A Novel Access Control Model for Cloud Computing

Rajat Saxena(✉) and Somnath Dey

Cloud Computing Lab, Department of Computer Science and Engineering,
Indian Institute of Technology Indore, Indore, India
{rajat.saxena,somnathd}@iiti.ac.in

Abstract. Cloud Computing is the fast growing and the dominant field of Information Technology (IT) industry. It proposes on demand and cost effective services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Many security challenges are drawn from these services in cloud environment such as abuse of cloud services, data security, malicious insider and cyber-attacks. Although various access control policies and models such as Mandatory Access Control (MAC) and Role Based Access Control (RBAC) are existing, but these are not suitable for Cloud access control requirements.

In this paper, we analyze and identify different important gaps of the conventional access control schemes based on the their demerits and cloud access requirements. We also propose a Combinatorial Batch Codes Based Access Control (CBCBAC) model, which fulfill all the cloud access control requirements.

Our approach ensures the secure and efficient sharing of resources among various non-trusted tenants and also has the capacity to support different access permission to the same user for using multiple services securely. We also implement a prototype of our work which depicts the effective access control in the cloud environment.

Keywords: Cloud computing · Collaborative simulation · Combinatorial Batch Codes (CBC) · Mandatory Access Control (MAC) · Role Based Access Control (RBAC) · Access control models · Task-Role Based Access Control

1 Introduction

Cloud computing [1] is defined as services and applications that are enforced on a distributed network using virtual resources and accessed by common networking standards and Internet protocols. In cloud, resources [2] are not only virtually and limitless, the implementation details of the physical systems on which software runs are abstracted from the user as well.

In cloud data [3] is stored and operated in multi-tenant systems, which are distributed and shared by unrelated users within a large area. In addition, maintenance of security audit logs may be difficult or impossible for a user that has

W. Li et al. (Eds.): IDCS 2016, LNCS 9864, pp. 81–94, 2016.
DOI: 10.1007/978-3-319-45940-0_8

limited resources. Thus, cloud service providers must devote proper security measures and resources to maintain privacy and data integrity. The customer also must ensure that the provider has taken the proper security measures to protect their information.

In a cloud environment [4], the users might have limited CPU, battery power and communication resources. So, effective access control is one of the basic security issues in the Cloud environment. Many access control models are existing for different scope, organizations, communities and environments, but each model has many drawbacks and limitations.

Organization. The rest of the paper is organized as follows: In the Sect. 2, we describe literature survey on access control with the current state of art works, identifying limitations of these models and a proposal of an efficient access control method. In the Sect. 3, we describe our Combinatorial Batch Codes Based Access Control (CBCBAC) model based on requirement observation. In Sect. 4 we describe implementation phases and performance analysis of our approach. In Sect. 5, we provide security analysis of our scheme. Finally, we conclude in Sect. 6.

2 Literature Survey

Some basic access control methods are followed:

1. **Mandatory Access Control (MAC) Model:** In this model [5], a central authority is responsible for taking access decisions regarding a subject which requests for accessing objects. For secure access MAC model assigns an access class to each subject and object. An access class provides a security level to secure information flow between subject and object with dominance relationship. Although we have Bell and lapadula [6] and Biba [7] as two distinct variants and improvement of this model and provides protection against indirect information flow and leakage, but both this variant have a guarantee of complete secrecy of information.
2. **Discretionary Access Control (DAC) Model:** In this model [8], owner of the objects have the authority and the ability to restrict access to their objects or membership in certain groups or information in the objects based on user identities. DAC model is implemented either via identity based access control or Access Control Matrix/Access Control List (ACL). DAC model is generally less secure than the MAC model, so it used in low level protected systems.
3. **Role Based Access Control (RBAC) Model:** In this model [9], "a subject's responsibility is more important than whom the subject is", so a subject can have more than one role or be a member of multiple groups. Thus, this model is more realistic way to control access of resources in organizations. But it has the following drawbacks:
   – Choosing the right roles for representation of a system is not an easy task and may occur a worse situation when subjects dividing into categories based upon roles.

