Database Forensics in the Service of Information Accountability

Kyriacos E. Pavlou
Department of Computer Science
The University of Arizona
P.O. Box 210077
Tucson, AZ 85721–0077
kpavlou@cs.arizona.edu

ABSTRACT

Regulations and societal expectations have recently expressed the need to mediate access to valuable databases, even by insiders. At one end of the spectrum is the approach of *restricting access to information* and on the other that of *information accountability*. The focus of the proposed work is effecting information accountability of data stored in databases. One way to ensure appropriate use and thus end-to-end accountability of such information is tamper detection in databases via a *continuous assurance technology* based on cryptographic hashing. In our current research we are working to show how to develop the necessary approaches and ideas to support accountability in high-performance databases. This will include the design of a reference architecture for information accountability and several of its variants, the development of forensic analysis algorithms and their cost model, and a systematic formulation of forensic analysis for determining when the tampering occurred and what data were tampered with. Finally, for privacy, we would like to create mechanisms for allowing as well as (temporarily) preventing the physical deletion of records in a monitored database. In order to evaluate our ideas we will design and implement an integrated tamper detection and forensic analysis system. This work will show that information accountability is a viable alternative to information restriction for ensuring the correct storage, use, and maintenance of databases.

1. INTRODUCTION

Corporate abuses by Enron and WorldCom have given rise to recent regulations which require many corporations to ensure trustworthy long-term retention of their routine business documents. The US alone has over 10,000 regulations [11] that mandate how business data should be managed [6, 30], including the Health Insurance Portability and Accountability Act: HIPAA [8], the Sarbanes-Oxley Act [26], the 1997 U.S. Food and Drug Administration (FDA) regulation "21 CFR Part 11" [9], and other laws requiring audit logs.

Due to widespread news coverage of collusion between auditors and the companies they audit, a fact which helped accelerate passage of the aforementioned laws, there has been interest within the file systems and database communities in built-in mechanisms to detect or even prevent tampering.

Compliant records are those required by myriad laws and regulations to follow certain "processes by which they are created, stored, accessed, maintained, and retained" [11]. It is common to use Write-Once-Read-Many (WORM) storage devices to preserve such records [31]. The original record is stored on a write-once optical disk. As the record is modified, all subsequent versions are also captured and stored, with metadata recording the timestamp, optical disk, filename, and other information on the record and its versions.

This way of ensuring record compliance, or information compliance in general, can be described as *information restriction* which entails rendering retained records immutable and controlling access to them. This approach appears to be the prevailing viewpoint for achieving privacy and security. We feel that the means of addressing security and compliance should be viewed as constituting a spectrum. If one asserts that information restriction lies at one end of the spectrum and then question which inevitably arises is what lies at the other end? In a recent article Weitzner et al. [29] argue that access control and cryptography are not capable of protecting information privacy and that there is a true dearth of mechanisms for addressing effectively information leaks. They propose that as an alternative information accountability "must become a primary means through which society addresses appropriate use" [29]. *Information accountability*, in this context, assumes that information should be transparent so as to easily determine whether a particular use is appropriate under a given set of rules.

We assert that a shift towards information accountability presents valuable advantages over information restriction in the particular area of correct storage, use, and maintenance of databases. An information accountability approach to database security is cheaper, can protect against a variety of threats (including insider threats), can successfully deal with the aftermath of information restriction failure and can render complex security problems tractable.

Information accountability is by no means a new idea. In fact it has been tried and tested successfully since ancient

times. The ancient Egyptians (c. 3000 B.C.E.) used lumps of clay to create tamper-indicating seals. Even though it was easy to break the seal and gain access to forbidden information, it was equally easy to detect tampering and hold the perpetrator accountable [16].

Information accountability has been applied in modern times and in many varied areas such as the Fair Credit Reporting Act of 1970 [1] and copyright protection via Creative Commons licensing [7]. Under the Fair Credit Reporting Act strict rules are imposed not on the collection of data or their analysis but on the way the data or the result of the analysis (e.g., credit report) can be used. For example, consumer reports can be used to determine the "consumer's eligibility for a [...] benefit granted by a governmental instrumentality" or for a "business transaction that is initiated by the consumer" but not for marketing purposes [1]. Furthermore, the data are *transparent*, that is, the consumers are allowed access to all the data the agencies maintain about them. Any agency using credit reports to make a decision is *accountable* for its action because it must be able to justify the basis of the decision based on specific details found in the report.

