Common Format for Information that is Digitally Signed: A Final Report

N. Hastings, W. Polk, and D. Cooper

November 1, 2001

1 Introduction

One of the more tantalizing and elusive goals of the computer age is the paperless office. The ubiquitous paper documents and forms are rarely the end product of an organization, but represent a substantial overhead cost in our daily processes. The paperless office represents an opportunity to increase efficiency of our everyday processes. Capitalizing on this opportunity would permit organizations to perform their work faster, reduce costs associated with storage of documents, and concentrate resources on their core businesses.

Succeeding in this quest is critical to many sectors, including our health care industry. While paper is never the final product, paper forms are present at every step in the process. Paper documents every health care procedure, every billing request, and every prescription. Patients file reams of paper to support insurance claims, and they receive reams of paper in return.

This mountain of paper is at times an impediment to provisioning health care. Processing of paper requests may delay approval of requests for needed care. If the patient is sent for consultations with specialists, the paper charts and analysis must flow as well. Processing of paper may delay reimbursement of patient expenses. Errors in reading or processing paper documents may result in denial of care, errors in filling medications, or mistakes in diagnosis. Scarce resources must be devoted to generating, processing, and transmitting paper instead of providing health care services.

The prospect of the paperless office remains elusive because of one key factor: signatures. One of the most common and widely accepted aspects of paper documents is the handwritten signature. The signature is used to establish who did what and why. Pharmacists rely upon the doctor's signature on a prescription form to ensure that they dispense medicine appropriately and legally. Nurses and doctors rely on signatures in charts to ensure that diagnostics have been performed, medicines dispensed, and to authenticate information in charts. Insurance companies rely upon signatures to identify applicants. People, companies, and our legal system, are exquisitely comfortable with the handwritten signature.

There is no ubiquitous, widely accepted analog for the handwritten signature in the electronic world. The most common analog is the PIN or password, but these are weaker than the handwritten signature. The digital signature is a much stronger analog but has not achieved wide acceptance. There are several factors for the slow acceptance of digital signatures. Some are social factors, and are outside the scope of this document.

One technical factor that is often overlooked is the lack of interoperability between formats for digitally signed information. Most common applications do not directly support digital signatures, so the signatures must be applied after document creation. Once the document has been signed, the document itself cannot be processed or displayed by the application that created it. That is, a signed WordPerfect document cannot be opened by WordPerfect. Even worse, the application of signatures is generally performed by a proprietary process and can only be verified using the same vendor's products.

This paper proposes formats for digitally signed objects that do not alter the document's ability to be processed or displayed by the application that created it. The format is based on open industry standards in hopes that a broad variety of products may be used interoperably to create and verify digital signatures.

1.1 History

This paper describes the final results of a two year research project. The goal of the project was the development of common signed object formats to provide security and interoperability to electronic documents and processes in the health care sector. In year 1, the researchers examined existing signing structures and defined several abstract models for signed objects. In year 2, the research objectives were two-fold: identification of a comprehensive set of digital signing methods that implemented all the abstract models; and definition of a signing structure in both ASN.1 and XML that supported these methods.

The findings and results of year 1 are documented in [5]. The researchers identified six signing models and examined nine standard digital signature formats. In addition, the research reviewed eight current products that provided the ability to digitally sign various objects. With this information, the researchers proposed to develop a common signed object format that could be used independent of the signing application. The signing object would enable interoperability of digital signature verification for each of the abstract signing models.

At the beginning of year 2, new resarchers were assigned to the project. These researchers noticed that the abstract signing models were focused solely on physical structure and the precise correlation between signatures and data. They observed that fundamental concepts from the world of physical documents were not addressed. They were also concerned with the compatibility of the research with existing applications and proprietary data formats. It seems unlikely that application developers will abandon their own internal data formats in favor of a generic format.

In year 2, the researchers began by reviewing the attributes of handwritten signatures on physical documents. After completing this review, they revised the abstract signing models. Next the researchers reviewed the attributes of digitally signed documents to identify the overlap in functionality. Where the digital objects represent a visually representable document, the attributes of physical and digital signatures may be used in a complementary fashion.

This document describes an abstract format for digitally signed objects. The signed object is encoded independently from the digital content. Two concrete formats that implement the abstract design are defined using standard ASN.1 and XML constructs.

2 Attributes of Physical Signed documents

A description of the features of a physically signed document should be reviewed in order to provide the parameters needed to determine what capabilities need to be supported by an electronically signed document. This will help map the physically signed documents to electronically signed documents.

The physical act of signing a document by hand demonstrates the intent of the signer to legally accept the information within a document associated with their signature. In general, each person that physically signs part or all of a document uses a unique mark to indicate their intent to accept specific information within a document. This unique physical mark must contain enough information so that it can be traced back to the creator of the mark using methods such as handwriting analysis or notarization by a trusted third party.

2.1 Visual Context

The visual context of a physically signed document is the only point of reference that verifiers of a signed document have to derive their interpretation of the information being relayed in the document including the acceptance by the document's signers. Some of the visual queues that are found in physical documents are handwritten corrections, location of specific information (date, who received copies, etc.), and the intent of the signer.

The location of dates contained within the document implies different things associated with that date. The date in the heading of a document generally indicates the creation date of the document. The date next to a signature generally indicates the date the signer signed information within the document. The visual queues associated with a signature, including its location within a document, are very important in determining the information signed by a signer. The signature at the end of the document generally indicates the signer accepts all the information within the document. The initials at the side of a handwritten modification generally means the signer accepts the modification. A signature on a line labelled “witness” indicates that the signer is only attesting to fact that they observed someone else sign the document. Signatures are frequently placed in signature blocks that contain text of the form “I hereby affirm that . . . ”, providing explicit information about the signer's intent.