- Classification by subject into a number of categories makes mandatory for each subject to have a role in order to access the system.
- This model does not provide any kind of sensitivity to the information.
- This model does not delegation principle which is applicable in case of absences of employees.
- In this model relationships define according to identities not just roles.
- 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) Model:** This model [10] is based on Role Based Access Control model and assigns permissions to the tasks instead of roles. So, the user is assigned roles and this role is assigned tasks that have permissions. It uses a workflow authorization model for synchronizing workflow with authorization flow. Thus, this model uses tasks to support active access control and roles to support passive access control.
5. **Attribute Based Access Control (ABAC) Model:** This model [11] is based on a set of attributes associated with a requester or resource to be accessed in order to make decisions. This attribute may or may not be related with each other. After defining attributes, each attribute is considered 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. This model may be either Policy Based Access Control (PBAC) or Claims Based Access Control (CBAC). Thus, for accessing the system, subject just only needs to authenticate with the system and then it provides its attributes. It is a crucial decision in cloud computing that how many and what kind of attributes should be used for making decisions.
6. **Risk-Based Access Control (RBAC) Model:** This model [12] handles different kinds of risk levels and used operational need principal for adoption of access decision. It has a dynamic security policy which changes according to risk levels. The model implementation is difficult for cloud computing because of the high amount of analysis is required for assessment of risk levels.
7. **Adaptive Access Control Model:** This model [13] is based on contextual information such as time and security information. In it authors build a trust relationship between Cloud Service Providers (CSP's) and its consumers with role based access control system. A trust management system is maintained, which update and change trust level after each transaction. In this scheme, it is assumed that An Authority Authorization Centre (AAC) is maintained by each cloud which calculate and modified trust level based upon the users behaviour. This model has suffered from potential single point of attack and policy information failure.
8. **Cloud Optimized Risk Based Access Control (co-RBAC) Model:** This model [14] inherits the features of distributing environment, merge distributed authentication services together and have the ability of issuing certificate same as Certificate Authorities (CA). In this model hierarchical cache have been embedded to improve overall efficiency of access control

system. Dependency on CA for issuing certificate might cause efficiency and scalability problems because for each access time new certificate is needed.

9. **Task-Role Based Access Control Method:** In this method [15], Access activation or deactivation of permission depends on current task or process state. This scheme uses workflow authorization with synchronization workflow. Thus, tasks support active access control and roles support passive access control. It is implemented with Amazon Elastic Compute Cloud (Amazon EC2). But it suffered from heterogeneity problem, thus no clear indication of semantic and separation problem between the roles and tasks is handled.
10. **Ontology Using Role Based Access Control (O-RBAC) Model:** This model [16] provides the appropriate policy with an exact role for every tenant. Every subject can have multiple roles in multiple sessions. Thus, a role hierarchy is based on domain ontology and can be transferred between various ontological domains. This model has to ensure granting access decisions in a reasonable time and according to system requirements.
11. **Attribute Role-Based Access Control (ARBAC) Model:** This model [17] is a combination of Attribute Based Access Control and Role Based Access Control. It is implemented using eucalyptus open source cloud infrastructure. The main objective of this model is protecting data privacy. But the model does not provide clear explanation or evidence how it is protected. The role of component privacy manager and how they will combine RBAC and ABAC is not clearly depicted.

These conventional access control methods are very prohibitive, time consuming and error prone for novice users. We observed following limitations on conventional access control methods.

1. Cloud environment is complex and sophisticated because of dynamic nature of the cloud resources.
2. Data location is hidden from cloud users and may be in different countries that have different regulations for the same data. They may not trust with each other and may cause Service Level Agreement (SLA) issues.
3. Conventional models would be suffering from lack of flexibility in scalability and attribute management.
4. Cloud computing has heterogeneity and variety of services.
5. Diversity in access control policies and interfaces can cause improper interoperability.
6. High frequency delegation of large number of users, different classification, high dynamic performance and mobility features.
7. Different access permissions to a same cloud user, and giving him/her ability to use multiple services with regard to authentication and login time.
8. Multi tenancy, virtualization, sharing of resources and credential transformation are crucial aspects of cloud environment.

