Integrity of Government Information
The VERS Experience

Andrew Waugh
VERS Centre of Excellence, Public Record Office Victoria
CSIRO Mathematical and Information Sciences

Executive Summary

This submission is concerned with the use of digital signatures to protect the integrity of government information over long periods (say greater than five years). Our experience with designing and implementing VERS is that digital signatures can be used for this purpose, but the application is considerably different to conventional digital signature applications. Careful thought needs to be paid to how public keys are to be preserved; how modifications to the signed object can be supported; and validation of the implementation. These issues are not considered in the Commonwealth Government's Gatekeeper requirements.

The Public Record Office Victoria and CSIRO have been investigating the use of digital signatures to ensure the integrity of government records since 1998 as part of the Victorian Electronic Record Strategy (VERS). Currently, there is an operational VERS system in the Victorian Department of Infrastructure using digital signatures to ensure the integrity of information generated in that agency.

Digital signatures are an important tool in ensuring that preserved digital objects retain their integrity; that is, they have not been modified in an unauthorised fashion after they were created. However, in implementing an electronic record system we have realised that using digital signatures in a preservation application raises challenges not faced by conventional digital signature applications. This submission lists these challenges and discusses the solutions we adopted.

The basic problem is that using digital signatures to preserve the integrity of digital objects over a long period is subtly different to using a digital signature to preserve the integrity of a message traversing a network – a typical application of digital signature technology. The key characteristic of using a digital signatures over a network is immediacy. That is, the signature is checked shortly after it is generated, the message is only locked for a short period, and if the verification fails the message can be resent. When using digital signatures for preservation, however, the signature may need to be verified a century later, the preserved object will be locked for this entire time, and there is no possibility of resending or repairing the object if the signature fails.

The consequences of this difference are that it is necessary to:

- Ensure that sufficient information is archived to verify the signatures. In fact, we can use the fact that the objects are stored in an archive to avoid using a traditional public key infrastructure.
- Allow the signed object to change while retaining the ability to verify the original digital signatures.
- Validate the digital signature using an independent implementation.

None of these issues are insolvable, but they do indicate the care that is necessary when using digital signature technology to ensure the integrity of government information.

In addition, we have considered the usefulness of non-repudiation and the issues around non repudiation and expiry of certificates.

Table of Contents

1. Introduction

The Public Record Office Victoria and CSIRO have been investigating the use of digital signatures to ensure the integrity of government records since 1998 as part of the Victorian Electronic Record Strategy (VERS). Currently, there is an operational VERS system in the Victorian Department of Infrastructure using digital signatures to ensure the integrity of information generated in that agency. In designing and implementing VERS we have encountered a number of issues with using digital signatures to ensure the long term integrity of electronic information. This submission summarises these issues. None of these issues are insolvable, but they do indicate the care that is necessary when using digital signature technology to ensure the integrity of government information.

VERS is designed to preserve electronic records for very long periods. Records are defined as information produced by an organisation for the purpose of its business (more formal definitions are given in the next section). Most government information consequently falls within the scope of records. Archival theory has the concepts of authenticity, reliability and integrity. An authentic record is one that is what it purports to be (e.g. was created by the person who apparently created the record). A reliable record is one that accurately documents the facts that occurred. Integrity is the ability to prove that no unauthorised modifications have been made to the record. With paper records the issues of authenticity, reliability and integrity are largely addressed by the policies and procedures used to create and manage the records and not by technology [InterPARES]. This approach is currently being extended to electronic records by InterPARES, a significant international theoretical research project.

From a practical perspective, electronic records are already being managed within record management systems based on approaches used in paper records. However, these systems do not seem to be widely used to manage electronic representations of emails and office electronic documents. These records are among the important in an agency as they document the development and application of policy. Our major concern, however, with the current use of these systems to manage the authenticity and integrity of records is that these systems have a finite life which is typically much shorter than the life of the record. Transferring records between management systems make the proof of authenticity, reliability and integrity complex. VERS takes a data centric approach to authenticity and integrity instead of this system centric approach.

A major design goal of the Victorian Electronic Records Strategy (VERS) is to ensure the preservation of electronic records without depending on software applications such as records management systems. In the VERS approach, authenticity and integrity is shown by the record itself, independent of the system that holds and manages the record for the time being. Digital signatures are a key technology in the VERS approach.

Over recent years some preservation literature has suggested the use of digital signatures to prevent objects from being undetectably modified. Hedstrom [Hedstrom], for example, suggests that digital signatures are part of a set of tools that can ' maintain the physical and intellectual integrity of the records'. Lynch [Lynch] has pointed out that using a digital signature is equivalent proving the authenticity of your copy of an object by comparing your copy with a master copy - the master copy being the object at the point of time when it was signed, and your copy either being the same object at a later time, or a different copy. The OAIS reference model [OAIS] discusses 'fixity' information and technology amongst which can be included digital signatures.

When we were developing the Victorian Electronic Record Strategy (VERS) [VERS1] in the late nineties we used digital signatures to detect modifications to the preserved objects. Subsequent implementation experience with VERS within the Victorian Department of Infrastructure (DoI), however, has highlighted a number of practical challenges in using digital signatures in preservation work.

Fundamentally, the problem is that using digital signatures to secure an object over a long period of time is subtly different to conventional applications of digital signatures. Conventional applications protect messages, such as Web pages, as they are being transmitted across a network. A characteristic of this usage is that the messages are checked for corruption very shortly after they are signed. If corruption occurred the message can be resent, and there is no need to keep information around for long periods. Compare this with the preservation usage where it is not

possible to 'resend' the message and the information required to verify the signature must be kept for as long as the item is preserved.

The submission describes a number of challenges in applying digital signatures in a preservation context, and outline the approaches we have taken to surmount these challenges. These challenges include:

- The requirement to preserve the necessary public keys to verify the signatures over a very long period of time. This is a challenge as conventional public key infrastructure organisations are unlikely to be suitable for preserving the necessary certificates for a century or more. We use the fact that an archive contains many examples of records signed by a user to verify the authenticity of the certificates which contain the necessary public keys.
- The difficulty of simultaneously allowing preserved objects to be modified while ensuring that the original digital signatures remain accessible. It is common to wish to modify preserved objects, and this is perfectly permissible provided the modification is authorised and documented. However, modifying an object protected by a digital signature means that the original digital signature can no longer be verified. VERS provides a mechanism whereby records can be modified while ensuring that the original digital signature can continue to be verified.
- The issue of ensuring that the software that produces the digital signature actually generates correct signatures. In implementing VERS at DoI we discovered that the software we were using did not generate valid digital signatures, although the software could verify signatures it generated. This is a particular issue as preserved records may not be validated by an independent implementation for years. We strongly recommend that any system that needs to keep digital signatures for a significant period of time have the signatures tested using a completely independent implementation.

