Privacy-aware OrLa Based Access Control Model in the Cloud

Pengfei Shao, and Shuyuan Jin*
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
GuangDong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou, China
Email: shaopf@mail2.sysu.edu.cn, jinshuyuan@mail.sysu.edu.cn

***Abstract*—In recent years, cloud computing has been widely adopted by an increasing number of enterprises and individuals because of its attractive features, such as its large scale, low costs, and pay-per-use. Nevertheless, traditional access control models cannot satisfy the security requirements of complex cloud environments. In this paper, a privacy-aware access control model (Pa-OrLaBAC) is proposed that emphasizes privacy protection and flexibility. This model combines Organization based Access Control (OrBAC) model with Label-based access control (LaBAC) model and retains their respective advantages, making it more suitable for the cloud. By introducing the concept of purpose, the issue of lacking privacy protection is well addressed and the problem of the separation of control and ownership is alleviated to some extent. In order to get a more precise access purpose, two methods (static declaration and dynamic acquisition) and a negotiation module are also applied in this model. Finally, we illustrate the use of Pa-OrLaBAC with a case study and summarize this model.**

***Keywords*—access control, cloud, privacy protection, flexibility**

I. Introduction

There are many definitions of cloud computing, the most widely accepted of which was proposed by the National Institute of Standards and Technology (NIST): "Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1]. In recent years, cloud computing has been adopted by an increasing number of enterprises and individuals as a new computing model because of its appealing characteristics, such as its ultra-large scale, high scalability, high dynamics, low costs and pay-per-use. In addition, users can obtain the required resources through the network without geographical limitations. Despite these benefits, the security of cloud computing remains a major factor hindering its development. Therefore, ensuring cloud security is one of the urgent tasks in cloud computing environments [2].

Access control [3] is the fundamental security method for the promotion and protection of network security and is used to confirm or deny a request for one subject to access an object. A proper access control model can prevent unauthorized users from maliciously or unintentionally obtaining data [4].

DOI reference number: 10.18293/SEKE2020-038

Moreover, there are a huge number of users and a huge amount of data in the cloud, making the traditional coarse-grained access models based on pre-defined rules inappropriate.

The cloud-specific outsourcing business model separates ownership and control. With the increasing number of companies and individuals migrating their data to the cloud, the protection of users' private information has become a major focus in this field. Privacy is defined as the rights of individuals to decide when, how, and to what extent their information could be shared with others [5]. The Organisation for Economic Co-operation and Development (OECD) Guidelines [6] are the most widely adopted principles of privacy protection, and they claim that purposes, conditions and obligations are the key elements of privacy preserving access control models. The primary concern of a privacy policy is the specific reason why the data have been collected or used [7]. However, the traditional access models are not designed to enforce privacy policies and satisfy privacy protection requirements [8]. That is because these models are based on guiding the operations of the user on relative data [9].

Nevertheless, for now, few existing studies on access control models focus on both privacy protection and flexibility. Considering the aforementioned reasons, the main aim of this paper is to address the problem of privacy disclosures in the cloud while ensuring the flexibility of the access control model.

This paper proposes an extensible access control model for privacy protection — the Privacy-aware OrLa based access control model (Pa-OrLaBAC) — for cloud environments. The features of this model are as follows.

1) Privacy-aware. By introducing the concept of purpose, privacy protection is strengthened and the shortcoming, which is the separation of ownership and control, of the cloud is alleviated to some degree.
2) Flexibility. The presented model preserves the extensibility of the Label-based access control model. In Pa-OrLaBAC, other proper labels could be added according to the specific requirements.

The rest of this paper is organized as follows: Section II summarizes related works. Section III presents necessary preliminaries to establish our model. Section IV details the Pa-OrLaBAC model. Section V introduces a usage scenario. Section VI concludes this paper and points out our future direction.

II. Related Work

Traditional access control models can be divided into three main classes: discretionary access control (DAC) [3], mandatory access control (MAC) [10], and role-based access control (RBAC) [9].

DAC restricts object access on the basis of the identity of the subjects or the groups to which they belong. For now, it is the most commonly used model in computer operating systems. However, it does allow legitimate users to pass permissions or rights to others, regardless of whether they are malicious or not.

