Policy Management as a Service: An Approach to Manage Policy Heterogeneity in Cloud Computing Environment

Hassan Takabi  
School of Information Sciences  
University of Pittsburgh  
Pittsburgh, PA, USA  
hatakabi@sis.pitt.edu

James B. D. Joshi  
School of Information Sciences  
University of Pittsburgh  
Pittsburgh, PA, USA  
jjoshi@sis.pitt.edu

Abstract—Security issues are delaying fast adoption of cloud computing and security mechanisms to ensure its secure adoption has become a crucial immediate need. On the other hand, cloud computing can help enable security controls to be delivered in new ways by service providers. To this end, we need frameworks for efficient delivery of cloud-based security services and for provisioning desirable solutions to customers based on their requirements. In this paper, we focus on policy management systems in cloud environments. Currently, users must use diverse access control solutions available for each cloud service provider to secure data. Access control policies may be composed in incompatible ways because of diverse policy languages that are maintained separately at every cloud provider. Heterogeneity and distribution of these policies pose problems in managing access policy rules for a cloud environment. In this paper, we introduce Policy Management as a Service (PMaaS), a cloud based policy management framework that is designed to give users a unified control point for managing access policies to control access to his resources no matter where they are stored. We present the framework and describe its components and protocols needed for various components to communicate.

Keywords—access control; heterogeneity; cloud computing; policy management; policy management as a service;

I. INTRODUCTION

Cloud computing aims to incorporate the evolutionary development of many existing computing approaches and technologies such as distributed services, applications, information and infrastructure consisting of pools of computers, networks, information and storage resources [1], [2]. It is still an evolving paradigm but has shown tremendous potential to enhance collaboration, agility, scale, and availability although its definitions, issues, underlying technologies, risks, and values need to be refined [5].

The US National Institute of Standards and Technology (NIST) defines cloud as follows: “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. This cloud model promotes availability and is composed of five essential characteristics, three delivery models, and four deployment models.” [1]. The five key characteristics of cloud computing include on-demand self-service, ubiquitous network access, location-independent resource pooling, rapid elasticity, and measured service, all of which are geared toward using clouds seamlessly and transparently [1]. The three key cloud delivery models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [6]. The cloud deployment models include public cloud, private cloud, community cloud, and hybrid cloud composed of multiple clouds [2].

Cloud computing has become a very attractive paradigm, with the potential to significantly reduce costs through optimization and increased operating and economic efficiencies [2], [3]. The architectural features of the cloud allow users to achieve better operating costs and be very agile by facilitating fast acquisition of services and infrastructural resources as and when needed. However, these unique features also give rise to various security concerns [6], [4]. Several surveys of potential cloud adopters also indicate that security and privacy is the primary concern hindering its adoption but cloud computing has become very popular because of its potential benefits [4], [3]. Hence, understanding the security and privacy risks in cloud computing and developing effective solutions are critical to the success of this new computing paradigm [5].

In this paper, we focus on access control in cloud computing environment. The cloud computing environment does not allow use of a single authorization mechanism, single policy language or single management tool for various cloud service providers. Each cloud service provider employs its own access control solution and authorization mechanism in a often tightly bound to a cloud service provider and has limited flexibility in terms of addressing a particular user’s security requirements [6]. This approach is not user/customer centric and hence, can be a significant bottleneck to its widespread adoption. The definition of access control policies clearly is the business of the organization/users deploying the cloud service. However, not only the cloud service providers themselves are dictating how these policies should be defined but
also each of them does it in its own way [5].

Currently, users must use diverse access control solutions available for each cloud service provider to secure his data and control its dissemination. Access control policies may be composed in incompatible policy languages and maintained separately at every cloud service provider. When such mechanisms are used on a daily basis, they add considerable overhead, especially since these mechanisms often lack flexibility with respect to functionality as well as user interfaces. This may frustrate users and make them feel that they have given up all control of where their data ends up and how it is used. The challenge here is to design an integrated access control system that can be used across services from different providers. An ideal access control scheme must be able to work with all types of content regardless of where they are stored. Users should be able to manage policies to govern access to their information and resources from a central location.

