RJ10331 (A0410-017) October 18, 2004
Computer Science

IBM Research Report

Fossilization: A Process for Establishing Truly Trustworthy Records

Windsor W. Hsu, Shauchi Ong
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Fossilization: A Process for Establishing Truly Trustworthy Records

Windsor W. Hsu and Shauchi Ong

IBM Almaden Research Center San Jose, CA 95120
{windsor, ong}@us.ibm.com

July 16, 2004

Executive Summary

Trustworthy records are vital to an organization. These records help to improve an organization's operations and aid in reducing its liability and costs. The fundamental purpose of record keeping is to establish solid proof and details of events that have occurred. A trustworthy record management system is, therefore, one that can be relied upon to provide irrefutable evidence of all of the events that have been logged. In other words, trustworthiness has to be established on 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. In this white paper, we show that the current limited focus on storing electronic records in Write-Once-Read-Many (WORM) storage is not adequate to ensure that such records are trustworthy. What is really needed is a process we call *fossilization*--a holistic approach to storing and managing records that ensures that they are trustworthy. Fossilization is composed of three parts. The first, *fossilization of storage*, guarantees that all records and their associated metadata are reliably stored and securely protected from any modification. The second, *fossilization of discovery*, ensures that all preserved records pertinent to an enquiry can be quickly discovered and retrieved. The third, *fossilization of delivery*, warrants that the exact pertinent records are delivered to the agent and that the records are delivered in an intact form. Because of the extremely high stakes involved in tampering with the records, fossilization must be realized very securely. The essential principles for securely implementing fossilization include 1) raising the barrier to any attack; 2) focusing on end-to-end trust; 3) limiting what has to be trusted; 4) using a simple, well-defined interface between trusted and untrusted components; and 5) verifying all operations.

1 Introduction

Records (or more generally, reference information) serve as institutional memory of events and actions and are a valuable asset to any organization. They represent much of the data on which key decisions in business operations and other critical activities are based. Having records that are accurate and readily accessible is therefore vital. Records also serve as evidence of activity and can be used both on the offensive and as protection against adverse litigation. To be effective, however, the records must be credible. Moreover, records that are not properly managed increase the risk of being cited for spoliation, which could not only result in severe sanctions and penalties, but also public relations disasters [1]. Ensuring that the records are trustworthy, *i.e.,* not only readily accessible and accurate, but also credible and irrefutable, is thus imperative.

On the other hand, records management is increasingly expensive, labor-intensive, prone to errors and susceptible to tampering. Today's information-centric global economy generates records at extremely high rates and requires records to be retained, protected and made accessible for longer periods of time. Our increasing reliance on information also means that the potential gain from tampering with the records is huge. Moreover, electronic records are not only convenient to create, but are also relatively easy to delete and modify without leaving so much as a trace. There is, therefore, an increasingly pressing need to be able to faithfully and cost-effectively preserve large volumes of records, and to ensure that the records are available and credible.

Furthermore, a growing fraction of the records is subject to regulations that specify how they should be maintained. In the US alone, there are currently more than 10,000 such regulations [3]. The penalties for failing to comply with such regulations are stiff. Unprecedented fines have recently been levied by several regulatory bodies, including the Securities Exchange Commission (SEC) and the Food and Drug Administration (FDA), for non-compliance. The bad publicity alone could cost an organization dearly. As information becomes even more valuable to organizations and with recent headlines of corporate misdeeds, accounting scandals and securities fraud, we can expect the number and scope of such regulations to grow. Having a trustworthy record management system would not only reduce regulatory risk, but also ease management burden and complexity.

In this paper, we show that the current focus of providing Write-Once-Read-Many (WORM) storage falls well short of what is required to achieve a truly trustworthy records management system. We contend that the issue of trust in record management must be approached from an end-to-end perspective. We stress that WORM storage is an important component of trustworthy electronic record management systems, but by itself effectively offers no value as far as trust is concerned. Based on these insights, we develop the basic requirements for a truly trustworthy record management system and devise the process--which we call *fossilization*--for achieving these requirements. We also analyze how implementations of the system affect trust and present several fossilization principles that can be used to realize a truly trustworthy record management system.