In addition, we consider two related issues that are not problems per se:

- Whether it is cost effective to use digital signatures for non-repudiation. In VERS we eventually decided that issuing individual private keys to users to support non-repudiation was not cost effective. Instead, the record keeping system signed each record when it was registered into the system. As part of this process the system recorded the account which registered the record. We felt that this gave most of the benefits of having individual users sign records at a fraction of the cost.
- The treatment of expired and revoked certificates. Essentially, VERS ignores the expiry dates of certificates as otherwise this would make all signatures fail after a short period of time. We do not use the revocation of certificates as the meaning of a revoked certificate is questionable.

These issues are not addressed in the Commonwealth Government's Gatekeeper requirements [Gatekeeper] which seems to be focussed on the conventional use of digital signatures to secure communication.

It should be noted that the ability to detect modifications to a digital object addresses only one aspect of the authenticity of the object. Other techniques must be used to address the other aspects and these techniques are beyond the scope of this submission.

2. Records and Government Information

As the name implies, Victorian Electronic Records Strategy (VERS) is concerned with records and archives. Archival theory has extensively considered the issues associated with integrity of electronic records.

Records are defined by Australian Standard AS4390 as 'recorded information, in any form, including data in computer systems, created or received and maintained by an organization or person in the transaction of business or the conduct of affairs and kept as evidence of such activity' [AS1]. The subsequent International Standard has a similar definition: 'information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in the transaction of business' [AS2]. As can be seen almost all information held by government can be classified under these definitions as a 'record'.

Governments hold a large amount of records on a wide variety of computer systems. These records range from financial information, through databases, to email and office documents. The computer systems in which these records reside range from application specific applications such as financial systems, case tracking systems, and human resource systems which are often custom built for agencies, through generic applications such as email systems, to the file system which most users would not even consider as an application.

In considering records, archivists and records managers distinguish between an authentic record, a reliable record, and the integrity of a record. These terms are defined in the previously referenced standards.

An *authentic* record is one that can be proven to be what it purports to be (i.e. the content is what it appears to be, it was created by the person who appears to have created it and it was created at the time it appears to have been created). A *reliable* record is one which contains a full and reliable representation of the facts which the record documents. Note that a record can be authentic, but not reliable. For example, the author of the record could have left out material facts, misrepresented the position, or simply lied. Such a record would not be reliable, but would be an authentic record as the content is as the author intended and it was created by the apparent author at the apparent time. Authenticity is concerned with the truth of the record as an object; reliability is concerned with the truth of the contents of the record. *Integrity* refers to the record being complete and with no unauthorised alterations. Note that records can be altered and retain their integrity provided the alterations are allowed by policy, are authorised, and are documented.

These three properties, authenticity, reliability, and integrity, are independent of whether the record is paper or electronic. In a traditional paper based system these properties are largely demonstrated by the procedures involved in the creation, storage, and handling of the record. For example, reliability is largely shown by the fact that the record was created by an organisation at the time of the event for its own future use; records that are unreliable could not be used as the basis for future work. Authenticity is often shown by the record being in the custody of an archive or records management system since creation. Ultimately, these procedures are backed up by conventional forensic tests; for example tests on signatures, the age of the paper, type of typewriter, and ink.

This reliance on procedures can be transferred to many electronic records, particularly those managed by application specific systems. Consider a financial system, for example. Only authorised users can perform actions within the system and all actions are logged. The records would be considered reliable because they are automatically generated by the system as a side effect of carrying out financial tasks. They are authentic because the actions can only be carried out via the financial system and the system keeps logs of who carried out the task, when it was carried out, and how the tasks are related together. Finally, the logs record any changes to the records and hence the records have integrity. Because these specific applications are usually designed to satisfy legal and accounting principles, we would expect them to satisfy archival requirements for authenticity, reliability, and integrity.

However, many electronic records in are not managed in such a formal way. This particularly applies to those records held in generic software applications (e.g. email systems) or on the general file system. Fundamentally, the problem is that these systems are not designed to ensure authentic

records or to ensure their integrity once created. These records can be the most important held by an agency; for example, they may document the development of government policy.

One method of ensuring authenticity and integrity of these records is to install an application that is designed to manage records and to ensure their authenticity and integrity (a records management system). Once records are registered with the records management system the system can ensure that the record is authentic and retains integrity. Essentially, the records management system acts as a vault, mediating and recording access to the records. Just like the financial system, the records management system only allows certain operations on the registered records, only allows authorised users to perform those operations, and keeps audit trails of all operations.

However, there are several issues with using a records management system to ensure the reliability and integrity of records:

- It depends on users placing their records under the control of the records management system. At some point, for example, they must move their emails from their mailbox to the records management system. The difference with a financial system, for example, is that agencies use a financial system to carry out the tasks associated with managing money, the records are automatically generated as a side effect. With a records management system the tasks are carried out in other applications and users have to consciously decide to place the records under the control of the records management system.
- Care needs to be taken that users with special access cannot subvert the system. Such users might be the records administrators or (computer) system administrators. However, it should be noted that such users can equally subvert traditional paper based records systems, so this issue is no different in the electronic environment. The question is whether advantage should be taken of technology to close this hole.
- Management by a records management system should be viewed as a medium term solution. Any computer system has a relatively short life – say five to ten years – and there must be a plan to extract records from a system and to migrate them to a replacement system (or to manage them by some other mechanism if there is no replacement system). This migration is likely to be complex as it is necessary to preserve sufficient information to show that the record was properly managed to ensure authenticity and integrity when under control of the original system. A particular concern about migration is that this may have to occur under extreme time or budgetry constraints. Typically this would occur if an agency (or section) was closed and the records were no longer considered of operational interest. An example would be a Royal Commission. Funding for migration is likely to be minimal in these circumstances, and the time available for migration very short.