Security solutions delivered as cloud-based services will have a dramatic impact on the industry. Cloud computing will enable security controls and functions to be delivered in new ways and by new types of service providers. It will also enable customers to use security technologies and techniques that are not otherwise cost-effective. Enterprises that use cloud-based security services to reduce the cost of security controls and address the new security challenges that cloud based computing will bring are most likely to prosper.

In this paper, we propose a cloud based policy management framework that puts users in full control of their resources which may be scattered across multiple cloud service providers. It is designed to give cloud users a unified control point for specifying authorization policies, no matter where all the data is stored and distributed on the Cloud. It facilitates the ability of users to manage access policies using a centralized policy manager which provides usable interfaces for specifying access policies and exporting them to the cloud service providers on behalf of the user.

The paper is organized as follows: In Section 2, we present two use case scenarios of using various cloud services and managing access control. In Section 3, we present our cloud based policy management framework that we propose. In Section 4, we discuss some of the advantages of the framework. Section 5 discusses related work and Section 6 presents conclusions and future work.

II. USE CASE SCENARIOS: HETEROGENEOUS ACCESS CONTROL MECHANISMS

In this section, we present two use case scenarios where our proposed framework of a cloud based policy management service is well-suited and then discuss how individual users and organizations that use cloud services can benefit from the proposed PMaaS framework.

A. Individual user

Alice is a PhD student and uses different applications and services for various purposes. She is working on a project and wants to have access to the project files from anywhere. Sometimes she works at her office using her PC and sometimes she works at home or a coffee shop using her laptop. Since she is frustrated with synchronizing project files using flash memories or other means, she decides to use Dropbox, a file hosting service which uses cloud computing to enable users to store and share files and folders with others across the Internet using file synchronization (http://www.dropbox.com). Since she collaborates with some other researchers on the project, sometimes she needs to share some of the files she stores at Dropbox with her colleagues.

Alice also has other documents and spreadsheets that include important content like financial data and she uses Google Docs, a service to create and share various types of files online and access them from anywhere (http://docs.google.com). Moreover, she stores some of her older files at Amazon S3, an online storage service that provides unlimited storage through a simple web services interface (http://aws.amazon.com/s3). She occasionally shares some of these files with family members. Furthermore, she uses Mint, an online personal finance service, to manage her financial and budget planning (http://www.mint.com). Sometimes she may want to share some of these files with a family member or a close friend.

Some of the above mentioned applications are used for convenience and others for sharing and collaboration purposes. With the increasing amount of data that Alice puts online, managing access control for her resources becomes difficult and time consuming. Each of these applications has its own access policy mechanism forcing users to specify separate access control policies for each application. However, with resources being scattered across multiple applications, it is difficult to manage access to them. Alice needs to understand the applications’ policy mechanisms and specify access policies in their specific policy languages which is a challenging task for her and most users like her. These mechanisms often are either very simple and inflexible or complicated and not very usable.

Moreover, introducing new access policy rules or modifying existing ones is problematic due to heterogeneity of these access policy mechanisms. Suppose Alice wants to modify an access policy rule to a resource or a set of resources, or a new colleague is added to her current project and she needs to share some resources with him. In order to do this, she needs to scan all her applications and services to modify access policy rules.

In order to better manage security, Alice needs a centralized policy management to have a better view on access policies applied to her resources. Our proposed cloud based
policy management service can help Alice centralize the policy management task and effectively identify errors or inconsistencies in the policies. Using the proposed service, the access control policies can be applied to a distributed set of resources hosted on various cloud service providers. It gathers information from all of the services Alice uses and provides her with an interface to centrally manage access to her resources regardless of where they are stored at and what cloud service provider they belong to.

B. Small and Medium Businesses

Suppose an organization wants to adopt cloud computing services. It uses Amazon S3 and FlexiScale, which are examples of IaaS for storage and maintaining virtual servers respectively. The Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web (http://aws.amazon.com/s3) while the FlexiScale provides virtual dedicated servers to its customers (http://www.flexiant.com/products/flexiscale).