Creative Commons establishes a set of copyright licenses which do not attempt to prevent the lawful use of works they protect by using technology, but rather set forth rules regulating the use of the works [7]. For example, one license named *Attribution-NoDerivs* "allows for redistribution, commercial and non-commercial, as long as [the work] is passed along unchanged and in whole, with credit to [the author]" [7]. Once again, the data is transparent and the emphasis is on holding the consumers accountable under the set of rules set forth by the license.

Lest a conclusion be drawn that accountability is only appropriate for information with a low associated risk, we offer as an example the widespread use of simple wire-loops as tamper-indicating seals for nuclear safeguarding [16].

A related concept is *continuous assurance technology*, defined as "technology-enabled auditing which produces audit results simultaneously with, or a short period of time after, the occurrence of relevant events" [3]. This concept is crucial because our research employs it to achieve a meaningful operationalization of information accountability.

In our current research we are working to show that information accountability can effectively realize appropriate use (i.e., guarantee no unauthorized modifications—insertions, deletions, updates) in high-performance databases. We will achieve this by developing an approach to tamper detection that provides continuous assurance, accommodates shredding and litigation holds, and includes a series of forensic analysis algorithms. An evaluation via a prototype implementation will demonstrate that this approach is a viable alternative to information restriction.

2. RELATED WORK

Each of the following three subsections feature published work describing the origin and evolution of audit log compliance, database tamper detection, and forensics.

2.1 Audit Log Compliance

In the context of audit log compliance, a "record" is a version of a document. Within a document/record management system (RMS), a DBMS is often used to keep track of the versions of a document and to move the stored versions along the storage hierarchy. Examples of such systems are the EMC Centera Compliance Edition Content Addressed Storage System[1], the IBM Information Archive [2], and NetApp's SnapLock Compliance[3]. These systems utilize magnetic disks, optical drives, and tape to provide WORM storage of compliant records. They are implementations of *read-only file systems* (also termed *append-only*), many of which have been presented in the research literature [10, 19, 22]. The file systems use cryptographic signatures to guarantee document integrity.

Hsu and Ong have proposed an end-to-end perspective ("from the proper preservation of all of the records to the subsequent delivery of the relevant records to an agent seeking the proof") for establishing trustworthy records, through *fossilization* [15]. The idea is that once a record is stored in the RMS, it is "cast in stone" and thus not modifiable. An index allows efficient access to such records, typically stored in WORM storage. Subsequently, they achieved fossilization of the index itself [31].

Agrawal et al. suggest that database technology can be used to assist compliance with the internal control provisions described in Sections 302 and 404 of the Sarbanes-Oxley Act [26]. The approach taken in this paper is an example of continuous assurance technology [3]. We adopt this approach in our research.

All the work presented above deal with records of coarse granularity, for instance, spreadsheets. None deal with fine granularity records like database tuples, which is what our research focuses on.

2.2 Database Tamper Detection or Prevention

We assert that every database tuple is a record, to be managed. This creates a two-fold challenge since tuples, unlike records in an RMS, are light-weight objects and tend to change rapidly in a high-performance transaction database. Achieving the functionality of tracked, tamper-free records with the performance of a DBMS is challenging.

The first work to show that records management could be effectively merged with conventional databases was that by Barbará et al. on using checksums to detect data corruption [4]. By computing two checksums in different directions and using a secret key, they were able to dramatically increase the intruder's required effort to tamper the database.

An example of a WORM-based, long-term high-integrity retention technique for fine granularity business records is the *log-consistent compliant database architecture* (LDA),

[1] `http://www.emc.com/products/detail/hardware/centera.htm`
[2] `http://www-03.ibm.com/systems/storage/disk/archive/`
[3] `http://www.netapp.com/us/products/protection-software/snaplock.html`
(all accessed April 1, 2011)

which extends immutability to relational tuples [20, 21]. This system stores a database snapshot on WORM at audit time, while an additional *compliance log* stored on WORM records database modifications. The snapshot plus the compliance log lets an auditor verify if a new database state is compliant.