To keep these limitations in mind, we proposed a novel Combinatorial Batch Codes Based Access Control (CBCBAC) model, which have 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 cloud. Our approach verifies and guarantees that the cloud service provider could not learn about any data content stored in the cloud server during the efficient access control. Specifically, our contribution in this work can be summarized as the following three aspects:

1. We motivate the access control of data in cloud computing, and provide a new access control scheme with Combinatorial Batch Codes (CBC).
2. To the best of our knowledge, our scheme is the first to support scalable and efficient access control with CBC in the cloud computing.
3. We analyze the security and performance of our proposed scheme with current state-of-the-art.

3 The Proposed Scheme

In this section, we present our access control scheme for cloud services with antecedent research goals in mind. First, we establish notation related to our scheme, then we explain the details about CBC. There after we describe our scheme with CBC. Thereafter, we discuss algorithms that subsequently represent our scheme.

3.1 Notation and Preliminaries

1. h : The maximum height of the hierarchy.
2. id : Identity tuple (id_1 .....id_τ), where $1 \leq \tau \leq h$.
3. PP : Public Parameters.
4. κ : Security Parameter.
5. Msk : Master Key.
6. $E_{id}(\bullet)$ and $D_{id}(\bullet)$: denote the encryption and decryption algorithms.
7. M : Message.
8. C : Cipher-text.
9. G and G_T : Cyclic and multiplicative group of prime order p.
10. g : Random Number Generator.
11. u and v : Prime numbers $\in G$ and G_T, respectively.
12. e : Bilinear map.
13. pk : Public Key.
14. sk : Private Key.
15. $\mathcal{C}$: Combinatorial Batch Codes.
16. n : Number of file blocks.
17. m : Number of Cloud servers.
18. N : Total storage over m servers.
19. k : Selected number of elements.
20. t : Number of file blocks that at most read from each server.
21. $\mathcal{F}$: Set of n elements (or file blocks).
22. $\mathcal{S}$: Collection of m subsets of $\mathcal{F}$.

3.2 Combinatorial Batch Codes

Combinatorial Batch Code $\mathcal{C}$ [18] (n, N, k, m, t) 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 $N = \sum_{s \epsilon \mathcal{S}} |s|$, 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.

3.3 Combinatorial Batch Codes Based Access Control (CBCBAC) Model

Combinatorial Batch Codes Based Access Control (CBCBAC) model is based on CBC for distribution of access control of Cloud Service Providers (CSP) servers. In CBCBAC model, the generation of the private key can be a computationally intensive task. The identity of an entity must be authenticated before issuing a private key and the private key needs to be transmitted securely to the concerned entity.

CBCBAC model reduces the workload of the PKG by delegating the task of private key generation and hence authentication of identity and secure transmission of private key to its lowest levels. However, only the PKG has a set of public parameters. The identities at different levels do not have any public parameters associated with them. In CBCBAC model, identities are represented as vectors. So for a maximum height h of hierarchy (which is denoted as h-CBCBAC) any identity id is a tuple $(id_1, ..., id_\tau)$, where $1 \leq \tau \leq h$.

Let, $id^{'} = id_1^{'}, ..., id_j^{'}, j \leq \tau$ be another identity tuple. We say $id^{'}$ is a prefix of id if $id_i^{'} = id_i$ for all $1 \leq i \leq j$.

In CBCBAC model, the PKG has a set of public parameters PP and a master key Msk. For all identities at the first level, the private key is generated by the PKG using Msk. For identities at the second level onwards, the private key can be generated by the PKG or by any of the ancestors of that identity. Figure 1 shows the CBCBAC model for the Cloud environment. In this scheme, the private key sk_{id} of id can be generated by an entity whose identity is a prefix of the id and who has obtained the corresponding private key.

Our CBCBAC 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 PP 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 cipher-text space C and the identity space I. Figure 2 (a) presents the steps of setup operation.