VERS was specifically designed to address the final two issues.

3. The Victorian Electronic Record Strategy

The Victorian Electronic Record Strategy (VERS) addresses the issues concerned with the long term preservation of electronic records. It is a strategy, not a system. VERS is a suite of recommendations, practices, and standards that deal with the issues of preserving access to electronic records for very long periods of time (a century or more). The techniques, however, have applicability for records that need to be kept accessible in the medium term (say over five years).

There are four basic issues that VERS addresses:

- Media Deterioration and Obsolescence. Records are ultimately stored on physical pieces of media (e.g. magentic discs, tapes, CDs). These deteriorate over time and ultimately become unreadable. This could occur over a very short period (say five to ten years) if the media was originally of poor quality or is kept in poor storage conditions. Even if the media remains readable, the rapid rate of development of computer storage technology means that it may be impossible to find the hardware necessary to read the media. For example, is it unlikely that CD readers will be obtainable in one hundred years, or possibly even ten years. The obsolescence of media can occur with disturbing speed, particularly if a technology is produced by one vendor that ceases to trade or decides to cease supporting the technology.

  The VERS recommendation for dealing with these issues is to actively manage the records; that is to keep the records on-line or near-line. Records should never be written to media and placed off-line on a shelf. Records kept on-line or near-line can be easily and transparently transferred from one technology to another. System administrators perform this task routinely in all computing installations.

- Application obsolescence. Accessing an electronic records requires that the record be interpreted by a software application which renders them for display. Software is dependent on the underlying environment for its operation, particularly the computer itself and the operating system. If the operating environment changes significantly the software may no longer function and the records are lost. It should be noted that this problem is becoming worse as the technology becomes more complex; for a Web browser to work ‘correctly’, for example, all the plug-ins for various formats must be available. Although new versions of software are normally available there is no guarantee that the organisation or individual trying to access records will have the funds to upgrade to the new version. In any case, there is no guarantee that new versions of the software will correctly read all of the old files (note that new versions may be able to read old files but fail to correctly render it). Alternatively, the software may go out of production.

  The VERS recommendation for dealing with this issue to convert the original file into one or more long term preservation formats. This conversion should be carried out as soon as possible after the creation of the record to minimise the possibility of conversion failure. As part of the VERS we have recommended what characteristics distinguish long term preservation formats.

- Loss of management system. A special case of application obsolescence is the obsolescence of the records management system itself. The records management system is an application, like any other, and has a life span. At the end of its operational life, the records in a system must be transferred to a new system. This transfer is of particular concern for a number of reasons. First, the issue of decommissioning a system and extracting the records is often not considered when building or running the system; this is particularly true of bespoke systems that are not based on commercial records management systems. This is a particular issue where the system is being decommissioned because it no longer has an operational purpose; the agency consequently has no strong reason to spend resources transferring the records. Second, the process of transferring is a weakness in proving authenticity and integrity. Authenticity and integrity is normally demonstrated by the fact that the records management system prevents unauthorised alterations to records. But the records may be completely unprotected during transfer between two systems.

  VERS specifies a standard external record format that is locked by a digital signature to ensure the maintenance of authenticity and integrity of records while they are outside the control of a record keeping system. Ideally, we recommend that records be held within a

record keeping system in this standard format. This means that records can be rescued from the system with no loss of authenticity and integrity even with a catastrophic failure of the record management system, or if it is necessary to quickly retrieve the records due to the agency being closed or record management system being turned off. Finally, the VERS standard format is specifically designed to be self documenting. The contents of the record can be examined using the simplest possible tools (i.e. text editors such as notepad) and the format contains textual descriptions of the technology used in the format. This means that it is easy to write programs to process the standard format and extract information from it.

- Loss of context. To interpret a record it is important to know its context; that is, who created the record, why was it created, when was it created, and how it relates to other records. Context has two important functions. The first is to understand the story. When answering a query, it is rare that it will be answered by a single record. Instead, it is far more common to have to read a number of related records. For example, an email may simply approve something; it is necessary to read the previous emails to understand what was approved and why it was approved. Second, knowledge of the context is essential to judge the authenticity of a record. This contextual information is typically not captured or stored by systems not designed as records management systems. Worse, the records documenting an event may be scattered across many systems; for example, some may be emails held in an email system, others may be reports or documents held on the file system or EDMS.

  VERS captures contextual information in a number of metadata elements which are based on the National Archives of Australia's Recordkeeping Metadata Standard [NAA]. These elements are stored in the VERS standard format; this is particularly important as it means that the contextual information cannot be separated from the content of the record.

Digital signatures are consequently a key technology in VERS.

4. Public Key Digital Signatures

For readers unfamiliar with public key digital signatures, we present the following simplified introduction. A detailed technical discussion of the application of digital signature technology can be found in Housley [Housley].

A digital signature has little in common with a physical handwritten signature. In particular, it is not a scanned image of a handwritten signature. Instead it is the result of a mathematical calculation which takes as input the digital object to be signed and a secret known only to the signer. The calculation has the property that the digital signature (the result) changes if the digital object is changed (even in a minor way) or if a different secret is used.

Generating and verifying digital signatures is technically simple.

Signing a digital object commences by 'hashing' the object. This involves running the object through a mathematical function (known as a hash function) to produce a number known as the hash value. The hash function has the property that even tiny changes in the message will result in a different hash value; this is how alterations in the object are detected. The output of the hash function is then encrypted using a secret (a key) known only by the signer to produce the digital signature.

Verifying a digital signature starts by decrypting the signature to give the original hash value. The digital object is then rehashed using the same hash function and the two hash values are compared. If they are the same the digital object has not been changed since it was signed. If the two hash values are different one of the following has occurred (it is not possible to determine which one):

- The object has been modified since it was digitally signed
- The digital signature has been altered (or substituted)
- The wrong secret key was used to decrypt the digital signature

Encryption of the hash value is done using public key cryptography. This type of cryptography uses keys that come in pairs: the public key and the private key. The private key is used to *encrypt* the hash value and the matching public key is used to *decrypt* the hash value. Use of two matched keys allows the signer to keep the private key secret while publishing the public key. It is not possible to work out the private key if you know the public key.

