Next: Executive Summary

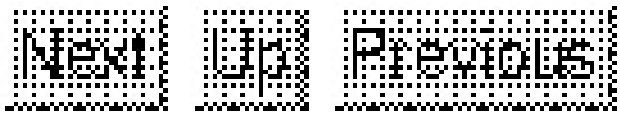
- Executive Summary
- Introduction
  - Brief History of the Significance of Signatures
  - Questions to be Resolved
- Electronic Signature Requirements
  - Definition of Digital Signature
    - Basic Concepts
    - Checksums

    - Standards
    - Legal Issues
        - Compliance Steps
        - Legal Admissibility
        - Written Agreement
        - Statute of Limitations
- Approaches to E-signatures
    - True Signature Schemes
    - Arbitrated Signature Schemes
    - Reuse of Signatures
    - Checksums
        - Properties of Checksums
        - Algorithms
- Approach to Implementation in PERM
    - Introduction
    - PERM Implementation Goals
    - PERM E--Signature Design

- In Search of a Secure E--Signature Protocol
- The PERM E--Signature Protocol
  - E--Signature Authentication Level 1
  - E--Signature Authentication Level 2
  - E--Signature Authentication Level 3
- Summary

- Conclusions
  - Checklist
- References
- About this document ...

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: Introduction **Up:** No Title
Previous: No Title

Executive Summary

A discussion of electronic signatures and a design for their implementation and use in an electronic records management environment are presented. A brief history of the significance of signatures is given and the requirements that an electronic version must meet are discussed. The legal issues and the various methods for possible compliance are reviewed. In an effort to adhere to currently accepted practices

and standards and to determine the legal precedents for electronic signatures, several federal agencies have been consulted including: National Archives and Records Administration (NARA), National Institute of Science and Technology (NIST), General Accounting Office (GAO), and the Department of the Treasury.

While certification efforts are currently underway, at this time there are no applicable standards for electronic signatures. However:

1. the Center for Electronic Records at NARA is preparing a draft standard that addresses the admissibility of electronic records

as court evidence,

2. NIST is currently working to develop a standard for public key encryption,

3. NIST is also studying message digest algorithms to determine their appropriateness for certification,and

4. the Department of the Treasury has a system that replaces written signatures on disbursements that has been accepted by the GAO. This system uses the Data Encryption Standard (DES) to calculate a message digest based on a binary key.

In the absence of defined standards, an attempt has been made to design a system that will meet the proposed legal criteria. The system is flexible enough in its use of both encryption and message digest algorithms to facilitate adherence to standards that will be defined in the near future in these two areas. Due to their involvement in this area, both NARA and the GAO have expressed an interest in the approach described in this document.

The legal issues, research issues, and design issues, for an electronic--signature system are quite complex if it is to be effective. Further research with the GAO, NARA, and NIST are required

in order that the decisions that the Prototype Electronic Records Management (PERM) design team will be valid for an indefinite and lengthy time period. The choice of message digest algorithm and the choice of signature sealing function are crucial to the mathematical vigor and security of the signature process. The approval of the PERM choices by these agencies provide a considerable durability to the lifetime of PERM. These processes of approval take time. For this reason PERM is being designed such that a choice of algorithm today will not be heavily impacted by the need to change an algorithm tomorrow.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Next: Brief History of **Up:** No Title
Previous: Executive Summary

Introduction

The use of signatures affixed to documents to certify their authenticity and ascribe agents for legal action has a history extending back to the late seventeenth century. As electronic means of preparing and disseminating information has advanced, methods to provide the same functionality as that provided by a signature on paper are being sought. A brief review of the significance of signatures based on the work of Meyer [1] is presented here to give a foundation for the discussion of electronic signatures in Sect. . Following that, the approaches to

electronic signatures will be presented in Section ▯. Section ▯ discusses the approach taken for the implementation of digital signatures for the Prototype Electronic Records Management (PERM) System for the U. S. Army Information System Command Missile Command (USAISC--MICOM). The final recommendations will be presented in the Conclusions (Sect. ▯), where a checklist of necessary procedures and actions remaining will be given.

- Brief History of the Significance of Signatures

- Questions to be Resolved

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Brief History of the Significance of Signatures

The legal significance of signatures and the use of documents bearing signatures is based on several branches of the law, including the Statute of Frauds, the Law of Acknowledgments, the Law of Agency, the Uniform Commercial Code (UCC), and others.