2 Trustworthy Record Management

When events such as financial transactions occur, records are created to serve as proof that those specific events happened at specific times. A trustworthy record management system is, therefore, one that can be counted on to provide irrefutable evidence of all of the events that have been logged. This means a system that faithfully records all relevant events as they occur, and one that not only reliably stores all of the records for an extended period of time, but also securely protects them from any modification. In addition, the system must be capable of quickly locating every record pertinent to an enquiry and of delivering the records intact to the agent seeking the proof. In other words, trustworthiness must be established on an end-to-end perspective based on the basic purpose of record keeping, namely to establish solid proof of all of the events that have occurred.

The process of creating faithful records for all of the relevant events as they occur is generally trusted. This is the case especially if the records are used in the normal course of business, and are hence required for the proper functioning of the organization. Furthermore, record creation is an ongoing process for which periodic audits are effective in ensuring proper execution. More importantly, it is extremely unlikely that one can anticipate at record creation time how a record could be modified for personal gain. In any case, if one is privy to such knowledge, one would arguably be better off by trying to influence the events themselves. The basic objective of record keeping is not to prevent the writing of history, but to prevent the changing of history; in other words, changing the records after the fact.

The requirement for reliably storing records over an extended period of time is similar to the basic requirement of any storage system and includes preventing any loss of data due to disasters, system failures, equipment obsolescence, *etc.* Loss of data could also result from intentional destruction of the storage system to remove damaging evidence. However, one considering such an action should instead safeguard the records because large scale destruction of records would be evident and could result in the presumption of guilt. Also, deliberate destruction of records can be effectively mitigated by imposing physical security and by remotely mirroring the data.

The more pressing need is to protect against clandestine modification of selected records; in other words, to ensure that all of the records are effectively immutable, readily accessible and deliverable in an unaltered form. Ensuring that the records are effectively immutable translates into preventing any physical modification, including selective destruction, of the records during both their storage and their delivery to the agent seeking the records. The readily accessible requirement means that all of the records relevant to an enquiry must be discovered and retrieved within days and sometimes even within hours [1]. With the large volume of records today, this specification necessitates some form of direct access mechanism, such as an index, for the records. Ensuring that the records are effectively immutable must, therefore, also include preventing any logical modification of the records by altering the access mechanism, including replacing them with new versions, performing logical deletions and employing other forms of record hiding.

Modification of the records could result from user errors, such as issuing the wrong commands and replacing the wrong disks during service actions, and from software bugs. Given the importance of

records, the potential gain from intentionally manipulating and modifying them is huge. Thus, the records would also have to be protected from intentional attacks. For electronic records, these malicious attempts to compromise the records could come in the form of hacking and viruses. The more menacing threat is that they could be inside jobs, launched by disgruntled employees, company insiders or even conspiring technology experts. A truly trustworthy record management system would have to withstand all of these threats.

3 Fossilization

Fossilization refers to the process of ensuring that all of the records are effectively immutable, readily accessible and deliverable in an intact form. Fossilization is an end-to-end process--the ends being the storage where records are kept, and the access point where records are received, such as by an agent performing an audit, a legal or regulatory discovery or an internal investigation. Fossilization ensures that all of the records that are relevant to the agent are quickly discoverable and that they are delivered unaltered to the agent.

Fossilization consists of three separate components. The first component is *fossilization of storage*, *i.e.,* ensuring that all of the records are reliably stored and securely protected from modification. The second is to be certain that every record that has been logged can be readily found and retrieved in a timely manner, in other words, that none of the preserved records can be hidden from the access or discovery mechanism used by the agent. This is known as *fossilization of discovery*. The third is to ensure that all of the retrieved records are not compromised along the delivery path, but are delivered in an immutable fashion to the agent. We call this *fossilization of delivery*.