2.2 Linking Content to Signatures

The visual context of a physically signed document has limitations, however. A verifier of a signature on a physically signed document may find it hard to determine which information was present when a signer applied their signature. A skilled malicious person may be able to add, delete, or modify information within a signed document through various techniques after a signer has applied a signature; because a physical signature lacks the capability to explicitly map a signature to the information covered by the signature. In addition, the verifier can have difficulties determining what information the signer was intending to sign. Did the signer mean to sign the original or modified verbiage of the document? When multiple signatures are present in a document, the mapping of the information covered by a specific signature or signatures becomes even more complex. Is the pricing information covered by the sales representative's signature, the sales manager's signature, or both?

With physical documents, it is possible to make changes to the document after signatures have been applied. The verifier cannot always discern which information was placed on the document before the signature, and which information was added at a later date. Of course, the signer was only attesting to the information that was present at the time the signature was applied.

Handwritten corrections and annotations to a type-written, physical document are generally easy to detect and usually have some indication, like the initials, of the modifier and acceptance of the information. The location of a document's information implies much about the specific information.

2.3 Strength of Signature Verification

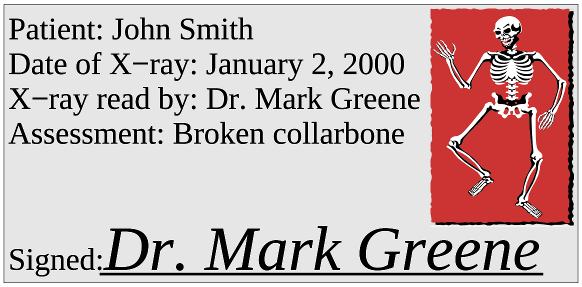
While people are comfortable with handwritten signatures, few can effectively evaluate the validity of a signature. This task depends upon handwriting analysis which is an inexact science. High profile fraud cases, such as the recent "Hitler Diaries", show that even the experts can be fooled. For the average person, handwritten signatures are accepted on an initial identification that a person is who they claim to be and by simple visual inspection of the signature. Only when the authenticity of a signature is brought into question to settle a dispute does rigorous analysis of a person's identity and signature take place.

2.4 Signature and Document Separation

When a document is signed in the physical world, the signature or signatures physically become part of that document. Wherever the document travels, the signature(s) on the document will go with it. So, there is a concept of not being able to physically separate the signature(s) on a document and the actual document which was to be signed. In general, a signature and the document associated with it are physically stored together. The physical inseparability of the document and its signature(s) provides a level of assurance both to the signer and verifier. The signer has the comfort of knowing that his signature cannot be easily removed from the document without some method of forgery. Likewise, the verifier of the document's signature is assured that a signature on a document is somehow associated with the information contained in the document. The verifier only has to interpret the information that was signed and its meaning but does not have to collect and/or construct the information because the signature and document are physically together.

3 Signing Models for Physical Signed documents

The requirement for signatures on electronic documents is not new. The goal of these processes is to support information flow models that currently exist in the physical world. In this section we will describe six increasingly complex usage models for signed documents.



Patient: John Smith
Date of X−ray: January 2, 2000
X−ray read by: Dr. Mark Greene
Assessment: Broken collarbone

Signed:*Dr. Mark Greene*

Figure 1: Annotated X-ray Signed by a Physician

The simplest usage models assume only one signer. The most complex models support multiple signers, each representing different assertions, and the document may change at every stage until the final signature is applied.

3.1 One Signature

The simplest usage models for signatures address situations where only one party needs to sign the document.

3.1.1 Simple Single Signature

This is the simplest usage model and applies where a single individual signs the complete set of information contained in a form, a letter, or a file. Figure 1 shows an annotated X-ray signed by a physician as an example of how this signature type is used in the health care industry.

3.1.2 Single Signature, Selected Content

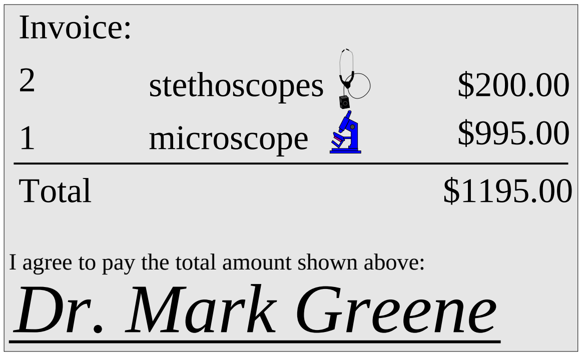
This usage model applies where a single individual signs a selected subset of the information contained in a form, a letter, or a file. In many cases, a form contains a mixture of content from multiple sources. The signer cannot attest to all of the information on the form, only a certain portion. An example of this is shown as Figure 2. The signer is attesting that he will pay the amount shown in the completed total. The signer is not attesting that prices on the receipt are appropriate!

3.2 Multiple Signatures, Static Content

Many paper processes require multiple signatures. This adds new levels of complexity, since different signers may be attesting to different information.

