SPROV 2.0: A Highly-Configurable Platform-Independent Library for Secure Provenance

Ragib Hasan
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
rhasan@cs.uiuc.edu

Radu Sion
Stony Brook University
Stony Brook, NY
sion@cs.stonybrook.edu

Marianne Winslett
University of Illinois at Urbana-Champaign
Urbana, IL
winslett@cs.uiuc.edu

ABSTRACT

Data provenance allows us to explore the lineage and derivation history of data objects. As data and its provenance flow between people and tasks in potentially untrusted environments, it becomes essential to provide integrity and confidentiality assurances for provenance. Any solution also needs to be efficient, modular, and easy to deploy. In this poster and demonstration proposal, we discuss deployment issues of secure provenance in existing provenance systems. We present the design and implementation of SPROV 2.0 – a highly configurable and modular library for secure provenance. SPROV 2.0 is designed in a platform-independent manner, and can be easily configured using plugins to utilize different cryptographic techniques and storage methods. We also show how SPROV 2.0 can be added to existing provenance frameworks.

1. INTRODUCTION

Provenance information summarizes the history of the ownership of objects and the actions performed on them. Widely used in arts, archives, and archaeology, provenance has recently gained widespread usage in digital data processing. Several provenance systems have been developed in scientific computing and workflow applications [5]. However, most of the existing research has focused on collection, annotation, and querying provenance, and little has been done to ensure the trustworthiness of data provenance through integrity and confidentiality assurances [1]. As data crosses application and organization boundaries and passes through untrusted environments, its associated provenance information becomes vulnerable to illicit alteration. With widespread use of provenance in business, medical, and government sectors, we must ensure that we can trust provenance. Healthcare and financial regulations (e.g., the Sarbanes-Oxley Act) often mandate provenance assurances. To make provenance trustworthy, we need to guarantee completeness - all relevant actions pertaining to a document are captured; integrity - adversaries cannot forge or alter provenance; availability - auditors can verify the integrity of provenance information; confidentiality - only authorized parties should read provenance; and efficiency - solutions should have low overheads.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CCS November 9-13, 2009, Chicago, IL, USA
Copyright 2009 ACM ...$5.00.

In [2], we presented a scheme for providing integrity and confidentiality assurances for data provenance. To evaluate our ideas, we developed SPROV – a high-performance proof-of-concept application layer library for recording file system provenance with integrity and confidentiality assurances. However, existing provenance systems have different security requirements and use many different storage mechanisms. To ensure deployment and compatibility with existing provenance management frameworks, we need a modular and configurable architecture.

In this poster and demonstration, we provide details about the design of SPROV 2.0 (henceforth called SPROV2) – an efficient, highly-configurable, and modular library for secure provenance. SPROV2 separates the provenance collection mechanism from the mechanism to provide security assurances. It is highly-configurable as different plugins can be added to provide various types of security guarantees, depending upon the application domain. In our demonstration, we also show the integration of SPROV2 with existing provenance frameworks such as the Provenance-Aware Storage System (PASS) [4]. The contributions of this poster and demonstration are as follows:

- We investigate issues related to the deployment of secure provenance;
- We design SPROV 2.0 – a high-performance, modular, and configurable architecture for secure provenance; and
- We show how we can integrate SPROV 2.0 into existing provenance frameworks.

2. BACKGROUND

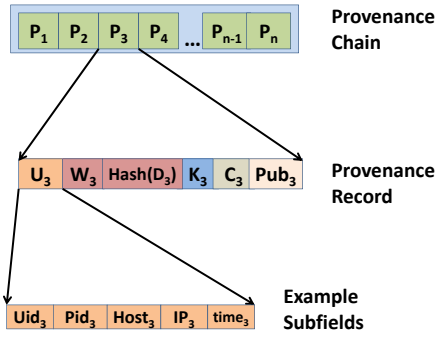

Figure 1: A provenance chain and a sample provenance record.

