A Secure and Efficient End-to-End Provenance System (EEPS)

Patrick McDaniel
Computer Science and Engineering
Department
Pennsylvania State University
mcdaniel@cse.psu.edu

Radu Sion, Erez Zadok
Computer Science Department
Stony Brook University
{sion,ezk}@cs.stonybrook.edu

Marianne Winslett
Computer Science Department
University of Illinois,
Champaign-Urbana
winslett@illinois.edu

Abstract

Work on the End-to-End Provenance System (EEPS) was begun in the later summer of 2009. The EEPS effort seeks to explore the three central questions in provenance systems; a) "where and how do I design secure host-level provenance collecting instruments (called provenance monitors)?", b) "how do I extend completeness and accuracy guarantees to distributed systems and computations?", and c) "what are the costs associated with provenance collection". This paper discusses our initial exploration into these issues and posits several challenges to the realization of the EEPS vision.

1 Introduction

Data provenance [1, 2, 3] traces the genesis and subsequent modification of data as it is processed within and across systems. Such information indicates the pedigree of data [4, 5, 6, 7, 8] and enhances, among other functions, system calibration [9], experimental replay [10], auditing [11], fraud and malicious behavior detection [12], and quota and billing management [13]. Because of the immaturity of the underlying technologies, provenance systems are at present largely experimental.

Practical provenance systems use a specialized *recording instrument* to collect information about data processing at run time. The instrument annotates data with information on the relevant operations performed on it. The ordered collection of provenance annotations becomes an unalterable record of data evolution, which we call a *provenance chain* [14, 15]. The scope of provenance is determined by the needs of its users. For example, it is sufficient in some database applications to record the queries that manipulate each table [16, 17, 2, 1, 18, 19]. Thereafter, anyone viewing the data and annotations has a complete record of how the table contents came into being and how it evolved over time. This forensic information is invaluable in repairing failures, understanding application usage, and identifying and undoing malicious behavior.

There have been longstanding calls for provenance in large scale systems systems. A recent report prepared for the chairman and ranking member of the US Senate Committee on Homeland Security and Governmental Affairs [20] highlighted provenance as one of three key future technologies for securing our national critical infrastructure. The report cited a need to ascertain the provenance of sensor data as it is recorded and aggregated in cyber-physical systems such as the electrical grid and SCADA environments. In a different domain, the scientific computing community has long urged the development of provenance systems. Experimenters desire to use provenance to track dependencies between data sources, experiments, and results. Whether tracing sensor data from a pipeline or tracing dependencies between clinical data in a drug trial, it is essential that the provenance be secure against manipulation. Failure to provide such protection leaves the supported system open to misuse. For example, sensor readings could be manipulated to induce or ignore catastrophic failures or mislead drug developers, researchers and agencies governing drug experiments (e.g., the FDA).

Although a number of systems have been developed to record provenance meta-data [21, 10, 22, 23, 11, 24, 14, 25, 26] (some securely), existing systems largely assume a trusted recording instrument. That is, they assume that the systems that are being monitored are (a) trustworthy enough to assert provenance data, and (b) are not compromised. The long history of security has shown that these assumptions are only reasonable in the most restricted of environments, and even there for a short time. Therefore, provenance must be tamper-proof and non-repudiable [27]. A consequence of this requirement is that the applications/systems whose actions are being recorded must not have dominion over the creation or management of provenance.

In recently begun work, we envision an *end-to-end provenance system* (EEPS). EEPS collects provenance evidence at the host level by trusted monitors. Provenance authorities accept host-level provenance data from validated monitors to assemble a trustworthy provenance record. Subsequent users of the data obtain a provenance record that identifies not only the inputs, systems, and applications leading to a data item, but also evidence of the identity and validity of the recording instruments that observed its evolution. Here EEPS addresses the critical open problem of showing that provenance information was recorded accurately *within* and *across* systems.

EEPS introduces the notion of a host-level *provenance monitor*. A provenance monitor acts as the recording instrument that observes the operation of a system and securely records each data manipulation. Like a more general-purpose reference monitor [27], a provenance monitor must preserve several basic properties to ensure accurate recording. Described below, these include tamper-proofness, complete mediation, and simple verification. Note that because the provenance monitor is a host system artifact, further services are needed to coordinate the provenance gathering across systems.

It is our intent to explore the security requirements and performance constraints of practical applications and environments. In this we are studying how policy compliance under regulatory constraints may be implemented in EEPS. We will provide interfaces to these devices that maintain regulatory conditions in the face of potentially adversarial operating systems.

