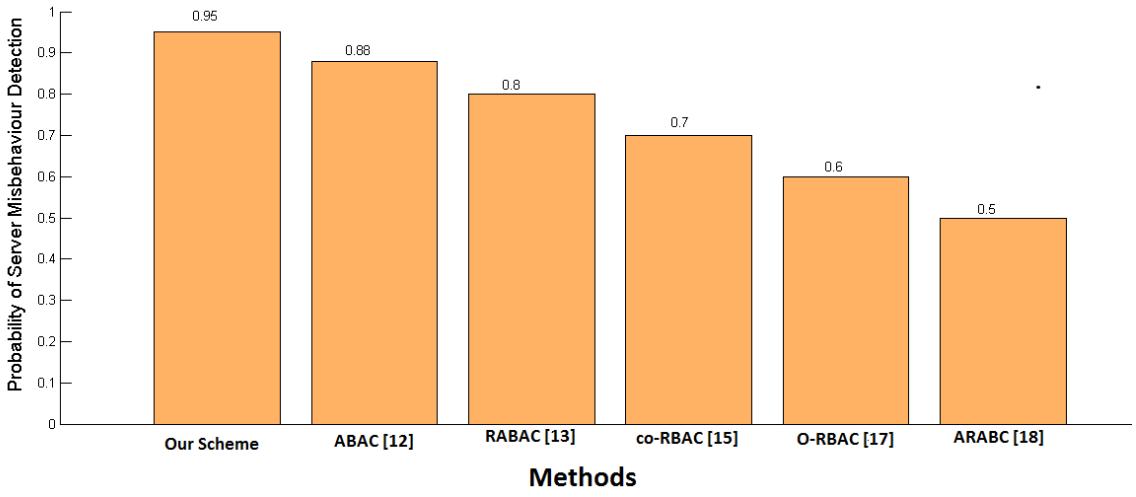

Figure 4: Comparative Analysis of Probability of Server Misbehaviour Detection

(5) **CSP Performance**

The Z_1, Z_2, Z_3, and Z_4 are the weighted values of TE, RA, STR, and IP, respectively.

$$CSPP = [(Z_1 * TE) + (Z_2 * RA) + (Z_3 * STR) + (Z_4 * IP)] * 100$$

Let t_n and t_{n-1} are the time interval for the current time window and previous time window, respectively. The Average CSP Performance (ACSPP) is evaluated by the following equation:

$$ACSPP = \gamma * (CSPP)_{t_n} + (1 - \gamma) * (ACSPP)_{t_{n-1}}$$

where γ and $(1 - \gamma)$ are the weighted values for t_n and t_{n-1} time interval, respectively.

Table III: Assessment of CSP Performance

$\mathbf{TAT_{Actual}}$	$\mathbf{TAT_{Est}}$	$\mathbf{TE(R_k)}$	$\mathbf{N_k}$	$\mathbf{A_k}$	$\mathbf{RA(R_k)}$	$\mathbf{S_k}$	$\mathbf{STR(R_k)}$	$\mathbf{C_k}$	$\mathbf{D_k}$	$\mathbf{IP(R_k)}$	**CSPP**
100	76	0.76	40	36	0.9	33	0.917	26	27	0.963	91.63
200	168	0.84	60	56	0.933	54	0.964	33	33	1	94.57
300	123	0.41	70	64	0.914	60	0.938	46	47	0.979	89.68
400	224	0.56	64	60	0.938	58	0.967	45	45	1	93.37
500	270	0.54	75	70	0.933	67	0.957	53	53	1	92.77
600	354	0.59	80	78	0.975	75	0.962	57	57	1	94.26
700	476	0.68	86	80	0.931	79	0.986	61	62	0.984	94.36
800	568	0.71	90	87	0.967	85	0.977	65	65	1	95.75
900	675	0.75	95	93	0.978	90	0.968	76	77	0.987	95.58
1000	780	0.78	100	98	0.98	95	0.969	85	85	1	96.47

We have chosen the value of Z_1, Z_2, Z_3, and Z_4 as follows: $Z_1 = 0.1$, $Z_2 = 0.2$, $Z_3 = 0.3$ and $Z_4 = 0.4$. The assessment of $CSPP$ is given in Table III.

