Two Layer Encryption for Delegated Access Control in Public Clouds

[1] G.Rakesh Reddy, [2]U.Usha Rani

[1]M.Tech Research Scholar, Priyadarshini Institute of Technology & Science, Chintalapudi, India
[2] HOD, Priyadarshini Institute of Technology & Science, Chintalapudi, India

***Abstract* –** *Delegation is a process of sharing access rights by users of an access control model. It facilitates the distribution of authorities in the model. It is also useful in collaborative environments. Despite the advantages, delegation may have an impact on the access control model's security. Allowing users to share access rights without the control of an administrator can be used by malicious users to exploit the model. Delegation may also result in privacy violations if it allows accessing data without the data provider's consent. Even Though the consent is taken, the privacy can still be violated if the data is used differently than the data provider agreed. Our work investigates data privacy in delegation. As a contribution, a privacy model is introduced that allows a data provider setting privacy policies that state how their data should be used by different organizations or parties who are interested in their data. Based on this setting, a delegation model is designed to consider the privacy policies in taking delegation decisions and also, to set the data usage criteria for the access right receivers. In addition to privacy policies, several delegation policies and constraint have been used to control delegation operations. Delegation is studied within a party and between two parties.*

Keywords – Delegation, privacy, access control, security, policy.

I. INTRODUCTION

In typical access control models, the set of access rights a user gets is predetermined. Predetermining a user's access rights is equivalent to anticipating possible usages of the system by that user. However, users may need new access rights due to the dynamic nature of their work. There are two ways to assign access rights. First, a system administrator acts every time a user needs an access right. Secondly, a user gets the right from another user who already possesses it. The latter approach is called delegation.

Delegation brings flexibility to access control models. Zhang *et al.* [22] identify three cases when delegation is necessary. In the first, an individual is absent from Their job and so, someone else should carry out the tasks. Secondly, delegation is allowed to decentralize the authority. Having one system administrator who assigns access rights to all the users in the system would decrease efficiency. In the final case, delegation is very useful in an environment where users collaborate with each other to complete a task.

Despite its usefulness, delegation may produce security risks for an organization. Consider the case where a system administrator does not assign some privileges to a user for security reasons. In a delegation enabled system, it may not be sufficient for security protection as the user may receive the privileges from other users. Delegation may also lead to data privacy violations. Delegation invites new users to access data which raises the question of whether the data provider's consent is taken. Even if the provider is informed, the issue of how data will be used is also critical. Any of these issues may violate data privacy if they are not resolved. The security requirement in delegation mainly comes from an enterprise's perspective while data privacy protection in delegation is required by the data provider. This work investigates data privacy protection in delegation.

To study privacy preserving delegation, we need an environment where data providers provide *privacy policies* for their data. The policies would state who can use the data and how the data should be used. The access control models in such environments control data accesses based on the privacy policies. These are known as privacy preserving access control models. Several models have been proposed [23, 26] in the literature. Most of the proposed models assume that data is accessed only by the collecting organization. However, in real life, many parties are interested in data including the collecting organization. One of our contributions is to define a privacy model that allows a data provider to set privacy policies for different organizations accessing their data.

We adapt an existing access control model Called P-RBAC [14] which is an extension of role-based access control model. The proposed privacy model is used in conjunction with the access control model to create privacy-aware access rights *i.e.*, the rights containing constraints that specify the valid use of data. Data users assigned to these rights use data according to the constraints. Based on these foundations, we propose a delegation model where access rights to a data item can be delegated only if it is allowed by the data item's privacy policies. Since the delegated rights contain privacy constraints, the users who receive the rights are bound by the privacy constraints when they use the data. We also investigate prohibiting certain delegation operations to maintain the access control model's security. The paper is organized in the following ways. Section II describes the privacy model and the access control model. Delegation model is presented in Section III. Revocation is described in Section IV. An overview of the relevant literatures is given in Section V. Conclusion and future Works are discussed in Section VI.

II. ACCESS CONTROL MODEL

In this section, we first propose a formal model to express privacy policies. How access rights or privileges are created based on the privacy policies are described in the privilege model. Finally, we describe how access control decisions are taken.