A more efficient architecture is the *transaction log on WORM* (TLOW) approach for supporting long-term immutability of relational tuples [14]. TLOW stores the current database instance in ordinary storage and the transaction log on WORM storage, while dispensing with the compliance log altogether. The audit process uses hash values representing the data rather than the data themselves. An audit is successful if the hash from the old database snapshot plus the hash of all the new tuples introduced in the transaction log match the hash of the current database instance. TLOW includes only a rudimentary forensic analysis technique.

Employing a DBMS in order to detect or prevent data tampering is a step in a right direction. Nevertheless, with the exception of TLOW, none of these approaches deals systematically with database forensics.

2.3 Database Forensics

Basu presents a method of forensic tamper detection and localization of corrupted data in SQL Server [5]. The solution is based on creating an interwoven chain of hash values used by a detection algorithm to determine if a particular audit log table row is modified, inserted, or deleted. Although this method has advantages (e.g., no special deployment strategy required), it suffers from the use of non-cryptographically strong hash functions, and the limited forensic strength of the detection algorithm.

An entirely different approach to tamper detection that can encompass database forensics is *database watermarking*. In general, digital watermarking for the purpose of integrity verification is called *fragile watermarking* whereas *robust watermarking* is used for copyright protection. Examples of robust watermarking schemes for databases include work by Agrawal and Kiernan [2], and Sion et al. [27].

Guo, Jajodia, Li, and Liu formulated a fragile watermarking scheme for databases [12, 17]. Their scheme is based on a watermark that is *invisible* (watermark does not distort data) and can be *blindly verified* (original unmarked relation is not required for verification). The watermark depends on the hash values of the tuples' primary key value, their attribute values, and a secret embedding key. During verification, the extracted watermark indicates the locations of alterations. This scheme wrests control of tuple placement from the DBMS and suffers from false positives.

All these database watermarking forensic techniques are valuable but tend to be evaluated by a probabilistic analysis. Furthermore, watermark embedding techniques may distort data and have a high overhead. Although watermarking techniques can provide spatial bounds on the tampered data, they provide no temporal bounds on when the tampering transpired. These are concerns we wish to avoid.

3. THE PROBLEM

We identify problem areas and needs, which when resolved will render information accountability-based security cheaper, protect against a variety of threats, successfully deal with the aftermath of information restriction failure, and render complex security problems tractable. This will demonstrate the advantages of information accountability over information restriction in the particular area of correct storage, use, and maintenance of databases.

As we have seen, there are two basic approaches to achieving information accountability in databases: fragile watermarking and cryptographic hashing. Each has its own benefits and challenges. In this proposed work we focus on the latter approach. Cryptographic techniques coupled with a carefully-considered architectural design solve one part of the information accountability puzzle: detecting tampering [28]. In our current work we address other parts of the puzzle that are still open.

Within the domain of cryptographic hashing techniques no high-level reference architecture exists that ensures this concept of information accountability. Moreover, there is a distinct dearth in the literature of the ways to develop, evaluate, and generalize forensic analysis algorithms. These forensic analysis algorithms must work efficiently within this framework and provide spatial and temporal bounds on a corruption event once tampering has been detected. Apart from threat analyses of specific systems no taxonomy of corruption types exists. By developing this taxonomy of corruption types in conjunction with a taxonomy of forensic analysis algorithms, one can systematically formulate the process of forensic analysis in databases, something which is also absent from current research.

In order to achieve enterprise-wide information accountability we must provide an implementation of forensic analysis algorithms which can identify the different types of corruption identified in the corruption type taxonomy. The algorithms need to be integrated into a working prototype system with enhanced capabilities based on the reference architecture developed.

4. APPROACH

The goal is to devise a set of ideas and concepts which can be used in high-performance relational databases to ensure end-to-end information accountability using cryptographic hashing techniques.

This conceptual framework is constructed by a careful consideration of core principles rather than by devising "add-on" solutions, the latter being a common, rather ad hoc, approach. This allows us to construct sufficient defenses against the most common security threats so that the cost of accountability outweighs the gains from a successful tampering.