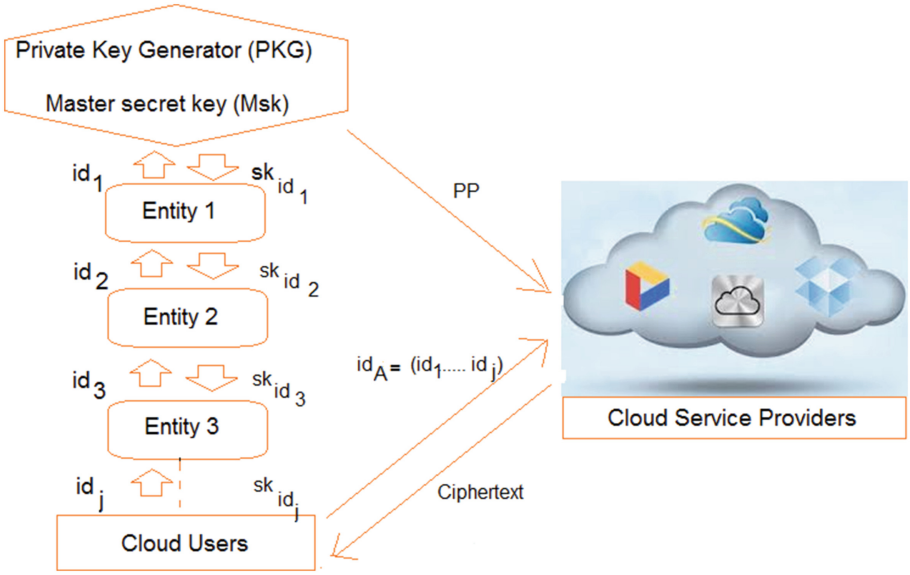

Fig. 1. CBCBAC model for cloud environment

Algorithm 1: Setup operation

Input: $\{0, 1\}^k$
Output: Initial Msk, PP

1. Initial $Msk \xleftarrow{R} \{0,1\}^k$
2. Initial $PP \xleftarrow{R} \{0,1\}^k$

(a) Algorithm for **Setup** operation

Algorithm 2: Key generation

Input: $id = (id_1, \cdots, id_j)$, where $j \geq 1$ and $sk(id_{j-1})$ for the identity $(id_1, \cdots, id_{j-1})$
Output: $sk(id)$

1. if j=1 then
2. $sk(id_1) \leftarrow BilinearPair(Msk, id_1)$
3. else
4. $sk(id_j) \leftarrow BilinearPair(sk(id_{j-1}), id_j)$
5. endif
6. return $sk(id_j)$

(b) Algorithm for **Key-gen** operation

Algorithm 3: Encryption operation

Input: PP, id_j *and* $\mathcal{M}$
Output: C

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

(c) Algorithm for **Encryption** operation

Algorithm 4: Decryption operation

Input: PP, id_j C and $sk(id)$
Output: $\mathcal{M}$ Ψ

1. if Cipher-text is not valid them
2 return Ψ
3. else
4. return $\mathcal{M} \leftarrow D_{id_j}(C)$
5. endif

(d) Algorithm for **Decryption** operation

Fig. 2. Algorithms for Combinatorial Batch Codes Based Access Control (CBCBAC) model

2. Key-Generation: This operation generates the private key $sk(id_j)$ corresponding to the jth identity. This method uses bilinear pairing [9] between identity tuple id $= (id_1, ..., id_j)$, $j \geq 1$ and the private keys sk$(d_{id|j-1})$ for the identities $((id_1, ..., id_{j-1}))$. Bilinear pairing defines a map between two cyclic

groups of some prime order and satisfies bi-linearity, non-degeneracy and efficient computability properties [9]. 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 an identity at any level can produce the decryption key. Key generation algorithm is given in Fig. 2 (b).

3. Encryption: This process encrypts a message M by a public parameter PP of an identity id and produces a cipher-text C. We use a standard encryption algorithm E. The steps of encryption operation are specified in Fig. 2 (c).

