Chapter 11
An Efficient, Robust, and Secure SSO Architecture for Cloud Computing Implemented in a Service Oriented Architecture

Khandakar Ahmed
RMIT University, Australia

Altaf Hussain
Shahjalal University of Science and Technology, Bangladesh

Mark A Gregory
RMIT University, Australia

ABSTRACT
Implementing Single Sign-On (SSO) in a Cloud space for a spectrum of services and applications is an interesting research avenue for scientific communities in the field of secure identity and access management for Cloud Computing. Using an SSO implementation, in the backend, users can navigate any or all of the supported applications or resources without the need to repeatedly provide credentials. In this chapter, the authors present an efficient and robust Cloud Single Sign-On Architecture (CSSOA) model based on a token security mechanism. Service Oriented Architectures (SOAs) are one of the enabling technologies for solving complex service oriented real world challenges, and hence, CSSOA has been implemented using SOAs. In the authors’ CSSOA model, a CSSO SOAP authentication service is distributed among the Cloud servers while the CSSO database service is centralized.

1. INTRODUCTION
Cloud Computing enables a method of on-demand convenient network access to a shared pool of configurable computing resources (Jaeger & Schiffman, 2010). According to IBM, it is a new utility computing model based on consumption and delivery, where the user only sees services and has no need to know anything about the underlying technology or implementation (Breiter & Behrendt, 2009). With pools of computation, network, information and storage resources the Cloud offers...
a collection of services, applications, information and infrastructure. With the promising potential of computing functions as a utility, customers are both excited and nervous. The excitement comes from the chance of disassociating themselves from infrastructure management, and being able to focus on core competencies with an emphasis on the business logic that should be implemented through IT rather than management of IT. On the other hand, customers are nervous with the security risks associated with Cloud Computing which have not been completely addressed yet (Hubbard, et al., 2010, March). However, today’s aggressive adoption of immature Cloud Computing services by enterprises creates the need for a strong Cloud-based Identity and Access Management (IAM) system that supports business needs ranging from secure collaboration with global partners to secure access for global employees consuming sensitive information using a range of devices from any location world-wide at any time of day. This strong IAM invariably requires users to enter credentials and login for each application accessed and hence it is quite cumbersome. In order to deliver a unified and seamless migration from one application to another inside a Cloud, an SSO would be a great solution. Implementing SSO in a Cloud environment will require a user to login once to a Cloud and be provided with access to all authorized applications without re-login.

Many enterprises exercise identity management to integrate applications into different domains through an application portal giving users an opportunity to hop among applications without the need for re-authentication. However, this feature cannot be extended to the Cloud Computing architecture. Existing identity management approaches work well for enterprise applications within a corporate data centre or within the same domain while services in Cloud Computing are typically external to the data centre and located within a different domain and hence a new SSO architecture is needed for the Cloud.

For multiple applications, SSO is a centralized authentication mechanism which could be independent and may or may not be interrelated. Here, a user is authenticated once and then all of the user’s subsequent service access requests via other applications within the enterprise system are handled seamlessly until the user’s login session is terminated. iGoogle login is the best appropriate analogy to visualize this concept where the user once logged in can access multiple applications like Gmail, Google Docs, Google+ accounts etc. without having to re-login to each individual application. SSO provides a few advantages: 1) In SSO users utilise a single username and password which is easier to manage and remember; 2) Additionally, the system administrator can manage user accounts centrally and thus it is easier to provision or de-provision passwords; 3) As SSO users are less likely to lose passwords it reduces the assistance provided to users by the IT helpdesk and hence reduces costs; and 4) The service provider manages a single access mechanism rather than developing and maintaining several identity modules for different applications. However, there are some drawbacks or constraints that need to be traded-off in order to get the benefits. A few of them are: 1) Due to password leaks or misuse, security breaches could affect multiple applications and resources; and 2) Due to the constraint of a single point of failure, there is a need for SSO IAM high availability.

There are various security mechanisms that can be used as part of an SSO mechanism. A few of them are Kerberos TGT (Migeon, 2008; Tagg, 2000; Upadhyay & Marti, 2001), Smart Card (Kao & Milman, 2001; Mauro, Sunyaev, Leimeister, Schweiger, & Krcmar, 2008; Rankl & Effing, 2010), One Time Password Token OTP (Blunk, Vollbrecht, & Aboba, 2002; Tiwari & Joshi, 2009) and Integrated Windows Authentication. Some of the basic implementation considerations include: 1) An ideal SSO should be aware of the applications that it is serving and hence also of encryption or decryption keys for handling the authentication requests; 2) The Authentication Token generated during the first successful login needs to be encrypted or decrypted while it travels.