3.2.1 Simple Multiple Signatures

This usage model applies where multiple individuals sign the complete set of information contained in a form, a letter, or a file. In this case, the individuals sign exactly the same set of information and all modifications to the information are made before any of the individuals



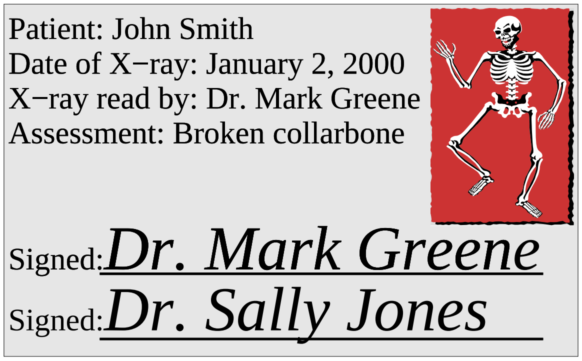
Invoice:

2	stethoscopes	$200.00
1	microscope	$995.00
Total		$1195.00

I agree to pay the total amount shown above:

Dr. Mark Greene

Figure 2: A Signed Credit Card Receipt



Patient: John Smith
Date of X−ray: January 2, 2000
X−ray read by: Dr. Mark Greene
Assessment: Broken collarbone

Signed: Dr. Mark Greene
Signed: Dr. Sally Jones

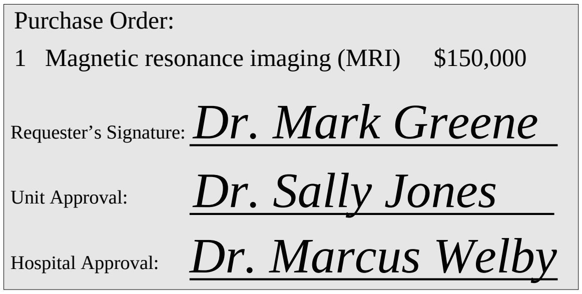
Figure 3: Annotated X-ray Signed by Two Physicians

sign the information. This implies that each signer is acknowledging only seeing the set of information signed, not the fact the other signatures are present. Figure 3 shows an annotated X-ray signed by two physicians as an example of how this signature type is used in the health care industry.

3.2.2 Simple Hierarchical Signatures

This usage model applies where multiple individuals attest to the same set of information plus the set of previously applied signatures. This usage model is important when a document must be reviewed and signed in a specific order, but reviewers cannot change the content. In this case, the individuals all sign the same set of information, with later signatures also signing previous signatures. This indicates that the document is both complete and entirely correct before any signatures are applied.

The first signer signs the basic set of information on the document. The second signer signs the basic set of information along with the first signature. The third signer signs the basic set of information along with the first and second signatures, and so forth. If a subsequent signer finds an error, the information must be modified and the process starts from the beginning. Figure 4 shows an example hospital purchase order signed by the requester and two levels of hospital management.



Purchase Order:

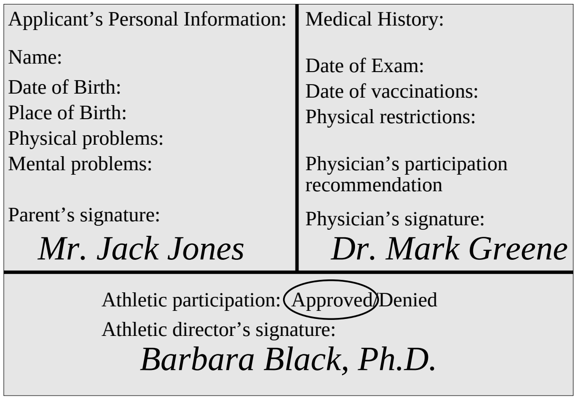
1 Magnetic resonance imaging (MRI) $150,000

Requester's Signature: Dr. Mark Greene

Unit Approval: Dr. Sally Jones

Hospital Approval: Dr. Marcus Welby

Figure 4: Purchase Order with Multiple Hierarchical Signatures



Applicant's Personal Information:

Name:
Date of Birth:
Place of Birth:
Physical problems:
Mental problems:

Parent's signature:
Mr. Jack Jones

Medical History:

Date of Exam:
Date of vaccinations:
Physical restrictions:

Physician's participation recommendation

Physician's signature:
Dr. Mark Greene

Athletic participation: [✓ Approved]/Denied
Athletic director's signature:
Barbara Black, Ph.D.

Figure 5: Application to Participate in Athletic Programs with Separate Signatures for Different Parts of the Application

3.3 Multiple Signatures, Dynamic Content

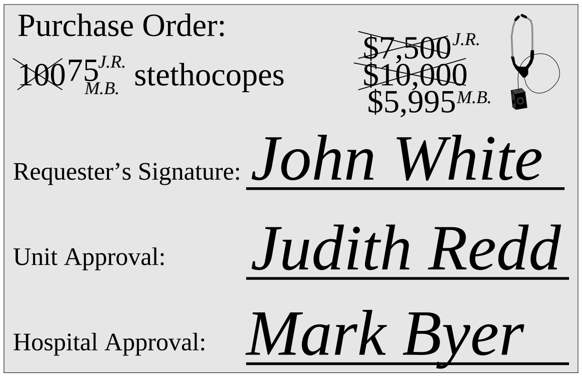
In truth, very few physical documents that require multiple signatures have static content. Multiple signatures imply that multiple people are reviewing and approving the content. The reality of such a workflow is that later signers will wish to modify or amend the content of the document. In this section, we define two usage models where the content is changing.

3.3.1 Multiple Signatures, Selected Content

This format applies where multiple signatures appear in one document, but each signature applies to selected content from the document. The selected content may include previous signatures where appropriate. Figure 5 shows an example athletic program participation application where each signer is responsible for the signing different information.

Mr. Jones and Dr. Greene are signing disjoint sets of information; Mr. Jones is attesting to the personal information and Dr. Greene is attesting to the medical history. Ms. Black is actually attesting to an overlapping set of information: Mr. Jones' and Dr. Greene's signatures, as well as her own approval of the application. (She cannot attest to the contents of either the personal information or medical history, but may be attesting that those sections were completed.)