4. Decryption: This process takes the public parameter PP, an identity id, a Cipher-text C and a private key $sk(id)$ as input and compute the original message M. If the cipher-text is not valid, this algorithm produces ψ. We use standard decryption algorithm D corresponding to E in the decryption process. Decryption algorithm is presented in Fig. 2 (d).

4 Implementation

To demonstrate our approach, we implement 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 [19] on one PC that is used for file storage. The second PC configured with Cloudera CDH 5.3.0-0 [20]. This is used as a Cloud Service Provider that provides access control of the stored files of cloud users.

The working of CBCBAC model is divided into 5 phases of access control. Fig. 3 describes these phases.

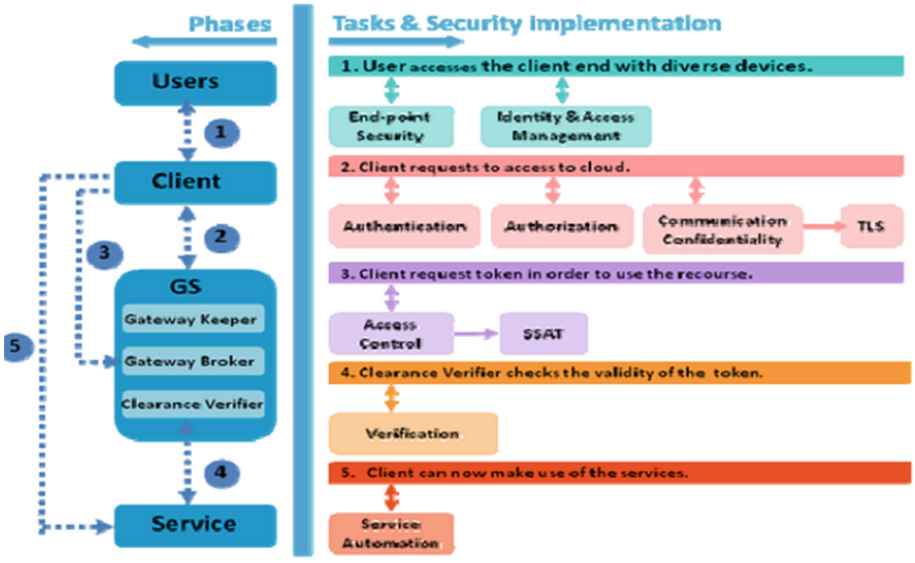

Fig. 3. Phases of access control for cloud environment

1. **Phase 1:** A client program is installed on or downloaded to every endpoint (laptop, cell-phone, etc.) when user accesses the client end. A server or gateway hosts the centralized security program, which verifies logins and sends updates and patches when needed.
2. **Phase 2:** In the second 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.
3. **Phase 3:** The GS needs to securely identify their users through authentication and after that, a user must gain authorization for doing certain tasks. With the single sign-on access control 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.
4. **Phase 4:** This step is a precaution against SSAT forging. If the CV reports back that the Gateway Server did not generate the SSAT, the request is blocked. If the SSAT is examined and proved valid, the CV attaches a verification token to the SSAT.
5. **Phase 5:** Client can now make use of the services.

4.1 Performance Analysis

We measure the performance of our proposed model by computing the efficiency of a user access control that is how efficiently and frequently a user can access data from the cloud service providers. In our scheme, the number of data access can be set flexible according to users requirement. We compute the efficiency on the basis of time of access, encryption and decryption. Efficiency of our scheme is calculated using Eq. 2. Figure 4 shows the performance comparison of our scheme with the different schemes.

$$Efficiency = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Clocks}{Instructions} \times \frac{Seconds}{Clocks} \quad (2)$$

The Advantages of a CBCBAC model are following.

1. It has dynamic performance and mobility features to support remote access to its resources.
2. Cloud based systems need reliable mechanisms for proving users' identities and authenticating them. login time and time of authentication is less to improve performance of the system.
3. A trust relationship between cloud user and cloud service providers is induced to get more attention.
4. The model have scalable in terms of users, enforcement points and policy evaluation.
5. The model have heterogeneous to adopt a vast number of diverse technologies and mechanisms.

