XBRL-Trail — A model for introducing digital forensic readiness to XBRL

Dirk Kotze & Martin S. Olivier

University of Pretoria, Lynnwood Road, Hillcrest, Pretoria, Gauteng, SOUTH AFRICA

djjkotze@kroon.co.za & molivier@cs.up.ac.za

Abstract

Business has become heavily reliant on Information Technology to process and share business data and financial data. Proprietary, non-standardised formats often hinder such sharing of financial and business data as not all stakeholders can read and access the data. As a result, a standardised, open format was needed in order to ensure that all stakeholders have the ability to read and process the relevant data.

XBRL (The eXtensible Business Reporting Language) is rapidly becoming the standard format for sharing financial data. This is mainly due to the easy sharing of information facilitated by it, as it is based on a publicised standard. Not only is this format easily readable, but it can also be interpreted by computer as it contains semantic data.

The usage of XBRL does however pose the significant risk of fraud, as it is rather trivial to edit the financial records in a fraudulent manner. Such cyber crime is typically investigated by a digital forensics team, whose duties are significantly complicated by XBRL's very scant retention of forensic evidence (forensic readiness).

This article addresses XBRL's lack of forensic readiness and proposes a model to enhance the forensic readiness of XBRL. Using a mediator, placed between the users of the XBRL data and the XBRL data itself, we show that forensic evidence can be captured in real time, which would significantly reduce the investigation time.

Keywords:

Digital Forensics, Forensic Readiness, XBRL

XBRL-Trail — A model for introducing digital forensic readiness to XBRL

Abstract

Business has become heavily reliant on Information Technology to process and share business data and financial data. Proprietary, non-standardised formats often hinder such sharing of financial and business data as not all stakeholders can read and access the data. As a result, a standardised, open format was needed in order to ensure that all stakeholders have the ability to read and process the relevant data.

XBRL (The eXtensible Business Reporting Language) is rapidly becoming the standard format for sharing financial data. This is mainly due to the easy sharing of information facilitated by it, as it is based on a publicised standard. Not only is this format easily readable, but it can also be interpreted by computer as it contains semantic data.

The usage of XBRL does however pose the significant risk of fraud, as it is rather trivial to edit the financial records in a fraudulent manner. Such cyber crime is typically investigated by a digital forensics team, whose duties are significantly complicated by XBRL's very scant retention of forensic evidence (forensic readiness).

This article addresses XBRL's lack of forensic readiness and proposes a model to enhance the forensic readiness of XBRL. Using a mediator, placed between the users of the XBRL data and the XBRL data itself, we show that forensic evidence can be captured in real time, which would significantly reduce the investigation time.

1 Introduction

XBRL (the eXtensible Business Reporting Language) was developed in response to the challenge of free sharing of financial information. XBRL, like XML, is a mark-up language that uses specialised tags to delineate financial structures or elements.

Although XBRL facilitates information sharing, its approach is not without problems. One of the key concerns is the financial source data's susceptibility to easy modification in an unauthorised manner. This is as a direct result of the human readability requirement. Furthermore, due to the monetary significance of business decisions, manipulation of the business decision making process by misrepresenting and/or falsifying the financial records is rather profitable. Modification of the XBRL financial data is thus easily done to the financial benefit of the party perpetrating the manipulation. This action is commonly known as fraud [3].

Due to the digital nature of the financial statements, the investigation of such fraud cases cannot be done in the traditional way. One needs the help of digital forensic experts — known as the investigative component of cyber law enforcement. As with traditional forensics, the focus of this approach is to gain evidence from the digital crime scene. Unlike traditional forensics however, it is often significantly harder to find meaningful evidence in the cyber world.

Furthermore, cyber investigations are complicated even more by the fact that it difficult to accurately assess the reliability of evidence found due to the ease with which digital evidence can be modified and tampered with [9]. The reliability and accuracy of investigation is of crucial importance in the successful investigation and prosecution of a digital crime. Casey state that forensic examiners have "a duty to estimate how closely ... their data approximate reality". In essence, the conscientious investigator must apply some form of rating to the certainty and reliability of his/her evidence.

Casey [9] proposes such a rating scale, called the Casey Certainty Scale (CCS). It rates evidence on a scale, from lowest level of certainty (C0 — erroneous evidence that contradicts known facts) to highest level of certainty (C6 — evidence that is absolutely certain, tamper-proof and unquestionable).

In addition, it is also worthwhile to ensure that processes are in place to facilitate the easy collection of digital forensic evidence in the event of a digital crime. This is known as forensic readiness [6].

By nature however, XBRL is not inherently forensically ready as it does not store any meta data. In the event of an investigation, one is reliant on applications that fullfill the role of arbiter (such as the Operating System logging function) for all digital evidence. Typically this evidence only provides the bare minimum of what is needed and is not sufficient to conduct a proper investigation. Casey rates this type of evidence as C1 (highly uncertain) on the CCS as it originates from only one source and may be manipulated in any way. One can thus conclude that XBRL in its current state is *not* reliable and *not* forensically ready (as in the vast majority of cases it does not provide sufficient useful forensic evidence).

This brings us to the problem that will be addressed by this article. XBRL is created in human readable clear-text, in order to allow for data exchange. Should fraud and/or illegitimate tampering occur with a company's financials (stored in XBRL), how can we extract usable digital forensic evidence from XBRL data? XBRL is merely a tagging and organisation scheme for data and does not contain any usable evidentiary or meta-data that can be utilised by forensic investigators.

