Towards a Secure and Efficient System for End-to-End Provenance

Appears in the Proceedings of the Second USENIX Workshop on Theory and Practice of Provenance (TaPP 2010)

Patrick McDaniel, Kevin Butler,
Steve McLaughlin
Computer Science and Engin. Department
Pennsylvania State University

Radu Sion, Erez Zadok
Computer Science Department
Stony Brook University

Marianne Winslett
Computer Science Department
University of Illinois, Urbana-Champaign

Abstract

Work on the End-to-End Provenance System (EEPS) began in the late summer of 2009. The EEPS effort seeks to explore the three central questions in provenance systems: (1) "Where and how do I design secure host-level provenance collecting instruments (called provenance monitors)?"; (2) "How do I extend completeness and accuracy guarantees to distributed systems and computations?"; and (3) "What are the costs associated with provenance collection?" This position paper discusses our initial exploration into these issues and posits several challenges to the realization of the EEPS vision.

1 Introduction

Data provenance [11, 12, 26] traces the genesis and subsequent modification of data as it is processed within and across systems. Such information indicates the pedigree of data [1, 7, 15, 19, 34] and enhances, among other functions, system calibration [13], experimental replay [5], auditing [2], fraud and malicious behavior detection [16], and quota and billing management [37]. 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 runtime. 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 called a *provenance chain* [17, 24]. The scope of provenance is determined by the needs of its users. For example, it is sufficient in some database applications to record only the queries [9–12, 32, 33]. Thereafter, anyone viewing the data and annotations has a complete record of how the table contents came into being and how they evolved over time. This forensic information is invaluable in repairing failures, understanding application usage, and identifying and undoing malicious behavior.

There have been long-standing calls for provenance in large-scale systems. A recent report prepared for the chairman and ranking member of the US Senate Committee on Homeland Security and Governmental Affairs [36] 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 smart-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, readings could be manipulated to induce or ignore catastrophic failures or mislead clinicians and regulators (e.g., the FDA).

Although a number of systems have been developed to record provenance meta-data [2, 5, 8, 14, 23, 24, 28, 29] (some securely), existing systems largely assume that the recording instrument is inherently trustworthy. That is, they assume that the systems being monitored are (a) trustworthy enough to assert their own provenance data, and (b) not compromised. However, the long history of security has shown that these assumptions are only reasonable in the most restricted of environments, and even there, only for a short period of time. Thus, a stronger set of security requirements are needed for provenance to be tamper-proof and non-repudiable [3]. The definition and enforcement of these requirements constitute the first major challenge of the work proposed here.

In this 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.

To address the need for stronger provenance security guarantees, 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 the well-known reference monitor concept for the enforcement of security policies [3], 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.

The key depature of this work from past efforts in provenance is in its focus; we are exploring the trust, security, and performance constraints of practical provenance applications and environments. In this, we are studying how policy compliance under regulatory constraints may be implemented in EEPS. We propose interfaces to these devices that maintain regulatory conditions in the face of potentially adversarial operating systems. This work faces three key challenges:

1. Provenance collection at the host level must meet the security guarantees of a reference monitor. We propose the host level provenance monitor as a method for achieving these guarantees.
2. The aggregation of provenance records must be kept secure and verifiable across domains with different security policies. We use the notion of a *plausible history* as a method for tracking a data item's history of domain traversals.
3. As the resources spent on provenance are pure overhead, it must be collected, stored, and audited in the most efficient means possible. For this challenge, we leverage our previous work with optimized cryptographic constructions for provenance data [18].

This short position paper reviews several of the challenges and designs of EEPS, and highlights 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 divided into three interconnected explorations loosely parallel to the challenges outlined above:

(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. Adopting a trust model similar to systems like PASS [24], 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 host level provenance monitor should enforce the classic reference monitor guarantees of complete mediation of relevant operations, tamper-proofness of the monitor itself, and basic verification of correct operation. For the purpose of the provenance monitor, we define these as follows. First, a provenance monitor should mediate all provenance-relevant operations, whatever these may be for a given application. Second, the provenance monitor must be isolated from the subjects operating on provenance-enhanced data, e.g. the OS kernel or storage device. Finally, the provenance monitor should be designed to allow for simple verification of its behavior.

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 are exploring solutions that avoid costly cryptographic operations on application critical paths and prevent provenance state explosion.