A. Privacy model

Privacy policies are the data provider's preferences for using their data. We can also say that privacy policies define the data collector's practice with the data. This work assumes that a data collector sets the privacy policies for the data items that they collect. Data providers accept the policies if they agree. In addition, the collector may allow the data providers to customize or even opt out from some policies. Examples of such practice include the privacy statements of Amazon.com [1], Toys.com [19], *etc*.

In this work we formalize the privacy Policies. There are several works that define the contents of the privacy policies [23]. Rather than proposing a new one, we choose an existing definition that says that each policy should consist of data, action, purpose, condition and obligation [24]. Here, data is the information collected about the data provider. Actions are read, update, *etc*. Purposes are the reasons for using the collected information. Conditions are Boolean expressions that are used to validate contextual information necessary to enforce a privacy policy. Obligations are the tasks that must be done as a result of accessing data items. Consider an imaginary policy of a website saying, "Every time we use your data for marketing purpose, we will inform you by emails". Here, the obligation of accessing data is to send emails.

Let the sets of data, actions, purposes, conditions and obligations be denoted by D, ACT, P, C and OB respectively. The set of privacy policies, denoted by PPolicy, is defined over the following range:

$$PPolicy \subseteq D \times ACT \times P \times C \times OB$$

Data is accessed by different *visibilities* [3, 28] which are parties involved in the business operations. For example, Toys.com states in its privacy statement that a customer's information are accessed by itself (*i.e.*, by its employees), service providers, business partners and advertisers [19], thereby defining four visibilities for the collected data. Among the visibilities, the organization responsible for collecting data is called *enterprise visibility* which is Toy.com in this example.

Privacy policies may vary for different visibilities. A data provider's preference for one visibility, say *service provider*, can be different than their preference for another visibility, say *third party*. Therefore, privacy policies should be defined for each visibility. Let the set of visibilities bevies. The set of privacy policies for a visibility i can be denoted by Policy where i € VIS. The union of privacy policies for all the visibilities amounts to the entire policy set of a data collection.

$$PPolicy = \bigcup_{\forall i \in VIS} PPolicy_i$$

The enterprise visibility collects the data and later, shares the data with other visibilities according to the privacy policies. They cannot share data with a visibility that is not listed in the privacy agreement. We assume that a trusted third party is employed to oversee the data sharing to ensure that they follow the privacy policies. We do not include the third party auditing to the scope of our work and assume that the visibilities will follow the privacy policies.

Data providers may be allowed to specify their preference about particular policies. For example, a privacy policy may state that the consent for using personal information for marketing purpose is optional and one can opt out from such use of their data. To support this in the privacy model, we introduce the notion of *privacy policy metadata* which are the data provider's preferences and other related information necessary to enforce privacy policies. Note that one of the elements of a privacy policy is condition that checks the values of these metadata. The metadata are stored in a set of context variables denoted by SV. Let the set of data providers be dProvider, then the set of privacy metadata for visibility I € VISis defined as follows:

$$StatDS_i \subseteq dProvider \times PPolicy_i \times \wp(SV)$$

Here, $\wp$ denotes power set. Example of the privacy model with a sample policy:

The example uses a real life privacy statement taken from the website Toys.com that sells toys online. "From time to time, you may receive periodic mailings, telephone calls or e-mails from "R" Us Family members with information on products or services, discounts, special promotions, upcoming events or other offers from an "R" Us Family member or its marketing partners. You may opt out of receiving e-mail communications by clicking the link at the bottom of the e-mail received.." [19].

The policy allows the website's employees (mentioned as "R" Us Family members) to access the customers' email addresses (along with other media) for sending promotional offers. The policy also says that if a customer opts out, their data will not be used for this purpose. Note that this policy does not impose any obligation. As the data is being used by the website's employees, let the visibility be termed as *website* (denoted by the acronym ws). Table 1.1 presents all types of information collected about the customers with a sample record. Table 1.2 is an instance of the relation $PPolicy_{ws}$ that stores the above privacy statement by breaking it into several formal policies. The conditions of the policies test the value of a variable called OptOut which represents the opt in/out preferences of the customers. Table 1.3 is an instance of the relation $StatDS_{ws}$ and stores the values of the variable OptOut When the variable has the value 'Y' for a particular data provider and a policy, it indicates that the provider has opted out from the policy.