In MAC, only the system administrator or central authority is responsible for designing and managing access control policies that cannot be changed or granted by the users. MAC is often used in military areas that require multi-level security. Thus, this centralized authorization approach can neither provide flexibility nor support the separation of duties or Least Privilege.

RBAC was designed to solve the shortcomings of the previous models. In RBAC, permissions are associated with roles, and users gain accesses to objects by acquiring the appropriate roles. RBAC satisfies the security needs of various organizations and also improves the efficiency and reduces the complexity of authorization management. However, due to the many-to-many mapping relationships between roles and users, when it is applied to the cloud, the role explosion problem may occur, and the flexibility may be affected by the millions of dynamic users and permissions that exist in commercial networks [11].

As is shown above, traditional access control models are not perfectly applicable to the cloud. Therefore, a variety of new access control modes have been proposed. Among which the attribute-based access control (ABAC) [12] model, the organization based access control (OrBAC) [13] model, and the label-based access control (LaBAC) [14] model are the most representative.

In ABAC, access is granted or denied according to a set of attributes that are associated with the subject, object and environment. It does possess more granularity and flexibility compared with traditional models. The main drawback of ABAC is how to accurately select the attributes for access decisions in a specific application environment such as the cloud [11]. Otherwise, designing a rich computational language to define attribute-based rules makes policy update and policy review NP-complete or even indeterminate problems [14].

Mustapha Ben Saidi et al. [15] proposed an access control model based on OrBAC that introduces the concept of the Trusted Third Party. Their goal is to better control the external connections of users with different accesses. To ensure a continuity of critical infrastructures, Nawal AIT AALI et al. [16] proposed an access control model based on trust management using the OrBAC model. This model can both manage different resource access policies from other organizations and keep the trust between collaborating organizations. By extending OrBAC with new entities and introducing a new trust relationship among tenants, MA Madani et al. [17] proposed an approach that ensures the access control to the shared resources in a collaborative session in cloud environments.

Roger E. Sanders [18] proposed a method for securing data using label-based access control (LBAC) in which data are protected by the security label. Only the administrator can modify the labels. In [19], access is managed based on the user label and the data label. Labels provide extra protection, especially for sensitive data such as credit cards and Social Security Numbers (SSNs). Chen et al. [20] proposed a novel framework, the multi label-based access control model, which uses different labels to provide access security for big data applications. Chinnasamy P et al. [21] proposed a solution to overcome data security defects by implementing multi label-based scalable access control as a service for the cloud. This model enables data owners to keep the authority over their resources.

Although the OrBAC model considers the context when making access requests and overcomes the limitation of directly binding permissions to roles, it is more suitable for centralized structure because of lacking flexibility [11] and it mainly restricts access control with respect to the subject.

LaBAC expresses authorization policies in the form of enumeration, and it is a variable-grained access control method that labels subjects and objects. Meanwhile the drawback, the separation of ownership and control, that is caused by cloud computing is alleviated. However, one concern about this model is that the costs of storing the potentially large number of enumerated tuples would be high [14]. Furthermore, neither of them takes both privacy protection and the flexibility of the access control model into account.

As we all know, utilizing multiple models with other enhancements may achieve a better result [22]. Inspired by this thought, in this paper, we propose a new access control model (Pa-OrLaBAC) for cloud computing. It could be an effective method to ensure the security of data in cloud environments.

III. Preliminaries

In this section, we introduce the required concepts that will be used in the Pa-OrLaBAC model.

A. Organization based Access Control (OrBAC)

The core feature of OrBAC [13] is the organization, and it defines a new level of abstract entities that are separated from concrete ones. The entities of subject, action and object are abstracted as role, activity and view, respectively. The other entity is the context, which is used to specify the concrete circumstances in which organizations grant role permissions to perform activities on views. Unlike RBAC, in OrBAC, after the subject is granted to the appropriate role, it no longer immediately obtains the access permission to the object. Instead, on the abstract level, the role obtains permission to perform an activity on the view in a certain context. Then, the access permission of the concrete level is derived from the abstract one. To make this transition, OrBAC also defines

some relationships that associate abstract entities with concrete ones. The framework of the model is shown as Figure 1.