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, and 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 to extend EEPS to support 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 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 author-

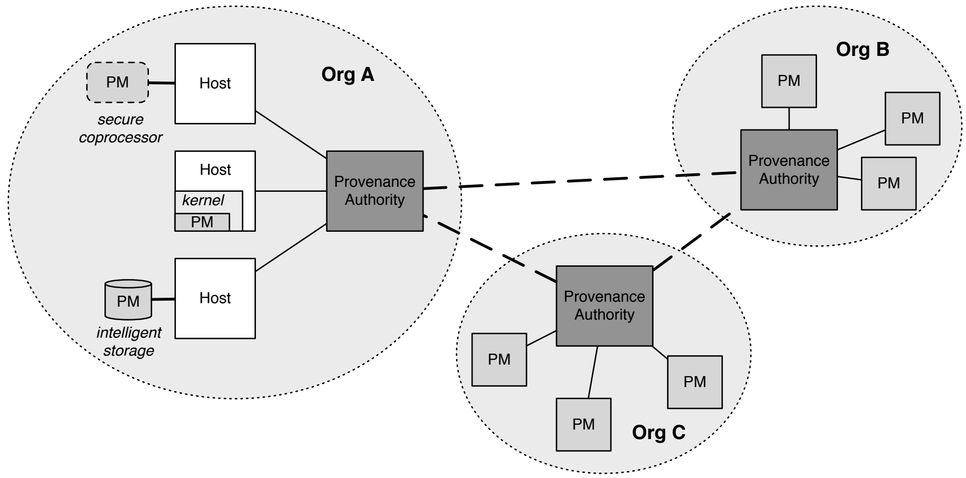

Figure 1: The end-to-end provenance system (EEPS) distributed provenance architecture, with provenance monitors (labeled PM) placed within the kernel and in trusted hardware. Provenance authorities negotiate cross-organization policy to ensure compliance.

ities 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.

As a first design requirement for a distributed provenance monitor, we are extending 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 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 consider not just provenance chains, but also the directed acyclic graphs (DAGs) that result from multi-party processing. We are designing 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 [6]).

(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 [27], to the opportunity cost of lost computing cycles and potentially reduced user productivity.

In response to this reality, the third 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 are profiling the CPU overheads, memory space, network bandwidth, and storage space required for every possible provenance item collected by EEPS [4, 20, 21, 25, 30, 31, 35]. We are enhancing our profiling apparatus to integrate with the provenance collection LSM methods and report associated space and time costs at a fine granularity. These tools allow us to pinpoint specific code paths and functions which are responsible for overheads.

Second, we 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 are collecting and analyzing 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 are building 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 are enhancing our tools to capture profiles in a distributed fashion. These profiles will be securely transmitted because they are provenance in themselves. Once profiles are collected from multiple locations, they can be merged to provide a distributed provenance view. Finally, we are developing and evaluating 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, we highlight our current preliminary system design. 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. Further modifications to the OS may allow even greater transparency; for example, a small modification to the host's USB stack can allow for the issuance of trusted commands directly to the disk, which can validate the integrity state of the host prior to allowing provenance operations to occur. The disk can then uniquely identify the host performing read or write operations even if the host is offline; an administrator can later retrieve the versioned history from the disk.

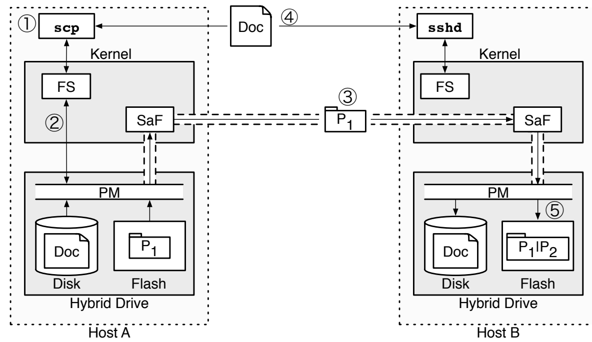

Figure 2: A provenance-enhanced file transfer.

3 Discussion and Conclusions

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 propose addressing these challenges through the creation, deployment, and measurement of an *end-to-end provenance system* (EEPS).

Note that we have yet to explore the security and costs associated with the consumption of provenance data. Issues such as privacy and confidentiality and the inherent information leakage associated with its collection are daunting. Applications of provenance such as regulatory compliance carry with it provisions not only for monitoring, but also for the correct handling of the provenance data itself. The exposure of relationships between organizations and data is often as damaging as its corruption. This is an open area of research we will embrace as application requirements arise from the use of EEPS.

Ultimately, societal trust in increasingly distributed information systems such as e-business and e-government requires better tools for accountability. As we move 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.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant numbers CCF-0937944, CNS-0643907, CCF-0937833, CNS-0845192, CNS-0708025, IIS-0803197, CNS-0716608, CNS-0614784, CCF-0621463, and CCF-0937854, as well as an IBM Faculty award, and support from Xerox and Network Appliance.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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 [22], pages 171–183.

[9] 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.

[10] Peter Buneman, Adriane Chapman, James Cheney, and Stijn Vansummeren. A provenance model for manually curated data. In Moreau and Foster [22], pages 162–170.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] P. Sehgal, V. Tarasov, and E. Zadok. Evaluating Performance and Energy in File System Server Workloads extensions. In *Proceedings of the Eighth USENIX Conference on File and Storage Technologies (FAST '10)*, San Jose, CA, February 2010. USENIX Association. To appear.

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

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

[28] 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.

[29] 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 [22], pages 203–211.

[30] 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.

[31] 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.

[32] Nithya N. Vijayakumar and Beth Plale. Towards low overhead provenance tracking in near real-time stream filtering. In Moreau and Foster [22], pages 46–54.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.