The content of the document evolves between applications of signatures; it is not stable.



Purchase Order:

~~100~~75$^{J.R.}_{M.B.}$ stethocopes ~~$7,500~~$^{J.R.}$ ~~$10,000~~ $5,995$^{M.B.}$

Requester's Signature: John White

Unit Approval: Judith Redd

Hospital Approval: Mark Byer

Figure 6: Purchase Order with Multiple Signatures and Dynamic Content

Mr. Jones supplies the personal information before signing it. Dr. Greene fills in the medical history before signing his section. Ms. Black must indicate her approval (or disapproval) by circling the appropriate information before she signs the form.

In this case, Ms. Black's signature is hierarchical but Mr. Jones' and Dr. Greene's are not. That is, Ms. Black cannot approve the application without signatures from a parent and doctor. However, the doctor may be first or second to fill in his information.

Note that Ms. Black cannot modify any of the information in Mr. Jones' or Dr. Greene's sections of the document. Each of the signers completes their own section of the document and does not change other fields. If Ms. Black had noticed a mistake, she would have to restart the entire process.

3.3.2 Multiple Signatures, Information Modification

This usage model applies where one document contains multiple signatures but signers can unilaterally modify (add/remove) information prior to signing. This information may be covered by one of the previous signer's signatures.

For example, consider the purchase order in Figure 6. Mr. White has drafted a purchase order for 100 stethoscopes. However, Ms. Redd has reduced the quantity to 75 stethoscopes before she signed off on the procurement. Finally, the purchase price was adjusted when the final order was placed to reflect a negotiated discount before Mr. Byer placed the order.

Mr. White initiates the order to fulfill a perceived business requirement. His signature indicates that he believes the equipment is needed and that he estimated (or confirmed) the listed price. Ms. Redd has modified the purchase request to indicate that only 75 stethoscopes are really required or that the institution can only spare funds to cover that quantity. Mr. Byer accepts that 75 stethoscopes are required based on Ms. Redd's signature, but recalculates the final price using his own pricing schedules or preferred sources.

4 Attributes of Electronic Signed Documents

The electronically signed documents must minimally support the attributes of physically signed documents, so that all the capabilities of physically signed documents can be captured and the signatures can fit into a legal framework based on physical signatures.

4.1 Context

The context of a digital signature has two aspects: the first is the electronic context, the second is the visual context. In combination, these aspects capture the intent of signer by the electronic signature. Unlike a physical signature, where the signer is in control of the instrument used to create a unique mark, the signer must depend a device to create the electronic signature correctly. What prevents the device from creating a signature without the signer's knowledge? In general, a pen cannot create a mark without an external manipulator such as a person.

4.1.1 Electronic Context

A digital signature is generated using a specific stream of data as an input. The signature cannot be verified without precisely reconstructing the data input stream. Digital signatures may be generated over an object as a whole, or may be generated over a subset of the data. Data that falls outside the signature input stream may be modified, deleted, or added without affecting the digital signature. Clearly, the digital signature cannot be construed as making any assertion regarding data outside the input stream.

We will refer to data included in the input stream as the electronic context of a signature.

4.2 Visual Context

A signer cannot examine the data which is contained in the input stream. In the electronic world, everything is represented in ones and zeros, which are not easy for a human verifier to interpret visually. As a result, the signer and verifier of an electronically signed document generally will need to rely on a device to render the document for interpretation. The signer is attesting to information that is displayed within the application that created or modified the signed object. The verifier evaluates that attestation on the basis of the object's displayed characteristics. As with physical documents, both the signer and verifier are primarily concerned with the visual context of their signature.

The electronic world does add some complexity here because two different electronically signed documents may render the same visual representation. Conversely, an single electronically signed document may be rendered into different visual representations depending on the device used to render the document. If the visual representations of a document are different for the signers and the verifiers, there is a risk that the signers' intentions in signing the document will not be properly conveyed to the verifiers.

So, the suitability of an application for use with digital signatures must be considered. While independent of the digital signature mechanisms, the consistent rendering of electronically signed documents is critical to the correct interpretation of a signature.

4.2.1 Contextual Conflicts

The electronic context and visual context of a document are evaluated independently. When the two contexts are in conflict, the verifier can only depend upon the intersection of the two contexts. It is critical that the signature verifier reconstruct the electronic context of a signature, and verify that it is consistent with or is a superset of the visual context.

4.3 Multiple Signatures with Different Context

For electronically signed documents that contain multiple signatures, it is required that each signature be generated over the same information. This is the feature of the signature providing integrity over the information which is signed. However, the smallest change to an electronically signed document requires all of the signers to generate their signature over the modified document. If a change occurs in a physically signed document, the signatures already on the document to not automatically become invalid. This could provide convenience to all the signing parties because they do not have to sign the new, modified document. However, if the modifications are not obvious, such as adding a zero to the end of a dollar amount, the verifier of the signature(s) on the document may not know that one or more of the signers may be unaware of the change to the document.

4.4 Shorthand Mechanisms for Document Modifications

In electronically signed documents, a mechanism to capture a simple change to a document is not as straightforward as with physically signed documents. As noted in section 2.1, it is generally accepted that initials of all signing parties of a physical document next to the modification within a document indicates acceptance of the change by the parties. This provides a shorthand mechanism to handle modification of a document without regenerating the document with the updated information. Currently, there is not an equivalent mechanism in electronically signed documents because one of the features of electronically signed documents is the integrity of the information that is signed. If the information associated with a signature has been modified in anyway, the signature will not verify properly.