Palmer [19] states that there is a dire need for "incorporating scientifically-based approaches to conducting forensic analysis in the digital world, rather than developing digital technologies and then adapting them to benefit forensic analysis techniques". This article aims to contribute to such a scientifically-based approach to enhance the forensic readiness of XBRL. This is accomplished by suggesting a pluggable model to provide relevant evidentiary meta-data that ranks high on the Casey scale in terms of certainty and reliability. This model is called *XBRL-Trail*.

In order to ensure that a valid set of meta-data is preserved for forensic analysis, XBRL-Trail should be the sole point of creation and modification of XBRL documents. It should enforce at least authentication and authorisation steps, tamper-proofing and version control. The latter should be performed on the financial transactions posted in the books by means of XBRL.

By enforcing all of the above elements, multiple sources of evidence are created that are protected against tampering. Furthermore, in conjunction with other logs (for example. Operating System logs) the evidence obtained from XBRL-Trail can now be rated as very certain on the CCS as it supports agreement of "multiple, independent sources that are protected against tampering" [9].

The remainder of this paper is structured as follows — in section 2 we supply a brief overview of XBRL and Forensic Readiness, continuing in section 2 with the development of the XBRL model. We introduce the idea of XBRL Version Control and Tamper-proofing in section 3 and proceed to discuss the derived characteristics and implementation of the XBRL model in section 4. We conclude by addressing the architecture of the model in section 5.

2 XBRL and Forensic Readiness

As mentioned in the introduction, the eXtensible Business Reporting Language was developed mainly to address information sharing concerns between businesses (business to business communication) [2].

XBRL implements semantic awareness of the relationships between different elements of data [2]. Semantic awareness serves to detect source data errors and enables complex calculations of different data groups and fields. Groupings of data for different purposes, like business units, organisational groupings or inter-company relationships can also be implemented by means of XBRL.

Not only does XBRL suggest over-all process improvement, it also enables performance improvement in the area of internal business processes. Financial service companies can now easily compile statistics of business sectors by electronically comparing financial statements of major business sector stakeholders and predicting trends based on historical data [2]. Furthermore, banks and other financial regulatory institutions can perform audit type functions with greater ease as the financial data is electronically available and interpretable. A simple example of such an audit function is that of an automated credit limit approval process, based on an automated asset and liability comparison of a debtor.

As with any system however, there are definite drawbacks to the mark-up and human readability components of the design of the format. XBRL has been widely criticised for being too verbose [11]. It is believed that the addition of tags create larger data sets that are more time-consuming to transmit via the internet. Other complaints have been raised regarding the cumbersome method of parsing and the performance penalty on the computation process (due to semantic interpretation of the tags).

It is beyond the scope of this paper to determine XML/XBRL's suitability to its purpose. Instead, this paper accepts XBRL as a solution to the problems created by proprietary formats, and focuses on the forensic issues inherent to XBRL.

Looking at XBRL from a forensic point of view, there are a number of serious concerns. Palmer highlights this fact by stating that any new technology can be utilised for unauthorised and illegal purposes. One of these concerns is the human readability component of XBRL.

The human readability requirement introduces the risk of fraud by means of editing the financial information stored in a system. Not only is such an act of fraud fairly easy to commit, XBRL lacks inherent support for the extraction of detailed forensic evidence in the event of fraud. This evidence is needed in order to prosecute the perpetrators of fraud that is committed in the above fashion.

As mentioned in the introduction, XBRL does not support the extraction of detailed forensic evidence from the tagging structure as it is simply not kept. In addition, modifying the XBRL standard to collect such forensic evidence is not feasible. As the standard has already been published, changing it is not really a viable option.

In normal circumstances, the lack of forensic support from data formats would not be problematic. Considering however that the format is intended to handle very sensitive data and that it embodies the weakness of human readability (and thus the strong possibility of human modification), this situation calls for some form of remediation.

The above problem is further exacerbated by the lack of certainty available in the current XBRL forensic evidence scenario. Casey [9] states that the reconstruction of digital crimes invariably contains some degree of error, be it the origin, time of events or even the perpetrator of the events. Furthermore, even lost data can provide an incomplete picture of the crime scene.

Casey establishes two types of uncertainty, namely temporal uncertainty and uncertainty of origin. These uncertainties deal with some aspects of the very core of a forensic investigation, namely the *who* and the *when* properties of the crime. These uncertainties form the main basis for the criteria of the CCS.

The only way to address the lack of forensic evidence available from XBRL is to obtain evidence by a third-party solution which is independent of XBRL. This provides the advantage of that evidence is gathered from an independent source, which enables a higher certainty rating according the CCS.

Furthermore, the third-party solution should also continuously collect forensic evidence as it is not available from the XBRL source data itself. The technique of having forensic data pre-gathered is a major component of forensic readiness.

Forensic readiness is defined by Rowlingson as the "ability of an organisation to maximise its potential to use digital evidence whilst minimising the costs of an investigation" [21].

Tan states that the default existence of relevant evidence in digital crime scenes is extremely rare [23]. Typically, extensive investigation is required in order to discover relevant evidence. Furthermore, a natural tendency exists to place a higher premium on continuance by containment and recovery than on the investigative process.

This is due to the criticality of business continuation, which is primarily responsible for revenue generation. A grave danger is posed by this tendency, as a trade-off exists between continuation and evidence recovery. Continuation is inversely proportioned to the evidence recovery — one cannot be favoured without impeding the other.