It is worth noting that the term 'digital seal' would have been a more accurate name than ‘digital signature’. Like a conventional seal, a digital signature can be mechanically applied by anyone who has access to the private key. It follows that the security of a digital signature is dependent on keeping the private key secret, just as the security of a conventional seal is dependent on keeping the seal physically secure to prevent misuse.

From the previous discussion it is clear that to verify a digital signature it is necessary to know the signer’s public key. The key here is ‘knowing’; if a forger can convince you to use their public key instead of the signer’s real public key they can convince you that any digital signature is valid. Reliably distributing public keys is undertaken by means of certificates and the public key infrastructure (PKI).

A certificate is a message containing a public key, the identity of the person or organisation who was issued with the matching private key, and the identity of the organisation who issued the certificate. To ensure its authenticity (and to prevent alterations) the certificate is signed by the organisation that issued the certificate. To verify the authenticity of the certificate requires the verification of the digital signature on the certificate; this in turn requires the public key of the organisation that issued certificate. This public key can be found in a another certificate. But this second certificate needs to be validated using a public key from a third certificate (and so on). Ultimately, this chain of certificates needs to be grounded by a certificate that is just trusted. In most desktop computer systems, for example, this chain of certificates is grounded in a ‘root’ certificate that the user manually loads in to the computer.

Issuing a certificate has potential financial and legal liabilities. The certificate appears to tie an organisation or individual to a public key and hence to a private key. The temptation is

consequently to believe that if a particular object is signed by a particular private key then the individual or organisation named in the certificate generated the signature. This is referred to as 'non-repudiation' as the signer cannot repudiate their signature. There are some issues with this belief, which we consider in section 5. However, the potential financial and legal issues have meant that the issuing and management of certificates is normally surrounded by significant safeguards, such as those required by Gatekeeper [Gatekeeper].

Public key digital signatures have the following characteristics:

- Access to the public key is necessary to verify the digital signature. The public key must consequently be accessible for as long as the signed object is kept. The implications of this will be considered in section 6.
- Any change in the preserved object (by even one bit) will result in a different hash value, and hence will cause the digital signature to fail. This implications of this will be discussed in section 7.

5. Identification of the creator of the record

Challenge: One part of authenticity is the ability to prove who created the record. In theory, it is possible to use digital signatures to provide non-repudiation; that is the ability for the creator to 'sign' an object in such a way that they cannot subsequently deny having done so. In practice, the required technical infrastructure is very complex (and hence expensive to implement).

Because the public and private keys are intimately related together, and the public key is tied to an individual or organisation by the certificate, it is theoretically possible to determine who created a digital signature. This allows non-repudiation; that is preventing the signer from subsequently denying that they generated the digital signature. Non-repudiation is a central aspect of testing the authenticity of a record.

In practice, however, there are several issues with this approach to showing authenticity.

The first issue is that digital signatures are really digital seals; they can be mechanically applied by anyone with access to the private key. The strength of non-repudiation is thus critically dependent on the security of the private key. Private keys are far too long for users to remember them and they must be stored on a computer or in a storage device and accessed by the user when they need to create a digital signature. Non-repudiation is thus critically dependent on the security of the computer system or storage device. Unfortunately, this security is typically not high. For example, users walk away from their computer while it is still logged in, share passwords, and share security devices. The security of many desktop computers is not high with many known security holes in the system software. Finally, system administrators usually have rights which grants 'backdoor' access to the contents of computer systems. For these reasons when we were implementing VERS in the Victorian Department of Infrastructure it was felt that digital signatures can only be used to identify the user account that was used to sign a digital record. Who was actually operating the account at the time, however, usually cannot be proven. Identifying the user account that created the records can be done by the system in other, simpler, ways than by using digital signatures.

The second issue is the cost and complexity of issuing and subsequently managing private keys for individual users. The creation of a certificate associating a public key with a particular individual or organisation may have considerable legal and financial implications. Certificates are jointly issued by Registration Authorities (which validate the identity of the organisation or individual requesting the certificate) and Certification Authorities (which undertake the technical task of issuing and managing the certificates). Because of the legal and financial implications, the Gatekeeper requirements covering Registration and Certificate Authorities are onerous [Gatekeeper]. Meeting these requirements means that running an authority is complex and expensive, whether it is done in house or contracted out. In addition to the basic cost of running an authority, there is on-going cost of managing staff: ensuring that they are issued with private keys when they join, cancelling the keys when the leave (or change jobs), and re-issuing keys that have been compromised or lost.

In summary, issuing individual private keys to each staff member is a complex and hence expensive task, and, ultimately, only identifies the account which applied the digital signature. In the VERS implementation at the Department of Infrastructure, we decided that issuing individual private keys was too expensive. Instead, records were signed by the records management system itself and the system recorded the account from which the record was created. We felt that this gave almost the same proof of authenticity as if we had issued individual users at a fraction of the price.

6. Authentication of the Public Key

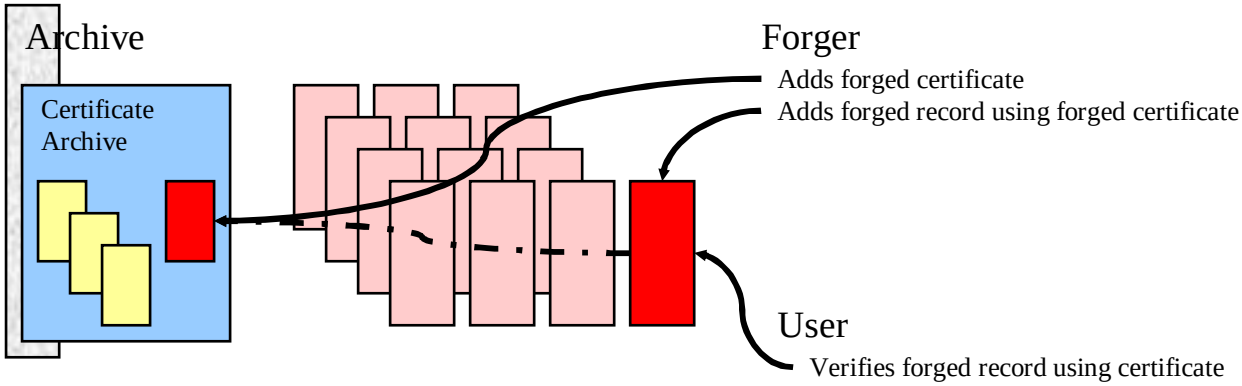
Challenge: Validation of a digital signature requires the authentic public key of the signer. In conventional digital signature applications this is obtained from a public key infrastructure (PKI) consisting of certificates issued by trusted certificate authorities. This is acceptable where the digital signatures have a short life, but certificate authorities are unlikely to keep certificates for the life of a preserved object. The challenge is to preserve these certificates for long periods.

