Building Regulatory Compliant Storage Systems

Zachary N. J. Peterson
The Johns Hopkins University
3400 N. Charles St.
Baltimore, MD, USA
zachary@cs.jhu.edu

Randal Burns
The Johns Hopkins University
3400 N. Charles St.
Baltimore, MD, USA
randal@cs.jhu.edu

1. PROJECT GOALS

In the past decade, informational records have become entirely digital. These include financial statements, health care records, student records, private consumer information and other sensitive data. Because of the delicate nature of the data these records contain, Congress and the courts have begun to recognize the importance of properly storing and securing electronic records. Examples of legislation include the Health Insurance Portability and Accountability Act (HIPAA) of 1996, the Gramm-Leach-Bliley Act (GLBA) of 1999, and the more recent Federal Information Security Management Act (FISMA) and Sarbanes-Oxley Act (SOX) of 2002. Altogether, there exist over 4,000 acts and regulations that govern digital storage, all with a varying range of requirements for maintaining electronic records.

Some legislation requires that systems provide confidentiality through encrypted storage and data transmission. Some legislation requires an auditable trail of changes made to electronic records that are accessible in real-time. Other legislation sets limits on the amount of time an organization may be liable for maintaining their electronic data. The list of requirements is comprehensive, however, distilling them into product requirements and implementing a system to meet all mandates is non-trivial. It is the goal of this project to make sense of the large body of requirements and develop technical solutions that help organizations manage their data and comply with federal regulations.

2. PROJECT HIGHLIGHTS

We have three major findings to date. We have developed and released an open-source versioning file system designed for regulatory compliance, added secure-deletion to the file system for privacy and compliance with legislation that mandates deletion, and developed a digital audit model that provides a secure record of how data changes over time.

We developed and released the ext3cow versioning file system [9] that addresses the mandated versioning and auditability requirements. The file system provides a *time-shifting* interface that permits a real-time and continuous view of data in the past. Ext3cow has hundreds of users from over one hundred different countries. It has served as tool for other academic research projects, and for us, ext3cow has continued to be a useful foundation for exploring technical solutions to other regulatory storage problems.

While versioning file systems are quickly being adopted by medical and commercial institution wishing to become federally compliant, most existing systems that advertise themselves as "compliant" overlook fine-grained secure deletion as an essential requirement. Secure deletion is the act of removing digital information from a storage system so that it can never be recovered. Fine-grained refers to removing individual files or versions of file, while preserving all other data in the system. We believe the reticence to integrate secure deletion into existing storage systems derives from the inefficiency and of current deletion techniques when applied to versioning systems. Our second major contribution is the development of two methods for the efficient secure deletion of individual versions of a file that are orders of magnitudes faster than existing techniques [10]. The first method uses all-or-nothing (AON) encryption [11] to create a small *stub* that, when *securely overwritten* [8], permanently deletes the corresponding data. The second technique uses random key generation to generate a stub, similar to *key disposal* [5]. Both techniques provide authenticated encryption [3], which provides both data privacy and authentication. To our knowledge, we are the first disk file system to adopt authenticated encryption. We collect and store stubs contiguously so that overwriting a small block of stubs deletes a large amount of file data, even when file data are non-contiguous. Our methods do not complicate key management.

Another challenge present in compliant storage systems lies in verifying the authenticity of data, *i.e.* making data safe from tampering and providing a proof of compliance. Both auditors and companies are required by SOX to keep strong audit trails on electronic records; for both parties to prove compliance and for auditors to ensure the accuracy of the information on which they report. A "strong" audit trail is a verifiable, persistent record of how and when data have changed. Our third contribution is a system for the verification of version histories in file systems based on generating message authentication codes (MACs) for versions and archiving them with a third party. A file system commits to a version history when it presents the MAC to the third party. At a later time, a version history may be verified by an auditor. The file system is challenged to produce data that matches the MAC, ensuring that the system's past data have not been altered. The MACs reveal nothing about the data contents and published MACS may even be stored publicly. Our design goals include minimizing the network, computational, and storage resources used in the publication of data and the audit process. To this end, we employ parallel message authentication codes [1, 2, 4] that allow MACs to be computed incrementally – based only on data that have changed from the previous version. Sequences of versions may be verified by computing a MAC for one version

and incrementally updating the MAC for each additional version, performing the minimum amount of I/O. With incremental computation, a natural trade-off exists between the amount of data published and the efficiency of audits.