As instances of PaaS, Google App Engine and LoadStorm are used for running web applications and testing their performance respectively. The Google App Engine provides web applications on Google’s infrastructure. When using Google App Engine, there are no servers to maintain; the organization just uploads its application to serve users (http://code.google.com/appengine) whereas the LoadStorm enables web developers to improve the performance of their web applications. Customers can create their own test plans, and generate concurrent users of http traffic in realistic scenarios (http://loadstorm.com).

It also uses Zoho, Zuora, Workday, Salesforce, and DocLanding that are instances of SaaS for different purposes such as email, billing, content management, human resource management, etc. Zoho offers a set of web applications geared towards increasing productivity and offering convenient collaboration (http://www.zoho.com) and Zuora offers online recurring billing and payment solutions for SaaS and subscription businesses (http://www.zuora.com). Workday provides human resource management, financial management, and payroll, and delivers the solutions on an SaaS model (http://www.workday.com). Salesforce provide customer relationship management (CRM) service to its customers (http://www.salesforce.com). DocLanding offers on-demand web-based document management service (http://www.doclanding.com).

Similar to individual user scenario, these applications have their own access policy mechanisms. The organization’s security personnel need to understand the applications’ policy mechanisms and are forced to specify separate access control policies for each application using different policy languages. Due to heterogeneity of these access policy mechanisms, introducing new access policy rules or modifying existing ones is challenging and could cause a lot of problems and inconsistencies.

A centralized policy management could help the security personnel to better manage security and provide them with a better view on access policies applied to the organization resources on different services. Our proposed service could be used to apply the access control policies to a distributed set of resources hosted on various cloud service providers. It also makes it easier for the organization to find errors or inconsistencies in the specified access policies.

III. THE CLOUD BASED POLICY MANAGEMENT FRAMEWORK

In this section, we present the proposed policy management as a service (PMaaS), a cloud-based framework that efficiently delivers policy management services. It is built on the concept of centrally expressing user’s security requirements that are applied to a user’s resources scattered across the Cloud. Such security requirements are expressed in the form of access control policies to protect users’ distributed resources. We define PMaaS as the capabilities provided to the customers to manage access policies for services and products running on a cloud infrastructure which are accessible through usable interfaces. The customers do not manage or control the underlying cloud infrastructure, network, servers, operating systems, storage, or even individual application capabilities. Figure 1 shows a high level view of the framework which includes four main components: cloud user, policy management service provider, cloud service provider, and requester. In the following, we provide a brief overview of each of these components.

- **Cloud User**: A cloud user uses different cloud service providers for various purposes. The cloud user is in charge of managing access policies on the policy management service provider which in turn will be used by cloud service providers to control access to the protected resources when a requester attempts to access
them. The cloud user is also responsible for registering the cloud service providers at the policy management service provider so they can communicate the specified access policies.

- **Policy Management Service Provider (PMSP):** A policy management service provider enables the cloud users to define, edit and manage their access policies. The cloud users can specify their policies in controlled natural language which in turn are translated by the policy management service provider into a machine readable policy language. It also conducts a conflict resolution on the policies to find and resolve possible conflicts and finally exports the policies into target cloud service providers. Therefore, a policy management service provider acts as a policy administration point (PAP) and a policy information point (PIP). Figure 2 illustrates the framework and its components in more detail which is described in the following sections.

- **Cloud Service Provider (CSP):** A cloud service provider offers one or more cloud services that are used by cloud users. A cloud service provider controls access to the protected resources based on the policies specified by the cloud users. It evaluates access requests made by a requester against applicable policies and is in charge of making access decisions and enforcing those decisions when a requester attempts to access the protected resources. Therefore, a cloud service provider acts as a policy decision point (PDP) and policy enforcement point (PEP).

- **Requester:** A requester is an application controlled by a person or a company that interacts with a cloud service provider in order to get access to a protected resource belonging to the specific cloud user. It can be a cloud service provider that accesses resources stored in another cloud service provider.

