Poster: Secure Provenance for Cloud Storage

Masoud Valafar (*student*) and Kevin Butler (*faculty*) Department of Computer and Information Science University of Oregon, Eugene OR 97403 Email: {masoud, butler}@cs.uoregon.edu

I. INTRODUCTION

Organizations are increasingly turning to the cloud for data processing and storage. Storing data in the cloud is advantageous for numerous reasons: the elasticity of cloud environments ensures that only storage used is paid for, while tasks such as backup, replication, and geographic diversification of data are effectively outsourced to cloud storage providers. However, unfettered access to this environment and arbitrary migration of data means that determining the origin of information, or equally importantly, determining the modifications and chain of custody undergone by this information before it has assumed its current form, is a virtually intractable problem given the current tools available to us. Such a state of the art becomes increasingly worrisome when issues such as regulatory compliance are brought to bear on information in the cloud: for example, if information is under certain regulations, we must ensure that such data does not automatically migrate to another data center within a cloud provider operating under stricter regulations.

Data provenance, which provides a full accounting of data curation from its creation to the present, provides a means of determining the authenticity of information as well as the manner in which it has been handled. Provenance systems have been well-studied in the context of scientific environments[3]. However, securing provenance, particularly within the context of the cloud environment, has received relatively little attention. Securing provenance is challenging for a number of reasons, including that it may require a separate security model from the data it describes[2]. This poster proposes a model for securing provenance collection, storage and management for cloud storage systems. We explain challenges of securing provenance in the cloud that our model addresses, and describe a prototype implementation.

II. SYSTEM ARCHITECTURE

Figure 1 describes our proposed architecture and its main components. Three main functionalities provided are: (A) secure provenance management, (B) organizational policy enforcement and, (C) scalable access control to provenance information. We discuss each of these in further detail.

A. Secure Provenance Collection

Clients use the cloud storage to retrieve stored files. To allow the transaction, the cloud storage checks the access rights of the client to the stored files. Having retrieved the files, clients



Fig. 1. Overview of an architecture for adding secure provenance collection and enforcement elements to a cloud infrastructure.

process the retrieved data and update the files on the cloud storage, once the processing is complete.

In the scenario described above, provenance is generated while the client processes data on the host system. To enable the cloud provenance system to track provenance generated at the host, the client must send the provenance of a data object uploaded to the cloud along with the object itself. The host system should ensure that (*i*) all processing of the data is thoroughly monitored and the associated provenance is completely recorded and, (*ii*) the integrity, consistency and non-repudiability of the provenance information is preserved. The combination of the host provenance system and the cloud provenance system provides these services.

To provide the desired functionality, we propose to add a security module to a conventional provenance system, such as PASS[6]. The conventional provenance system records provenance information and the security module ensures provenance integrity. Next, the data and its provenance are sent to cloud storage, following the protocol described in Table I.

The cloud provenance system, which consists of collector and authority components, intercepts all queries to the cloud storage system. The former component extracts provenance information and the latter is in charge of storing and managing provenance information. Upon receiving the information query from the client, the provenance collector strips the provenance information from the query and checks its integrity. Next, the collector sends the data and its provenance information to the cloud storage and the provenance authority, respectively. The process is aborted if the cloud storage does not recognize the client's right to update the file or the authority realizes that the operation is against that organization policies. Note that the cloud provenance system and the client need to authenticate each other by a means such as PKI.

Table I demonstrates the described protocol. At the first two

1.	$C \rightarrow PS$:	$n_c, Guid_c, oid$
2.	$PS \rightarrow C$:	$n_{ps}, PR_{oid,t-1}, Sign[K_{ps}^{-}, (PR_{oid,t-1} n_c)]$
3.	$C \rightarrow PS$:	$Sign[K_c^-, PR_{oid,t} n_{ps}], PR_{oid,t}, Object$
4.	$PS \rightarrow C$:	$Sign[K_{ps}^{-}, PR_{oid,t}]$
TABLE I		
PROVENANCE COLLECTION PROTOCOL - CLIENT AND THE CLOUD		

PROVENANCE SYSTEM ARE REPRESENTED BY C and PS.

steps, client agent and cloud provenance systems exchange initialization information. At step 3, the client sends provenance information with the update query to the collector component. By sending back the signed operation to the client at step 4, mutual non-repudiability is ensured. Furthermore, by signing the provenance information added to the previous one and creating a hash chain, integrity is ensured. Confidentiality can be assured by encrypting data in transit (e.g., using SSL).