Information accountability in this context of database compliance can be thought to apply to two usually different parties. The system has to be able to hold accountable the people who were charged with curating the database and failed to do so and also hold accountable those responsible for the unauthorized use of the database. In some cases these parties can coincide, in which case insider corruption is much

harder to detect. We also address this difficult case.

As we have seen the first mechanisms developed towards information accountability were applied to audit log security. Audit log security is one component of more general *record management systems* that track documents and their versions, and ensure that a previous version of a document cannot be altered. As an example, *digital notarization services* such as Surety (`www.surety.com`), when provided with a digital document, generate a *notary ID* through secure one-way hashing, thereby locking the contents and time of the notarized documents [13]. Later, when presented with a document and the notary ID, the notarization service can ascertain whether that specific document was notarized, and if so, when. Such approaches cannot be applied directly to high-performance databases. A copy of the database cannot be versioned and notarized after each transaction. Instead, audit log capabilities must be moved into the DBMS.

A previous paper by our research group on tamper detection accomplished exactly that. It also removed one assumption, that the system could keep a secret key that would not be seen by insiders [28]. That paper proposed an innovative approach in which cryptographically-strong one-way hash functions prevent an intruder, including an auditor or an employee or even an unknown bug within the DBMS itself, from silently corrupting the audit log [18, 28]. This is accomplished by cumulatively hashing all data manipulated by transactions as they become available to the system. This generates a hash chain which at each time instant its value represents all the data in the database. A module called a *notarizer* periodically performs a notarization by sending that hash value, as a digital document, to an external digital notarization service, and obtaining a notary ID. The notary ID returned along with the initially computed hash values are stored in a separate smaller database. This database, termed the *secure master database*, is assumed to exist in a different physical location from the database under audit. When at a later point in time the validity of the monitored database must be checked, a *validator* application rescans the monitored database, hashes the scanned data and sends, to the notarization service, the new hash value along with the previously obtained notary ID. The notarization service then uses the notary ID to retrieve the corresponding hash value stored during notarization, and checks if the old and the new hash values are consistent. If not, then the monitored database has been compromised.

We go beyond our previous work to introduce a two-dimensional spectrum which captures the nature of the data and methodology used to ensure their appropriate use. This spectrum is comprised of a vertical axis which specifies the granularity of the data that must be protected. On one extreme we find the notion of a file while on the opposite end that of a single tuple. The other axis characterizes the methodology. On one extreme we find information restriction while on the other information accountability. Once we have established the coordinates of the proposed research as being (information accountability, tuple) we develop a high-level reference architecture for databases which ensures the concept of information accountability. This is followed by the creation of the fundamental algorithmic tools used in tamper detection and forensic analysis. Specifically, we design several forensic analysis algorithms differing in forensic strength and efficiency. We propose to further generalize these algorithms so that spatial bounds of the detected corruption can be expressed in multiple ways beyond just commit time. We develop a taxonomy of forensic algorithms which along with a taxonomy of corruption types eventually culminate in the creation of forensic analysis protocols. These forensic protocols constitute a systematic formulation of forensic analysis.

In order to address issues of privacy and liability we provide mechanisms by which records can be physically removed from the database (shredding). The litigation hold mechanisms also proposed are not jeopardized by shredding.

Finally, we evaluate our proposed architecture and forensic tools by designing, implementing, and analyzing a prototype system titled Dragoon (Database foRensic Analysis safeGuard Of arizONa) that encompasses concepts put forth first in our research. Dragoon is evaluated using both quantitative and qualitative criteria. We use measures like space and running time efficiency, algorithm cost, completeness in terms of range of security risks addressed, understandability to user, ease-of-use, scalability, security, lightweightness, and amenability to hardware solutions. In doing so we show that information accountability is a viable and high performance alternative to information restriction for ensuring appropriate use in databases.

5. RESULTS ACHIEVED THUS FAR

In previous papers we developed fundamental algorithmic tools used in tamper detection and forensic analysis. Specifically, we developed several successively more sophisticated forensic analysis algorithms, including Monochromatic [23], RGB [23], RGBY [24], Tiled-Bitmap [25], and a3D [24]. These forensic algorithms determine when the tampering occurred, and what data was tampered with. A schematic representation, called the *corruption diagram* was used to capture the structure of the forensic analysis algorithms as well as the spatial and temporal bounds of the corruption.