3.1 Fossilization of Storage

The basic requirement for preserving records is to be able to prevent physical deletion and modification of the records. For electronic records, this requirement means protecting the bits and bytes that represent the records from any change, and is typically satisfied by storing the records in WORM storage. There are, however, different ways to achieve WORM storage, and the degree of protection they offer varies widely. As discussed earlier, a trustworthy record management system must protect against both accidental and malicious alteration of records, including attacks from the "inside" and from conspiring technology experts. Thus, it needs an extremely secure mechanism for enforcing the WORM property.

Fossilization of storage requires that the entire record be securely protected from modification. An electronic record is typically composed of a sequence of data blocks. Ensuring that the data blocks are immutable does not guarantee that the record cannot be altered. For example, Figure 1 shows that a record with blocks 1, 2, 3, 4 can be modified to comprise blocks 1, 2, 3', 4 if the metadata that describes the blocks and their sequence is not immutable. More generally, a record must be preserved on WORM storage together with all of its associated metadata, including information that describes the structure and attributes of the record such as its creation date (Figure 2). Such a process of fossilizing the storage of records ensures that no part of a record can be altered, replaced or removed, and that the record is preserved completely with all of the information necessary to ensure its long-term usefulness.

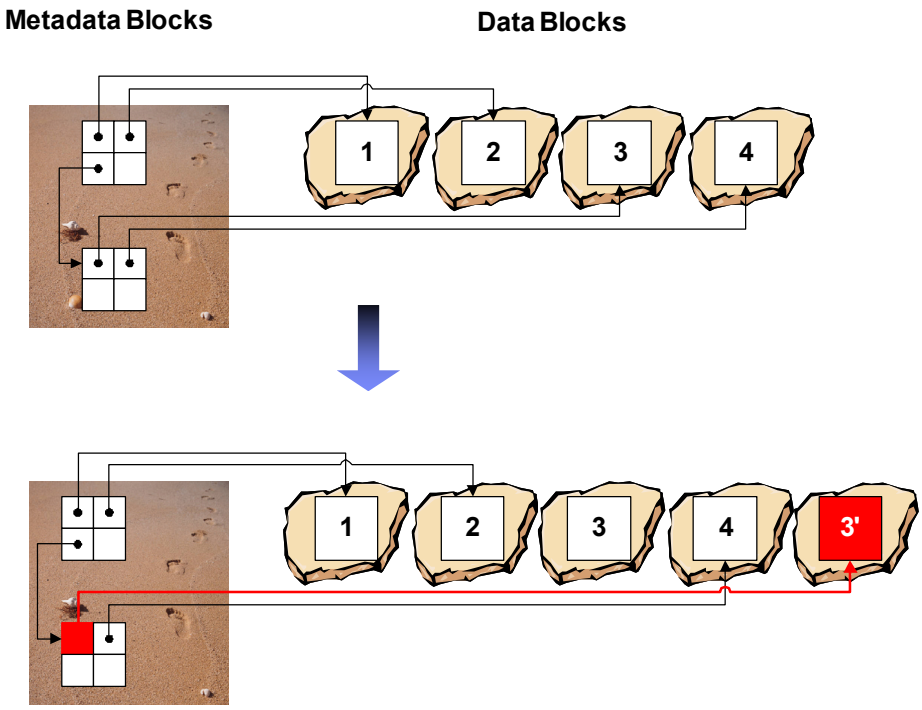

Figure 1: **Block Immutability is not sufficient.** *This diagram shows that a record stored in secure WORM storage can be altered if its metadata is not likewise preserved. Storing the data blocks in secure WORM storage is analogous to casting the data in stone. Such blocks cannot be modified once they have been written. Keeping the metadata blohcks in rewritable storage is akin to writing them in sand. These blocks can be easily changed.*