B. Policy Enforcement

We now describe how provenance can be used to provide finer grained, attribute-based access control. Upon intercepting client queries, the cloud provenance collector informs the authority of the transaction and provenance information. The authority, then, retrieves the policies of the organization that the data falls under from the policy database and checks the client query against them. The authority extract data attributes from its provenance information and hence, can provide attribute-based access control.

We extend the language introduced by Ni et al.[7] to enforce policies. The example dataset consists of data records coming from various oceanic sensors and its provenance contains information about sensor attributes. Figure 2 demonstrates a policy example in XACML. The first condition tag (lines 5,6) indicates that users with certain roles can not have access to sensor data with certain attributes. The obligation tag (lines 7-10) restricts data access from certain regions. Upon getting a query on sensor data, the requester's region is evaluated and the operation will be aborted in case the request is coming from a restricted region.

C. Access Control

The ultimate goal for a provenance system is to enable data owners to examine sources and evolution of data and audit how data was produced. However, an access control scheme that can thoroughly address provenance requirements is yet to be proposed due to challenges, such as the potential difference in the access model of the data and its provenance [2].

```
1
      <target>
2
              <subject> any user </subject>
3
              <record> sensor data <record>
4
      </target>
5
      <condition> sensor.attribute == some-value &&
6
                  user.role != some-role </condition>
7
      <obligation>
8
              <condition> access-region == some-region </condition>
9
              <operation>abort-operation </operation>
10
      </obligation>
```

Fig. 2. An example of policy using XACML



Fig. 3. Performance of cloud storage with and without provenance system.

Beyond the conventional challenges that we briefly mentioned above, providing access control for the provenance information on the cloud has its own unique challenges, such as scalability. Data owners should be capable of hierarchically delegating their rights to their clients because it can become extremely hard to apply a fine-grained access control in a diverse and distributed environment, such as a cloud.

Based on the challenges posed above and the fact that the access to provenance information should be determined based on the auditor's role, we envision a distributed rolebased access control (dRBAC) scheme, similar to the one proposed by Freudenthal et al.[4], in our model. As Figure 1 illustrates, auditors can query the provenance database by using a provenance query language, such as PQL[5]. The authority evaluates the auditor's permissions based on the certificate that they provide. It is possible that the auditor gains this right through a hierarchical delegation of rights, which addresses the scalability issues posed above.

III. IMPLEMENTATION AND FUTURE WORKS

We have created a prototype of our proposed model in Python. The prototype is implemented over Cumulus, the storage system for the Nimbus open source cloud toolkit[1]. We evaluated the prototype across files with various sizes and measured the performance with and without the provenance prototype in place. Figure 3 illustrates the results. This figure shows that our prototype introduces less than 10% overhead in overall transaction time.

The prototype currently does not have the access control and policy enforcement modules. We are planning to hook our prototype to a real-world provenance system and add policy enforcement and access control modules.

REFERENCES

- [1] Nimbus project. http://www.nimbusproject.org/.
- [2] U. Braun, A. Shinnar, and M. Seltzer. Securing Provenance. In USENIX HotSec, 2008.
- [3] J. Freire, D. Koop, and et al. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10(3), 2008.
- [4] E. Freudenthal and T. P. et al. drbac: Distributed role-based access control for dynamic coalition environments. In *ICDCS*, 2002.
- [5] D. A. Holland and U. B. et al. Choosing a data model and query language for provenance. In *IPAW*, 2008.
- [6] K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer. Provenance-aware storage systems. In USENIX ATC, 2006.
- [7] Q. Ni, S. Xu, and et al. An Access Control Language for a General Provenance Model. In Secure Data Management, 2009.