We characterized the algorithms' "forensic cost" under worst-case, best-case, and average-case assumptions on the distribution of corruption sites. We also validated cost formulae for these algorithms and provided recommendations for the circumstances under which each algorithm is indicated. Specifically, it is best to provide users with three algorithms: Monochromatic, a3D and, depending on the application requirements, RGBY or Tiled Bitmap. The Monochromatic Algorithm is by far the simplest one to implement and it is best-suited for cases when multiple corruptions are not anticipated or when only the earliest corruption is desired. The a3D Algorithm is the second easiest algorithm to implement and it is the only algorithm that exhibits all three of the most desirable characteristics: (i) it identifies multiple corruptions, (ii) it does not produce false positives, and (iii) it is stable and optimal for large number of corruptions. Hence this algorithm is indicated in situations where accuracy in forensic analysis is of the utmost importance.

6. CURRENT WORK

We are working on refinements on our previously-published forensic analysis techniques. Specifically, we have intro-

duced *page-based partitioning* as well as *attribute-based partitioning* along with their associated corruption diagrams.

In essence, our previous algorithms all view the data as partitioned on a particular attribute within each tuple: the commit time. We have significantly generalized this approach by partitioning the database on any attribute that can be correlated with real time. We have applied the same techniques to a database partitioned into pages, thereby analyzing each corruption from a different, "spatial" perspective.

Moreover, we have extended the partitioning of the data according to a chosen attribute. Note that this refers to a single table whose schema includes that attribute. This is in direct contrast to the commit time- and page-based schemes which are schema agnostic, a fact that potentially renders them global schemes (i.e., all data are under audit). Attributes, on the other hand, are restricted to a single table and therefore the scheme cannot be global; it only pertains to a section of the data. The advantage of the attribute-based scheme over the others is that, by taking into account the database schema, it allows for finer control over what part of the data is under audit. Attribute-based partitioning has allowed us to introduce an entirely new algorithm termed the Static-Level a3D Algorithm. It is not a natural extension of the commit-time- or page-based schemes. This algorithm retains the construction of a single binary tree of hash chains built on top of the data in a similar way as in the original a3D algorithm. The main difference here is that the leaves of the tree are no longer hash chains of page write events or time intervals. Instead, each leaf corresponds to a particular subset of the domain values of the partition attribute.

We have developed a comprehensive *taxonomy* of the types of possible corruption events, along with an associated *forensic analysis protocol* that consolidates all extant forensic algorithms and the corresponding type(s) of corruption events they detect. This has allowed us to formally and systematically define forensic analysis as a map from the protocol/algorithm observables to the elements in the taxonomy. The result is a generalization of these algorithms and an overarching characterization of the process of database forensic analysis, thus providing a context within the overall operation of a database for all existing forensic analysis algorithms.

We have established an architecture and an associated threat model that supports the forensic tools developed. The different components of the architecture have been associated with the execution phases so as to support tamper detection and forensics. In addition, we have provided multiple design choices for setting up variations of the core architecture such as types of hashing with respect to when the hashing occurs, and where the hash values are stored. We have also described the structure of forensic analysis algorithms in temporal terms. All these features will be supported in subsequent releases of DRAGOON.

DRAGOON is a prototype auditing system that is highly customizable in terms of offering a tunable trade-off between level of security and monetary/forensic cost. A beta version of DRAGOON is already available[4]. It is lightweight and scalable and hence is able to adequately address aspects of information accountability. We intend to expand our prototype to an enterprise-wide information accountability solution that can effectively realize appropriate use (i.e., guarantee no unauthorized modifications–insertions, deletions, updates even by insiders) in high-performance databases. This enterprise-wide solution will feature a replication service, a secure master database, and enterprise-level interfaces between the components of the architecture and the company's Chief Security Officer (CSO) who states enterprise-wide security policies, the database administrators (DBA) who are responsible for specific database(s), and one or more crime scene investigators (CSI) who investigate tampering and other corruptions. Moreover, to address some of the issues of privacy and liability we will equip the new system with mechanisms by which records can be physically removed from the database (shredding). Once this capability exists we will provide litigation hold mechanisms so as to secure court-mandated evidence. In the future we will consider expanding the current DRAGOON architecture to support databases deployed on the cloud as well providing audit capabilities to Apache access logs and log4j.