The above problem is worsened by extended investigation times. A proper investigation typically takes exponentially longer than it took the perpetrator to commit the crime [22]. The longer the investigation, the longer the down-time for the business.

It is thus clear that it is beneficial to ensure that a strong emphasis is placed on forensic readiness within the implementation of XBRL. Furthermore, the implementation of forensic readiness via an arbiter addresses the problem of the lack of availability of forensic data in XBRL.

We continue to develop a model to address these concerns in the next section.

3 Development of the XBRL model

As mentioned in the earlier, forensic readiness in XBRL is of crucial importance. Rowlingson [21] defined a process to establish forensic readiness in a corporation, aimed at demarcating the applicable business problems and developing strategies to address these. In this section we utilise a reduced version of Rowlingson's process to develop requirements for the XBRL-Trail model.

The Oxford dictionary defines a model as "*a simplified or idealised description or conception of a particular system, situation, or process*". In addition, Olivier [18] mentions the following three characteristics of a model:

1. a model should serve to *clarify and delineate the problem space*;
2. a model should *explain the proposed solution* to such an extent that it is clearly understood;
3. a model should be *generic* enough for easy application in practice.

Using the Rowlingson process, we aim to distill the challenges of human readability and the lack of forensic evidence posed by XBRL. We then proceed to use the output of the Rowlingson process to create a model solution that satisfies the criteria posed by Olivier [18].

Rowlingson states that the first step is "*the definition of a business scenario that requires digital evidence*" [21]. Applying this model to the business environment, where XBRL is most likely to be employed, we define the scenario as the *collection and processing of financial information*, used to create financial statements. The risk involved in this scenario is the fraudulent modification of financial books and statements.

In order to address this risk, it is clear that we need to monitor the transactions to the source financial data in some manner. As mentioned earlier, the best way to achieve this is via a monitoring agent acting as an arbiter, as XBRL does not inherently support monitoring or the collection of meta-data. Our first requirement is thus stated to be a central mediator that monitors XBRL transactions.

The second step is to "*identify available sources and different types of potential evidence*" [21]. As XBRL is used as the vehicle for the collection and processing of financial information, XBRL source data is the obvious available source of evidence. As stated previously however, pure XBRL data is not very useful for forensic investigations.

We therefore need meta-data — details about changes to the file, on which date and by whom changes were made, which changes were made and what their effects were. As mentioned earlier, recording all access data mandates that all modifications to the financial source data should occur through a single point. The single point requirement also holds in order to ensure that the entire population of edit operations are recorded.

This approach mandates the usage of a central mediator which should be able to successfully collect the meta-data of all accesses to the financial data.

The third step is that of the "*evidence collection requirement*" [21]. The evidence collection requirement for the business scenario is that of meta-data. In our case no business interruption or interference occur by silently logging meta-data in the background. The cost of keeping evidence on hand is thus absolutely minimal — the cost of hard drive space and the potential cost of developing the reference monitor being the only cost concerns. It is thus safe to say that no additional evidence collection requirements exist, other than ensuring the existence of the central mediator, mentioned earlier.

The penultimate step in the Rowlingson programme is that of "*establishing a capability for securely gathering legally admissible evidence*" [21]. The main problem is that all recorded evidence is stored (together with the source data) in a human readable XML-type text file, and might simply be removed or altered by a malicious agent.

This is a complex problem, requiring separate discussion. This is addressed in section 4.

The final step to establish forensic readiness is to "*ensure that monitoring and auditing is targeted to detect and deter major incidents*" [21]. This is done by means of a two-pronged approach — all XBRL changes are silently logged by means of the mediator; and access to the data is restricted on the basis of restriction of access to the mediation agent.

In the next section we discuss the complex problems of Tamper-proofing data and recording forensic data by means of Version Control.

4 Version Control and Tamper-proofing

We can infer two main requirements for XBRL data integrity by means of the discussion in the previous section. The first requirement deals with recording all access requests to the source data. This should be done in such a manner that forensic evidence is available at all times, adhering to the forensic readiness concept. The second requirement considers the protection of the XBRL financial data from unauthorised access.

In this section we examine each of the requirements in detail, determining the approach required for each.

In order to record all access to the financial data, stored in XBRL format, a number of fundamental pre-requisites should be in place. Firstly, all data modifications should be required to take place through a central point at which monitoring can occur (this is discussed in detail in Section 3). Secondly, some form of meta-data capturing should occur.

Successful meta-data capturing should not be too computationally intensive, in order to avoid an unnecessary performance penalty on each data access operation. Furthermore, for the purpose of fraud detection, only edit (create, update and delete) operations should be recorded[1].

Several approaches are available with regards to logging meta-data. One could either keep a list of the names of users who edited the financial source data; or one could keep a list of edits combined with the users responsible for each edit. For the purposes of forensic evidence, it is necessary to not only know *who modified the data*, but also *what was modified* and *when it was modified.*

The recording of such detailed meta-data dictates an approach of storing a the edit operation as well as the user responsible for it. Doing this in a linear fashion where each change and the username is recorded, is at best tedious and ineffective. Rather, a process such as a Version Control System (VCS) should be used, as a VCS is explicitly intended to record changes. A VCS is defined as "a mechanism that allows one to audit changes to a particular document or source by being able to see who changed what" [20]. Furthermore, it allows for automatic roll-back to different versions.