This short paper reviews several of the challenges and designs of EEPS, as well as highlight some of our early progress. We begin in the next section by describing the three main thrusts of the work.

2 EEPS

We are in the initial stages of developing the EEPS system, and are exploring the technical and logistical issues surrounding design alternatives. This current investigation can be loosely divided into three interconnected explorations, as follows:

(1) Host level provenance monitor architecture. The creation of a host level provenance monitor presents several interesting design challenges. A first question is where to place the monitor. We consider two alternatives: an in-kernel provenance monitor and an off-processor monitor. The former requires hooks into the system call interfaces that serve as an application to maintain provenance data, whereas the latter uses secure co-processors or intelligent storage (advanced disk-controllers) as provenance-aware trusted computing bases (TCBs). Figure 1 shows how each of these types of provenance monitors may be deployed within an organization.

The second major design question involves the substance and location of the provenance chain information associated with application data. Developing techniques to store system-level provenance data in ways that will not be resource intensive yet semantically rich enough to support diverse applications is a core requirement. In particular, we explore solutions that avoid costly cryptographic operations on application critical paths and prevent provenance state explosion, which are key to creating a viable system.

Lastly, any provenance system must be built upon a policy facility that flexibly specifies, for a given host/application/data context, what provenance information to record, at what granularity, with what security guarantees. The provenance enforcement policy must be driven by (often distributed) authorities. Identifying those authorities and providing the credentials by which they are validated is essential. Equally important is the investigation of techniques to securely identify and store, among other attributes, process data (unique program and library identity), system integrity state (OS attestations), timestamps, and host and user identity information within the provenance history.

(2) Distributed provenance systems. The next challenge is to extend the reach of the provenance monitor to a system of monitors. Here we seek ti will extend EEPS to support operating within distributed environments. Operation in these environments is complicated by the existence of multiple administrative authorities, coupled with the heterogeneity of platforms and policy. Existing tools do not address these challenges. We will thus explore new architectures and techniques, such as the use of provenance authorities shown in Figure 1, which communicate and disseminate policy across organizational boundaries.

The move from individual hosts to distributed systems spanning administrative domains presents new challenges. The existence of multiple administrative authorities coupled with heterogeneous platforms and policies mandates the exploration of new architectures and techniques building upon the host-level infrastructure.

Consider the version history in a distributed environment that would result if a document were created and subsequently edited and transferred across different autonomous systems' boundaries, with provenance information correctly and indelibly recorded all along the way. We call this a *plausible history* for the resulting document and its chain. We target applications whose provenance integrity needs are met by the following guarantee: *if a document and its associated provenance chain has no plausible version history, it will be detected*. Such applications are common; for example, a retail pharmacy will not accept a shipment of drugs unless it can be shown that the drugs have passed through the hands of certain middlemen. If a criminal wants to sell drugs manufactured by an unlicensed company, he will want to forge a provenance chain that gives the drugs a more respectable history, so that he can move them into the supply chain. This forgery is a condition that a secure distributed provenance system must be able to detect.

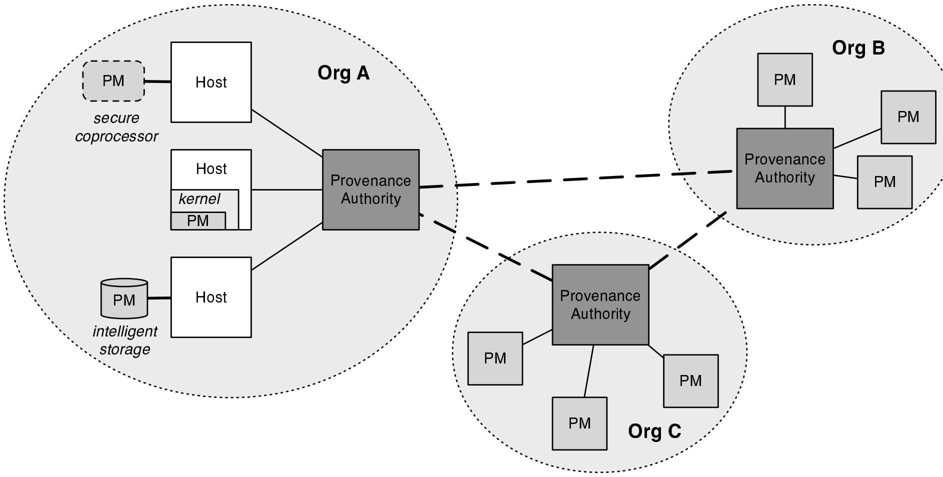

Figure 1: The proposed *end-to-end provenance system* (EEPS), a full-scale distributed provenance architecture, with provenance monitors (labeled PM) placed within the kernel and in trusted hardware. Provenance authorities negotiate cross-organization policy to ensure compliance.