Validation of a digital signature requires the public key of the signer. If the public key has been lost or discarded the integrity of the preserved object cannot be verified. Further, verification depends on being certain that the stored public key actually belonged to the purported signer (otherwise the preserved object could be modified, resigned, and the public key replaced). Public keys must consequently be securely stored for the lifespan of the signed objects; this could be for a century or more. Note that private keys should not be archived; indeed proof of authenticity is improved if it can be shown that private keys are destroyed once their use has ceased.

In a conventional digital signature application, public keys are obtained from certificates produced and stored by certificate authorities (a discussion of certificates and certificate authorities can be found in [VERS2]). However, it is open to question whether a certificate authority can (or should) be trusted to store the certificates it produced for the very lengthy periods of time required for preservation activities. Certificate authorities are usually commercial organisations and there is no guarantee that if the organisation fails or exits the business that the certificate store will be retained. How many commercial organisations are still in existence after 100 years? Note that there is little commercial pressure to provide cast iron guarantees of long term access to certificates as most digital signatures have a relatively short life.

One solution to this challenge requires an organisation holding preserved digital objects to also store the necessary public keys to verify the preserved objects. The public keys would normally be held within certificates. This should not be an onerous requirement as certificates are simply digital objects and can be preserved within the same archive system that manages the actual preserved objects.

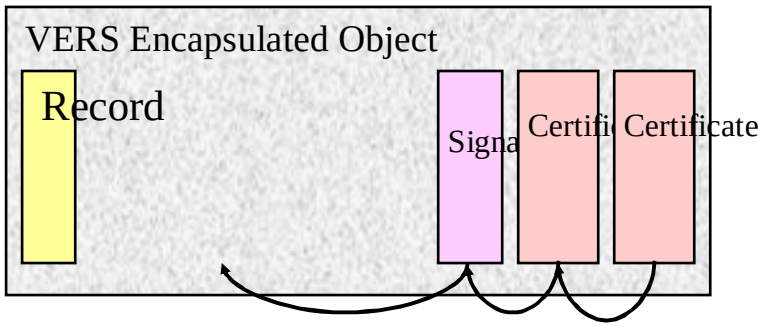
Care needs to be taken that the necessary certificates are actually captured into the system. Custom verification software will need to be written to obtain the certificates from the archive system rather than from conventional certificate authorities. If preserved objects are moved from one system to another the relevant certificates must be identified and moved with the preserved object. Finally, very great care needs to be taken to prevent the unauthorised addition of certificates to the system. If a forger can add certificates to the archive then they can forge or modify any preserved object (just as conventional digital signature applications such as SSL will fail if a forger can convince a user to install a fake root certificate on their computer).



If a forger can add fake certificates to the archive, they can forge any records. In theory, users can detect the forgery by noticing that the certificate used to verify the signature is not the same certificate used to validate other records. In practice, users are unlikely to notice this, but we use this principle as an alternative to verifying signatures.

A second option is to hold the necessary certificates within the preserved object itself. This was a particularly attractive option within VERS as a key assumption of VERS was that preserved

objects would outlive the archive system that held them, so the preserved objects should stand alone from the archive system. Including the certificates within the preserved object reduces the dependency of the preserved object on other objects, ensures that the certificates are captured when the digital object is preserved, and that the certificates are transferred with the preserved object. There are two problems, however. The minor problem is the inefficiency involved in storing multiple copies of certificates, though this is not serious as certificates are quite small. The major problem is that it is not secure. A little thought reveals the circular argument that you are validating the contents of a preserved object by means of a signature which, in turn, is verified by the contents of the object.

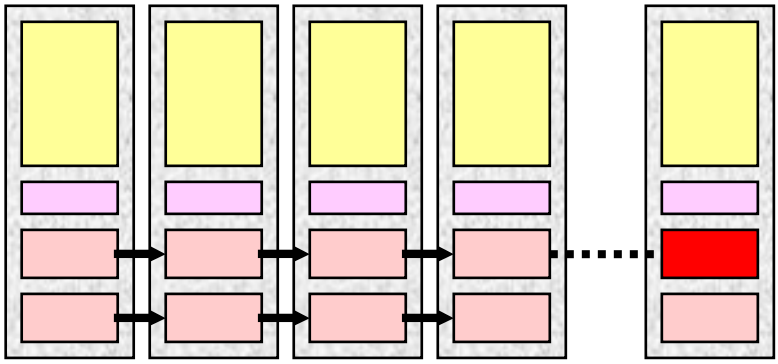


The VERS Encapsulated Object includes the digital signature and all of the certificates required to verify the digital signature. This makes archiving and subsequently managing the certificates trivial. This does not, however, prevent forgery as a forger can simply include their own certificates in the record.

A solution to this circular argument is to discard the conventional concept of digital signature verification by means of a certificate chain. An alternative is to adapt the process used to verify handwritten signatures in a paper based archive. When it is necessary to verify a handwritten signature, the suspect signature is compared with other examples of the signature in the archive. If they match, the handwritten signature is treated as valid, otherwise the signature is considered suspect.

With electronic records we compare the certificates stored with the records, not the digital signatures themselves. Clearly the digital signature will be different for each record (as the signature depends on the record). All the records signed by a user with a particular private key should contain the same certificates.

To verify the integrity of a digital signature on an electronic record the first step is to verify the digital signature using certificates contained in the record. This shows that the content of the record has not changed since the record was signed and that the certificates actually belong to the record. The second step is to choose (at random) another record signed by that user around that time and compare the certificates in the two records. The certificates should be identical. If they are, then either a forger has forged both records or both records are authentic. (In practice the test is slightly more involved than this as a user's private key is periodically replaced and that the certificates will validly change.) Clearly the certificates in the suspect record should be compared with those in more than one record signed by that user; the more records compared the more likely the records are valid. Clearly this is a probabilistic approach, but with a sufficiently large number of digital objects there would be strong evidence that the records have not been tampered with. The security could be increased further by arranging for a record to be signed multiple times.