4.5 Signature and Document Separation

The ability to separate a signature and the document and information associated with it is easily done with electronically signed documents. Unlike with physically signed documents where the signature is stored with the document and information associated with it, electronically signed documents do not necessarily require a signature and the associated document and information to be stored together. Because the information used to generate a signature may not be located with the signature, the verifier of the signature may have to gather the information needed to verify the signature. The information may not be available to the verifier, in which case the verifier would be unable to verify the signature. On the other hand, having the ability to separate the signature and document provides the flexability to manipulate these two pieces of information. A document can be reviewed without revealing the parties involved and only when a signature is necessary is it provided to a verifying party.

5 Electronic Signing Models

Electronic signatures can be used to provide many of the properties that are available with physically signed documents. Some of the properties that are provided by physically signed documents do not map to electronic signatures very well, however. On the other hand, some properties that are not available with physically signed documents can be implemented with electronic signatures.

5.1 Signature Verification

As was mentioned in section 2.3, it is not possible for the average person to verify that a signature on a piece of paper is not a forgery. With digital signatures, however, a verifier's computer can check that the signature on the document was actually created by the person who appears to have signed the document.

A digital signature is generated by computing a cryptographic hash (thumbprint) of the information to be signed and then protecting that hash using a private key that belongs to the signer. The verifier, who possesses a copy of the signer's public key can verify that the document was signed using the signer's private key. Since the signer is the only one who possesses the private key, only the signer could have created the signature.

Another feature of digital signatures is that they provide integrity protection. If any part of the document is changed after it has been signed, the signature will not verify. As a result, a user who verifies the digital signature on a document can not only determine the identity of the signer, but can also ensure that the document has not been changed since it was signed. As was mentioned in section 2.2, this is not necessarily the case with physically signed documents.

Even though changes can not be made to the document without invalidating the signature, it is still possible to support dynamic content with digital signatures. In order to do so, however, it is necessary to maintain multiple copies of the document. If the second signer of a document makes changes to the document before signing it, a copy of the original document must be maintained in order to be able to verify the first signer's signature. This copy may be in the form of a complete copy of the document or may be a description of the differences between the two versions of the document. By maintaining two copies, one can not only verify both signatures, but one can also unambiguously determine what the document looked like when it was signed by each signer. This is a useful property that may not always be available with physically signed documents.

5.2 Complete vs. Selective Data Coverage

With many physically signed documents, visual context plays an important role in the interpretation of the meaning of the document. In some cases, a signer may not be attesting to all of the information in a document. In a physically signed document, this may be conveyed by the location in which the signature is placed. The signature may, for example, be placed in a "signature block" that states "I affirm that the information in boxes 1 - 5 above is correct to the best of my knowledge."

With digital signatures, it is possible to sign only a portion of a document. When a cryptographic hash is computed, it can be computed over any subset of the document that the signer wishes. By signing only those portions of the document to which the signer is attesting, the signer can ensure that there is no ambiguity about what was being signed.

The problem with this approach, however, is that the signature verification is being performed by a computer, not by the person who will be relying on the signature. It is very difficult, in general, for a computer program to convey to a human user what subset of a document has been digitally signed and what portion of the document has not been signed.

One solution to this problem is to take advantage of the visual context notion from physically signed documents. When a user signs a document, that user can place a mark on the document at the place in the document that he or she intends to sign it. This mark could be a graphical image of a handwritten signature or simply the signer's type-written name. The resulting document, with the signer's mark, would then be digitally signed. The digital signature would allow the verifier to determine that the document being verified is the one that was signed by the signer and to verify the identity of the signer. The visual context surrounding the signer's mark could then be used to determine the signer's intent.

5.3 Encapsulated Signed Data vs. Detached Signatures

With physically signed documents, the signature is applied to the document and then becomes a part of the document. A digital signature, on the other hand, may either encapsulate the object that has been signed or be stored separately from the object. When the digital signature encapsulates the signed object, this most closely models physically signed documents. Encapsulating the signed object with the signature provides some degree of convenience. The signed object may easily be stored or transferred as a single object.

The problem with encapsulating the signed object within the signature is that the resulting object can no longer be processed by the signature unaware application that was used to create the object. For example, if the contents of a WordPerfect file is signed, and the result is written to a file as a single object, the resulting file can not be read by the WordPerfect application. The signature information would have to be stripped from the file before the WordPerfect application would be able to process it.

Unlike with physically signed documents, a digital signature may be stored separately from the object that was signed without diminishing the binding between the object and the signature. If the digital signature is stored as a separate, detached, object from the object (e.g., document) that was signed, then the object that was signed can still be read by signature unaware applications. This means that the document processing application would not need to be modified to handle signatures. Signature processing could be handled separately, by a different application, and users who did not have the signature verification software could still read the document.

The drawback to detached signatures is that it requires two separate objects, the signed document and the signature, to be maintained separately. The signature by itself is useless and the document by itself bears no indication that it was ever signed.

6 An Abstract Format for Digital Signatures

In this section we will derive an abstract format for digital signatures. We will select attributes for the format to satisfy the following requirements:

1. the object covered by the digital signature may be created by any application, and the processing of this object must be unaffected by the signature;

2. the context of the digital signature must be clear; and

3. all the abstract signing models must be supported.

6.1 Maintaining Application Independence

To maintain application independence without affecting processing, we will take advantage of the physical separability of digital signatures. There is no requirement that the digital signature and the signed object be combined. This imposes a penalty in increased overhead, since the system or user must manage two linked objects. We believe the ability to create and modify signed objects with applications that are not signature aware is adequate compensation for this cost.

To achieve this goal, we will use detached signatures exclusively.