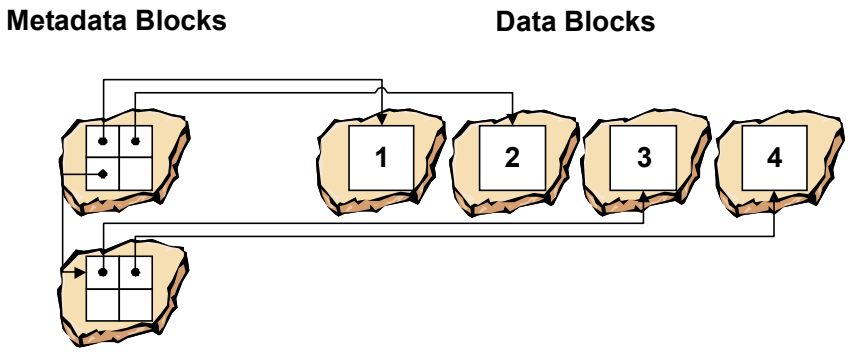

Figure 2: **Fossilization of Storage.** *This diagram shows a record preserved securely and completely with all the information necessary to ensure its long-term usefulness.*

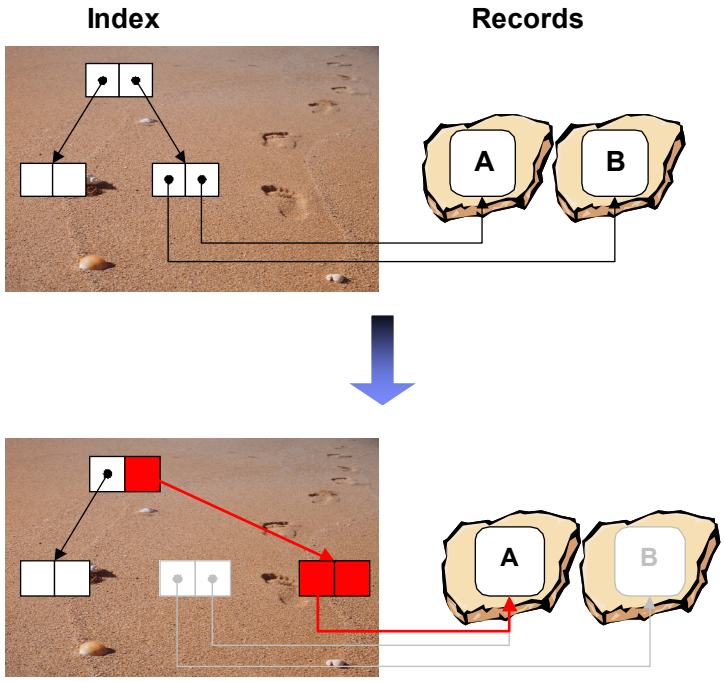

Figure 3: **The Need for Fossilization of Discovery.** *This diagram shows that record B can effectively be deleted if the access or discovery mechanism is not fossilized. The records are depicted as cast in slabs of stone to indicate that they are not modifiable. The index blocks are illustrated as written in sand to show that they can be easily changed.*

3.2 Fossilization of Discovery

Fossilization of storage protects any part of a record from physical modification such as having its bits changed. This alone, however, does not mean that the system can be trusted to provide undeniable evidence of all of the events that have been logged because the records could be logically modified. For example, they could be replaced by new versions. Physically preserving a record is effectively useless if the preserved record can be replaced by a new version, hidden, or otherwise made inaccessible. For instance, Figure 3 depicts the situation where record *B* is physically immutable, but can be hidden or effectively removed simply by modifying the access mechanism (*e.g.,* an index) through which the records are located. By a similar process, any record can be replaced with a new version if the access mechanism is not properly designed.