The four records on the left are probably valid as they were signed using the same private key; this is shown by the fact that they contain the same certificates. The record on the right was signed using a different private key, as shown by the fact that it contains a certificate that is different to the other records in the archive. This makes that record suspect. Clearly this is a probabilistic argument, but given a sufficiently large number of records, the proof can be quite strong. It can be strengthened by signing each record multiple times.

This is a particularly useful approach as it exploits the strengths of the archive and avoids having to trust the integrity of an archive of certificates. In the VERS system at the Department of Infrastructure where the system signs all records, there should be a very large number of records containing the same certificates.

One useful side effect of this 'comparison' verification is that it is possible to demonstrate when a record was created – essentially providing a notarization service. This is achieved by setting up a policy and procedure whereby the private key used to sign the records is regularly changed; say every month. Once the private key is changed the old private key is destroyed (in fact, the copy of the private key should be destroyed once it is loaded into the system and it should not be possible to extract the private key from the system). Every record signed during that period would contain the same certificates. Each record must have been created in that period (as the necessary private key is not available at any other time).

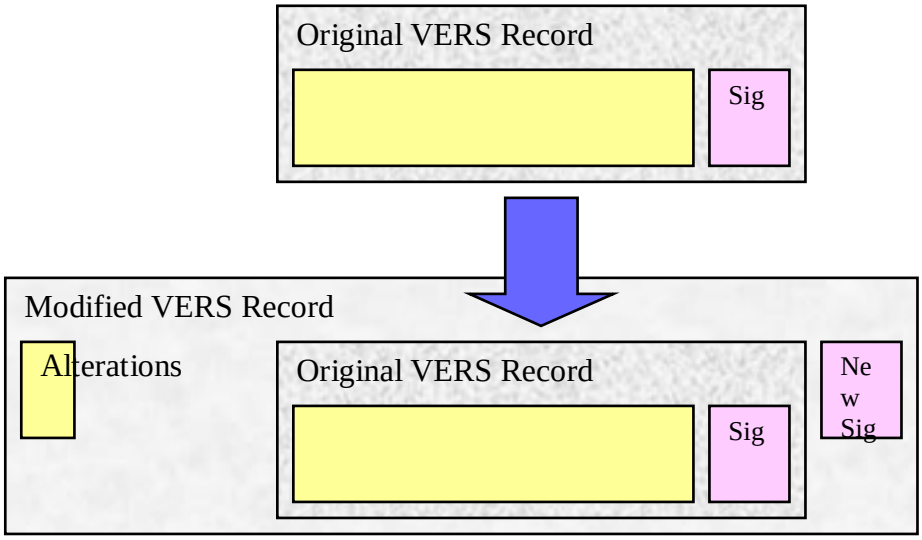
7. Changing an unchangeable object

Challenge: Digital signatures detect any change to a signed object; but it is not possible to discover what has been changed. This means that when an object has been locked by a signature, any change (even valid changes) will break the digital signature and it will be henceforward useless for protection.

A digital signature locks the signed object. The signature will detect any modification to the signed object – be it a change to one bit or to thousands of bits. When signature verification fails it is impossible to determine what has been changed or how much has been changed; all that is known is that something has changed.

This would be irrelevant if it were never necessary to modify a preserved digital object. This, however, is not the case. A preserved object within VERS, for example, contains a complete collection of metadata for the record (this supports our goal of being independent of the record management system that holds the object). It is quite legitimate to change some of this metadata, for example, to correct a spelling mistake, or to add additional descriptive information. This metadata forms the context of the preserved object (i.e. what it is and how it relates to other preserved objects), and in archival theory the context of a preserved object is as important as the content of the preserved object. For example, changing the date a record was created may be more important for a forger than changing the content. It is also appropriate, in some situations, to change the content provided that the change is authorised and documented.

Many archival systems manage the metadata about a preserved object separately from the content of the preserved object. The metadata is directly held by the system; this allows it to be modified and protects it from unauthorised modification. However, apart from making the preserved object dependent on the archive system, this raises the questions: if the archive system can be trusted to protect the metadata from modification, why can it not be trusted to protect the content as well, and if it can protect the content directly, why do we need digital signatures at all?



To modify a signed object VERS creates a new object that contains the original object (complete with original signature), the alterations, and a new signature. Because the original object is included unchanged, the original digital signature can be validated at any time. The new signature has to cover a portion of the original record (the original signature would be sufficient) in order to prevent the original record from being replaced. This technique of encapsulating the changes around the original record can be applied repeatedly. Various mechanisms can be used to reduce the amount of space required to store the alterations.

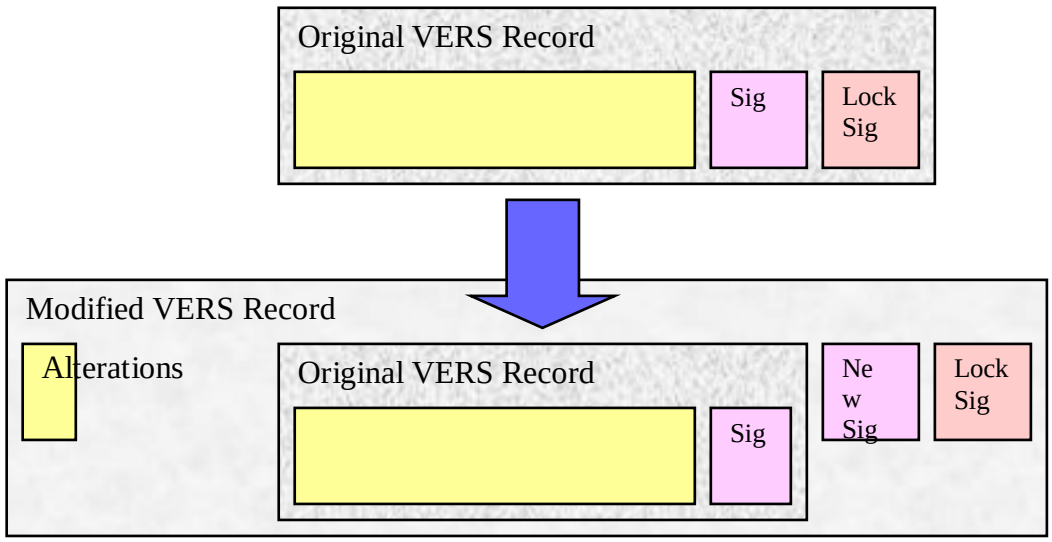
In VERS we made the choice that the archive system should not be trusted to protect the integrity of either the content or the metadata. We consequently faced a complex challenge. We wished to use a digital signature to detect unauthorised modifications to both the metadata and the content of

the record. On the other hand, the digital signature prevented users from performing legitimate modifications to the metadata.