6.2 Clear Signature Context

As defined above, the signature context is the intersection of the electronic context and the visual context of a signature. Electronic context is difficult to demonstrate to users. On the other hand, all users are familiar with visual context. Signature context would be clearest if we could ensure that the intersection of the electronic context and the visual context of a signature is always the visual context.

This may easily be achieved through over-signing. If the electronic context is the entire object, the visual context must be the intersection. As a result, we will require that every digital signature be calculated over the entire document as it exists at the time the signature is applied. We will depend upon forms and document designers to clearly indicate intent through visual context.

6.3 Supporting Hierarchical and Parallel Signatures

Some of the signing models described in this document support both hierarchical and parallel signatures. Where signatures are hierarchical, the order of signing is preserved and each signature covers the preceding signatures. Where signatures are parallel, the signatures attest to the same content, but not the signatures themselves.

There are few requirements (if any) for truly parallel signatures. In general, a document flows from one signer to the next. Order may not be important, but there is still a temporal order to the application of signatures. As we are depending upon visual context, it is important to ensure that the visual context be the same as the context of the signature.

The solution is to require all signatures to be hierarchical to the preceding signatures. Hierarchical signatures may be thought of as the temporal equivalent of over-signing. By insisting on hierarchical signatures, we can ensure that the visual context does not provide temporal information that is misleading.

6.4 Supporting Abstract Signing Models

Given detached signatures and over-signing, the next challenge is supporting all of the complex signature models presented in section 3. This is simpler than it sounds. With over-signing, we do not depend upon structure in the signature construct itself. All that is required to establish the context of the signature is reconstructing the contents of the document at the time the signature was generated.

6.4.1 Archive Signatures and Corresponding Documents

The simplest and most straightforward solution assumes unlimited storage capacity. Each time a signature is created, the corresponding document is archived. A “current” copy remains available for processing. When a user decides to verify a particular digital signature, the signature verification module would verify the signature using the appropriate archived document. If the signature verified, the user could determine the signature context by opening the archived document in its native application.

While systems do not have unlimited storage capacity, this may be a viable solution for some cases. Storage has become relatively inexpensive. Thanks to 40 and 80 gigabyte hard disks, most desktop systems have sufficient storage to maintain several old copies of important files. Digital signatures are measured in kilobytes, so adding signatures to the equation should not present a problem.

Still, duplication of files lacks elegance and could have a severe impact as business processes increasingly migrate to paperless systems. Fortunately, more elegant and practical solutions exist.

6.5 Document Recovery Strategies

A more elegant solution maintains the current copy of the document along with sufficient rollback information to support file reconstruction for signature verification and determination of visual context. The rollback technique must reliably produce the same binary image so that the signature verifies using the reconstructed document.

6.5.1 Application Specific Rollback

Rollback/history to support document reconstruction may be implemented using application specific commands if the application uses a distinguished encoding technique. Many applications maintain rollback information for the convenience of users; where applications expose this information this is an attractive solution.

Unfortunately, many applications use encodings that are not distinguished. In such cases, two different documents may be considered equivalent even though the binary encodings are different. For example, whitespace may be compressed in HTML documents. That is, the strings "Bob   White" and "Bob White" are considered equivalent. Applications that use encodings that are not distinguished usually cannot rollback to the same binary image.

Even where applications use distinguished encodings, they may not expose rollback information. This forces us to an alternative solution, where rollback information is created independently from the application.

6.5.2 Application Independent Rollback

Rollback/history to support document reconstruction may always be implemented using an independent binary differences program. The program must be used to maintain a history/rollback as the document evolves. The associated reconstruction program uses the binary differences to reconstruct the binary image at signing time. The user can then verify the signature and establish the signature context using the native application.

Application independent rollback is attractive because it provides a complete solution. Binary differences may be used even where the application does not employ a distinguished encoding technique. However, it should be noted that binary differences programs are far less mature than text-based differences programs.

7 Concrete Formats for Application Independent Digitally Signed Objects

In [5], two basic encoding techniques were identified for the formatting of digitally signed objects. The two techniques were the Cryptographic Message Syntax (CMS) and the eXtensible Markup Language (XML). CMS is a binary encoding technique based on Abstract Syntax Notation One (ASN.1) while XML is a text based encoding technique.

We begin this section by identifying the CMS features that are required to support the abstract structure described in the preceding section. Next we describe the XML-based analog to these features.

7.1 CMS

The Cryptographic Message Syntax (CMS) is a widely used standard encoding for signed data [4]. The CMS syntax is defined using ASN.1. A number of encoding rules have been defined for data that is described using ASN.1, but CMS values must be generated using the Basic Encoding Rules (BER). The Distinguished Encoding Rules (DER) are a subset of BER, and are the most popular encoding rules for ASN.1-based data that is to be signed.

With CMS, a signed document may be encapsulated with the signature or the signature may be detached. As was noted in section 6, we consider detached signatures to be the most appropriate for the given application. A CMS message, when detached signatures are used, consists of list of signers. For the purposes of this application, each CMS message will contain exactly one signer. The signer's information in the message includes that signer's signature and, optionally, a set of attributes. The attributes are associated with the signature and may be either signed or unsigned. The signing time is one example of an attribute that can be associated with a signature.

CMS messages may be used to support models in which more than one signer signs a document. If a CMS message is created in which the data that is signed is itself a CMS message, then the signatures in the "outer" CMS message signs the signatures in the "inner" CMS message in addition to the data that was signed by the "inner" CMS message. This can be used to support the multiple signature models in sections 3.2 and 3.3. By using nested CMS messages, all multiply signed messages are technically hierarchically signed. In other words, each successive digital signature sign's all of the previous signatures. When it is not the signer's intent to sign previous signatures, visual context can be used to convey the intent of the signer.