- The *Employ* relationship
  In this model, a Subject is an active entity, i.e., a user. The entity Role indicates the status of the subject in the organization. If *org* is an organization, *s* is a subject and *r* is a role, then *Employ (org, s, r)* means that *org* employs subject *s* in role *r*.
- The *Use* relationship
  The entity Object is the resource being accessed. A View corresponds to a set of objects that satisfy a common property. If *org* is an organization, *o* is an object and *v* is a view, then *Use (org, o, v)* means that *org* uses object *o* in view *v*.
- The *Consider* relationship
  The entity Action contains computer actions such as read, write, and send. In some cases, different organizations may decide that the same action comes under different activities. Therefore, if *org* is an organization, α is an action and *a* is an activity, then *Consider (org, α, a)* means that *org* considers that action α falls within activity *a*.
- The *Define* relationship
  Contexts could be used to specify the concrete circumstances where organizations grant role permissions to perform activities on views. If *org* is an organization, *s* is a subject, *o* is an object, α is an action and *c* is a context, then *Define (org, s, o, α, c)* means that within organization *org*, context *c* is true among subject *s*, object *o* and action α.
- The *Permission* relationship
  This is the access authorization at the abstract level. If *org* is an organization, *r* is a role, *v* is a view, *a* is an activity and *c* is a context, then *Permission (org, r, v, a, c)* means that organization *org* grants role *r* permission to perform activity *a* on view *v* within context *c*.
- The *Is_permitted* relationship
  This is the concrete authorization that can be derived from the abstract one. *Is_permitted (s, o, α)* means that subject *s* is permitted to perform action α on object *o*.

The procedure through which a subject can obtain permission to perform an action on the object is as follows:

Employ (org, s, r) $\wedge$ *Use (org, o, v)* $\wedge$ *Consider (org, α, a)* $\wedge$ *Define (org, s, o, α, c)* $\wedge$ *Permission (org, r, v, a, c)* $\longrightarrow$ *Is_permitted (s, o, α)*.

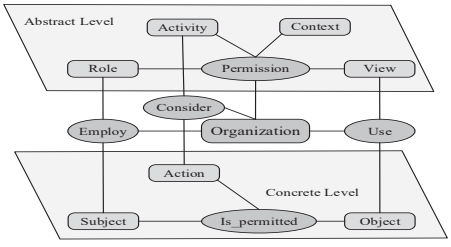

Fig. 1. Basic framework of OrBAC model.

B. Label-based access control (LaBAC)

The LaBAC [14] model expresses policies in the form of enumeration. Every subject and object is tagged using labels. A Label in LaBAC is a precise type of attribute. Values can be assigned by the administrator. The basic framework of LaBAC is shown in Figure 2.

In this model, the sets of users, objects and actions are denoted by U, O and A, respectively. Users are associated with a label function named uLabel, which maps the user to one or more values from the finite set UL (user label values). Similarly, the objects use oLabel to map the object label values (OL). A policy consists of a subset of tuples from the set of all tuples UL x OL. Only one policy can be defined for each action, which is denoted as Policy_a. If and only if the two-tuples group (ul, ol) $\in$ Policy_a is true will the related action be authorized.

An issue in LaBAC is that a complex access policy may need many or a significantly large number of enumerated policies to be defined. This may lead to a situation that the number of labels is greater than the number of entities in the system.

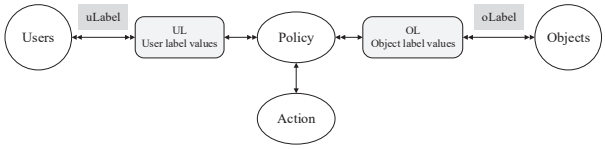

Fig. 2. Basic framework of LaBAC model.

C. Privacy protection

Typical privacy policies for data include purposes, conditions and obligations. The obligation designates the actions that must be followed after access is allowed. Conditions are prerequisites that should be satisfied when any action can be performed [23]. Purposes describe the reasons why the data are collected or used [24]. Platform for Privacy Preferences (P3P) defines the purposes as "the reason(s) for data collection and use" and specifies a set of purposes (World Wide Web Consortium). In commercial situations, purposes normally have hierarchical associations, i.e., generalization and specialization relationships. For instance, a group of purposes such as direct-marketing and third-party marketing can be represented by a more general purpose, marketing. We adopt the purpose definition from Byun et al. [7].