The technique we used to resolve this contradiction is to always retain the original preserved object intact with its digital signature. Thus, it is always possible to verify the integrity of the preserved object. Modifications are added outside the original preserved object and protected by their own digital signature. The original object and the collection of modifications are integrated into a single object by digital signatures.

One problem with this approach is that the original object is still a valid object within the modified object. A forger could discard any modifications by simply extracting the original object and discarding the modifications. The solution to this problem is to remove a part of the original object when it is modified so that it is no longer a valid object; the original object consequently cannot be extracted from the modified object as a valid object. The part that is removed cannot be covered by the digital signature of the original object (otherwise the digital signature will no longer verify). It also must not be possible to be subsequently be recreated (otherwise it could simply be added back to the object).

The answer to this problem is to add a special 'lock' digital signature. This is applied when the object is created, and removed when the object is modified. A new lock signature is applied to the modified object. When examining the authenticity of an object, the first test is to examine if a lock signature is present. If it is not present then the object is invalid. The lock signature is tested for validity and the identity of the signer is checked (this prevents a forger from just applying a lock signature). The lock digital signature must cover a different part of the object than the standard integrity signatures (otherwise the outer signature would be identical to the integrity signature and hence could be easily added again).



A lock signature is a normal signature that only appears on the outermost (most recent) layer of changes and it is discarded when the original record is encapsulated within a modified record. This prevents a forger from extracting the original record and discarding the modifications; if this happened the 'record' would lack the lock signature. The lock signature must not be easily replaced nor forged. VERS recommends that the lock signature be generated by the same private key that generates the normal integrity signature, but over a slightly different content.

We came up with two practical implementations of this general technique which we call the onion and the skewer. In an onion implementation the modified object is wrapped around the original object. Modifications can be repeatedly applied and the resulting object resembles an onion with the original object at the centre surrounded by successive layers of modifications. The digital signature at each level protects the modifications at that level from change and also protects the relationship between this version of the object and previous version of the object. In a skewer implementation, the modification is appended to the end of the original object and protected by a digital signature. A final digital signature is then applied to 'lock' the original object and the

modifications into one object (this lock signature can be used as the outer signature). When it is necessary to add a new modification the archival system checks the lock digital signature (to ensure the integrity of the object) and discards it. The new modification is added and the resulting collection relocked by a new digital signature. This approach prevents a forger from discarding modifications, but requires the archival system to be trusted while the object is 'unlocked'. It is possible to come up with subtle modifications of this scheme to improve utility: for example the lock signature might only cover the digital signatures of the components of the object. This provides equivalent protection, but is much faster to calculate as much smaller amounts of data need to be signed.

In either the onion or skewer implementations thought needs to be given as to whether the modifications contain everything (including information that is unchanged) or just the changes. The latter is much more efficient in storage, but requires a complex process to determine the 'current' value of a record. Essentially, the system must start with the original object and then apply each modification in turn to produce the current value. We are currently updating the VERS specification and, after examining this option, decided not to implement it. We felt that it would be far too likely that errors in the system generating the change, or applying the changes, would mean that resulting 'current' value of the record would be incorrect.

The second implementation question is how often changes result in modifications to the preserved object. One extreme is, of course, that each change results in a modification to the object. In practice this is likely to result in very large objects, particularly if the modification contains all information. The alternative is to store changes in the archive system and to periodically flush the changes through to the preserved object. This flush could be done upon demand by a system administrator, when a sufficient quantity of changes had accumulated, or when important changes have occurred.

In summary, providing the ability to modify a signed object while retaining the original digital signature is possible, but requires a certain amount of complexity. It is interesting to trace the development of our thinking in VERS. Initially we were not going to allow any changes at all. When this proved impracticable, we moved to the onion model with each layer containing all of the metadata. This is the current situation, but when we moved into implementation it was pointed out that the outermost layer could be undetectably discarded. If we were designing VERS today, we would use the skewer model and we might switch to only storing the changes. The lesson we learnt is to think carefully about what is protected by the digital signature and how to efficiently manage changes.

8. Revocation lists and expired certificates

Challenge: To limit the damage a forger can accomplish with a compromised private key, digital signature technologies limit the life of a certificate (hence the associated private key) by including an expiry date, and also have provision to revoke a certificate before it reaches its expiry date. The application of both of these techniques in an archival environment is questionable. Certificates must be retained to validate digital signatures, possibly for centuries – long after the certificate has expired. The meaning of an 'revoked' certificate is also questionable.

The original concept of a certificate was that it would not be necessary to fetch the certificate from the certificate authority's database each time a digital signature was verified. Instead, certificates could be fetched and stored locally for as long as required. This lead to the problem of how to cancel certificates if the associated private key was compromised. The techniques adopted were rather similar to those used for credit cards before the adoption of on-line authorisation: certificates have an expiry date and should not be used after this date; and lists of revoked certificates can be generated and circulated.

The application of both techniques for archived digitally signed objects is questionable. Clearly if a signed digital object is kept for a century the necessary certificates will have long expired. In practice, at VERS we ignore the expiry of a certificate in the archive.

We also ignore revocation lists; although they could be captured in the archive if required. In practice the value of a revocation list is dubious in an archival environment. If a private key is compromised, it will be cancelled and subsequent records signed using a new key. However, there is no reason to then stop using the original certificate to verify records signed with the original private key.

9. Implementation validation

Challenge: Applications that use digital signature libraries can be difficult to implement correctly. It is possible to create signatures that another implementation cannot verify. In a conventional networking application this problem will become quickly apparent due to interworking with other implementations. In a digital preservation application, however, it might be years or decades before the signature is checked by an independent application and the problem comes to light.

This challenge covers two separate problems. The first is that the implementation of the archival application may be incorrect and calculates the digital signature over the wrong object. The second is that the implementation of the underlying digital signature software may be incorrect.