In order to support multiple signatures with static content, it is sufficient to use nested CMS messages. In order to support dynamic content, a more complicated scheme must be employed. In order for a verifier to be able to determine what version of a document each signature was computed over, the CMS message must provide the verifier with this information. In order to maintain compliance with the standard, this information must be provided as an attribute in the signer's information field. The CMS syntax allows for arbitrary attributes to be associated with signatures. However, only a few attributes are defined within the document that defines CMS [4]. Since none of the standardized attributes can be used to convey the necessary versioning information, a new attribute must be defined.

As was described in section 6.4.1, the easiest way to maintain information about each version of a document that was signed is to store an entire copy of each version of the document. In order to allow a verifier to determine which version of the document each signer signed, the CMS message must include references to each version of the document, except the final version.

Figure 7 shows an example of how our newly defined, reference attribute can be used to maintain version information. In this figure, Mr. White has created and signed a purchase order for 100 stethoscopes. He forwards the document to Ms. Redd, who changes the number of stethoscopes in the order from 100 to 75 and then signs the document. In order to sign the document, Ms. Redd first adds an unsigned, reference attribute to Mr. White's signature in the CMS message that Mr. White created when he signed the document. The reference attribute is a pointer to a copy of a version of the document as it existed when Mr. White signed it. Since the reference attribute is unsigned, Ms. Redd's adding this attribute to Mr. White's signature information will not invalidate Mr. White's signature. After adding the reference attribute, Ms. Redd can then sign the resulting CMS message. When she computes the cryptographic hash for her signature, she will compute the hash over her own version of the document instead of the version created by Mr. White. In other words, she will perform the computation as if the **eContent** in the **encapContentInfo** in the "inner" CMS message contained her version of the document. Figure 7 contains a graphical representation of the resulting CMS message.

Signature verification is performed starting with the "outermost" CMS message and working inward toward the "innermost" CMS message. The "outermost" signature is verified by computing a cryptographic hash over all of the CMS messages that are encapsulated within it. The hash is computed as if the **eContent** in the **encapContentInfo** in the "innermost"

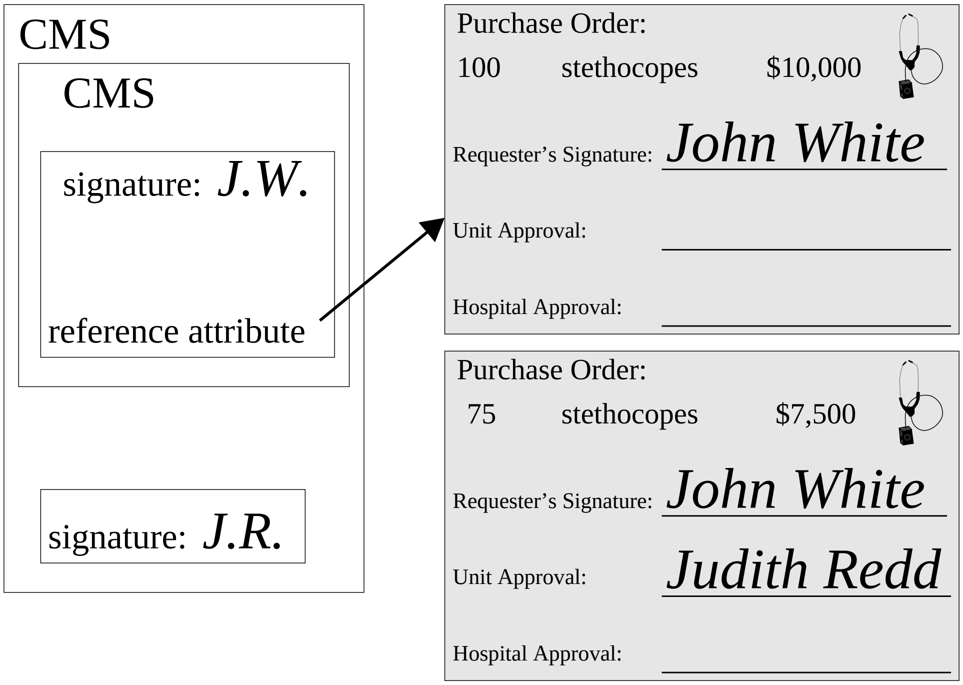


Figure 7: Purchase Order with Multiple Signatures and Dynamic Content

CMS message contained the current version of the document. The "outermost" CMS wrapper is then removed so that the remaining signatures can be verified. If the next signer's information in the next outermost CMS message does not contain a reference attribute, then the signature can be verified just as the "outermost" signature was verified. If the signer's information contains a reference attribute, then the cryptographic hash that is computed for signature verification should be computed as if the **eContent** in the **encapContentInfo** in the "innermost" CMS message contained the version of the document pointed to by the reference. The remaining signatures would then be verified using the referenced document as the current version. It is the responsibility of the verifier's application to display the different versions of the document in such a way that it is clear which versions of the document were signed by which signers.

There are three basic ways in which the reference attribute may refer to a version of a document. One could mimic the concept of encapsulated signatures by including a copy of the document in the reference attribute. Alternatively, the document could be stored separately, and the reference attribute could include a pointer to the document. The pointer could take the form of a Uniform Resource Identifier (URI).