3. PROJECT STATUS AND ACTIVITIES

Throughout the course of this research, we have been able to effectively collaborate with other researchers at Johns Hopkins and aboard. With secure deletion, we worked with co-PI Avi Rubin and Adam Stubblefield on constructing secure deletion algorithms with authenticated encryption for a version file system. For digital audit trails, we collaborated with co-PI Giuseppe Ateniese and Steve Bono to help in formulating efficient authentication algorithms in a versioning environment.

All projects are being implemented, not simulated, in the ext3cow file system. The ext3cow version file system and an implementation of secure deletion is available for download at *www.ext3cow.com*. The goal is to provide an open-source implementation of a storage system that meets the requirements of electronic records legislation. This will make compliance available to all, minimizing the costs involved.

We have shared our findings with the public and academic community through publications and conference presentations. Details of the ext3cow file system have been published in the journal, *ACM Transactions on Storage* [9]. Our secure deletion work was most recently published and presented at the USENIX File And Storage Technology (FAST) in December [10]. We have also presented our work on authenticators for versioning file systems at the ACM CCS Workshop on Storage Security and Survivability (StorageSS) [6]. A comprehensive presentation of our research to date was recently made to Digital Archives consortium at the Library of Congress in December of 2005.

4. FUTURE WORK

The development of these tools has left us with many opportunities for continued research in this field. Activities for the next project year include:

- **Secure deletion in managed environments:** We are expanding secure deletion constructs to delete data even when it has been replicated across multiple sites (for backup). We are also developing methods so that users may delete data without physical access to the media, *e.g.* a patient could use this construct to delete portions of her medical records from storage owned by doctors and insurance companies.
- **Unification of secure deletion algorithms:** We are collaborating with Giovanni Di Crescenzo of Telcordia to prove the security of our algorithms for secure deletion and create a unified framework among our work and his work on erasable memories [7].
- **Implementation and release of digital audits:** In the next year, we will complete our implementation of verifiable audit trails in ext3cow and make this available to the public through an open-source license.
- **Approximate MACs:** We have begun an investigation of security constructs that allow for digital audits to be conducted by sampling only portions of the data in a system. Such a construct would greatly improve the efficiency of audits and provide probabilistic guarantees that data have not been modified or lost.

5. REFERENCES

[1] M. Bellare, O. Goldreich, and S. Goldwasser. Incremental cryptography and application to virus protection. In *Proceedings of the ACM Symposium on the Theory of Computing*, pages 45–56, May-June 1995.

[2] M. Bellare, R. Guérin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Advances in Cryptology - Crypto'95 Proceedings*, volume 963, pages 15–28, 1995. Lecture Notes in Computer Science.

[3] M. Bellare and C. Namprempre. Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm. In *Advances in Cryptology - Asiacrypt'00 Proceedings*, volume 1976, 2000. Lecture Notes in Computer Science.

[4] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *Advances in Cryptology - Eurocrypt'02 Proceedings*, volume 2332, pages 384 – 397. Springer-Verlag, 2002. Lecture Notes in Computer Science.

[5] D. Boneh and R. Lipton. A revocable backup system. In *Proceedings of the USENIX Security Symposium*, pages 91–96, July 1996.

[6] R. Burns, Z. Peterson, G. Ateniese, and S. Bono. Verifiable audit trails for a versioning file system. In *Proceedings of the ACM CCS Workshop on Storage Security and Survivability*, November 2005.

[7] G. Di Crescenzo, N. Ferguson, R. Impagliazzo, and M. Jakobsson. How to forget a secret. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, volume 1563, pages 500–509. Springer-Verlag, 1999. Lecture Notes in Computer Science.

[8] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In *Proceedings of the USENIX Security Symposium*, pages 77–90, July 1996.

[9] Z. Peterson and R. Burns. Ext3cow: A time-shifting file system for regulatory compliance. *ACM Transcations on Storage*, 1(2):190–212, 2005.

[10] Z. N. J. Peterson, R. Burns, J. Herring, A. Stubblefield, and A. Rubin. Secure deletion for a versioning file system. In *Proceedings of the USENIX Conference on File And Storage Technologies (FAST)*, pages 143–154, December 2005.

[11] R. L. Rivest. All-or-nothing encryption and the package transform. In *Proceedings of the Fast Software Encryption Conference*, volume 1267, pages 210–218, 1997. Lecture Notes in Computer Science.