Definition 1 (Purpose and Purpose Tree): A purpose describes the reason(s) for data collection and data access. A set of purposes, which is denoted as Ω, is organized in a tree structure, which is referred to as a Purpose Tree and denoted as Φ. Each node in the Purpose Tree represents a purpose in Ω and each edge represents a hierarchical relation (i.e., specialization and generalization) between two purposes.

Figure 3 shows an example of a purpose tree. For instance, p_i and p_j are two purposes in Φ, and we say that p_i is an ancestor of p_j (or p_j is a descendent of p_i) if there is a downward path from p_i to p_j in Φ.

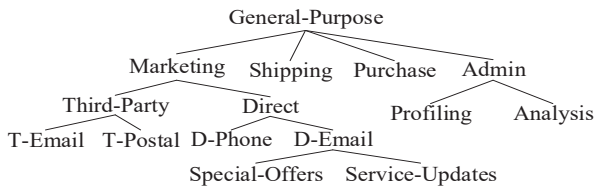

Fig. 3. Purpose Tree.

For a set of purposes, R, in Φ, the following notations will be used.

$R^{\blacktriangle}$ is the set of all nodes that are ancestors of the nodes in R, including the nodes in R themselves.

$R^{\blacktriangledown}$ is the set of all nodes that are descendants of the nodes in R, including the nodes in R themselves.

$R^{\blacklozenge}$ is the set of all nodes that are either ancestors or descendants of the nodes in R, that is, $R^{\blacklozenge} = R^{\blacktriangle} \cup R^{\blacktriangledown}$.

Definition 2 (Access Purpose, AP): An access purpose is used to access data objects, and it should be confirmed when data are requested.

Definition 3 (Intended Purpose, IP): The intended purpose is the data-related purpose that regulates data access. When access is requested, the access purpose is checked against the intended purposes. An intended purpose consists of two components: the Allowable Intended Purposes (AIP for short) and the Prohibited Intended Purposes (PIP for short).

Allowable Intended Purpose (AIP): Data providers explicitly allow data access for a particular purpose.

Prohibited Intended Purpose (PIP): Data providers strictly disallow data access for a particular purpose.

Therefore, an intended purpose (IP) is a tuple $\langle$AIP, PIP$\rangle$, where AIP $\subseteq \Phi$ and PIP $\subseteq \Phi$ are two sets of purposes. We adopt the denial-takes-precedence policy that PIP overrides AIP if there are conflicts between the AIP and the PIP for the same data element.

Definition 4 (Access Purpose Compliance): Let Φ be a purpose tree. IP = $\langle$AIP, PIP$\rangle$ be an intended purpose and AP be an access purpose that are defined over Φ, respectively. AP is said to be compliant with IP according to Φ if and only if the following two conditions are satisfied:

1. AP $\in AIP^{\blacktriangledown}$, and
2. AP $\notin PIP^{\blacklozenge}$.

IV. Our Proposed Model

The access control model that is proposed in this work combines the OrBAC with LaBAC and introduces the concept of purpose. Its main framework is illustrated in Figure 4.

This paper only uses the "Purpose" label to protect privacy. By integrating the advantages of OrBAC and LaBAC, flexibility and fine-granularity can be achieved. It should be noticed that other proper labels could be added according to the specific requirements. In the following statement, the same parts that were previously depicted will not be described again, and new components that are extended or modified in Pa-OrLaBAC will be explained in detail.

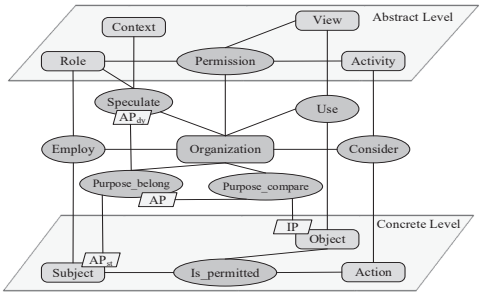

Fig. 4. The main framework of Pa-OrLaBAC.

A. Context

In the traditional OrBAC, the actual circumstances in which organizations grant role permissions to perform activities on views could be clarified by using the entity context. In Pa-OrLaBAC, contexts will be used to specify the concrete conditions that organizations use to determine the dynamic access purpose (AP_{dy}) of the role. The process of inferring the AP_{dy} using the Context will be given later.