A VCS is traditionally created by incrementally storing the *deltas* (or changes) between different documents in a tree format. In addition, the date of the change and details of the person who made the change are stored with the change. This approach facilitates version-switching by applying the deltas in chronological order [7].

In the traditional VCS approach no consideration is given to the type of text data that a VCS is applied to [4]. Unfortunately this approach is not efficient, as numerous changes to XBRL source data can be made without changing the semantic meaning. We refer to these changes as *non-semantic changes.* Typically, a non-semantic change constitutes a change in spacing, or a change in the order of tags (whilst not changing the tag data or the parent tag).

Such changes are negligible and should not be stored as they do not change the semantic meaning of the data. It therefore is advisable to use a delta-algorithm specifically created for XML/XBRL files. For a full discussion on various XML VCS and delta algorithms, see [4] as such a discussion is beyond the scope of this article.

[1] This requirement may vary with regards to application. If the financial data is sufficiently sensitive, read operations should be recorded as well.

Using such an algorithm, it is possible for the arbiter module to not only *keep a record of changes* to the financial source data, but also to be able to *interpret changes*. Furthermore, forensic investigators can now determine *who* changed *what*, instead of simply being able to prove that user x had access to the files during the time of the suspected incident. Being able to obtain such a detailed set of evidence will significantly shorten investigation time and further aid in the successful prosecution of a malicious/fraudulent user. Lastly, this approach once again bolsters the CCS rating of the forensic evidence as it establishes a multiplicity of elements that compose evidence and addresses the concern of uncertainty of origin. These elements are comprised of the detail of *what was changed*, *how it affected the statements* and *who changed it*.

The use of a VCS system brings us one step closer to Casey's ideal [9] of evaluating computer generated records "based on the reliability of the system and process that generated the records".

This discussion brings us to the second requirement in securing the XBRL data, namely preventing unauthorised access and direct modification of the data. This is known as *tamper-proofing* [16]. Casey [9] clearly states that should be "protected against tampering" in order to be reliable, making tamper-proofing another requirement for a high rating on the CCS.

As stated earlier, in order to address the problem of human readable (and editable) XBRL data, all access/edit operations should be forced to occur through the central mediator. Furthermore, this application is subject to several requirements, listed below:

Human readability of data — XBRL requires the source data to be human readable;

Highly tamper-resistant — Due to the sensitivity of the source data, namely financial information, it is imperative that this approach has a very high success rate; and

Computationally cheap — The process employed to tamper-proof the XBRL data must have virtually no impact on the computational complexity of processing. This is due to the already computationally expensive operation of interpreting XBRL source data.

As can be seen, the problem of tamper-proofing XBRL data is rather unique and complex. It should be noted that the authors do not use the term tamper-proofing in the absolute sense (as there is no such thing as an infallible system), but rather as a reference to a sufficiently tamper-resistant system.

Tamper-proofing is traditionally enforced by means of password protection [13]. In the case of XBRL, such an approach is not applicable as the raw data is editable and cannot be password protected *per se* without using an arbiter. As a result, a different approach to tamper-proofing is needed to ensure that even in the event of access being obtained to the source data, the data cannot be changed.

Another popular choice for ensuring tamper-proofing is that of encryption [13]. Encryption protects the confidentiality of the data by using a cipher to transform plain text into encrypted text, which is unreadable without being in possession of the key to the cipher function.

Encryption is however also not a viable method to enforce tamper-proofing as the encrypted data does not comply with the XBRL criteria of being human readable.

At this point it is useful to introduce the idea of Digital Rights Management (DRM). DRM refers to "the control and protection of digital intellectual property (content), including documents, images, video and audio" [10]. The purpose of "protecting digital property" is in close alignment with our objective of ensuring that XBRL data is protected from direct access and editing.

The main reasoning behind DRM is to allow the owner of information to control the information usage by the user. This is done in such a way that usage in violation of the usage agreement is not allowed [12]. The Webopedia Computer Dictionary further remarks that this objective is mainly achieved by means of two methods — encryption of the content, which allows for usage only when the correct key is given by the authorised user; and marking the content with a digital watermark in order to verify ownership of the content and validate it [15].

It should be noted that instead of restricting usage of XBRL data, we want to *restrict access to the contents.* In keeping with this objective, it should be noted that (as mentioned earlier) encryption is not a viable avenue of preventing direct access to the XBRL data. The argument of affixing some form of validation to a document by means of *digital watermarking* is however a promising technique for dealing with access control. As such, we will now investigate the concept of digital watermarking in detail.

Bansal and Bhadouria [5] define a digital watermark as "a piece of information which is embedded in the digital media and hidden in the content in such a way that it is inseparable from the data".

Bansal and Bhadouria [5] further note that digital watermarks are used for a variety of applications of interest to the problem of tamper-proofing, such as *digital signatures*, *fingerprinting* and *authentication.*

We now briefly discuss the impact of each of these applications on the problem of tamper-proofing:

Digital Signatures — A digital signature is defined as "extra data appended to a message which identifies and authenticates the sender and message data using public-key encryption" [1]. Digital signatures in the form of watermarks serve to identify the owner of the content and can also be used to indicate the originator of the data.

Authentication — Watermarks can be used in the process of authentication, designed in such a manner that any alteration or editing of the data results in either the destruction of the watermark or a mismatch between the watermark and the content.

