

An Approach to Achieve Delegation of Sensitive RESTful Resources on Storage Cloud

Kanchanna Ramasamy Balraj

Engineering Ingegneria Informatica Spa, Rome, Italy

Abstract. The paper explains a simple approach to achieve delegation of RESTful resources in a storage cloud using query string[6]. It is browser based and is well suited for delegating read access to RESTful resources and supports simple as well as chained delegation. As more and more cloud storage providers are moving towards providing RESTful web service interfaces, it is imminent that delegation approaches are defined to suit the adoption.

Keywords: User centric delegation, Chained delegation, Multi-tenant storage cloud, Delegation in RESTful approaches using Browser, Access control

1 Introduction

The objective of this paper is to define an approach to delegate read access to sensitive RESTful resources securely in Storage cloud. Using RESTful web service, allows storage data being accessed from URIs by passing query string as parameters. Query string is the part of a Uniform Resource Identifier (URI) that contains data to be passed to web applications. The length of the URI is theoretically infinite. But today's browsers and web servers do limit the length of the URI. For instance, url max length supported by IE is 2,083 characters, header max size supported by apache is 8k and that of IIS(Internet Information Services) server is 16k.

When a storage data needs to be shared that is sensitive and that allows only read access, query string authentication is used. The delegation approach defined in this paper is an extension to the Query String authentication that provides chained delegation and respects the compliance regulations. The compliance requirements in cloud, require the tracing of information like, which users access the resource, for auditing purposes. The approach defined in this paper, will allow reconstruction of the delegation chain and thus verification of the resource owner's consent and this retracing is useful for auditing and accounting purposes in cloud.

2 Cloud Storage Environment

A new storage infrastructure is emerging, as an embodiment of Service Oriented Architecture, with a set of services that are delivered via APIs. Such a cloud

Storage infrastructure allows users to store their data at remote disks and access them anytime from any place. This service is commonly known as Storage as a Service, and it facilitates cloud applications to scale beyond their limited servers. Examples of commercial cloud storage systems are Amazon's S3 [2] and EMC Storage Managed Service[1].

Scalability, performance and pay for use are attributes of traditional and cloud storage solutions, but Web services APIs is a distinguishing feature of cloud storage. Most of the Cloud Storage industry strongly believe that a key capability of a storage cloud is REST style Web Services API.

In a cloud storage environment, delegation is needed when enterprise users store their data on the cloud and want to share the data only to certain users. Moreover, compliance regulations mandate the tracking of users who access the resource for accounting, billing and auditing purposes.

3 Delegation Model

3.1 Existing approaches

Delegation can be achieved through the following methods

- Certificates
 - Many end users are not familiar with x509 certificates, and many users need to be educated on the usage of certificates.
 - X509 certificates need the public certificate to be passed for signature verification and are not suited to be used in query string due to size limitations.
- ACL-based mechanism (e.g.) in Amazon's S3.
 - Needs updation of ACLs for delegating access, this is not flexible and requires deletion of ACLs to revoke access delegation

When a user/resource owner shares access to a RESTful resource, one of the common methods used is providing browser access with URIs containing Query String parameters. This is called Query String authentication[5]. The user creates a Query String, with the name-value pairs of the parameters like name of the storage object, container/bucket etc., adds its signature and passes it to other users for sharing access to the resource. i.e Any user in possession of the URI is able to access the resource. This works when the storage object shared is not sensitive and can be shared with anonymous people. But often, especially in the enterprise scenario, this is not the case. Often the Resource owner wants to be sure that the URI is being used only by the delegatee or any other entity delegated to by the delegatee and not by any other unauthorised entities.

3.2 Proposed approach

Assumptions

- Signatures in the paper refer to SHA-HMAC hashing of the query string using a secret key unique to each user

- Users are in possession of secret key that they use for signatures in URI and eventual authentication.
- Secure Access Systems are in place that verify the signatures of users
- There is a shortening service which does shortening of URIs.

Methodology In this methodology, a delegator(User A) who wants to delegate his resource Ra, creates a query string with the delegatee's(User B) unique id and UserA's signature. When the delegatee(User B) accesses the shared URI, the Secure Access Service includes a redirection URI and redirects the user agent of User B to the Idp of User B . The Idp authenticates User B, on successful authentication, adds a query parameter that asserts the authentication of User B in the redirection URI, and redirects the user agent of User B to the redirection URI specified by the Secure Access Service. The Secure Access service now verifies the authentication assertion, and the User B is allowed access to the resource. This is simple delegation. This flow is similar to the OAUTH [3] user agent flow and is outlined in the figure 1.

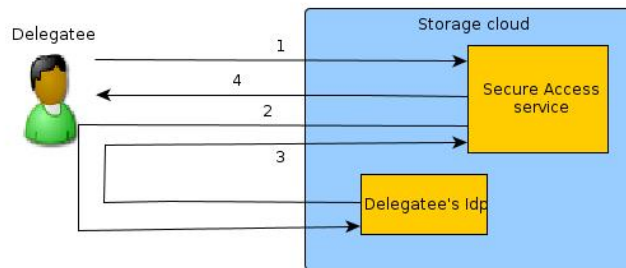


Fig. 1. Flow of delegatee's access to delegated resource

- 1) Delegatee(User B) accesses the URI passed by User A (Delegator)
- 2) The Secure Access Service at the Storage cloud, verifies the signature of the delegator and redirects the delegatee to his Idp for authentication with a Redirection-URI.
- 3) The Idp on successful authentication of Delegatee, adds a parameter to the query string of the Redirection-URI parameter, that asserts the delegatee is authenticated, and redirects the delegatee to the Redirection-Uri provided by the Secure Access Service.
- 4) On verification of the authentication assertion added by the Idp, the delegatee is allowed access to the delegated resource.