A first requirement design a distributed provenance monitor. We extend host-level provenance monitors with channels for transmitting and receiving distributed provenance information in a manner that is transparent to applications. A second, related goal is to define how distributed protocols and associated policy will be coupled with distributed access control mechanisms. This includes protocols for setting up and maintaining cross-domain communications, as well as communications between provenance monitors and their corresponding domain authorities. We will also leverage work on distributed reference monitors to provide baselines for negotiating trust between provenance authorities and provide for distributed RBAC capable of expressing complex policy.

Distributed systems necessarily require increased provenance expressiveness. In addressing this need, we will consider not just provenance chains, but also the directed acyclic graphs (DAGs) that result from multi-party processing. We will design cryptographic constructions that mitigate costs of these operations, looking initially at “co-provenance” through entangled provenance chains and then designing and implementing DAG constructions. Applying concepts from distributed systems will be essential to making these processes efficient (e.g., virtual synchrony [28]).

(3) Performance/cost modeling and profiling. Collecting and processing provenance can be very costly. However, richer provenance can lead to better security. The choice of how much provenance to collect not just has security implications, but it also affects usability. Moreover, these factors have real dollar costs that can be associated with them, from the cost of storage to hold large provenance data, to total costs of ownership [29], to the opportunity cost of lost computing cycles and potentially reduced user productivity.

In response to this reality, the thirds thrust of the EEPS work is to create an extensive framework to measure performance and other costs. Here we wish to answer, for a given environment and set of request, ”how much does provenance cost?” EEPS is instrumented with sensors profiling of every possible provenance collection decision we build in this project; this would be helpful in performance optimizations and cost modeling. Using collected data, we intend to build cost models to help users decide how much real money they want to spend to collect a certain amount of provenance. This effort can further be divided into four sub-tasks.

We will begin by first profiling the CPU overheads, memory space, network bandwidth, and storage space required for every possible provenance item collected by EEPS. We will build upon Zadok’s extensive experience in high-quality, low-overhead performance tracing and profiling [30, 31, 32, 33, 34, 35]. We will enhance our tools to integrate with the provenance collection LSM methods and report associated space and time costs at a fine granularity. These tools will allow us to pinpoint specific code paths and functions which are responsible for overheads.

Second, we will use the profiling information we collect in two ways: (a) to find out where EEPS adds the most overhead, and focus on optimizing those code paths, and (b) to allow users to make meaningful decisions. We collect and analyze profiling information on a large set of micro- and macro-benchmarks belonging to different scientific domains: bio-informatics, cosmology, data mining, atmospheric modeling, quantum chemistry, fluid dynamics, molecular dynamics, etc.—as well as traditional file system/storage benchmarking—and finally on POSIX compliance test suites.

Third, we will build several cost models that associate real dollar costs with provenance collection and processing. To empower users to make the best provenance-collection decisions, we will associate as many real dollar costs as possible to individual provenance-collection and processing tasks. We will allow users to input and update these costs, and also provide our own cost tables, based on trends and industry best practices. With this cost model, and our exhaustive performance profiles, users could pose “what if” questions to EEPS—reviewing the potential impact on real costs before choosing any provenance-security policy.

Fourth, we will enhance our tools to capture profiles in a distributed fashion. We will transmit these profiles securely because they are provenance in themselves. Once profiles are collected from multiple locations, we will merge them to provide a distributed provenance view. Finally, we will develop and evaluate distributed cost models that incorporate network wide parameters.

2.1 Example Operation

There are many possible use models of a provenance system, each of which dictates different system designs. For illustrative purpose, highlight one possible system design in this section. Here we assume the existence of a trustworthy and tamper-proof smart storage device. This device coordinates the collection of provenance information with other storage devices in the same system.

Consider an example file transfer between two hosts in this hypothetical system illustrated in Figure 2. Documents are kept on disk and provenance chains in the flash of a hybrid drive. (1) A program on Host A initiates the transfer with a system call to the FS. (2) The FS notifies the drive of the transfer. (3) The drives establish a secure tunnel for out-of-band transfer of provenance chains, which are transmitted via a store and forward (SaF) driver in the OS. The tunnel protects the provenance chains from tampering by the untrusted OS. (4) The document transfer occurs as normal. (5) The destination drive verifies the integrity of the document against the provenance chain and adds a new record to indicate the transfer. The entire exchange remains transparent to applications.