A digital signature will detect any change to the object it protects. This object is ultimately represented as a sequence of octets. To verify the signature it is necessary to use exactly the same sequence of octets that was used when signing. Although this may seem easy, our experience has shown that it is surprisingly difficult to achieve. Octets that have no significance in normal processing (e.g. spaces, line feeds, null bytes) do have a significance when calculating the digital signature and must be carefully processed. This is particularly true when the digital signature only covers a part of the preserved object.

Even when the archive software correctly verifies the digital signatures it calculates, it is worth thinking about whether a future implementator can determine exactly what octets are to be fed into the verification software. Some form of specification is required, and in VERS we include this specification in each preserved object to ensure that it is available to a future implementor. The specification must be unambiguous and the implementation must implement it correctly.

The second problem is incorrect implementations of the digital signature algorithm itself. We did not expect this problem; there are only a few digital signature implementations and they must all interwork in conventional digital signature applications. However, when we tested the signatures generated by the VERS implementation for DoI we could not verify them using an independent implementation (and vice versa). After several weeks work, we discovered that the signature generated by the digital signature software used by the VERS implementation did not conform to the digital signature standard.

These two problems (buggy cryptographic software, and buggy use of the software) are particularly dangerous for digital preservation. Simply testing that an implementation can verify the signatures that it produces does not show that it has implemented the specification correctly, nor that the underlying cryptographic software is correct. This can only be shown by testing against an independent implementation. However, it may be years before an independent implementation exists and until that happens all the signatures a system generates may be incorrect and hence worthless. Part of the acceptance testing for any archival system that uses digital signatures should be a verification of the signatures using an independent implementation. This independent implementation should be implemented by a different software engineer from a specification; this will help ensure that the specification is adequate and that it is correctly implemented. The independent implementation should use a completely different cryptographic implementation to ensure that the implementation the system is using is correct.

10. Conclusions

Digital signatures can be used to preserve the integrity of preserved digital objects, but care needs to be taken in the design and implementation of such a system. Preservation applications have different characteristics to conventional digital signature applications which usually verify signatures shortly after the signature has been applied. Preservation applications require the signature to be verified long after the signature has been applied (possibly centuries later) by software that may not be written for years into the future.

A key design issue of the preserved object is whether it will be necessary to modify parts of the preserved object after it has been signed. If so then these parts must either be outside the protection of the digital signature (and protected by other means) or a the preserved object must be designed allow modification of the preserved object. We have suggested using layers of modifications with each layer digitally signed and a special outer signature on the final layer to prevent discarding modifications. We believe that the requirement to modify a preserved object is the most difficult aspect of using digital signatures in preserving electronic objects.

A second key design issue is to ensure that the necessary public keys to verify the signature are preserved for as long as it is required to preserved the signed objects. Essentially the only method of ensuring this is for the preservation organisation to act as an archive for the public keys. In VERS we store the public keys in the preserved objects, and use a probabilistic approach to verifying these keys where the purported public key is compared with the public key of other objects signed by the same signer around the same time.

The resulting implementation must be carefully tested, ideally by a completely independent test program. Independent means written by separate developers from a specification using different cryptographic software. If the signature is not tested by means of an independent test program it is quite possible that the digital signature will be incorrect.

References

[AS1] Australian Standard, Records Management, Part 1: General, AS 4390.1-1996, Standards Australia, ISBN 0 7337 0306 2

[AS2] Australian Standard, Records Management, Part 1: General, AS ISO 15489.1-2002, Standards Australia, ISBN 07337 4346 3 (This is the Australian badged version of ISO Standard 15489-1, Information and documentation - Records management – Part 1: General)

[Bearman] Authenticity of Digital Resources, David Bearman and Jennifer Trant, D-Lib, June 1998, http://www.dlib.org/dlib/june98/06bearman.html visited 4 March 2002

[CLIR] Authenticity in a Digital Environment, Council on Library and Information Resources, May 2000, ISBN 1-887334-77-7, http://www.clir.org/pubs/reports/pub92/pub92.pdf visited 18 March 2002

[Gatekeeper] Gatekeeper Accreditation, NOIE, http://www.noie.gov.au/projects/confidence/Securing/GatekeeperAccreditation.htm visited 12 December 2002

[Hedstrom] Building Record-Keeping Systems: Archivists Are Not Along on the Wild Frontier, Margaret Hedstrom, Archivaria, No 44, Fall 1997, p 44-71

[Housley] Internet X.509 Public Key Infrastructure, Certificate and CRL Profile, Housley, Ford, Polk & Solo, Internet Engineering Task Force RFC 2459, Jan 1999, http://src.doc.ic.ac.uk/computing/internet/rfc/rfc2459.txt visited 11 December 2002

[InterPARES] The Long-term Preservation of Authentic Electronic Records: Findings of the InterPARES Project, InterPARES, http://www.interpares.org/book/index.htm visited 11 December 2002

[Lynch] Authenticity and Integrity in the Digital Environment, Clifford Lynch, in Authenticity in a Digital Environment, Council on Library and Information Resources, May 2000, ISBN 1-887334-77-7, http://www.clir.org/pubs/reports/pub92/pub92.pdf visited 18 March 2002

[McCullagh] Non-Repudiation in the Digital Environment, Adrian McCullagh and William Caelli, First Monday, Vol 5 No 8 (August 2000), http://firstmonday.org/issues5_8/mccullagh/index.html visited 8 August 2000

[NAA] Recordkeeping Metadata Standard for Commonwealth Agencies, Version 1.0, National Archives of Australia, May 1999, ISBN 0 642 34407 8, http://www.naa.gov.au/recordkeeping/control/rkms/rkms_pt1_2.pdf visited 11 December 2002

[OAIS] Reference Model for an Open Archival Information System (OAIS), Consultative Committee for Space Data Systems, Red Book, Issue 2, July 2001 http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html visited 23 March 2002

[VERS1] Victorian Electronic Record Strategy Web page, http://www.prov.vic.gov.au/vers/ visited 18 March 2002

[VERS2] Standard for the Management of Electronic Records, PROS 99/007 (Version 1), Public Record Office Victoria, April 2000. http://www.prov.vic.gov.au/vers/published/publcns.htm#PROStandards visited 18 March 2002.