We consider a hierarchical relation among the purposes. Purpose hierarchy is denoted by (p, ≤) where ≤ is a partial relation defined over the set of purposes P. The relation is reflexive, transitive and anti-symmetric. Fig. 1 shows a sample purpose hierarchy. In our interpretation of the hierarchy, any node includes all of its connected predecessors.

For example, Sales includes Business operation which also represents (Sales ≤ Business operation). The use of purpose hierarchy makes the policies more expressive. Consider the policy #4 in Table 1.2 which is (Email address, read, Promotion, OptOut ≠ Y, φ). It allows accessing email address for purpose Promotion as well as for purposes Contest and eMarketing as both of them include Promotion in Fig. 1. Condition and

Table 1.1: Customer information

ProviderID	Name	DofB	Gender	Home address	Telephone number	Email address
235	Smith Jonas	02/23/1970	Male	60 Essex st., Toronto	89450	Smith78@search.com

Table 1.2: Privacy policies, $PPolicy_{ws}$

Policy ID	Data Name	Action	Purpose	Condition	Obligation
1	Name	Read	Promotion	$OptOut \neq Y$	$\emptyset$
2	Home address	Read	Promotion	$OptOut \neq Y$	$\emptyset$
3	Telephone number	Read	Promotion	$OptOut \neq Y$	$\emptyset$
4	Email address	Read	Promotion	$OptOut \neq Y$	$\emptyset$

Table 1.3: Privacy metadata, $StatDS_{ws}$

Provider ID	Policy ID	$OptOut$
235	1	N
235	2	Y
235	3	Y
235	4	N

Obligations for these purposes are the same as for purpose Promotion.

A policy can be overridden by a more restrictive policy. Consider the following policy.

Privacy policy#5:

$$(Email\ address, read, eMarketing, OptOut \neq Y \wedge Age > 18, \emptyset)$$

B. Privilege model

Typically, a privilege consists of data and action. In a privacy preserving access control model, privileges also contain privacy restrictions specifying the valid use of data items. These privileges are created from the privacy policies. For instance, a privilege consists of data, action, purpose, condition and obligation in P-RBAC [14]. We adapt this structure to propose a more simple privilege structure. Like P-RBAC, we also group the privileges into roles which are in turn assigned to users.

When an enterprise creates privileges for the data users (*e.g.*, its employees), it may create privileges more restrictive than the privacy policies. If a privacy policy allows using a data item for a set of purposes, the enterprise .May choose a subset of the allowed purposes to create a privilege. We define purpose range to specify a subset of purposes in a privilege. A purpose range is given by pr = <p_u, p_b> where p_b, p_u € p and $p_b \leq p_u$. If $p_u = p_b$, the range has only one purpose. The set of privileges, DP, are defined over the following range:

$$DP \subseteq D \times \bar{ACT} \times \bar{PR}$$

Here, D, ACT, and PR are the sets of data, actions and purpose ranges, respectively. As an example, dp = (Email address, read, (Promotion, eMarketing)). Users assigned to this privilege can use the data for these Purposes {Promotion, eMarketing}. What condition and obligation users should fulfill will depend on the purpose they use to access data. The system will find the appropriate condition and obligation based on a user's access purpose.

Dropping condition and obligation allows us to create a single privilege based on multiple privacy policies. For example, privilege dp is based on policy #4 and policy #5 described in the previous section. A user assigned to the privilege should satisfy (OptOut ≠ Y ∩ Age > 18) when they access data for eMarketing while they need to satisfy only (OptOut ≠ Y) when they use data for Promotion.

C. Formal specification of the access control model

The access control model uses the following entities:
- VIS is the set of visibilities
- D*, ACT, P, C* and OB are the sets of data, actions, purposes, conditions and obligations, respectively.
- U, R and DP are the sets of users, roles and privileges.

The dot operator indicates a specific component of a privilege; *e.g.*, dp. d denotes the data contained in the preveliege dp. Following components are defined for a visibility I € VIS:
- U_i *,* R_i and DP_i are the sets of users, roles and privileges for visibility i
- Role hierarchy is (R_i, $\leq_{Ri}$) where $\leq_{Ri}$ is a partial relation defined over the set of roles R_i. The relation is reflexive, transitive and anti-symmetric.