[4]`http://www.cs.arizona.edu/projects/tau/dragoon/`

7. CONTRIBUTIONS

This conceptual framework on information accountability architecture, forensic analysis tools and their evaluation, together with solutions for shredding and litigation holds will be extremely valuable and applicable to a variety of sectors. For example, they can help ensure record compliance for financial and medical institutions. They can serve as an unbiased witness to any type of database storing sensitive information. These may include court-submitted data from police databases or biological research results. The latter can be of particular use to a biosciences lab because it can ensure non-deviation from protocols thus providing a certain type of provenance for their final results. Furthermore, they can be utilized for the improvement of software and the protection of databases from bugs silently corrupting the system by potentially providing hints for isolating the piece of code responsible.

The techniques proposed will not just protect data but also through continuous assurance will be able to detect corruption shortly after tampering as well as automate to a great extent the work required in the aftermath of a database corruption. This obviously saves both time and money for those affected. The techniques will also highlight the advantages over approaches relying heavily on information restriction through either hardware which can have prohibitive costs for small institutions, have a limited shelf-life and are relatively complex; or cryptography which does not adequately offer remedies after a leak.

DRAGOON is lightweight, scalable, and highly customizable in terms of offering a tunable trade-off between level of security and monetary/forensic cost.

8. SUMMARY

As we have seen, within the domain of cryptographic-based hashing techniques used to achieve information accountability in databases, there exists no accountability-based reference architecture, no systematic formulation of forensic analysis, nor any fully-analyzed fundamental algorithmic tools.

Lastly, the taxonomy of corruption types has not been characterized.

The approach and techniques we are developing aim to remedy the above. They, and in conjunction with DRAGOON's evaluation will demonstrate that information accountability is a viable alternative to information restriction for ensuring appropriate use in databases.

Ultimately, our proposed techniques and approaches based on information accountability can solve security issues which seem intractable if seen as access control problems while striving to mirror the relationship between the law and human behavior more closely than existing approaches based on information restriction.

9. ACKNOWLEDGEMENTS

NSF grants IIS-0415101, IIS-0803229, and a grant from Surety, LLC provided partial support for this work. We also thank Richard T. Snodgrass, who has supervised this work, Peter Downey, Nirav Merchant, Soumyadeb Mitra, Radu Sion, Joseph Watkins, and Marianne Winslett for numerous and very helpful discussions on compliant databases and on tamper detection and prevention, as well as the reviewers.

10. REFERENCES

[1] 15 U.S.C.1681. Fair Credit Report Act. `http://www.law.cornell.edu/uscode/15/usc_sup_01_15_10_41_20_III.html` (accessed April 1, 2011).

[2] R. Agrawal and J. Kiernan. Watermarking Relational Databases. In *Proceedings of the International Conference on Very Large Databases*, pages 155–166. VLDB Endowment, 2002.

[3] M. Alles, A. Kogan, and M. Vasarhelyi. Black Box Logging and Tertiary Monitoring of Continuous Assurance Systems. *Information Systems Control Journal*, 1, 2003.

[4] D. Barbará, R. Goel, and S. Jajodia. Using Checksums to Detect Data Corruption. In *Proceedings of the International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*. Springer, March 2000.

[5] A. Basu. Forensic Tamper Detection in SQL Server, November 2006. `http://www.sqlsecurity.com/images/tamper/tamperdetection.htm` (accessed April 1, 2011).

[6] C. C. Chan, H. Lam, Y. C. Lee, and X. Zhang. *Analytical Method Validation and Instrument Performance Verification*. Wiley-IEEE, 2004.

[7] Creative Commons ©. `http://creativecommons.org` (accessed April 1, 2011).

[8] U.S. Department of Health & Human Services. The Health Insurance Portability and Accountability Act (HIPAA), 1996. `http://www.cms.gov/HIPAAGenInfo/` (accessed April 1, 2011).