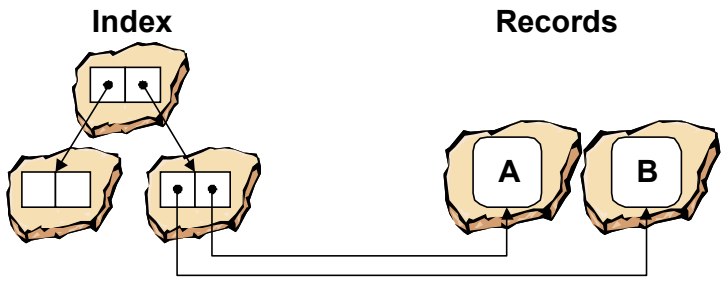

Figure 4: **Fossilization of Discovery.** *This diagram shows that with a fossilized access mechanism, all of the records are guaranteed to be quickly discoverable.*

Therefore, a trustworthy record management system must ensure that all of its records are organized in such a way that they are guaranteed to be quickly discoverable. More specifically, any mechanism that is trusted by the agent to find and access records should always be able to locate and retrieve any record in the system for which it is accountable. For example, Figure 4 illustrates a system that achieves such fossilized discovery by making sure that after a record is entered into the system, both the index entry for that record, as well as the path through the index to that entry, are immutable.

Note that fossilization must be applied to any trusted means of finding and accessing a record. Examples of access mechanisms include a file system directory, which allows records to be located by file name; a database index, which enables records to be retrieved based on the value of some specified field or combination of fields, and a full-text index, which allows records containing some particular words or phrases to be found.

3.3 Fossilization of Delivery

With fossilized storage and discovery of records, we can be certain that all of the records are preserved and that any preserved record can always be located quickly. However, a weak link in the system still remains--the records may be susceptible to alteration during transit to the agent conducting the search.

For example, Figure 5 shows a typical electronic record management system today where the retrieved records are handled by an elaborate stack of software components on their way to the agent. To enable the system to be composed in a modular fashion, the software stack is designed to accept new components and to allow existing components to be replaced. In fact, the components often have to be updated to increase their functionality and, more commonly, to fix bugs and security risks. This dynamic nature of the software stack makes it vulnerable to compromise. For example, as illustrated in Figure 5, an additional component or software patch could be installed in the system to filter and modify selected records so that even though the records are preserved in fossilized storage, the agent would receive tampered records.

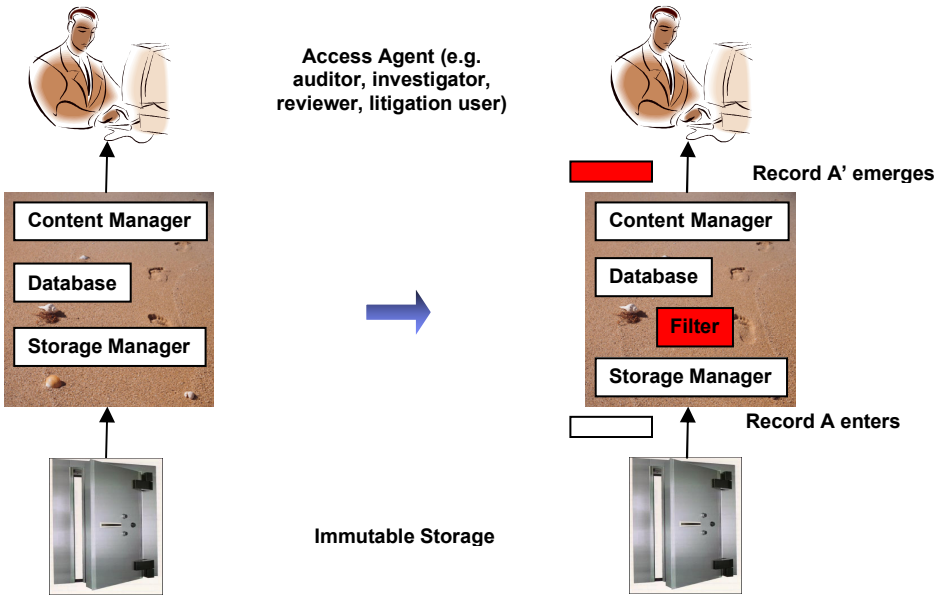

Figure 5: **The Need for Fossilization of Delivery.** *This diagram shows that the delivery path can be compromised such that records retrieved in an unaltered form from the immutable storage are modified as they are delivered to the agent. The delivery path is shown as written in sand to illustrate that it can be easily changed.*

