Towards a Flexible Fine-Grained Access Control System for Modern Cloud Applications

Reza Shiftehfar
Department of Computer Science
Uni. of Illinois @ Urbana-Champaign
Email: sshifte2@illinois.edu

Kirill Mechitov
Department of Computer Science
Uni. of Illinois @ Urbana-Champaign
Email: mechitov@illinois.edu

Gul Agha
Department of Computer Science
Uni. of Illinois @ Urbana-Champaign
Email: agha@illinois.edu

***Abstract*— The fast growth of cloud applications highlights the requirement of appropriate security controls to restrict access to shared resources limited to authorized users. Existing authorization systems are not primarily designed for cloud environments and do not provide the required flexibility, adaptability, elasticity, scalability, or fine-grainedness of cloud applications. This paper outlines an ongoing effort in development of a flexible fine-grained access control system for modern cloud-based applications. Modern cloud applications are distinctive in that the required authorization rules are defined by the organizations owning data and resources, before the application logic can be developed by their programmers. Although this simplifies cloud application development and provides flexibility and adaptability to potential future policy changes, it highlights the need for an adaptive flexible authorization system.**

I. Introduction

The widespread application of cloud platforms for managing large datasets has brought an increasing awareness of the security requirements of cloud-based applications. Traditionally, cloud solutions are developed in a closed trusted environment without providing high level of security protection [3]. Such a trusted environment can only be reached when the cloud is isolated from the outside word and used by a small group of authorized people within the company. However, with the cloud acting as the heart of many organizations, the number of users accessing cloud services has dramatically increased. In addition to users accessing the cloud from within the company, new external clients also interact with the cloud via cloud-based applications. It is imperative to adopt a flexible and fine-grained authorization system to regulate accesses to different cloud resources. The required levels of accesses differ for internal programmers developing applications for the cloud and for external clients reaching the cloud through applications running on their untrusted personal devices.

Traditional authorization solutions, such as Discretionary Access Control, Mandatory Access Control, and Role-Based Access Control, are all identity-based access control models and fail to address the dynamic nature of cloud applications. Modern access control systems, such as Attribute-Based Access Control systems and its published standard, eXtensible Access Control Markup Language (XACML), support dynamic decision-making capabilities but lack a comprehensive implementation that naturally supports multiple policies as required by cloud environment. Previous efforts to extend XACML to support multiple policies require large set of prior-arrangements making it impractical and inefficient for large-scale applications [1]. More importantly, these solutions have a data-centric view of the cloud and focus on developing an authorization system that allows revealing stored data only to the authorized users, whereas modern cloud applications require access control for both data and code components moving transparently across cloud spaces and user devices.

Our objective is to develop a flexible fine-grained access control framework for regulating accesses to cloud resources used by modern cloud applications. The framework aims at allowing enterprise developers to efficiently develop their cloud application without worrying about distribution and access restrictions. It emphasizes on the principle of separation of concerns by separating the application logic, to be developed by the programmers, from the access control policy, defined beforehand at a higher level by the organization. The framework provides fine-grainedness, flexibility, scalability, and confidentiality for organization-wide cloud applications with components storing data, executing code, and transparently moving between cloud spaces and user devices. It accommodates the needs of both organization-wide access control policy makers and end-programmers, automatically taking both into account to define who can exercise which access privileges and when.

II. Cloud Authorization System

Cloud Model: Cloud services are transitioning from the model of public cloud spaces to private clouds, and recently to a hybrid model combining both. In order to preserve privacy and confidentiality of the data, cryptographic methods are widely used to encrypt data stored in the cloud while decryption keys are only disclosed to authorized-users [2]. This solution has the traditional data-centric view on cloud computing with the focus on storing data and providing services to access the stored data. However, in modern cloud applications, resources stored in the cloud contain more than just data. These resources contain part of the application code that results in access operation meaning execution of the code inside the cloud. It is obvious that certificate-based authorization systems fail to address this type of applications, as the encrypted code within the cloud cannot be executed without decryption and revealing its content to the cloud