Fingerprinting — This technique is employed by using a hidden watermark to establish the creator or owner of the content protected by the watermark.

By combining the applications of a watermark as specified, one can employ watermarks to establish the identity of the last legitimate editor of the document by using Fingerprinting and Digital Signatures. The next paragraph discusses how these techniques can be used as a form of tamper-proofing.

Tamper-proofing is enforced by ensuring that the users of the XBRL data only accepts data last modified by a certain set of individuals. Should the digital signature on the document not match any of the signatures in the set, the document is simply rejected as invalid or non-trusted.

Digital signatures on their own are however not an absolute form of protection as these can be forged. This inherent weakness is addressed by the Authentication application. Authentication involves creating the watermark (containing the digital signature) in such a manner that alteration to the document either *destroys the watermark* or creates a *mismatch* between the content and the watermark. The latter is easily achieved by using some form of hash of the XBRL data in the digital signature, ensuring that modification of the data results in a copy that will be rejected by the requesters of the XBRL data.

The use of watermarks for access denial or tamper proofing is further supported by Kundur and Hatzinakos in [16]. They state that the approach, as outlined in the above paragraph, should be divided into three steps. The first step is to extract the relevant content which should be watermarked — in our case this is the XBRL financial data. The second step is to hash the results from step one in order to reduce the size and to find a unique way of referencing the source data. The last step is to encrypt the hash with the author's private key.

The resulting watermark can now be affixed to the data whilst the original problem requirement (ensuring that the source data is human readable), is adhered to.

Kundur and Hatzinakos, Johnson et al and Bansal and Bhadouria further mention several important requirements to determine the suitability of a digital watermark for the purpose of tamper proofing [16, 14, 5]:

1. The watermark should indicate with a great level of certainty whether tampering has occurred;

2. The watermarks should validate and authenticate the source data without requiring input from an additional source or extra maintenance;
3. The watermark should not cause an alteration of the value of the source data (described as the *fidelity* of the watermark);
4. The watermark should not be susceptible to a high false-positive rate;
5. The watermark should be computationally inexpensive to apply; and
6. The watermark should be tightly integrated with the source data content in order to ensure that it cannot be removed without degrading or destroying the source data.

The last consideration for the watermark's suitability is its *robustness*. Robustness is the quality of a watermark to be resistant to processing and attack that does not affect the value of the source data, such as compression operations [24]. It is clear from the definition that the suitable watermark should be robust.

Research by Kundur & Hatzinakos [16] and Johnson et al [14] suggests that watermarks deal well with the requirements above, and as such it is the authors' opinion that watermarks provide a very viable solution to the problem of tamper-proofing XBRL data.

There are several techniques for the application of digital watermarking, but these are outside of the scope of this article. Please refer to Johnson et al's discussion of digital watermarking in electronic commerce [14] for a full discussion of the subject.

Lastly, it should be noted that using digital watermarks for tamper-proofing is a rather novel application of the watermarking concept [16]. Our research suggests that using the watermarking technique to secure XBRL source data is the first application of the tamper-proofing qualities of digital watermarks for non-image related data.

In the above sections we have derived a set of criteria to address the concerns with XBRL data and we have formalised a process for dealing with version control and the tamper-proofing of data. This was necessary in order to establish a tamper-proof method of capturing meta-data to be used as forensic evidence. Both of these strategies succeeded in establishing a greater degree of certainty with regards to the evidence as per the guidelines set by Casey [9].

We proceed by supplying an overview of the characteristics of the model that embodies these criteria and processes in the next section.

5 Characteristics and implementation of XBRL-Trail

As noted in section 3, there are several core components that give rise to the XBRL trail model. In summary, these are:

1. The need for recording all transactions to data — requiring a central mediator;
2. The need for recording meta-data to serve as forensic evidence — requiring a central mediator, a version control system (needed to record the component of *what happened*) and an authorisation system (which records *who committed the change*); and
3. The need for the safe-keeping of evidence and source data — requiring techniques such as digital signatures and digital watermarks.

Figure 1 illustrates the cohesion between the core components (as mentioned above) in the XBRL-Trail model.

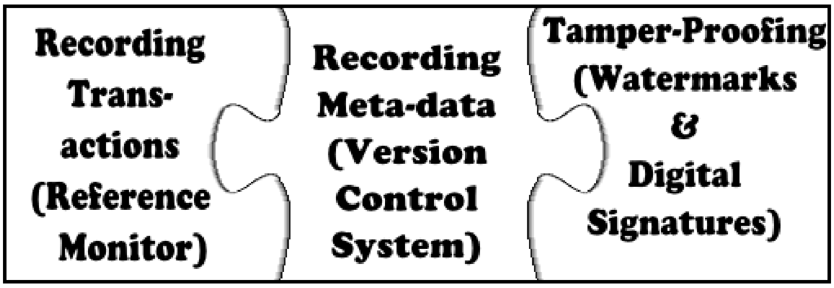
Recording Trans-actions (Reference Monitor)
Recording Meta-data (Version Control System)
Tamper-Proofing (Watermarks & Digital Signatures)

Figure 1: An illlustration of the cohesion between the components of XBRL-Trail.

Let us now examine each of these components and focus on how they can be applied and implemented in the XBRL-Trail environment.