The probability of server misbehavior detection for all the methods under the same condition has compared in Figure 4.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an access control method called as the Light-Weight Access Control (LWAC) method. We have surveyed existing access control techniques for cloud environment and noted the limitations of these methods for working in the Light-Weight environment.

To resolve these limitations, we have proposed the LWAC as a solution. We implemented the prototype of the proposed method in Hadoop and MapReduce framework and evaluated the user and CSP performances with the different parameters under the different conditions. We have also done the comparative analysis of our scheme with other methods. From the result analysis it is evident that our scheme is more effective and efficient solution in the Light-Weight environment.

References

Al-Kahtani, M., Sandhu, R., et al. 2002. A model for attribute-based user-role assignment. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*. IEEE, 353–362.

Ausanka-Crues, R. 2001. Methods for access control: advances and limitations. *Harvey Mudd College 301*.

Cloudera. 2014. Cloudera downloads get started with hadoop @ONLINE.

Lampson, B. W. 1974. Protection. *SIGOPS Oper. Syst. Rev. 8,* 1 (Jan.), 18–24.

Laurie, B. 2009. Access control (v0. 1).

Ruj, S. and Saxena, R. Jan 2015. Securing cloud data. *Cloud Computing with e-Science Applications*, pp 41–72.

Saxena, R. and Dey, S. 2014. Collaborative approach for data integrity verification in cloud computing. In *Recent Trends in Computer Networks and Distributed Systems Security - Second International Conference, SNDS 2014, Trivandrum, India, March 13-14, 2014, Proceedings*. 1–15.

Saxena, R. and Dey, S. 2015. Cloud shield: Effective solution for ddos in cloud. In *Internet and Distributed Computing Systems - 8th International Conference, IDCS 2015, Windsor, UK, September 2-4, 2015. Proceedings*. 3–10.

Saxena, R. and Dey, S. 2016. A novel access control model for cloud computing. 81–94.

Saxena, R. and Dey, S. 2017. A curious collaborative approach for data integrity verification in cloud computing. *CSI Transactions on ICT* .

Stinson, D., Wei, R., and Paterson, M. B. 2009. Combinatorial batch codes. *Advances in Mathematics of Communications 3,* 1, 13–27.

Sun, L., Wang, H., Yong, J., and Wu, G. 2012. Semantic access control for cloud computing based on e-healthcare. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*. IEEE, 512–518.

Tianyi, Z., Weidong, L., and Jiaxing, S. 2011. An efficient role based access control system for cloud computing. In *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*. IEEE, 97–102.

Tsai, W.-T. and Shao, Q. 2011. Role-based access-control using reference ontology in clouds. In *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*. IEEE, 121–128.

Wang, W., Han, J., Song, M., and Wang, X. 2011. The design of a trust and role based access control model in cloud computing. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*. IEEE, 330–334.

XenServer. 2014. Download xenserver 6.2 @ONLINE.

Rajat Saxena received the B.E degree from Jawaharlal Institute of Technology, Khargone and M.E degree from Shri Govindram Sakseria Institute of Technology and Science, Indore. He is working towards his P.hD under the supervision of Dr. Somnath Dey. He has ten years teaching experience in different colleges of India. His research focused on the computer network, security issues in Cloud computing and ad-hoc networks. Rajat is a member of IEEE computer society.

Somnath Dey is working as an Assistant Professor in the Discipline of Computer Science and Engineering at the Indian Institute of Technology, Indore. He has completed his B. Tech. from University of Kalyani in the year 2004 . He has earned an M.S. and P.hD from Indian Institute of Technology, Kharagpur in the year 2008 and 2013, respectively. His research focused on security issues in cloud computing, biometric security, image processing and human-computer interaction.