B. Access Purpose Authorization

a) Access Purpose: There are three possible ways to confirm the access purpose [7]. First, the users can be required to explicitly declare their access purposes along with the requests. Obviously, this method is the easiest to implement. Most privacy preserving access control models are based on it. Nevertheless, it demands the complete trust of the subjects, which is not suitable at all for an open environment. The second possible method is that the system registers a special access purpose for each application or stored procedure in advance. It may not be used in complex applications or stored-procedures scenarios in which subjects may access different objects for multiple access purposes. The third is that access purposes can be dynamically determined based on the current context.

Considering that not all the users in the cloud are absolutely trustworthy, in this paper, we use both the first and third methods to identify the access purpose. Those access purposes that are declared by users are named Static Access Purposes (AP_{st}). Similarly, those purposes that are dynamically determined are named Dynamic Access Purposes (AP_{dy}). One thing that should be noted is that only the AP_{st}s are dispensable.

Therefore, two new relationships and the Negotiation module are defined as follows.

- The *Speculate* Relationship
  This relationship is used to generate dynamic access purposes. *Speculate (org, c, r, AP_{dy})* means that if *org* is an organization, *c* is the current context, and *r* is a role, the dynamic access purpose AP_{dy} is true.
  Suppose that an employee of a delivery company is asking for access to a customer's address using a specific application during normal business time. We could

speculate that the AP_{dy} of this employee is shipping in such a situation.

- The *Purpose_belong* Relationship
  Purpose_belong(org, AP_{st}*,* AP_{dy}*)* $\rightarrow$ {True, False} is used to determine the affiliation between AP_{st} and AP_{dy}. If the user does not declare his own AP_{st}, we consider that *Purpose_belong* is always True. Under this circumstance, AP is AP_{dy}. Considering that this may lead to a situation in which some malicious users could intentionally hide their real access purposes, we introduce the Negotiation module as a reward mechanism. Otherwise, *Purpose_belong (org,* AP_{st}*,* AP_{dy}*)* = True iff $AP_{st} \in AP_{dy}^{\blacktriangledown}$. Meanwhile, AP is AP_{st}.
- Negotiation module
  This module is activated only when the user declares his AP_{st} and $AP_{st} \notin AP_{dy}^{\blacktriangledown}$. In this case, the data request is not immediately terminated. Instead, the user can get a second chance to modify his AP_{st} or the Context, which means another opportunity to access the data item that he is requesting.

Example 1. Suppose AP_{dy} = "Third-Party" is defined over the purpose tree given in Figure 3.

Therefore, $AP_{dy}^{\blacktriangledown}$= {Third-Party, T-Email, T-Postal}.

1) If the user does not declare his AP_{st}, then his AP is "Third-Party" by default.
2) If AP_{st} = "Direct" and $AP_{st} \notin AP_{dy}^{\blacktriangledown}$, the Negotiation module is activated, and the user will get a second chance to modify his AP_{st} or the Context.
3) If AP_{st} = "T-Email", $AP_{st} \in AP_{dy}^{\blacktriangledown}$, then *Purpose_belong (org,* AP_{st}*,* AP_{dy}*)* = True, the AP of the user would be "T-Email".

b) Intended Purpose: Before migrating data to the cloud, a label named "intended purpose" is set for each item based on the data owner's privacy preferences. As described above, an intended purpose (IP) is a tuple ⟨AIP, PIP⟩.

Example 2. Suppose IP = ({Admin, D-Email}, {Third-Party}) is defined over the purpose tree that is given in Figure 3. Thus,

$AIP^{\blacktriangledown}$ = (Admin)$^{\blacktriangledown}$∪(D-Email)$^{\blacktriangledown}$ = {Admin, Profiling, Analysis, D-Email, Special-Offers, Service-Updates}

$PIP^{\blacklozenge}$ = (Third-Party)$^{\blacklozenge}$ = {Third-Party, Marketing, T-Email, T-Postal, General-Purpose}

c) Authorization: The relationship of *Purpose_compare (org, AP, IP)* $\rightarrow$ {True, False} is defined to determine the compliance between the user's AP and the object's IP. The access request could be allowed if and only if the AP satisfies the pre-set rules of the intended purpose. That is, AP $\in AIP^{\blacktriangledown}$ $\wedge$ AP $\notin PIP^{\blacklozenge}$.