Firstly, let us examine the central mediator. It is of critical importance to note that not any mediator will suffice. Due to the criticality of the financial data that is handled by the mediator, the mediator should be *trusted* to report *all* transactions that occur. These transactions should be reported in an *accurate*, *valid* and *complete* manner — meaning that all transactions should be captured correctly; the record should contain no transactions that did not occur; and there should be only one entry for each transaction. Once again, these requirements form part of the foundations of Casey's evidence certainty [9], as missing entries from log files immediately lowers the certainty rating to C1 on the CCS, which means that very little reliance can be placed on the evidence.

Pfleeger & Pfleeger [20] define such a trusted mediator as "a system that meets the intended security requirements, is of high enough quality, and justifies the user's confidence in that quality".

It is now useful to introduce the concept of a Reference Monitor. A reference monitor is defined as "a portion of code that controls accesses to objects" [20]. As per this definition, the central mediator is in fact the same as a trusted, two-way reference monitor. This is due to the fact that both these constructs function as trusted intermediaries that regulate the flow of data. As such, the authors will from now onwards refer this concept as the reference monitor.

Pfleeger & Pfleeger [20] further state that each reference monitor should comply with three requirements in order to be effective:

1. It should be *tamper-proof*;
2. It should *always be invoked* when access to the info it protects is required; and
3. It should *be small enough* to be subjected to thorough analysis and testing, in order to ensure that all components are functioning correctly and can be trusted.

Secondly, we discuss the need for the retention of meta-data. As we already require a reference monitor, in addition to also requiring the logging of meta-data, it seems logical that we assign the recording of meta-data to the reference monitor as well.

We define meta-data to be consisting out of three components, namely details as to *what was changed*; details as to *who performed the change*; and finally details as to *when the change was performed.*

As discussed in Section 4, we make use of a Version Control System in the reference monitor to log the data pertaining to what was changed.

The logging of who changed it, however requires extra functionality. This is called an authorisation system. Such a system requires the definition of two concepts, namely authorisation and authentication.

Gollman [13] defines authentication as the process of ensuring that the credentials (supplied to the reference monitor by the user) are valid. Authorisation, in turn, is defined as the process of ensuring that the user has access to the information requested [13].

Authentication and/or authorisation is typically used to provide access control. Furthermore, this should be used to restrict access from within the reference monitor, only allowing authorised users to access the financial data. This has two advantages — 1) a reduction in the risk of fraud, as only trusted parties are allowed access to the financial data; and 2) a comprehensive evidentiary record, allowing forensics investigators to deduce exactly who had access to the financial data in the event of an incident taking place. This also provides certainty as to the origin of evidence with regards to evidence certainty.

This brings us to the third component of meta-data, namely timestamps. We recommend that in addition to an access log, timestamps should be kept in order to facilitate an evidentiary chronologic timeline of events [6]. In this case, the source of the time-stamp should be trusted as well, in order to ensure that timing cannot be manipulated. This can be achieved in one of two ways — one can either utilise a webservice to obtain the time from an atomic clock website; or one can use the time on the local machine where the reference monitor resides.

Using a web-service requires constant web access, which leaves us with the caveat of what to do should the web access not be available. This renders this option unsuitable for use, as we require a constantly available source of time.

Instead, we advocate setting up the environment for the reference monitor so that only a user with administrative privileges can change the time. Furthermore, it should be ensured that the host server for the reference monitor uses periodic time syncing (preferably to an atomic clock by means of a webservice) to guarantee that the correct time is used. In this way we can trust the timestamp, as long as other controls sufficiently regulate the users with administrative privileges.

Time stamps fulfil a crucial role in investigations as it enables the creation of a time line which in turn enables investigators to accurately reconstruct the events of the crime. Casey [9] states that even small time discrepancies can be important and that accurate time-keeping is thus very important. The quality of proper time keeping addresses the temporal uncertainty (as explained in section 2) which further increases the certainty rating on the CCS.

Our final component is that of the safe-keeping of evidence and source data. This is embodied by the requirement that meta-data should be stored in a tamper-proof format. We do this by storing the data in plain text, as per XBRL requirement and by then utilising digital watermarks combined with digital signatures to protect the data. In doing this, a watermark is created that is invalidated whenever the data changes, as discussed in section 4. By employing this methodology, we do not allow for forgeries and we stay consistent to XBRL's requirement of human readability.

Let us now determine the effect of our model components and requirements on the certainty rating of our available forensic evidence. Due to our safe-keeping of evidence and source data by means of tamper-proofing the evidence, we increase the certainty of the evidence to a very high degree, securing either a C4 (Probable) or C5 (Almost certain) rating.. Furthermore, XBRL-Trail offers multiple sources of evidence in the form of authorisation and authentication information, Version Control information and standard forensic sources (such as Operating System logs). which further increases the certainty rating that can be applied to evidence gathered from XBRL-Trail. Lastly, due to the requirements and integrity of our architecture, XBRL-Trail ensures that a complete capturing of all transactions occur. This characteristic almost eliminates the chance of errors and loss of evidentiary data.

We thus conclude that evidence gathered from XBRL-Trail is worthy of a very high degree of certainty as it contains evidence from multiple sources that is protected against tampering [9]. This is certainly a major enhancement over the relatively low level of certainty afforded to evidence gathered without XBRL-Trail.

We now proceed to investigate the architecture of the XBRL-Trail model.