**A. The Cloud Service Provider (CSP)**

As shown in Figure 2, each cloud service provider keeps a repository of all the resources that cloud users store and has its own access control system that makes decisions and enforces them based on input from the policy management service provider. PMaaS does not impose any restrictions on what access control model the cloud service providers use and how they make decisions to whether grant an access request or deny it, and enforce those decisions. It means that each service provider has its own policy engine and may use a simple access control matrix or a complex flexible policy engine. Each cloud service provider also has a local policy base to store policies and an authorization API that is used by the PMSP to export access policies into the cloud service providers.

**B. The Policy Management Service Provider (PMSP)**

The policy management service provider (PMSP) is the most important part of the framework and as shown in Figure 2, has two main components, the policy editor and the policy server as explained next.

1) **The Policy Editor:** The policy editor acts as policy administration point (PAP) and provides interfaces for cloud users to manage access policies in a single centralized location. It facilitates the policy management process for cloud users by allowing them to specify their policies in controlled natural language. It also handles the cloud service provider registration process which will be explained in Section III.C. Moreover, it includes a policy recommendation unit that uses information related to the cloud user and its resources to recommend some policies; needless to say that the cloud user can accept these recommended policies, modify them or ignore them.

2) **The Policy Server:** The policy server acts as the policy information point (PIP) and is responsible for interacting with the policy editor and the cloud service providers as well as translating the policies specified by the cloud user into a machine readable policy language. It keeps a repository of registered cloud service providers associated with each cloud user. It is also responsible for the resource discovery process which we will explain in Section III.C. After the cloud user registers its cloud service providers at the PMSP, the PMSP communicates with each cloud service provider to find resources and stores them in a global resource repository which contains all resources and their association with cloud users and cloud service providers. These resources are presented in policy editor interface to the cloud user to help him/her in specifying policies.

Moreover, it receives the policies specified by cloud user in policy editor, parses them, transfers them into machine readable policy language and stores them at a policy base. The output policy language could be XACML [16] or an OWL-based policy language. Since there is no agreed-upon policy language that all cloud service providers use, the framework should have capability to provide the output policies in multiple languages to support as many cloud service providers as possible.

Next, the policy server detects and resolves possible conflicts among access policies. And the final step is to export policies into the cloud service providers; the policy server first separates the policies related to each cloud service provider based on the resource-provider association and export them into the associated cloud service provider using its authorization API as explained in Section III.C.

**C. Policy Management Process**

At a high level, interactions among components of the proposed framework include the following steps: registering cloud service providers, resource discovery, specifying poli-
cies, and exporting policies. In the following sections, we discuss each of these steps in more detail.

**Step 1: Registration of CSPs at PMSP**

In this step a cloud user registers all cloud service providers he/she uses at the PMSP. As shown in Figure 3, this can be achieved by providing the location of the cloud service provider to PMSP.

When the location of the cloud service provider is provisioned to the PMSP, it uses the host-meta discovery mechanism, such as one proposed in [19], to obtain a host-meta document from the cloud service provider. Such a document defines the location of a user authorization URL among other items. The PMSP then uses the user authorization URL to initiate the process of acquiring authorization to communicate with particular cloud service provider. When a PMSP receives the host-meta document from the cloud service provider, it obtains the cloud user’s authorization information to communicate with this cloud service provider. This is achieved by receiving a verification code authorized by resource owner from the cloud service provider. At the end of this step, a PMSP is able to communicate with the cloud service provider to discover resources and also export access policies specified by the cloud user.

**Step 2: Resource Discovery by PMSP**

After a cloud user registers cloud service providers at the PMSP, we have a repository of all cloud service providers at the PMSP. Next, the PMSP discovers the resources stored in each CSP.

We recommend using POWDER-S or Semantic POWDER [18] for resource description. The Protocol for Web Description Resources (POWDER) “facilitates the publication of descriptions of multiple resources such as all those available from a Web site” [17]. These descriptions are always attributed to a named individual, organization or entity that may or may not be the creator of the described resources. Its main unit of information is the Description Resource (DR), one or more of which are contained in a POWDER document. Processing such a document yields RDF triples describing the resources that are within the scope of the DRs. POWDER documents are written in XML and have relatively loose semantics, however, POWDER-S is developed to support Semantic POWDER of Description Resources.