Example 3. Suppose $AIP^{\blacktriangledown}$ and $PIP^{\blacklozenge}$ are discussed above. Then, the access purposes that meet the authorization conditions are {Admin, Profiling, Analysis, Special-Offers, Service-Updates}.

C. Security Policy

We can now give the security policies that apply to such an organization by adding our new entity AP to the access policy. The relationship *Permission(org, r, v, a, AP)* means that organization *org* grants role *r* permission to execute activity *a* on view *v* based on the access purpose *AP*.

D. Concrete authorization

In Pa-OrLaBAC, the procedure through which a subject can obtain permission to perform on the object is as follows:

Employ (org, s, r) $\wedge$ *Use (org, o, v)* $\wedge$ *Consider (org,* α*, a)* $\wedge$ *Speculate (org, c, r,* AP_{dy}*)* $\wedge$ *Permission (org, r, v, a, AP)* $\wedge$ *Purpose_belong (org,* AP_{st}*,* AP_{dy}*)* $\wedge$ *Purpose_compare (org, AP, IP)* $\rightarrow$ *Is_permitted (s, o,* α*).*

This means that if *org* employs subject *s* in role *r*, if *org* uses object *o* in view *v*, if *org* considers that action α falls within activity *a*, if organization *org* within the current context *c* speculates the dynamic access purpose of role r is AP_{dy}, if organization *org* grants role *r* permission to perform activity *a* on view *v* for access purpose AP, if *Purpose_belong* is true, and if *Purpose_compare* is true, then *s* has permission to perform α on *o*.

V. A Case Study

Medical informatization has become an inevitable trend of modern medical care. Electronic medical record (EMR), as the main carrier of medical information, plays an important role in modern medical treatment. The EMR itself contains a large amount of private information of the original owner, such as name, date of birth, home address, and sensitive information that is unwilling to be known to the outside world, such as marital status and disease information. The leakage and illegal use of this information may cause irreparable losses. However, patients have limited control over their medical data, which may lead to the disclosure of privacy information when EMR is consulted.

The proposed method can help solve the above problem.

An Application Scenario: Hospital *hosA* uses this method to manage its EMRs. John is treated in *hosA* whose attending internist is Tim. Figure 5 shows the purpose tree of *hosA*.

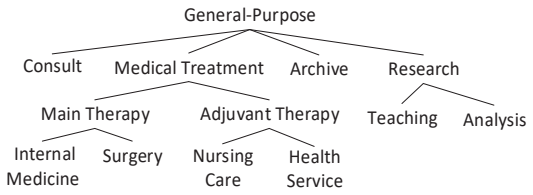

Fig. 5. The Purpose Tree of *hosA*.

Before submitting his EMR, John wanted to protect his "personal information" as much as possible, so he set the "intended purpose" label of it as ({Main Therapy, Archive}, {Research}). That is to say, for John's personal information,

$AIP^{\blacktriangledown}$ = (Main Therapy)$^{\blacktriangledown}$ ∪ (Archive)$^{\blacktriangledown}$ = {Main Therapy, Internal Medicine, Surgery, Archive}.

$PIP^{\blacklozenge}$ = (Research)$^{\blacklozenge}$ = {Research, Teaching, Analysis, General-Purpose}.

Tim is doing some researches at home. He requests access to John's EMR and declares his AP_{st} as "Medical Treatment".

Meanwhile, *hosA* determines Tim's AP_{dy} based on contextual information that whether John is receiving treatment right now or not, Tim's geographic location, etc.

Assuming that the inferred AP_{dy} of Tim is "Teaching". Obviously, $AP_{st} \notin AP_{dy}^{\blacktriangledown}$. Under this circumstance, the negotiation module is activated. Tim could have a second chance to modify his AP_{st}. If the condition is still not satisfied, access will be denied. In the meantime, John's "personal information" could not be accessed by Tim. In other words, John's "personal information" was protected as he wished.

The proposed method can also help us protect our privacy information from being leaked in other applications, e.g., banks and logistics. Depending on the actual requirement of different usage scenarios, this access control model can be adjusted dynamically by modifying the labels assigned to the data which could be validity, security level, risk value, etc.