[9] F.D.A. Title 21 code of federal regulations (21 cfr part 11) electronic records; electronic signatures, 2003. `http://www.fda.gov/ICECI/EnforcementActions/default.htm` (accessed April 1, 2011).

[10] K. Fu, M. F. Kaashoek, and D. Mazières. Fast and Secure Distributed Read-Only File System. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, pages 181–196, 2000.

[11] P. A. Gerr, B. Babineau, and P. C. Gordon. Compliance: The effect on information management and the storage industry. Research Report, Enterprise Storage Group, May 2003.

[12] H. Guo, Y. Li, A. Liu, and S. Jajodia. A fragile watermarking scheme for detecting malicious modifications of database relations. *Inf. Sci.*, 176(10):1350–1378, 2006.

[13] S. Haber and W. S. Stornetta. How To Time-Stamp a Digital Document. *Journal of Cryptology*, 3:99–111, 1999.

[14] R. Hasan and M. Winslett. Efficient Audit-based Compliance for Relational Data Retention. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 238–248, New York, NY, USA, 2011. ACM.

[15] W. W. Hsu and S. Ong. Fossilization: A Process for Establishing Truly Trustworthy Records. Technical Report 10331, IBM Research Report RJ, 2004.

[16] R. G. Johnston. Tamper-Indicating Seals. *American Scientist*, 94(6):515–524, Nov–Dec 2006.

[17] Y. Li, H. Guo, and S. Jajodia. Tamper Detection and Localization for Categorical Data Using Fragile Watermarks. In *Proceedings of the 4th ACM Workshop on Digital Rights Management*, pages 73–82, 2004.

[18] M. Malmgren. *An Infrastructure for Database Tamper Detection and Forensic Analysis*. Honors thesis, University of Arizona, 2007. `http://www.cs.arizona.edu/projects/tau/tbdb/MelindaMalmgrenThesis.pdf` (accessed April 1, 2011).

[19] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel. Separating key management from file system security. In *Proceedings of the ACM Symposium on Operating Systems Principles*, pages 124–139, Dec. 1999.

[20] S. Mitra. *Trustworthy and Cost Effective Management of Compliance Records*. PhD dissertation, University of Illinois at Urbana-Champaign, Department of Computer Science, 2008.

[21] S. Mitra, M. Winslett, R. T. Snodgrass, S. Yaduvanshi, and S. Ambokar. An Architecture for Regulatory Compliant Database Management. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 162–173, 2009.

[22] A. Muthitacharoen, R. Morris, T. M. Gil, and B. Chen. Ivy: A Read/Write Peer-to-Peer File System. In *Proceedings of USENIX Operating Systems Design and Implementation*, volume 36, pages 31–44, 2002.

[23] K. E. Pavlou and R. T. Snodgrass. Forensic Analysis of Database Tampering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 109–120, June 2006.

[24] K. E. Pavlou and R. T. Snodgrass. Forensic Analysis of Database Tampering. *ACM Transactions on Database Systems*, 33(4):30:1–30:47, November 2008.

[25] K. E. Pavlou and R. T. Snodgrass. The Tiled Bitmap Forensic Analysis Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 22(4):590–601, April 2010.

[26] U.S. Public Law No. 107–204, 116 Stat. 745. The Public Company Accounting Reform and Investor Protection Act, 2002.

[27] R. Sion, M. Atallah, and S. Prabhakar. Rights Protection for Relational Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 98–109, June 2003.

[28] R. T. Snodgrass, S. S. Yao, and C. Collberg. Tamper Detection in Audit Logs. In *Proceedings of the International Conference on Very Large Databases*, pages 504–515, September 2004.

[29] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman. Information Accountability. *Communications of the ACM*, 51(6):82–87, June 2008.

[30] G. Wingate, editor. *Computer systems validation: Quality Assurance, Risk Management, and Regulatory Compliance for Pharmaceutical and Healthcare Companies*. Informa Health Care, 2003.

[31] Q. Zhu and W. W. Hsu. Fossilized Index: The Linchpin of Trustworthy Non-Alterable Electronic Records. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 395–406, 2005.