For the cloud service providers that do not support/use POWDER-S, the host-meta discovery mechanism [19] can be used for resource discovery. It contains information about individual resources controlled by the cloud service provider.

**Step 3: Specification of Access Policies at PMSP**
There has been some efforts to allow users to specify policies in controlled natural language [15]. IBM’s SPARCLE tool aims to enable users to enter policy rules in natural language. It automatically parses the policies to identify policy elements. It includes a set of grammars which execute on a shallow parser designed to identify the rule elements in policy rules. Then, it generates a machine readable (XML) version of the policies that can be used by any enforcement engine that can handle the standardized XML format. The policy creation portion of SPARCLE provides visualizations to help the users ensure that the specified policies are what they intended them to be.

Our framework does not impose any constraints on how cloud users specify access policies and our goal here is to allow cloud users to specify their policies in controlled natural language without any restriction. The same system can be adopted to be used in our framework for policy specification. Controlled natural languages (CNLs) are subsets of natural languages, obtained by restricting the grammar and vocabulary in order to reduce or eliminate ambiguity and complexity [14].

**Step 4: Translation of access policies by PMSP and exporting them into CSPs**

After the access policies are specified in natural language, the PMSP parses the policy, identifies policy elements, and transforms the policy into machine readable language. Next, a conflict detection and resolution is done on the policies to remove potential conflicts. Then, using the association between resources and cloud service providers, the policies are separated based on target cloud service provider. Finally, the access policies are exported into their related cloud service provider using an authorization API.

In order to export policies into the cloud service providers, we use the W3C Rule Interchange Format (RIF) which is a format to exchange rules between rule engines that operates over both XML and RDF data [20]. It is a standard for exchanging rules among rule systems, in particular among Web rule engines. Even though the RIF provides more than just a format, the concept of format is essential to the way it is intended to be used. The ultimate medium of exchange between different rule systems is XML and central idea behind the rule exchange through RIF is that different systems will provide syntactic mappings from their native languages to RIF dialects and back [20]. The systems can talk through a suitable dialect, which they both support. In order to be able to communicate rule sets from one system to another, the mappings should be semantics-preserving. Due to its extension mechanisms, RIF is an ideal language to investigate machine-readable first-order logic rules.

**D. The Framework Deployment**

In general, for the PMSP to be able to export policies into the cloud service providers, it should be able to exchange data with the CSPs. The ideal situation for PMSP is that cloud service provider supports well known policy
languages such as XACML and provides required APIs to exchange data with the PMSP. However, some cloud service providers may not be willing to do any changes and do not support well known policy languages. In this case, it is the responsibility of the PMSP to generate policies in a format that the cloud service provider understands. It is clear that the more machine readable formats the PMSP provides and the more CSPs offer APIs, the more PMaaS would be deployed.

The requester sends access requests to the CSP, and the CSP handles them locally. Our proposed framework does not impose any limitations on the requester and the CSP’s decision making functionality. The requester can send a request to access protected resources similar to every other system. The CSP, checks the request against its policy base; makes decision, and grants or denies the access to the requester based on access policies defined by the cloud user.

Whenever cloud user changes his policies, the PMSP applies the required updates and communicate them with the target cloud service providers. This could be done by push and pull strategy. In push strategy, whenever there is a change, the PMSP updates the policies and exports them to target CSPs while in pull strategy, the CSP initiates the communication and checks for possible policy changes/updates in certain time periods. However, we believe that push strategy is more efficient in this situation. Since the associations between policies, resources, and cloud service providers have been already identified, the PMSP only needs to relate the changes to target cloud service providers and export them.

Similarly, whenever cloud users add/remove resources to/from cloud service providers, appropriate updates need to be done. However, in this case we believe that pull strategy is better because resources are stored in cloud service providers and whenever there is a change, CSPs can inform the PMSP to get updated policies.