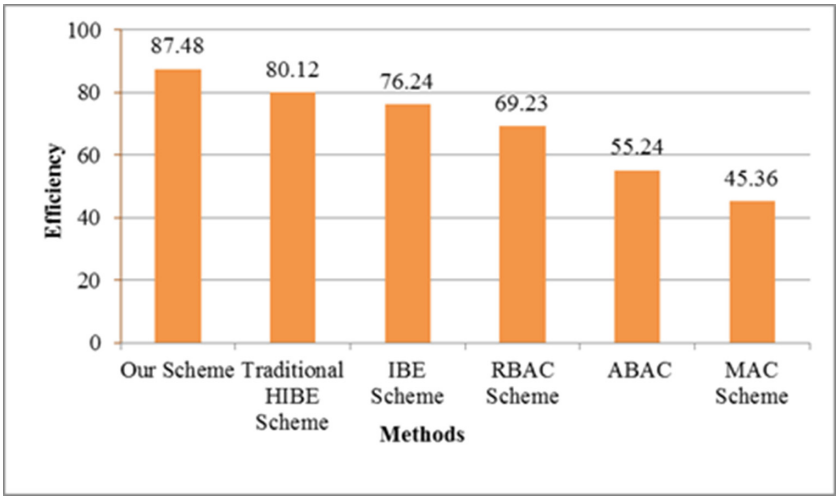

Fig. 4. Efficiency comparison of different methods

6. The model have interoperable among different consumer requirements and resources.
7. It has less computation complexity and response time for providing better Quality of Service (QoS).
8. It must be able to handle the flexibility in virtualization, sharing of resources and attribute management.
9. In case of using multiple cloud, users should be able to assign and ease privileges and transfer their credentials across different layers of clouds.
10. It has auditing and delegation capabilities for policy management to add, delete, change, import, and export of user's file.
11. It has flexibility in configuration, operation and situational awareness for users.
12. It has operating system compatibility and support for passive and active workflows.

5 Security Analysis

In this section we discussed about the standard, soundness requirement of our scheme. In the later part, we test our scheme against Chosen Ciphertext attack.

5.1 Standard Soundness Requirement

Standard soundness requirement of our CBCBAC model is following:

Algorithm 1. Standard soundness requirement of our CBCBAC scheme

1: **if** (PP,msk) is output by Set-Up;
d_{id} is a private key corresponding to the identity tuple id generated by the Key-Generation algorithm;
and $\mathcal{C}$ is the output of the Encryption algorithm for a message M $\epsilon\mathcal{M}$ using id as a public key and PP; **then**
2: The Decryption algorithm must return M on input d_{id} and $\mathcal{C}$.
3: **end if**

5.2 Test of Our Scheme Against Chosen Ciphertext Attack

At the basic level, the security model of CBCBAC has formalisation of the adversarys inability to distinguish between ciphertexts arising out of two equal length messages M_0 and M_1.

For this, an identity is chosen by the adversary as the target identity, i.e., the goal of the adversary is to compromise the security of the identity it chooses as the target identity. A random bit γ is chosen and challenge the ciphertext is produced by encrypting M_γ under the target identity. The adversary wins if it can predict γ with a probability significantly away from half.

Let our CBCBAC scheme as defined in the previous section is $\mathcal{H}$. The IND-ID-CCA security (Indistinguishability under Adaptive Identity and Adaptive Chosen Ciphertext Attack) for $\mathcal{H}$ is defined in terms of the following game between a challenger and an adversary of the CBCBAC. The adversary is allowed to place two types of oracle queries decryption queries to a decryption oracle $\mathcal{O}_d$ and key extraction queries to a key-extraction oracle $\mathcal{O}_k$.

Figure 5 shows a schematic diagram of the security game defining the security of our CBCBAC scheme.