Such a system is clearly not trustworthy. A system is only as trustworthy as its weakest link. If we cannot ascertain that records will be faithfully delivered to the agent, fossilization of their storage and discovery is effectively futile. Therefore, a trustworthy record management system requires a fossilized delivery mechanism that will convey in verbatim to the agent all of the records retrieved from the fossilized storage and only these records.

4 Principles and Practices

As discussed earlier, a trustworthy record management system must protect against accidental record modification that could result from user errors, software bugs, *etc.* The system must prevent any user or software errors that occur over the records' long retention period from inadvertently affecting their preservation, discovery and delivery. In addition, any software change that occurs during the retention period must not cause the trust to be compromised. Given the importance of records, the potential gain from intentionally manipulating and modifying them is huge. Thus, even more importantly, the system must guard against malicious attacks, even those from the "inside," and from conspiring technology experts. In other words, an extremely secure mechanism is required to enforce the end-to-end trust in record management.

In the following list, we highlight some principles that are adapted from computer security for securely implementing fossilization. A more complete discussion of these principles is in [2].

- *Increase barrier to attack.* Raise the cost and conspicuity of any attack against the system by, for instance, reducing the pool of people and organizations with the relevant expertise.

- *Focus on end-to-end trust.* Take a holistic approach to increase the trustworthiness of the overall system or component.

- *Limit what has to be trusted.* Isolate the trust-critical modules and make them simple, verifiable and correct.

- *Use a simple, well-defined interface between trusted and untrusted components.* Tightly control what can be entered into the trusted components to reduce the possibility of compromising the trusted components and to limit any error propagation.

- *Trust, but verify*. Verify each component to ensure that it works as intended. Preferably, the verification should be performed on each and every operation so that any fault can be quickly discovered, isolated and corrected.

These basic principles indicate that for electronic records, the component for enforcing the WORM property should be as small as possible, both to reduce the probability that something could go wrong or be compromised, and to increase our ability to verify the correctness of the component. In addition, the component should have a simple, well-defined interface to robustly restrict traffic into the component to only legitimate requests. In other words, from the trust perspective, block-level WORM storage, by virtue of its basic functionality and narrow interface, would be preferable to a system where the overwrite prevention is embedded within a feature-rich component such as an application or a general purpose file/object system. Furthermore, the WORM component should be implemented with customized hardware or firmware to reduce the resource pool capable of launching an attack, and to cause any such action to be more costly and conspicuous.

Further applying the principle of limiting what has to be trusted, fossilized discovery should be achieved by relying only on the non-rewritability of WORM storage. Specifically, any access mechanism, such as an index, that is trusted by the agent for locating records should have the property that the insertion of a new record into the system does not in any way affect how previously inserted records are accessed through that mechanism. This means that once the insertion of a record into the access mechanism has been committed to WORM storage, the record is guaranteed to be quickly discoverable using that mechanism unless the WORM storage is compromised. In other words, the accessibility of the record is dependent only on the non-rewritability of the WORM storage. To be exact, the access mechanism must still operate correctly, meaning that the entries that have been committed to WORM storage must be faithfully read and interpreted. This can be accomplished in a similar manner as the faithful delivery of records, which we consider next.