IV. DISCUSSION

In this section, we discuss some of the advantages our proposed cloud based policy management service has over existing systems and its weaknesses.

- Policy specification functionality is external to cloud service providers and can be done in a centralized location for all cloud providers. Decisions about who has access to what resources are made locally and enforced by each cloud service provider. However, the specification of policies for all resources and services is done centrally in a single location.
- Cloud users use a unified policy management system to control access to all their resources scattered over the Cloud. They do not need to deal with various policy management systems bound to each cloud service provider.
- Cloud users compose access control policies using controlled natural language and do not need to use various policy specification languages. They do not need to learn different policy languages employed by various cloud service providers and can simply specify policies using controlled natural language.
- Cloud users use a single management tool to compose access policies which allows them to have a consistent user experience when managing these policies. They do not need to learn to work with different interfaces and tools that may even not be usable.
- Since access policies are composed using a single policy management tool and hosted in a single location, cloud users have a consolidated view of the access policies applied to their resources.
- If cloud users move their resources from one cloud service provider to another for any reason, they do not need to redefine all the policies again. For example, if Alice moves one document from GoogleDocs to Amazon S3, she does not need to redefine the policies associated with that document in Amazon S3.
- It is easier for users to introduce new access control policies and modify existing ones when needed.
- With existing systems, the cloud user is limited to the functionality provided by the cloud service providers’ policy engine. However, our proposed framework may be able to apply extra policies by transforming them into the provider’s policies. For instance, in Amazon S3 or Dropbox, if Alice wants to share a file with a group of users, she has to define different policies for each user. In our system, however, she can specify a policy to share the file with a group of users called “colleague” and our system exports required policies (one policy per each user of the group) into the Amazon S3 or Dropbox.
- Our proposed framework supports reactive access control model [13]. The reactive access control model aims to help non-expert users to create accurate access policies. In this model, cloud users are not required to determine all access control policies a priori, but may instead update their policy dynamically in response to access requests that would not otherwise succeed. It has been shown that when making policy decisions, people want more control and interactivity and rely on social norms, so reactive policy creation can contribute to an access control system [13].

Despite various advantages our proposed framework provides, some concerns should first be addressed to make sure our proposed framework is efficiently used in cloud computing environment. For a successful deployment of the framework, the CSPs should be able to communicate with the PMaaS via API or can read the policies from output languages (like XML, XACML, OWL, etc) of the PMaaS.
and update their policy base using that information. Note that PMaaS does affect the authorization decision making and enforcement processes of the CSPs and only policy specification is externalized; policies are specified in the PMaaS but access decisions are still made and enforced by CSPs.

A question may arise regarding privacy of cloud user’s identity and privacy of his policies and the concern is that cloud users have to trust PMaaS provider to provide their identifications in different cloud service providers and specify their access policies using the PMaaS. We assume that there is enough level of trust between cloud user and PMaaS provider to deploy the service. However, the PMaaS could be deployed as private cloud within an organization’s premise or fully controlled by an individual user to avoid privacy concerns.

Another concern is that if the PMaaS which is a centralized policy manager gets compromised what happens. Since the PMaaS only stores the policies and resources are stored at CSPs, the resources won’t be in any danger. However, the PMaaS provider should take all necessary steps to make sure it is secure enough and CSPs can communicate with ease of mind. In general, as any other service, PMaaS provider should strengthen security and privacy of its service to ensure that the PMaaS is secure enough that the CSPs can trust the providers to allow their users to use the PMaaS service.

V. RELATED WORK

The National Institute of Standards and Technologies (NIST) cloud efforts intend to promote the effective and secure use of the technology within government and industry by providing technical guidance and promoting standards. The NIST has released an early definition of cloud computing and also documents on how effectively and securely use the cloud computing paradigm [1]. The Cloud Security Alliance is an effort to facilitate the mission to create and apply best practices to secure cloud computing [2]. Its initial report, “Security Guidance for Critical Areas of Focus in Cloud Computing”, outlines areas of concern and guidance for organizations adopting cloud computing. The intention is to provide security practitioners with a comprehensive roadmap for being proactive in developing positive and secure relationships with cloud providers [2].