1. **Set-Up**: The challenger takes as input a security parameter 1^k and runs the Set-Up algorithm of the CBCBAC. It provides $\mathcal{A}$ with the system parameters PP while keeping it the master key Msk.
2. **Phase 1**: Adversary $\mathcal{A}$ makes a finite number of queries where each query is one of the following two types:
key-extraction query (id): This query is placed to the key-extraction oracle $\mathcal{O}_d$. Questioned on id, $\mathcal{O}_k$ generates a private key d_{id} of id and returns it to A. The Key-Generation algorithm is probabilistic and so if it is queried more than once on the same identity, then it may provide different (but valid) decryption keys.
decryption query (id,C): This query is placed to the decryption oracle $\mathcal{O}_d$. It returns the resulting plaintext or $\perp$ if the ciphertext cannot be decrypted. $\mathcal{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers.
3. **Challenge**: When $\mathcal{A}$ decides that Phase 1 is complete, it fixes an identity id^* and two equal length messages M_0, M_1 under the (obvious) constraint that it has not asked for the private key of id^* or any prefix of id^*. The challenger chooses uniformly at random a bit $\gamma\varepsilon 0, 1$ and obtains a ciphertext

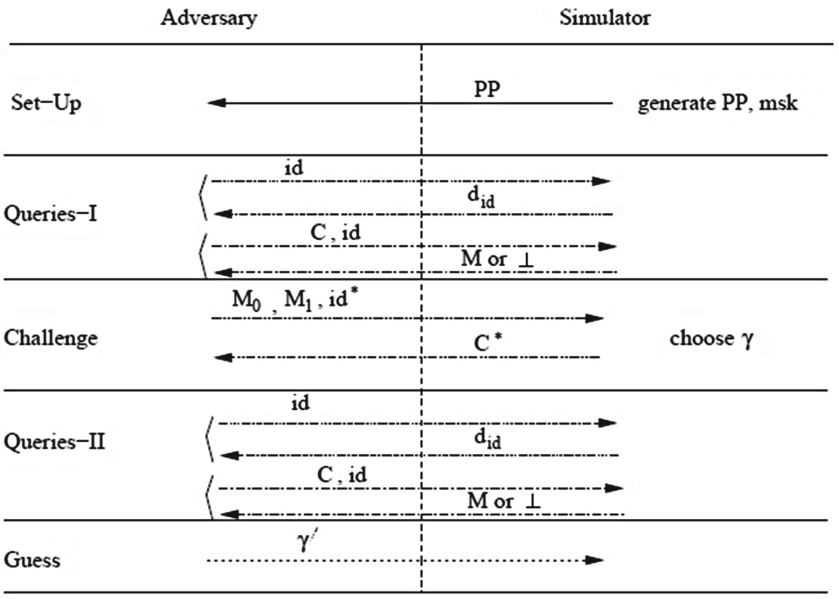

Fig. 5. Work flow of security game for our scheme.

C^* corresponding to γ, i.e., C^* is the output of the Encryption algorithm on input (γ, id^*, PP). It returns C^* as the challenge ciphertext to $\mathcal{A}$.

4. **Phase 2**: $\mathcal{A}$ has now issued additional queries just like Phase 1, with the (obvious) restriction that it cannot place a decryption query for the decryption of C^* under id^* or any of its prefixes nor a key-extraction query for the private key of id^* or any prefix of id^*. All other queries are valid and $\mathcal{A}$ can issue these queries adaptively just like Phase 1. The challenger responds as in Phase 1.
5. **Guess**: $\mathcal{A}$ outputs a guess γ' of γ. The advantage of the adversary $\mathcal{A}$ in attacking the CBCBAC scheme $\mathcal{H}$ is defined as:

$$Adv_{\mathcal{A}}^{\mathcal{H}} = |Pr[(\gamma = \gamma')] - 1/2|.$$

Our CBCBAC scheme $\mathcal{H}$ is said to be (t, q_{id}, q_C, ϵ)-secure against adaptive chosen ciphertext attack ((t, q_{id}, q_C,ϵ)-IND-ID-CCA secure) if for any t-time adversary $\mathcal{A}$ that makes at most q_{id} private key queries and at most q_C decryption queries, $Adv_{\mathcal{A}}^{\mathcal{H}}$. In short, we say $\mathcal{H}$ is IND-ID-CCA secure or CCA-secure.