6 XBRL-Trail architecture description

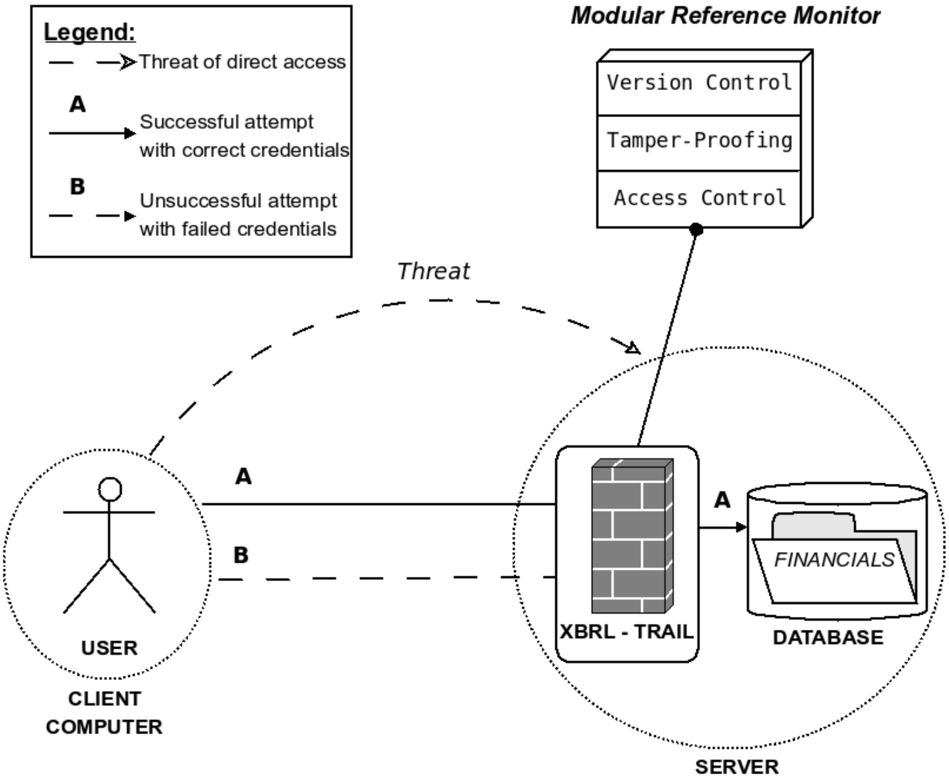

Figure 2: A graphical representation of how the reference monitor restricts and regulates access to the financial data in XBRL format.

So far we can derive that the XBRL-Trail model should be built around a reference monitor that provides intermediary access to the XBRL file. Furthermore, it should not be possible for the user to directly access the file. This is depicted in Figure 2 — access operation A shows the path of an access request where correct credentials are presented; whereas access operation B shows the rejection of an access request with incorrect credentials. Furthermore, the attack vector whereby a user wants to directly access the XBRL source data is shown by the access path entitled *Threat.*

In the case of XBRL-Trail, it is not possible to completely prevent access to the source data. It is however possible to prevent the promotion of an *unauthorised modification* as an *authorised modification*, thereby negating the motive for direct file access.

Furthermore, the reference monitor should be modular in order to allow for different applications, such as recording of evidentiary meta-data in addition to authorising and facilitating access. This calls for a pluggable reference monitor as base, allowing the business owner (who sets up XBRL-Trail) to add on different modules, customising functionality as needed.

XBRL-Trail should be transparent with regards to computational complexity, not adding significant complexity to the system. Keeping in mind that every access operation is logged, the requirement of computational transparency is of significant importance as every transaction requires use of the reference monitor. It is thus clear that even the smallest computational complexity can drive down the system performance as the reference monitor can easily become a bottle-neck.

The reference monitor should also be able to deal with concurrent access requests. We propose a simple system utilising a semaphore and a First-In-First-Out queue associated with the semaphore for dealing with concurrent access requests. A semaphore is defined as "a protected variable counting the instances of resource use within a program" [1].

Lastly, the reference monitor should be installed on a separate server, of which the timestamp can be trusted (refer back to section 5 for an explanation). As such, we advocate a client-server type environment, whereby the XBRL data and the reference monitor is located on a separate server, which in turn provides access to clients needing to access the source data. Not only does this ensure that the users cannot modify the timestamp data on the server, it also adds a further level of protection to the source data as users now need to access the server directly in order to modify the XBRL data. This can in turn be regulated by means of restricting access permissions to the folder containing the XBRL data so that only the reference monitor is allowed to access the data.

7 Conclusion

In this article we investigated the Extensible Business Reporting Language and its impact on business. We noted that due to XBRL's standardised and accessible format, it is especially well-suited towards propagating the sharing of financial information between various business stakeholders. This is accomplished by publishing information in a human readable format that can be semantically interpreted by a computer.

As a result of XBRL's inherent openness and human readability, we noted that it is easily modified, which introduces the possibility of fraud. The fraud risk is due to the resulting ease with which unauthorised modification of financial data can take place. The impact of such fraud may be significant to business operations utilising XBRL, as financial statements often determine the market conception of a company in addition to being a tool for assessing the financial health of a company. Such fraud thus presents both a reputational and financial risk.

Cyber fraud is investigated by digital forensics experts who need to rely on forensic data gathered from the crime scene in order to successfully solve the crime. Due to XBRL's basic mark-up structure, very little forensic information is available by default to forensic investigators. XBRL is not forensically ready and some extension and/or modification needs to be effected on XBRL in order to facilitate forensic readiness.