Things get complex when we consider the Chained delegation scenario, where the delegatee decides to delegate the resource delegated to him to another

user(User C). This is chained delegation outlined in figure 2 . User B in order to delegate, creates a query string for the resource Ra, adds the delegatee’s(User C) unique Id, adds the shortened URI of the original query string received from User A and passes to User C.

When User C accesses the URI, the same steps are repeated as that for simple delegation, the user is first redirected to the Idp for authentication and then on successful authentication, the Secure Access service checks the resource owner for the resource being accessed and constructs the delegation chain using the info in the URI. The Secure Access service checks the shortened URI, which states the delegation of Ra from User A to User B, and the current delegation of Ra from User B to User C, on successful establishment of chain, the access to resource is allowed for User C.

If a chain cannot be established between the resource owner and the user accessing the resource, the access to the resource is denied. Since the URI shortening service is used, there can be more levels of delegation without any significant increment in the size of the URI.

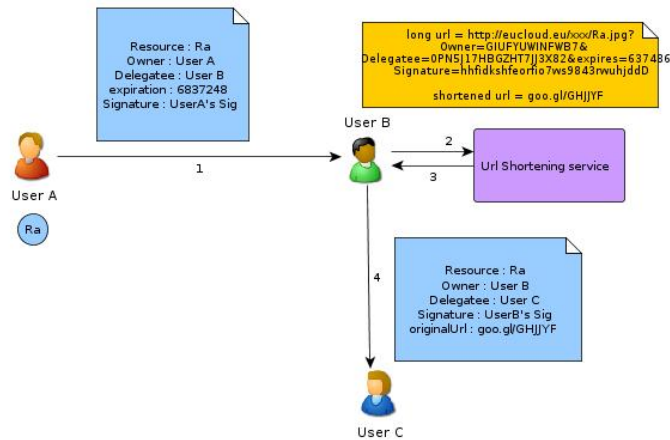


Fig. 2. Chained delegation

Revocation The resource owner when creating the query string, includes a parameter "expires" that contains the time limit upto which the resource can be accessed by the delegatee. Any further levels in the delegation, are supported only till the expiry time stated by the original resource owner. After which the access permission is revoked.

4 Technology or Business Case description

Many of the most popular storage cloud services include or exclusively use REST, including SoftLayer's CloudLayer, Amazon S3, Nirvanix SDN and Rackspace Cloud Files[4]. Cloud storage is becoming popular, and there should be ways to make access delegations simple and trackable. End Users are not willing to use X509 proxy certificates like methods that they are not familiar with. There are multiple tenants in a cloud, there are cases when a resource owner inside a tenant wants to share his resource with another user(delegatee) belonging to same or different tenants just by passing the URI. Although the Resource owner wants to be sure that the URI is being used only by the delegatee or any other entity delegated to by the delegatee and not by any other unauthorised entities.

The approximate URI length for a GET request, in case of simple delegation is 150 characters including the parameters - signature, owner's id, delegatee's id, expiry time, resource name. Including the shortened URI to achieve chained delegation, the approximate URI length becomes 180 characters, as opposed to URI length of 300 characters in case the shortening service is not used. Without the shortening service, the URI length increases linearly with the number of levels of delegation and leads to limitations in the levels of delegation that can be achieved. Whereas using shortening service, additional parameters can be comfortably added based on the necessity, as part of the URI. With multiple levels of delegation, the approximate URI length remains the same, thus ensuring smooth access to the delegated resources.

5 Conclusions and Summary Recommendations

Most of the Cloud Storage industry strongly believe that a key capability of a storage cloud is the REST style Web Services API. REST is a simple and powerful architecture. Moving towards cloud, REST is being increasingly considered because they are easy to use, the capabilities can be easily used in mashups and they expose significant capabilities. We believe the solution proposed is a good start in achieving additional features to authorization, like Delegation using REST style web services. The solution now addresses delegation for only HTTP GET requests by users. There is scope for this kind of delegation to be extended for other HTTP methods.

6 Acknowledgement

This research has been partially funded from the European Community's Seventh Framework Programme (FP7/2007-2013) under the project Vision-cloud(grant agreement no 257019)

References

1. EMC Managed Storage Service. <http://www.emc.com/>

2. Amazon Simple Storage Service. <http://aws.amazon.com/s3/>
3. OAUTH 2.0 Specification. <http://tools.ietf.org/html/draft-ietf-oauth-v2-15>
4. <http://blog.programmableweb.com/2007/11/26/10-online-storage-apis/>
5. <http://docs.amazonwebservices.com/AmazonS3/latest/dev/index.html?S3-QSAuth.html>
6. http://en.wikipedia.org/wiki/Query_string