SPROV2 implements the provenance security architecture that we presented in [2]. In our model, each *document* (i.e., any container of data) is associated with a provenance chain. Changes

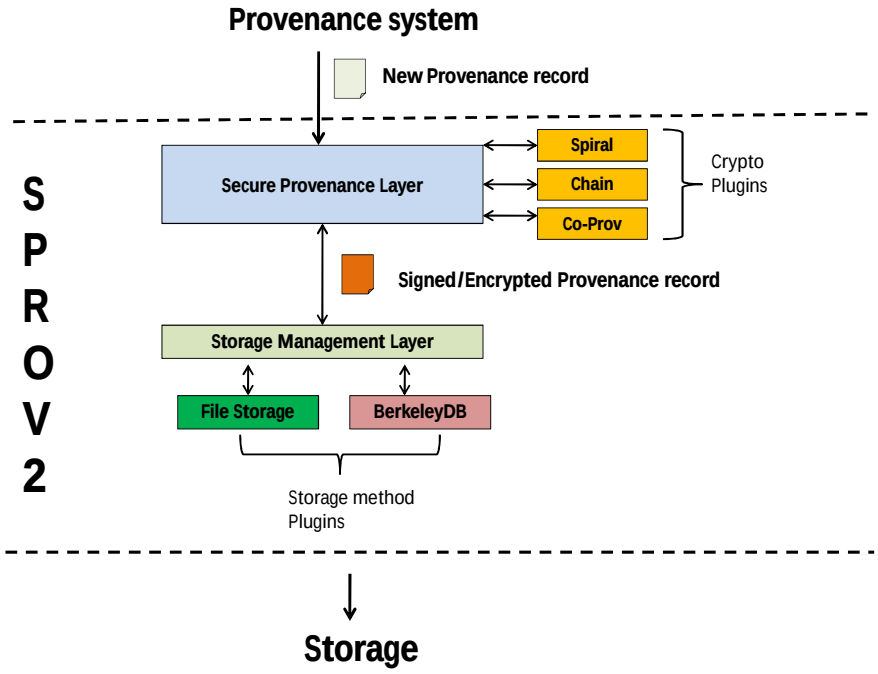

Figure 2: The SPROV2 System Architecture.

made to a document by users are recorded in provenance records. A chronologically ordered sequence of provenance records forms the provenance chain. Integrity is protected using a signature-chaining mechanism. Each user can selectively choose the confidentiality of their actions by encrypting the modification details. Each provenance record P_i consists of the identity component U_i, a log of the user actions W_i (which may be encrypted), a hash of the current version of the document, key distribution component K_i, the checksum component C_i, and an optional public key component $Public_i$. Further, U_i may have different attributes, such as the user ID and timestamps. The checksum C_i is computed by first concatenating the previous checksum C_{i-1} with a hash of the first four components of a provenance record, and then signing it with the current user's private key. The modification log W_i may be encrypted with a key, and the key distribution component K_i is used to allow trusted auditors to access the key. An example provenance record in this model is shown in Figure 1.

In addition, we proposed several refinements and additional functionalities in [2, 3]. For example, a *spiral provenance chain* has multiple checksums per entry, where each checksum is computed by using a checksum from a previous entry in the chain. This allows us to perform integrity preserving summarization of chains, i.e., systematically remove entries from the chain while preserving the integrity-auditing properties. We also discussed the *Co-provenance property* [3], which allows the entanglements of provenance chains of multiple related documents.

3. DEPLOYMENT ISSUES

A number of provenance collection and management systems have already been designed for different application domains [5]. However, none of them consider security issues. While designing SPROV – the proof-of-concept library for secure provenance [2], we identified some deployment issues in augmenting the existing provenance frameworks with security assurances. Here, we discuss some of the main deployment issues:

- **Configurable Security Mechanisms.** The security requirements of different application domains vary widely. For example, the Karma provenance framework is geared towards provenance collection in scientific applications [6]. PASS is designed for provenance collection in commodity operating systems [4]. Some application areas require recording only data modification information, while others may require recording all accesses to data (both read and write). Therefore, the security mechanisms should be configurable in order to customize the integrity and confidentiality assurances.

- **Storage Mechanisms.** Existing provenance frameworks use many different types of storage mechanisms. For example, PASS uses an in-kernel port of Berkeley DB. There are proposals to use a cloud to store provenance data under PASS. Karma stores provenance in relational databases. Therefore, any generic system for secure provenance should separate the provenance collection, storage, and security functionalities. It should also provide support for different types of storage systems.

4. DESIGN

Design Philosophy. Our main goals in the design of SPROV2 was to make it platform-independent, highly-configurable, and efficient. To achieve these goals, we separated the storage and security mechanisms, and also delegated functionality to customized plugins. SPROV2 is independent of any provenance format requirements. All it requires is a finite-length provenance record that the overlying provenance system would generate. After applying

the cryptographic and storage plugins, SPROV2 outputs a secured provenance record ready for the configured storage method.