III. DELEGATION

Data collector may share data with unknown parties if they do not follow the privacy policy. In the proposed model, delegation follows privacy policy which allows onlylegitimate parties accessing the data. It also sets the data usage guidelines for them. We refer the data sharing between two parties as *inter-visibility delegation*. The party or visibility which shares data is called *source visibility* while the visibility that receives data is called *destination visibility*. In addition, we study *intra-visibility delegation* where two users within a party share the access rights with each other. Users who delegate the rights are called *delegators* while users who receive the rights are called *delegatees*.

Delegation policies are the rules that state what delegation operations are valid. These rules are used to control delegation among the users. In this work, we define delegation policies for intra- and inter-visibility delegation. Besides the delegation policies, a security policy is used to maintain the separation of duty among the users. The security policy is often referred as the security constraint to differentiate it from the delegation policies. All the delegation operations in our model are controlled by these policies and constraint.

Fig. 2 presents an outline of the proposed delegation model. The *delegation agent* processes a delegation request by retrieving the delegation policies and constraint from module *DRP*. To test these rules, the agent uses the authorization records (module *AR*), privacy policies (module *PP*) and delegation histories (module *DRH*). Here, the authorization records include the authorization relations described in Section II (C) and Section III (A). These include both regular and delegated user-access right assignments. Delegation histories are the records of the delegation events that have taken place in the system so far.

If a delegation request satisfies all the policies and constraint, a new entry is added to the authorization records assigning the requested right to the delegatee. The new record is later used for taking access control decisions. The delegation event is also logged in the delegation histories. This was an overview of the proposed delegation model. The model will be described in detail over the next few sections.

A. Specification of the delegation model

This section describes the notations and relations used in the delegation model.

Delegation units: A role-based access control model is used in this work. In role-based access control models, access rights are typically delegated using two units – role and privilege. In role delegation, all the privileges of a role are delegated while a single privilege is given away in privilege delegation. Role and privilege delegations are often termed as full and partial delegations, respectively. The privacy policies are same for all the users with the same visibility; so, role delegation within a visibility is not affected by the privacy policies. Study of role delegation in this work would then be no different than the existing literatures [6, 22]. On the other hand, role delegation between two visibilities is a challenging problem. The delegated role should be mapped to one of the existing roles of the destination visibility. Since privacy policies for the source and destination visibilities can be different, it may not be possible to find all the privileges of the delegated role in one of the roles of the destination visibility. To keep the model simple, we study only partial delegation where a user can delegate individual privileges to other users.

B. Delegation policies and constraint

We apply several policies to control delegation operations among the users. Of them, delegation policies are used to specify what delegation operations are valid. A security policy is applied to maintain the separation of duty among the users. These policies are expressed in a declarative logic language which is described as follows:

Policy language: A slight variant of First Order Logic is used as the policy language. The language consists of a set of variables and constants. Let v represents a variable and cn represents a constant. A term tm is either a variable or a constant.

The predicates used in the delegation policies and constraint can be categorized into four classes: utility, specification, decision and constraint predicates. Though some of these classes have the same names as the ones in [4], the semantics are not the same. The semantics of the predicate classes are defined as follows. Utility predicates provide functionalities like set operations, counting and comparisons. Specification predicates are used to retrieve information from the system configuration. For example, the set of privileges assigned to a user is returned by one of the predicates of this class. Decision predicates define the enforcement of the rules. Being used as the head of a rule, these predicates denote the consequence when they become true. Constraint predicate is used to enter the security policies into the system.

Policy for inter-visibility delegation:

In an inter-visibility delegation, access rights to data items are delegated from one visibility to another. The scope of our work is limited to the case where access rights can be delegated to a non-enterprise visibility only from the enterprise visibility. Recall the data collection by the website Toys.com in Section II (A) where the enterprise visibility is the website itself. All other visibilities receive access rights from it.