Takabi et al. discuss security and privacy challenges in cloud computing environment [5]. They present unique issues of cloud computing that exacerbate security and privacy challenges in clouds, discuss these challenges, some possible approaches and research direction to address them. They have also proposed SecureCloud, a comprehensive security framework for cloud computing environments [6]. The framework consists of different modules to handle security, and trust issues of key components of cloud computing environments. These modules deal with issues such as identity management, access control, policy integration among multiple clouds, trust management between different clouds and between a cloud and its users, secure service composition and integration, and semantic heterogeneity among policies from different clouds.

Jaeger et al. discuss security challenges in the cloud, foundation of future systems’ security and key areas for cloud system improvement [7]. Kandukuri et al. present security issues that have to be included in service layer agreement (SLA) in cloud computing environment [12]. Jensen et al. provide an overview on technical security issues of the cloud [11]. They start with real-world examples of attacks performed on the Amazon EC2 service, then give an overview of existing and upcoming threats to the cloud. They also briefly discuss appropriate countermeasures to these threats, and further issues to be considered in future research. Gruschka et al. present taxonomies and classification criteria for attacks on cloud computing based on the notion of attack surfaces of the cloud computing scenario participants and try to anticipate the classes of vulnerabilities that will arise from the cloud computing paradigm [9].

Zhang et al. describe an electronic health record (EHR) security reference model for managing security issues in healthcare clouds [8]. They discuss important concepts related to EHR sharing and integration in healthcare clouds and analyze the arising security and privacy issues in access and management of EHRs.

Tian et al. present a new database service provider (DSP) re-encryption mechanism to implement flexible access control enforcement in the database as a service (DaaS) paradigm [10]. The basic idea of the approach is that the DSP uses different re-encryption keys for users of the system. Their approach can relieve the users from the complex key derivation procedure while implementing the selective access control of the encrypted data by the DSP.

IBM’s SPARCLE policy workbench “simplifies how people manage organizational policies across the enterprise, improve the quality of policy rules, and enable those rules to be implemented through technology to ensure consistency, reliability, and compliance” [15]. It allows users to author policy rules in natural language. It automatically parses the policies to identify policy elements and enables the user to review and modify the rules. It includes a parser designed to identify the rule elements in policy rules. Then, it generates a machine readable (XML) version of the policies. Karat et al. have designed a system to facilitate effective privacy policy authoring, implementation, and compliance monitoring [14]. They introduce two methods for policy authoring using natural language and structured entry method and conducted a usability study to show effectiveness of their privacy policy tool.
VI. CONCLUSION AND FUTURE WORK

Although security and privacy issues of cloud computing are delaying its fast adoption, it has become very popular and we need to provide security mechanisms to ensure its secure adoption. While security and privacy services in the Cloud can be fine-tuned and delivered in new ways and by new types of service providers, there need to be frameworks that efficiently deliver cloud-based security services and provide a desirable solution to customers based on their requirements.

We have proposed Policy Management as a Service (PMaaS), a cloud based policy management framework that puts users in full control of their resources which may be scattered across multiple cloud service providers. It is designed to give cloud users a unified control point for specifying authorization policies, who and what can get access to their data, content, and services, no matter where all those things live on the Cloud. It relies on a user’s centrally located policy manager of those resources and enables users to manage access policies using a centralized policy manager which provides usable interfaces for specifying access policies and exporting them to the cloud service providers on behalf of the user. Furthermore, we have discussed advantages that the proposed framework has over existing systems.

We are currently implementing a prototype of the proposed framework to show its applicability using real world cloud service providers and evaluate its performance.

ACKNOWLEDGMENT

This research has been supported by the US National Science Foundation award IIS-0545912. We would like to thank the anonymous reviewers for their helpful comments.

REFERENCES


[17] Protocol for Web Description Resources (POWDER): Description Resources http://www.w3.org/TR/powder-dr

[18] Protocol for Web Description Resources (POWDER): POWDER-S Vocabulary (WDRS) http://www.w3.org/2007/05/powder-s