6 Conclusions and Future Work

In this paper, we provide an efficient CBCBAC approach for access control in cloud computing. We survey all the current techniques. We accomplish that none of previous existing techniques practically feasible in cloud context.

We also provide brief security analysis of our scheme. In the near future, we planned to work on implementation and performance analysis of our scheme. This will provide effective and efficient authentication in a cloud environment.

References

1. Saxena, R., Dey, S.: Cloud shield: effective solution for DDoS in cloud. In: Di Fatta, G., Fortino, G., Li, W., Pathan, M., Stahl, F., Guerrieri, A. (eds.) IDCS 2015. LNCS, vol. 9258, pp. 3–10. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23237-9_1
2. Saxena, R., Ruj, S., Sarma, M., Collaborative model for privacy preservation, data integrity verification in cloud computing. In: Proceedings of the Security and Privacy Symposium, IIT Kanpur, Kanpur, India, February 2013
3. Ruj, R., Saxena, R.: Securing cloud data. In: Cloud Computing with e-Science Applications, pp. 41–72, January 2015. doi:10.1201/b18021-4
4. Saxena, R., Dey, S.: Collaborative approach for data integrity verification in cloud computing. In: Martínez Pérez, G., Thampi, S.M., Ko, R., Shu, L. (eds.) Recent Trends in Computer Networks and Distributed Systems Security. Communications in Computer and Information Science, vol. 420, pp. 1–15. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54525-2_1
5. Ausanka-Crues, R.: Methods for access control: advances and limitations, Harvey Mudd College 301
6. LaPadula, L., Bell, D.E., LaPadula, L.J.: Secure computer systems: Mathematical foundations, Draft MTR, The MITRE Corporation 2
7. Biba, K.J.: Integrity considerations for secure computer systems. Technical report, DTIC Document (1977)
8. Lampson, B.W.: Protection. SIGOPS Oper. Syst. Rev. **8**(1), 18–24 (1974). doi:10.1145/775265.775268
9. Laurie, B.: Access control (v0. 1) (2009)
10. Oh, S., Park, S.: Task-role-based access control model. Inf. Syst. **28**(6), 533–562 (2003)
11. Al-Kahtani, M., Sandhu, R., et al.: A model for attribute-based user-role assignment. In.: 2002 18th Annual Proceedings of Computer Security Applications Conference, pp. 353–362. IEEE (2002)
12. Brucker, A.D., Brügger, L., Kearney, P., Wolff, B.: An approach to modular, testable security models of real-world health-care applications. In: Proceedings of the 16th ACM Symposium on Access Control Models and Technologies, pp. 133–142. ACM (2011)
13. Wang, W., Han, J., Song, M., Wang, X., The design of a trust, role based access control model in cloud computing. In: 2011 6th International Conference on Pervasive Computing and Applications (ICPCA), pp. 330–334. IEEE (2011)
14. Tianyi, Z., Weidong, L., Jiaxing, S.: An efficient role based access control system for cloud computing. In: 2011 IEEE 11th International Conference on Computer and Information Technology (CIT), pp. 97–102. IEEE (2011)
15. Sun, L., Wang, H., Yong, J., Wu, G.: Semantic access control for cloud computing based on e-healthcare. In: 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 512–518. IEEE (2012)
16. Tsai, W.-T., Shao, Q.: Role-based access-control using reference ontology in clouds. In: 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS), pp. 121–128. IEEE (2011)

17. Mon, E.E., Naing, T.T.: The privacy-aware access control system using attribute- and role-based access control in private cloud. In: 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), pp. 447–451. IEEE (2011)
18. Stinson, D., Wei, R., Paterson, M.B.: Combinatorial batch codes. Adv. Math. Commun. **3**(1), 13–27 (2009)
19. XenServer, Download xenserver 6.2 @ONLINE (2014). http://xenserver.org/open-source-virtualization-download.html
20. Cloudera, Cloudera downloads get started with hadoop @ONLINE (2014). http://www.cloudera.com/content/cloudera/en/downloads.html