978-1-4799-5063-8/14 $31.00 © 2014 IEEE
DOI 10.1109/CLOUD.2014.144

provider. As a result, some companies have gradually moved toward building their own private clouds. Storing sensitive data within private clouds aligns with the traditional on-premise application deployment model, where sensitive data resides within the enterprise boundary and is subject to its physical, logical and personnel security and access control policies. However, private clouds are usually not as efficient, scalable, nor reliable as the public ones and are gradually being replaced by a hybrid model that benefits from the power of public services while keeping the confidential or sensitive data and algorithms in-house. Our framework views the cloud in the most general form, combining one or more private and public cloud spaces. This allows for the creation of a flexible elastic hybrid cloud space to address the various needs of different organization. Different users, including both internal staff and external clients, can access cloud resources but with different access levels and restrictions.

Modern Cloud Application Model: With the growing popularity of mobile applications and the cloud acting as a major supporting back-end, traditional data-centric view of cloud services needs to be replaced with a more general data/computation-centric view that dynamically and transparently leverages cloud resources to support resource limitations on the end-user device. As part of such elastic application, components storing data or performing computations move between different cloud spaces and the end-user device. An elasticity manager monitors the environment and decides the timing of the move-around of the application components based on user preferences, existing workloads, or performance goals. However, access privilege of a component might change based on its execution location and an authorization system must be combined with the elasticity manager to regulate that. In order to support component migration, modern cloud applications avoid shared memory model and restrict component interactions to communicating using messages. This perfectly aligns with the actor model of computation that views distributed components, called actors, as autonomous objects operating concurrently and asynchronously. The model provides natural concurrency, resiliency, elasticity, decentralization, and location transparency that ease the process of scaling-up or out and is used as the underlying application development model for our framework.

Cloud Authorization Framework: Our cloud authorization model is composed of the following parties: the owner organization, that owns data and governs cloud infrastructure, different programmers inside the owner organization, who access cloud resources for different purposes and develop applications using cloud resources, and external clients who use the developed cloud applications as services. Cloud applications have different components running on the end-user device or the company-managed cloud spaces, with the capability of migrating between them. Thus, an authorization framework is required to regulate and manage component migration and resource usage for different types of users and applications. The required authorization policy regulating access to different resources by different users must be defined at a higher level by the organization that owns both data and cloud resources. However, individual applications can further tighten these organization-wide accesses by defining their application-specific policy. The framework currently supports both hard organization-wide and soft application-specific policies.

Having modern elastic application model supported by the authorization framework allows bypassing end-user device hardware restrictions by using cloud resources. This leads to device-independent applications that can dynamically be adjusted based on the capabilities or surrounding context of the end-user [4]. An elasticity manager can maintain a cost model to find the optimal component distribution incorporating factors such as performance, latency, network and cloud usage cost. Including the elasticity manager as part of the authorization framework provides automatic regulated code offloading that allows programmers to only focus on the application logic.

In the actor programming paradigm, components can send or receive messages, create new actors, or migrate to new locations. Our authorization framework supports all these actions. Access control decisions are made based on the attributes of the requester, the resource, and the requested action using the predefined policy rules. The framework combines the advantages of attribute-based authorization system, actor programming paradigm, and elastic application development to support modern mobile-cloud applications. Current implementation uses the Akka programming language and regulates interaction between different application components distributed between end-user device and cloud spaces in addition to dynamic migration of components according to predefined organization-wide regulations or end-user device limitations.

III. Future Work

To fully realize the benefits of a flexible fine-grained access control framework for modern cloud applications, we are testing the framework prototype in different real-world applications, with the goal of evaluating and quantifying the capabilities of the framework with respect to fine-grainedness, flexibility, practicality and scalability, while masking the heterogeneity of the underlying infrastructure.

IV. Acknowledgments

This work is partially funded by AFOSR contract FA8750-11-2-0084.

References

[1] H. Takabi, J. B. Joshi, and G.-J. Ahn. Securecloud: Towards a comprehensive security framework for cloud computing environments. In *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*, pages 393–398. IEEE, 2010.

[2] G. Wang, Q. Liu, and J. Wu. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 735–737. ACM, 2010.

[3] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[4] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong. Securing elastic applications on mobile devices for cloud computing. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 127–134. ACM, 2009.