As was mentioned in section 6.5, the reference could also be a differences file. In this case, the reference attribute could contain a description of the differences between the version of the document that was signed by the signer whose signer's information contains the reference attribute and the version of the document that was signed by the signer whose CMS message immediately surrounds this CMS message. Since the differences may be described in either an application-specific or application-independent manner, the reference attribute must specify what method was used to create the differences file. Just as if the entire document had been included, the differences file may either be included in the reference attribute or the reference attribute may simply include a URI specifying the location of the differences file.

If, in the process of verifying a nested CMS message, a verifier encounters a reference attribute in which the signed version of the document is described using a differences file instead of by including the entire document, the verifier simply needs to create the signed version of the document by applying the differences file to the version of the document that was used to verify the previously verified signature.

7.2 eXtensible Markup Language

The eXtensible Markup Language (XML) is a document markup language. XML is a structural and semantic markup language. That is, its tags describe the structure and specify semantics associated with the data in an XML document. This is inherently different from HTML, where the tags describe the presentation of information (e.g., bold or italic font, or the font size). XML documents are encoded as text.

Security was not initially an objective of the XML standards. Security services, such as authentication and confidentiality, were not directly supported. Security services can, of course, be layered on top through complementary protocols such as SSL, but there was a desire for native XML security services. The XML Digital Signature Specification is defined in RFC 3075, XML-Signature Syntax and Processing, or XMLdsig [3]. This specification was designed to "provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere."

7.3 Profiling XML

The XML Advanced Electronic Signatures (XAdES) specification [2] defines an XML format for advanced electronic signatures, including an XML schema for new XML types able to contain the information conveyed by CMS **signedData**. These signatures are built on XMLDSIG with the addition of specifications for two main types of properties: signed properties and unsigned properties. The first are additional data objects that are also secured by the signature produced by the signer on the <SignedInfo> element, which implies that the signer has these data objects, computes a cryptographic hash for all of them and generates the corresponding <Reference> element. The unsigned properties are data objects added by the signer, by the verifier, or by other parties after the production of the signature. They are not secured by the signature in the <Signature> element (the one computed by the signer); however they can be signed by other parties.

The ETSI-defined attributes and schema provide most of the tools needed to encode digital signatures. However, XML is very rich in features, and some of them prove counterproductive to digitally signed messages. The authors suggest that several additional features of XML must be profiled for interoperable and secure use.

First, there are many different ways to encode documents that are functionally equivalent. For example, single and double quotes may be used interchangeably around attribute values in XML. Documents may be reformatted before processing or transmission by clients or servers to facilitate local processing or to compress data. This flexibility is considered a feature, but presents a major impediment to the use of digital signatures. A digital signature can not be verified if the document has been modified.

XMLdsig addresses this problem through the use of canonicalization algorithms. A canonicalization routine translates equivalent XML documents to a consistent state. If the signer and relying party translate the document to the same state (before generating and verifying the digital signature, respectively) then the digital signature will verify. Unfortunately, XMLdsig did not select a single canonicalization algorithm, and a number of algorithms exist. The specification permits the signer to select the canonicalization algorithm and specify it as part of the signature. This introduces interoperability problems: the verifier may not support the specified algorithm. For the purposes of this document, the only canonicalization routine that may be used is the c14n algorithm specified in [1].

The XMLdsig specification also permits users to generate three types of signatures: detached, enveloping, or enveloped. For this specification, we will only use the detached signature. In this case, the signature is over content external to the <Signature> element. Consequently, the signature is "detached" from the content it signs. This corresponds precisely to the CMS detached signature.

8 Conclusion

The investigation supported by the Advanced Technology Program (ATP) has yielded several significant conclusions. As described in this report, digital signatures can be used effectively to support the features present in physical signatures used today. The investigators determined that when migrating from physical to digital signatures, digital signatures cannot be used without regard to the signer's visual frame of reference that is inherent with physically signed documents. However, digital signatures need not explicitly mimic every aspect of the visual context. In fact, the electronically signed documents may be easier to interpret by the verifier than their physical counterparts in some cases. The critical factor when using digital signatures is that the visual context never exceeds the scope of the digital signature.

From a technical perspective, the investigators determined that it is possible for digital signatures to be computed and verified independent of the application used to create, display, or modify the electronically signed document by using a standard signed document format as described in section 7. By using a standard signed document format, digital signature technology can be used effectively to provide security to the electronic analog of today's paper processes. In fact, it is the belief of the investigators that security provided by digital signature technology is greater than that currently found using physical signatures.

The conclusions found through this investigation can be leveraged to further develop applications that use public key cryptography, public key infrastructures, digital signatures, and electronic document systems.

References

[1] Canonical XML, W3C working draft, 1999. See <http://www.w3.org/TR/1999/WD-xml-c14n-19991115>.

[2] XML advanced electronic signatures (XAdES), draft ETSI technical specification, ETSI TS 101 903 draft v0.0.8, July 2001.

[3] Donald E. Eastlake 3rd, Joseph M. Reagle Jr., and David Solo. XML-signature syntax and processing, RFC 3075, March 2001.

[4] Russell Housley. Cryptographic message syntax, RFC 2630, June 1999.

[5] Kathy L. Lyons-Burke and Nelson Hastings. Common format for information that is digitally signed: An interim report, August 2000.

A Meeting the ATP Proposal Milestones

This section of the report describes how the milestones set in the ATP proposal for the Development of a Common Format for Digitally Signed Objects as required by the ATP program were met. The first milestone for FY01 was the identification of a comprehensive set of digital signing methods that would be used to implement the signing models required by the healthcare industry. This milestone was met by the investigation that was used in the development of section 3 of this report. The second milestone for FY01 was the definition of a signing object structure in both ASN.1 and XML that supports the identified signing methods. This milestone was met by the investigation that was used in the development of section 7 of this report.