Furthermore, we introduced Casey's concept of evidence certainty [9] — a measurement of how much reliance can be placed upon evidence. We noted that it is the responsibility of the good investigator to detect, quantify and compensate for loss and error in evidence, and to introduce a measure of reliance (together with the evidence) to the court. Furthermore, we established that the default evidence available in XBRL rates very low on the Casey Certainty Scale, making it rather unsuitable for presentation in a court of law.

We proceeded to derive the foundation for the requirements needed for a model to successfully solve the problem of evidence availability, by addressing several core needs. These are: 1) the need for recording all transactions to data; 2) the need for recording meta-data as forensic evidence; and 3) the need for the safe-keeping of evidence & source data.

The model addresses these risks by utilising a reference monitor that controls access to the XBRL source data and that uses version control, timestamps and authorisation to record the meta-data that is necessary as forensic evidence. Furthermore, the data is kept safe by means of digital signatures and and the application of a digital watermark.

By utilising these different concepts, we succeeded in proving that a significantly higher level of reliance or certainty can be placed upon forensic data that is gathered from XBRL-Trail. Due to our model's addressing of temporal uncertainty and uncertainty of origin, as well as establishing evidence from multiple sources and the establishment of a reliable time-line, our data is proven to be highly reliable. As a result, our evidence reliability is rated very high, compared to the very low level of certainty (C1 classification) achieved by forensic data available from XBRL under normal circumstances.

At this point it should be noted that this solution is still in the concept phase and has not yet been implemented. It is not possible at this time to present empirical results regarding XBRL-Trail's practicality.

The resulting model presented in this article seems to adequately address the identified research problem, in turn motivating further work in refining the XBRL-Trail concept. Currently we are particularly interested in accurately assessing the practical validity of the proposed reference monitor by establishing and evaluating a prototype of the model.

Furthermore, definite potential exists in developing further modules to enhance the functionality of the reference monitor. Examples include a module to introduce role-based access which enforces segregation of duties and a module to apply custom logic to each transaction type.

Finally, we would like to develop a viable method for efficiently and accurately auditing the validity of atomic transactions in real time. This particular problem domain would require knowledge from a vast number of disciplines to solve it, namely algorithmics, compiler theory/semantics and accounting, to name but a few [17, 8].

References

[1] Free on-line dictionary of computing. Online, Mar. 2001. http://foldoc.doc.ic.ac.uk/foldoc/Dictionary.gz.

[2] An Introduction to XBRL. Online, Feb. 2007. http://www.xbrl.org/WhatIsXBRL/.

[3] *Oxford English Dictionary.* Oxford University Press, Mar. 2007. http://dictionary.oed.com/cgi/entry/50088116?

[4] R. Al-Ekram, A. Adma, and O. Baysal. diffX: An algorithm to detect changes in multi-version XML documents. *Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research*, 2005.

[5] A. Bansal and S. Singh-Bhadouria. Network security and confidentiality with digital watermarking. In *IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2007)*, page 325 to 328. IEEE, 2007.

[6] V. Baryamureeba and F. Tushabe. The Enhanced Digital Investigation Process Model. *Digital Forensics Research Workshop*, Aug. 2004.

[7] P. Baudis. Git Fast Version Control System. Online, June 2008. http://git.or.cz/.

[8] J. E. Boritz and W. G. No. Security in XML-based financial reporting services on the Internet. *Journal of Accounting and Public Policy*, 24:11 to 35, 2005.

[9] E. Casey. Error, uncertainty and loss in digital evidence. *International Journal of Digital Evidence*, 1(2), 2002.

[10] Commonwealth of Australia. Resources glossary, Jan. 2008. http://www.acma.gov.au/WEB/STANDARD..PC/pc=PC_1498.

[11] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing*, page 63 to 74, Sept. 2000.

[12] J. S. Erickson. Fair use, DRM, and trusted computing. *Communications of the ACM*, 46(4), Apr. 2003.

[13] D. Gollman. *Computer Security.* Wiley and Sons, second edition, 2005.

[14] N. F. Johnson, Z. Duric, and S. Jajodia. A role for digital watermarking in electronic commerce. *ACM Computing Surveys*, 1999.

[15] Jupitermedia Corporation. DRM. Online, 2008. http://webopedia.internet.com/TERM/D/DRM.html.

[16] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper-proofing and authentication. *Proceedings of the IEEE*, 87(7):1167 to 1180, July 1999.

[17] C. Nobes. Rules-based standards and the lack of principles in accounting. *Accounting Standards*, 19(1), Mar. 2005.

[18] M. S. Olivier. *Information Technology Research.* Van Schaik, second edition, 2004.

[19] G. Palmer. A Road Map for Digital Forensic Research. Technical Report DTR-T0010-01, DFRWS, 2001.

[20] C. P. Pfleeger and S. L. Pfleeger. *Security in Computing.* Prentice Hall, third edition, 2003.

[21] R. Rowlingson. A ten step process for forensic readiness. *International Journal of Digital Evidence*, 2(3), Winter 2004.

[22] J. Tan. Forensic readiness. *Proceedings of the CanSecWest Computer Security Conference*, Apr. 2001.

[23] T. Tan, T. Ruighaver, and A. Ahmad. Incident Handling: Where the need for planning is often not recognised. *Proceedings of the 1st Australian Computer, Network and Information Forensics Conference*, 2003.

[24] J. Zhao and E. Koch. A generic digital watermarking model. *Computers & Graphics*, 22(4):397 to 403, 1998.