**Industry Standard Authorization For Sensitive Resources**

**ABSTRACT**

Role based access control has become as the industry standard for authorization, and it is widely deployed as it provides organizations with a simplified mechanism for granting privileged access to sensitive resources. Although Rabc systems traditionally consider only identity attributes, such as job title, the emergence of location-based applications has led to the enrichment of the model with spatial features. A number of Rabc extensions incorporate location constraints into access control policies providing organizations with the ability to control resource access based on the physical coordinates of the requesting users.

**EXISTING SYSTEM**

While the challenge of creating enforcement architectures has received some focus, existing solutions require disclosure of the requesting user’s logical location or physical coordinates. Previous work has acknowledged the severe privacy threats that may occur as a result of disclosing fine-grained location information with high frequency.

**PROPOSED SYSTEM**

We propose a framework for privacy preserving Geo-rabc that enforces location-based authorization without revealing the identity attributes or physical coordinates of requesting users to the policy enforcement point. Our approach relies on a combination of cryptographic techniques and separation of functionality among several distinct components. The trust assumption in our model is reduced to one single component, which generates all cryptographic secrets required for evaluation but *does not participate* in the on-line operations.

**MODULE DESCRIPTION**:

1. **Cloud Computing**
2. **Role authority (*RA*)**
3. **Cryptography**
4. **Service provider**

**Cloud Computing**

Cloud computing is the provision of dynamically scalable and often virtualized resources as a services over the internet Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing represents a major change in how we store information and run applications. Instead of hosting apps and data on an individual desktop computer, everything is hosted in the "cloud"—an assemblage of computers and servers accessed via the Internet.

Cloud computing exhibits the following key characteristics:

**1. Agility** improves with users' ability to re-provision technological infrastructure resources.

**2. Cost** is claimed to be reduced and in a public cloud delivery model [capital expenditure](http://en.wikipedia.org/wiki/Capital_expenditure) is converted to [operational expenditure](http://en.wikipedia.org/wiki/Operational_expenditure). This is purported to lower [barriers to entry](http://en.wikipedia.org/wiki/Barriers_to_entry), as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a [utility computing](http://en.wikipedia.org/wiki/Utility_computing) basis is fine-grained with usage-based options and fewer IT skills are required for implementation. The e-FISCAL project's state of the art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.

**3.** [**Virtualization**](http://en.wikipedia.org/wiki/Virtualization) technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another.

**4.** [**Multi tenancy**](http://en.wikipedia.org/wiki/Multitenancy) enables sharing of resources and costs across a large pool of users thus allowing for:

**5. Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)

1. **Utilization and efficiency** improvements for systems that are often only 10–20% utilized.
2. [**Reliability**](http://en.wikipedia.org/wiki/Reliability_(computer_networking)) is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for [business continuity](http://en.wikipedia.org/wiki/Business_continuity) and [disaster recovery](http://en.wikipedia.org/wiki/Disaster_recovery).

**8.** [**Performance**](http://en.wikipedia.org/wiki/Computer_performance) is monitored and consistent and loosely coupled architectures are constructed using [web services](http://en.wikipedia.org/wiki/Web_services) as the system interface.

**9.** [**Security**](http://en.wikipedia.org/wiki/Computer_security) could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security [audit logs](http://en.wikipedia.org/wiki/Audit_log) may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

**10.** [**Maintenance**](http://en.wikipedia.org/wiki/Software_maintenance) of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

**Role authority (*RA*)**

A centralized RBAC server that is tasked with authenticating users, as well as creating and maintaining session identifiers.

**Cryptography**

The art of protecting information by transforming it ([*encrypting*](http://www.webopedia.com/TERM/E/encryption.htm) it) into an unreadable format, called [cipher text](http://www.webopedia.com/TERM/C/cipher_text.htm). Only those who possess a secret *key* can decipher (or [*decrypt*](http://www.webopedia.com/TERM/D/decryption.htm)) the message into [plain text](http://www.webopedia.com/TERM/P/plain_text.htm). Encrypted messages can sometimes be broken by cryptanalysis, also called *code breaking*, although modern cryptography techniques are virtually unbreakable.

**Service provider (*SP*)**

A server that controls access to a protected resource. We assume that there are

multiple such servers, and the client knows which *SP* to contact when retrieving a desired service

**System Configuration:-**

**H/W System Configuration:-**

**Processor - Pentium –III**

Speed - 1.1 GHz

RAM - 256 MB (min)

Hard Disk - 20 GB

Floppy Drive - 1.44 MB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

**S/W System Configuration:-**

Operating System : Windows95/98/2000/XP

Application Server : Tomcat5.0/6.X

Front End : HTML, Java, Jsp

Scripts : JavaScript.

Server side Script : Java Server Pages.

Database : My sql

Database Connectivity : JDBC.

**CONCLUSION**

We propose three directions for future work. First, we find the notion of transient attribute credentials to be very powerful; applying this concept to other domains could produce desirable results. Second, our work demonstrates the feasibility of reconciling robust privacy guarantees with strong security requirements; this raises the question of what other settings could benefit from such results. Finally, our security and privacy guarantees rely on two assumptions: the presence of a trusted third party (*IA*) and there is no collusion between *SP* and the authority servers (*RA*, *LA*, and *IA*). A solution that reduces these assumptions would be very attractive and would aid in the adoption of this work.