Inter-visibility delegation is divided into two sub-classes: *collaboration* and *exchange*. In collaboration, the source visibility shares data with the destination visibility so they can collaborate to achieve the same goal. For example, an organization allows its employees and an external service provider to access its customer’s information for doing market research. However, in exchange the source and the destination visibilities have separate business operations; but they share their customer’s information for their mutual benefits. For example, an organization shares its customer’s information with their parent and subsidiary companies.

Delegation policy for exchange:

In exchange, the source and destination visibilities have separate business process. Data items are not necessarily visible to both of them with the same privacy policies. A delegation policy (like the policy for collaboration) asking that a delegator must have access to the requested data item with the requested action and purposes, may not be useful in exchange. We can also use the privacy policy as the only delegation policy which means that access to a data item with an action and a set of purposes will be given to a delegate if there is a privacy policy for the destination visibility supporting such data usage. In such case, no users from the source visibility will be involved in delegation.

To ensure the involvement of the users from the source visibility, we define a delegation policy that is more relaxed than the policy for collaboration. Instead of requiring that a delegator have access to the requested data with the requested action and purposes, the new policy requires that the delegator have access to the requested data only; this may be with different action and purposes. Like the collaboration policy, it also checks if the delegation request is supported by a privacy policy of the destination visibility.

IV. REVOCATION

Revocation is the process of removing delegated access rights from a delegate. In the proposed delegation model, one possible revocation policy would be allowing a delegator to revoke the privileges they delegated in the past. We can call this type of revocation as *forced revocation* as the delegators can revoke the privilege at any time they want. Since we consider temporal delegation where a privilege is delegated for a specific period of time, another revocation policy would be allowing the system to revoke the delegated privileges from the users when the valid time ends. This type of revocation is termed as *auto revocation*. These revocation policies can be formally expressed by the policy language. Due to space limitation, we do not present the details of these policies here.

Successful inference of a revocation policy will initiate the cleanup task that would remove the delegated privilege from the delegate. The revocation event should also be logged using a relation similar to the delegation history that would contain who initiated the revocation - either a user or the system itself, what privilege was revoked, who the delegate was, and when the revocation took place.

V. RELEVANT WORKS

Many delegation models have been proposed in the literature covering different aspects of delegation including role and privilege delegation [6, 20]. The assignment of delegated authorizations to users and their enforcement through access control decisions have also been investigated [6, 11, 22]. There are proposals [20-21] that study multi-step delegation where a delegated access right is further delegated. Some delegation models (*e.g.,* [10]) are unconstrained while others [2, 21] apply delegation policies that allow or deny a delegation operation. Mechanisms to revoke delegated access rights have been studied by several research works [2, 21].

However, there are a few works in the literature that investigate data privacy in delegation. One of these models [9] is based on the identity management systems where Data providers get services from different service providers by giving access to their data. To provide the service, a service provider may rely on other service providers Transitively and so, the data is shared with them too. The model does not reveal the data provider's identity (ID) to any of the service providers. Instead, it uses a trusted third party to maintain a pseudo ID of the data provider for each service provider. When the data provider wants to provide access to their data to a service provider, the third party issues a credential [7] containing the data provider's pseudo ID for that service provider. Credentials are authorization certificates used in the identity management systems. If a service provider wants to delegate the credential to another service provider, a new credential is created containing the data provider's pseudo ID for the new service provider. The data providers remain anonymous in the entire process, so The authors claim that the privacy is protected. However, hiding only the identity information may not be sufficient because the exposure of other information like address and date of birth can lead to the identification of a data provider [18]. In addition, the data provider's identity is required to provide some services, *e.g.*, delivering products. So, some details may need to be exposed to receive the service. Instead of removing the identity information from the data, our model requires that the data provider be aware of the privacy policies that state who will access their data and how their data will be used. These policies later control the requests to use and share the data.

Bussard *et al.* [5] propose an XML-based policy language to encode the privacy preferences of a data provider. The language is designed with multiple data transfers in mind *i.e.*, a provider can specify if one visibility can allow other visibilities to access the data and if so, how these new recipients should use the data. The authors also propose to use the privacy policies as the access control policies for data. In our model, we separate the access control policies from the privacy policies. This separation gives an organization better control over the data usage because it can create more restrictive privileges than the privacy policies. We also study hierarchical purposes and roles in our work while Bussard *et al.* consider flat purposes and roles.