Components. We designed SPROV2 as an application layer library written in approximately 2000 lines of C++ code. The SPROV2 architecture is shown in Figure 2. SPROV2 consists of the following components:

- **Secure Provenance Layer.** This layer provides an interface to existing provenance systems. It takes a provenance record as input, and applies the configured *cryptographic plugins* to the record to create a signed and/or encrypted version of the provenance record.

- **Storage Management Layer.** This layer manages the storage of the provenance record in different types of underlying storage systems. *storage method plugins* can be configured to use different types of storage mechanism.

Plugins. SPROV2 delegates its main functions to cryptographic and storage method plugins.

- **Cryptographic Plugins.** The cryptographic plugins provide the integrity and confidentiality assurances for a given provenance record. For example, each type of integrity-preserving chaining mechanism can be implemented as a plugin. Similarly, different types of confidentiality and key distribution mechanisms are implemented as plugins. SPROV2 also allows extensibility, allowing a developer to design her own plugin with a customized security mechanism.

  At this moment, SPROV2 supports the following cryptographic plugins: (for integrity) basic chaining, spiral chaining, co-provenance chaining; (for confidentiality) default encryption, broadcast encryption, threshold encryption, and bit commitment.

- **Storage Method Plugins.** SPROV2 allows selecting one or more storage method plugins. Each storage method plugin corresponds to a different storage mechanism (e.g., file, relational database, cloud).

  At this moment, we support the following storage method plugins: file system, Berkeley DB, MySQL, and Amazon S3.

Integration with Existing Systems. SPROV2 exposes a very simple API to existing provenance systems. The openProv(metadata) method returns a pointer to the provenance chain for the specified object. The writeProv(provenance record, pointer) method stores the given provenance record into the provenance chain of the object identified by the pointer. The readProv(pointer[,index]) retrieves a provenance record (indicated by index) for the given provenance chain (indicated by the pointer). The closeProv(pointer) method is used to close the opened provenance chain. Finally, the verifyProv(pointer) method can be used to audit a provenance chain.

Using this simple API, it is easy to integrate SPROV2 with existing provenance systems. For the purposes of this demo and poster, we show how SPROV2 can be integrated with the Provenance-Aware Storage System (PASS) [4].

5. SPROV2 DEMONSTRATION

The SPROV2 demonstration will include a short video on the architecture of SPROV2. It will also include an example of how SPROV2 can be integrated into existing provenance architectures. We will also demonstrate visually how SPROV2 works. We will provide detailed performance evaluation of SPROV2 using several benchmarks. Finally, we will present a practical demonstration of SPROV2 integrated into the PASS framework.

The SPROV2 demonstration will require a desktop or a laptop computer with Internet access.

6. CONCLUSION

In this poster and demonstration proposal, we presented the design of SPROV 2.0 – a highly-configurable modular architecture for secure provenance. SPROV2 aims at providing secure provenance functionality to a wide number of existing provenance systems. SPROV2 is designed to be independent of the underlying provenance format and model. We showed guidelines of how this can be integrated with existing provenance systems such as PASS. The configurability of SPROV2 makes it very easy to add different types of security and storage mechanisms, depending upon the application scenario. This will in turn lead to widespread deployment of security assurances to existing provenance frameworks.

7. REFERENCES

[1] U. Braun, A. Shinnar, and M. Seltzer. Securing provenance. In *Proc. of USENIX HotSec*, pages 1–5, July 2008.

[2] R. Hasan, R. Sion, and M. Winslett. The case of the fake Picasso: Preventing history forgery with secure provenance. In *Proc. of USENIX FAST*, pages 1–12, February 2009.

[3] R. Hasan, R. Sion, and M. Winslett. Preventing history forgery with secure provenance. UIUC Department of Computer Science Technical Report. Online at `http://www.ideals.uiuc.edu/handle/2142/13004`, May 2009.

[4] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer. Provenance-aware storage systems. In *Proc. of the USENIX Annual Technical Conference*, pages 43–56, 2006.

[5] Y. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36, September 2005.

[6] Y. Simmhan, B. Plale, and D. Gannon. Karma2: Provenance Management for Data-Driven Workflows. *International Journal of Web Services Research*, 5(2):1–22, 2008.