To complete the end-to-end trust, exactly the retrieved records must be conveyed to the agent, and they must be conveyed without any alteration. The key concepts for securely fossilizing the

delivery are to limit what has to be trusted in the delivery process, and to design the delivery system such that it is difficult to replace, modify or otherwise compromise. In this case, these principles mean decreasing the complexity and generality of the software stack that is on the path to the agent, or providing a shortcut to the agent when the highest degree of trust is required. For example, the layout of the records and indexes on the fossilized storage could be disclosed so that the agent could use his own trusted code to interpret the layout and to retrieve the records. The agent could also be given the option to install and use his own, presumably pristine, version of the software stack to access the records. Alternatively, the code for retrieving the records could be made public so that its correctness and integrity could be independently examined and verified.

As added assurance, the system should verify that every operation is performed correctly. In particular, it should ascertain that every record in the system is trustworthy from an end-to-end perspective, meaning that each record is completely preserved, quickly discoverable, and will be faithfully delivered in an unmodified form to the agent. For example, after each record is written into the system, a verification engine could locate that record using the same mechanism that an agent would use, and compare the retrieved record with what was written into the system.

5 Other Systems Fall Short

Current electronic record management systems fail to provide end-to-end trust. Some simply write records to WORM storage, which, as we have established in this paper, is clearly insufficient to ensure that the records are trustworthy. Others do not even store records in WORM storage. Moreover, several recently introduced WORM storage systems that are designed specifically for record storage rely on function-rich software to prevent records from being overwritten. As we have discussed, such an approach offers inadequate protection against physical modification of the records, especially in view of the high stakes involved and the likelihood of malicious and inside attacks.

Note that in the past, storing records on WORM storage could guarantee that the records are accessible because even if a record were to be logically modified, all versions of the record would be preserved so that one could theoretically scan all the stored data to discover the original record. For example, if we store a file on CD-R, we can later write a newer version of the file and have it logically replace the first file. If we took the time to scan through the entire contents of the CD-R, we would be able to find the first file. The problem is that with the huge volume of records today and the complexity of the records as well as the systems managing the records, scanning all of the data to find and piece together possible versions of a record has become far from practical.

6 Summary

Records are a vital asset to an organization. They form the institutional memory on which key decisions are based, and serve as evidence of activity. Maintaining trustworthy records is, therefore, imperative. Moreover, many records are governed by regulations that specify how they should be properly stored and managed.

The primary objective of record keeping is to establish solid proof and details of events that have occurred. A trustworthy record management system is, thus, one that can be relied upon to provide indisputable evidence of all of the events that have been recorded. In other words, trustworthiness must be established on 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. The current limited focus on merely storing electronic records in WORM storage is clearly far from adequate to ensure that such records are trustworthy. What is really needed is *fossilization*, a holistic approach to storing and managing records such that they are trustworthy.

Fossilization is composed of three parts. The first, *fossilization of storage*, guarantees that all of the records and their associated metadata are not only reliably stored for an extended period of time, but are also securely protected from any modification. The second, *fossilization of discovery*, ensures that every preserved record that is relevant to an enquiry can be readily located and retrieved in a timely fashion, while the third, *fossilization of delivery*, warrants that exactly the relevant records are delivered to the agent, and that the records are delivered in an unaltered form.

Due to the extremely high stakes involved in tampering with the records, fossilization must be realized very securely. The key principles for securely implementing fossilization include 1) increasing the cost and conspicuity of any attack against the system, such as by using non-universal components and custom hardware for key parts; 2) taking an end-to-end approach to increase the trustworthiness of the overall system; 3) isolating the trust-critical modules and making them simple, verifiable and correct; 4) using a simple, well-defined interface between trusted and untrusted components to tightly control access to the trusted components and to limit error propagation; and 5) verifying every operation so that any fault can be quickly discovered, isolated and corrected.

References

[1] COHASSET ASSOCIATES, INC. The role of optical storage technology. White Paper, April 2003.

[2] WINDSOR W. HSU, LAN HUANG, AND SHAUCHI ONG. Content immutable storage: Truly trustworthy and cost-effective storage for electronic records. White Paper, IBM Research, July 2004.

[3] THE ENTERPRISE STORAGE GROUP, INC. Compliance: The effect on information management and the storage industry, May 2003.