VI. CONCLUSION

An effective way for privacy protection is to set the privacy policies that are the agreements between the data provider and collector about that can use the personal information and how the information should be used. Accesses to the information are then controlled by these policies. In this work, we propose a privacy model to formalize privacy policies for multiple parties who access the data .In a privacy preserving access control model, access rights contain privacy restrictions. Data users must satisfy the restrictions in order to get access. Another contribution of our work is to propose a delegation model that facilitates access rights sharing in this type of access control model. The proposed delegation model takes privacy policies into consideration for taking delegation decisions. The model also ensures that when an access right is delegated, it is constrained by the appropriate privacy policy for the receivers. We study delegation within a visibility and between two visibilities.

REFERENCES

[1] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in the storage as a service model," in *EEE International Conference on Information Reuse and Integration (IRI)*, 2012.

[2] E. Bertino and E. Ferrari, "Secure and selective dissemination of XML documents," *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 3, pp. 290–331, 2002.

[3] G. Miklau and D. Suciu, "Controlling access to published data using cryptography," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 898–909.

[4] N. Shang, M. Nabeel, F. Paci, and E. Bertino, "A privacy-preserving approach to policy-based content dissemination," in *ICDE '10: Proceedings of the 2010 IEEE 26th International Conference on Data Engineering*, 2010.

[5] M. Nabeel, E. Bertino, M. Kantarcioglu, and B. M. Thuraisingham, "Towards privacy preserving access control in the cloud," in *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, ser. CollaborateCom '11, 2011, pp. 172–180.

[6] M.Nabeel,N.Shang,andE.Bertino, "Privacypreservingpolicy based content sharing

[7] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in *Proceedings of the 33rdInternationalConferenceonVeryLarge DataBases*, ser.VLDB '07. VLDB Endowment, 2007, pp. 123–134.

[8] M. Nabeel and E. Bertino, "Towards attribute based group key management," in *Proceedings of the 18th ACM conference on Computer and communications security*, Chicago, Illinois, USA, 2011.

[9] A. Fiat and M. Naor, "Broadcast encryption," in *Proceedings of the 13th Annual International Cryptology Conference on Advances inCryptology*,ser.CRYPTO '93. London, UK:Springer-Verlag, 1994, pp. 480–491.

[10] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '01. London, UK: Springer-Verlag, 2001, pp. 41–62.

[11] J. Li and N. Li, "OACerts: Oblivious attribute certificates," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 340–352, 2006.

[12] T.Pedersen,"Non-interactiveandinformation-theoreticsecure verifiable secret sharing," in *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1992, pp. 129–140.

[13] M.NabeelandE.Bertino, "Attributebasedgroupkey management," *IEEE Transactions on Dependable and Secure Computing*, 2012.

[14] A. Shamir, "How to share a secret," *The Communication of ACM*, vol. 22, pp. 612–613, November 1979.

[15] V. Shoup, "NTL library for doing number theory," http://www.shoup.net/ntl/.

inpublic clouds," *IEEE Transactions on Knowledge and Data Engineering*, 2012. 14

[16] "OpenSSL the open source toolkit for SSL/TLS," http://www.openssl.org/.

[17] "boolstuff a boolean expression tree toolkit," http://sarrazip.com/dev/boolstuff.html.

[18] A. Schaad, J. Moffett, and J. Jacob,"The role-based access control system of a european bank: a case study and discussion," in *Proceedings of the sixth ACM symposium on Access control models and technologies*, ser. SACMAT '01. New York, NY, USA: ACM, 2001, pp. 3–9.

[19] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz, "Verification and change-impact analysis of access-control policies," in *Proceedings of the 27th international conference on Software engineering*, ser. ICSE '05. New York, NY, USA: ACM, 2005, pp. 196–205.

[20] S.Coull,M.Green,andS.Hohenberger,"Controlling accessto an oblivious database using stateful anonymous credentials," in *Irvine: Proceedings of the 12th International Conference onPractice and Theory in Public Key Cryptography*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 501–520.

[21] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious transfer with access control,"in *CCS '09: Proceedings of the 16th ACM conference onComputer and communications security*. New York, NY, USA: ACM, 2009, pp. 131–140.