VI. Conclusions And Future Work

For now, the access control models that are used by most Cloud Service Providers are based on RBAC. As an extension of RBAC, OrBAC overcomes the drawback that permissions are directly bound to roles. However, lacking flexibility makes OrBAC unsuitable for the cloud. In this paper, we presented Pa-OrLaBAC, which integrates OrBAC with LaBAC and introduces the concept of "purpose" as an effective means of privacy protection.

Before data are migrated to the cloud, a label named "intended purpose" is set for each data item based on the data owner's privacy preferences. Only when the access purpose of the subject is fully compliant with the intended purpose will the request be allowed. As for access purpose, two approaches and the Negotiation Module were applied to determine the most reliable one.

Compared with the traditional access control models, Pa-OrLaBAC alleviates the shortcomings of data control and ownership separation while ensuring flexibility by introducing purpose. Thus, it could be an effective method to protect resources in cloud environments. However, obviously, the Pa-OrLaBAC model still needs to be improved. In the future, we plan to formally analyse the properties of the proposed model compared to those of existing access control models, develop an XACML profile of the proposed model and enable this model to achieve dynamic access control.

VII. Acknowledgment

This research was supported by the following Grants: the National Natural Science Foundation of China (Grant No.61672494) and the Key Research and Development Program for Guangdong Province (Grant No.2019B010136001).

References

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[2] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: from single to multi-clouds," in *2012 45th Hawaii International Conference on System Sciences*. IEEE, 2012, pp. 5490–5499.

[3] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE communications magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[4] R. Sandhu, "Engineering authority and trust in cyberspace: The om-am and rbac way," in *Proceedings of the fifth ACM workshop on Role-based access control*. ACM, 2000, pp. 111–119.

[5] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic databases," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 2002, pp. 143–154.

[6] OECD, "Oecd guidelines on the protection of privacy and transborder flows of personal data," 1980.

[7] J.-W. Byun and N. Li, "Purpose based access control for privacy protection in relational database systems," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 17, no. 4, pp. 603–619, 2008.

[8] S. Fischer-Hübner, *IT-security and privacy: design and use of privacy-enhancing security mechanisms*. Springer-Verlag, 2001.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[10] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," MITRE CORP BEDFORD MA, Tech. Rep., 1973.

[11] J. Lopez and J. E. Rubio, "Access control for cyber-physical systems interconnected to the cloud," *Computer Networks*, vol. 134, pp. 46–54, 2018.

[12] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone *et al.*, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, 2013.

[13] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*. IEEE, 2003, pp. 120–131.

[14] P. Biswas, R. Sandhu, and R. Krishnan, "Label-based access control: An abac model with enumerated authorization policy," in *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*. ACM, 2016, pp. 1–12.

[15] M. B. Saidi, A. A. Elkalam, and A. Marzouk, "Torbac: A trust organization based access control model for cloud computing systems," *Int J Soft Comput Eng*, vol. 2, no. 4, pp. 122–130, 2012.

[16] N. A. Aali, A. Baina, and L. Echabbi, "Tr-orbac: A trust model for collaborative systems within critical infrastructures," in *2015 5th World Congress on Information and Communication Technologies (WICT)*. IEEE, 2015, pp. 123–128.

[17] M. A. Madani and M. Erradi, "How to secure a collaborative session in a single tenant environment," in *2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*. IEEE, 2015, pp. 1–6.

[18] R. E. Sanders, "Securing data with label-based access control," http://www.tridug.org/wp-content/uploads/2012/05/Understanding LBAC.pdf, 2012.

[19] J. Leffler, "Label-based access control with ids cheetah," http://www.iiug.org/webcasts/replay/30may07.pdf, 2007.

[20] H. Chen, B. Bhargava, and F. Zhongchuan, "Multilabels-based scalable access control for big data applications," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 65–71, 2014.

[21] P. Chinnasamy and P. Deepalakshmi, "A scalable multilabel-based access control as a service for the cloud (smbacaas)," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 8, p. e3458, 2018.

[22] A. Li, Q. Li, and V. Hu, "Access control for distributed processing systems: Use cases and general considerations," in *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2017, pp. 117–125.

[23] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C.-M. Karat, J. Karat, and A. Trombeta, "Privacy-aware role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 3, p. 24, 2010.

[24] E. Bertino, J.-W. Byun, and N. Li, "Privacy-preserving database systems," in *Foundations of Security Analysis and Design III*. Springer, 2005, pp. 178–206.