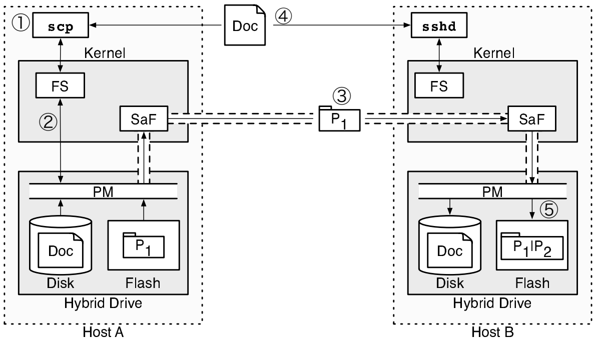

Figure 2: A provenance-enhanced file transfer.

3 Discussion and Conclusions

The value of data maintained by a computing system can only be determined by understanding its origins and pedigree. *Data provenance* provides this information by documenting the entities, systems, and processes that operate on data of interest—in effect providing a historical record of the lifetime of the data and its sources. The generated evidence supports important forensic activities such as data-dependency analysis, error detection and recovery, and auditing and compliance analysis. Although widely sought after in high-end computing systems supporting applications such as bioinformatics, scientific computing and intelligence systems, existing services for data provenance are limited in scope and depth.

The challenges preventing widespread deployment of provenance systems include a lack of services for a) securely and accurately generating provenance information within a computing system, b) securely coordinating that collection within distributed systems, and c) understanding and controlling the storage and computational overheads of managing the provenance information. In this work we will address these challenges through the creation, deployment, and measurement of an *end-to-end provenance system* (EEPS).

Societal trust in e-business and e-government requires accountability. As we move toward becoming an electronic society, as more data will be produced, processed and stored digitally, secure and pervasive provenance assurances will be vital in ensuring public trust and ferreting out corruption and data abuse. We hope this work to constitute a first step in that direction.

References

[1] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Data provenance: Some basic issues. In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (FST TCS)*, pages 87–93, London, UK, 2000. Springer-Verlag.

[2] Peter Buneman, Sanjeev Khanna, and Chiew Tan, Wang. Why and where: A characterization of data

provenance. In *ICDT '01: Proceedings of the 8th International Conference on Database Theory*, pages 316–330, London, UK, 2001. Springer-Verlag.

[3] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36, 2005.

[4] Allison Woodruff and Michael Stonebraker. Supporting fine-grained data lineage in a database visualization environment. In *ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering*, pages 91–102, Washington, DC, USA, 1997. IEEE Computer Society.

[5] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. *The VLDB Journal*, 12(1):41–58, 2003.

[6] Rajendra Bose and James Frew. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[7] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha Nabar, Tomoe Sugihara, and Jennifer Widom. Trio: a system for data, uncertainty, and lineage. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 1151–1154. VLDB Endowment, 2006.

[8] Thomas Heinis and Gustavo Alonso. Efficient lineage tracking for scientific workflows. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1007–1018, New York, NY, USA, 2008. ACM.

[9] R. Cavanaugh, G. Graham, and M. Wilde. Satisifying the Tax Collector: Using Data Provenance as a way to audit data analyses in High Energy Physics. In *Workshop on Data Derivation and Provenance*, October 2002.

[10] Roger S. Barga and Luciano A. Digiampietri. Automatic capture and efficient storage of e-Science experiment provenance. *Concurrency and Computation: Practice and Experience*, 20(5):419–429, April 2008.

[11] Rocio Aldeco-Perez and Luc Moreau. Provenance-based Auditing of Private Data Use. In *BCS International Academic Research Conference, Visions of Computer Science (In Press)*, September 2008.

[12] Francisco Curbera, Yurdaer Doganata, Axel Martens, Nirmal K. Mukhi, and Aleksander Slominski. Business Provenance – A Technology to Increase Traceability of End-to-End Operations. In *On the Move to Meaningful Internet Systems: OTM 2008*, Monterrey, Mexico, November 2008.

[13] Wenchao Zhu, Eric Cronin, and Boon Thau Loo. Provenance-aware Secure Networks. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE 2008)*, Cancun, Mexico, April 2008.

[14] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer. Provenance-Aware Storage Systems. In *Proceedings of the 2006 USENIX Annual Technical Conference*, Boston, MA, USA, Jun. 2006.

[15] Ragib Hasan, Radu Sion, and Marianne Winslett. Introducing Secure Provenance: Problems and Challenges. In *Workshop on Storage Security and Survivability (StorageSS 2007)*, Alexandria, VA, USA, Oct. 2007.

[16] Peter Buneman, Adriane Chapman, and James Cheney. Provenance management in curated databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 539–550, New York, NY, USA, 2006. ACM.

[17] Peter Buneman, Adriane Chapman, James Cheney, and Stijn Vansummeren. A provenance model for manually curated data. In Moreau and Foster [36], pages 162–170.

[18] Nithya N. Vijayakumar and Beth Plale. Towards low overhead provenance tracking in near real-time stream filtering. In Moreau and Foster [36], pages 46–54.

[19] Jennifer Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR '05)*, January 2005.

[20] Martin N. Wybourne, Martha F. Austin, and Charles C. Palmer. National cyber security research and development challenges. Institute for Information Infrastructure Protection, 2009.

[21] Uri Braun, Simson L. Garfinkel, David A. Holland, Kiran-Kumar Muniswamy-Reddy, and Margo I. Seltzer. Issues in automatic provenance collection. In Moreau and Foster [36], pages 171–183.

[22] Luc Moreau, Paul Groth, Simon Miles, Javier Vazquez-Salceda, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, and Laszlo Varga. The provenance of electronic data. *Commun. ACM*, 51(4):52–58, 2008.

[23] Martin Szomszor and Luc Moreau. Recording and reasoning over data provenance in web and grid services. In *International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE'03)*, volume 2888 of *Lecture Notes in Computer Science*, pages 603–620, Catania, Sicily, Italy, November 2003.

[24] Victor Tan, Paul Groth, Simon Miles, Sheng Jiang, Steve Munroe, Sofia Tsasakou, and Luc Moreau. Security issues in a soa-based provenance system. In Moreau and Foster [36], pages 203–211.

[25] Ragib Hasan, Radu Sion, and Marianne Winslett. The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST 2009)*, San Francisco, CA, USA, February 2009.

[26] Adriane P. Chapman, H. V. Jagadish, and Prakash Ramanan. Efficient provenance storage. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 993–1006, New York, NY, USA, 2008. ACM.

[27] J. P. Anderson. Computer security technology planning study, volume II. Technical Report ESD-TR-73-51, Deputy for Command and Management Systems, HQ Electronics Systems Division (AFSC), L. G. Hanscom Field, Bedford, MA, October 1972.

[28] K.P. Birman. The process group approach to reliable distributed computing. *Communications of the ACM (CACM)*, 16(12), December 1993.

[29] D. Simpson. Corral your storage management costs. *Datamation*, 43(4):88–98, 1997.

[30] A. Traeger, N. Joukov, C. P. Wright, and E. Zadok. A nine year study of file system and storage benchmarking. *ACM Transactions on Storage (TOS)*, 4(2):25–80, May 2008.

[31] A. Traeger, I. Deras, and E. Zadok. DARC: Dynamic analysis of root causes of latency distributions. In *Proceedings of the 2008 International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2008)*, pages 277–288, Annapolis, MD, June 2008. ACM.

[32] C. P. Wright, J. Dave, and E. Zadok. Cryptographic File Systems Performance: What You Don't Know Can Hurt You. In *Proceedings of the Second IEEE International Security In Storage Workshop (SISW 2003)*, pages 47–61, Washington, DC, October 2003. IEEE Computer Society.

[33] N. Joukov, A. Traeger, R. Iyer, C. P. Wright, and E. Zadok. Operating system profiling via latency analysis. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI 2006)*, pages 89–102, Seattle, WA, November 2006. ACM SIGOPS.

[34] N. Joukov, T. Wong, and E. Zadok. Accurate and efficient replaying of file system traces. In *Proceedings of the Fourth USENIX Conference on File and Storage Technologies (FAST '05)*, pages 337–350, San Francisco, CA, December 2005. USENIX Association.

[35] A. Aranya, C. P. Wright, and E. Zadok. Tracefs: A file system to trace them all. In *Proceedings of the Third USENIX Conference on File and Storage Technologies (FAST 2004)*, pages 129–143, San Francisco, CA, March/April 2004. USENIX Association.

[36] Luc Moreau and Ian T. Foster, editors. *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, volume 4145 of *Lecture Notes in Computer Science*. Springer, 2006.