Due to peculiar rules of evidence used by English courts during the seventeenth century, it was possible for suits to be tried in which there was no way to

counter an argument that someone had overheard a verbal contract since that would involve producing a witness who could testify that he did not hear such a contract. The Statute of Frauds [2] was enacted in 1677, required written evidence that contracts were actually entered into, specifically excluding from consideration by the courts legal actions on certain contracts unless there was written evidence of the agreement signed by the party to be charged or his duly authorized agent.

Certain documents require that the person who signs the document prove his identity and the date on which the document was signed by him. Often

these **acknowledgments** are witnessed by a judge, an official examiner of title, an official referee, or a notary public and are recorded in an official registry.

The principles of agency law [2] are essential for the conduct of business transactions. Agency is the fiduciary relation (involving a confidence or trust) which results from the manifestation of consent by one person to another that the other shall act on his behalf and subject to his control, and consent by the other to so act [3]. No particular formalities are required to create an agency relationship except for two situations: (1) a formally acknowledged instrument is used for conferring authority for a power of

attorney, and (2) in a few states it is required that the act which confers authority to perform a certain act must possess the same formalities as the act to be performed. Generally, a principal is bound by the duly authorized acts of his agent. However, if the agent does not possess the requisite authority, the principal in most instances will not be bound and instead the agent will himself be liable to third parties. Thus, the correct way for an agent to execute a contract is to affix the name of his principal followed by his own signature and the capacity in which it is made: ``P'' Principal, by ``A'' as Agent.

The UCC [4] is a comprehensive

modernization of various statutes relating to commercial transactions. It has been adopted in all states except Louisiana. The present articles relating to commercial paper, banking transactions, and investment securities are paper-based. A special committee was formed to prepare amendments to these laws to accommodate electronic funds transactions. While these are generally applicable to the transfer of securities without paper, many technical and mechanical changes are still required. However, the current (1972) version of the UCC gives a definition of ``Signed,'' viz.:

> ``Signed'' includes any symbol

> executed or adopted by a party with present intention to authenticate a writing.

and in the case of commercial paper,

> A signature is made by the use of any name, including any trade or assumed name, upon an instrument, or by any word or mark used in lieu of a written signature.

The inclusion of the word authenticate in the definition of signed clearly indicates that a complete handwritten signature is not necessary. The question is always whether the symbol used was executed or adopted by the party with the intention at that time of authenticating the writing.

The laws covering commercial paper also recognize that the drawer---the one who creates a negotiable instrument---has voluntarily entered into relationships beyond his control with subsequent holders of the instrument. The law imposes on the drawer the responsibility to assure that his own negligence does not contribute to the possibility of material alteration of the instrument later in the chain of transfer. In other words, the drawer should take precautions to avoid the charge of **contributory negligence**.

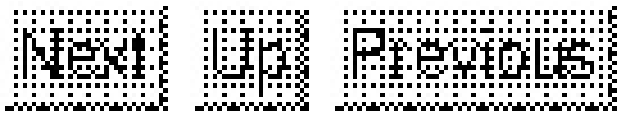
John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Questions to be Resolved

Throughout the remainder of this report, several questions will be raised. These questions will involve the selection of a suitable algorithm for encryption, procedures that are necessary to ensure acceptance of electronic documents as evidence in legal disputes, and what constitutes a reasonable effort to prevent unauthorized use of an electronic signature. The answers to many of these questions are spread throughout the document and will be summarized in the Conclusions. In addition, a final checklist of procedures and actions (see

Sect. ▨) will be presented for the reader's benefit in applying the principles described herein.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: Definition of Digital **Up:** No Title
Previous: Questions to be

Electronic Signature Requirements

The issues involved in using electronic signatures are complex at this time, as they encompass applying seventeenth century principles to twentieth century electronic data storage. In the original uses of signatures affixed to documents, it was virtually impossible to duplicate a document or a signature without detection. In the electronic world, data

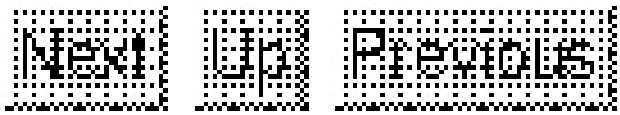
is routinely exactly duplicated. If a means exists for copying out the portion called a signature and applying it to another electronic document, how could this be detected or prevented? In this section, the definition of a digital signature is given and the requirements that a digital signature must possess are described. It should be noted that this is a technical discussion of the issues involved with electronic signatures and some of the legal aspects of solving these problems. The material contained in this section does not constitute advice, and those who intend to be responsible for electronic signature systems should first consult their legal counsel before finalizing their implementation of the

procedures described herein.

- Definition of Digital Signature
  - Basic Concepts
  - Checksums
- Standards
- Legal Issues
  - Compliance Steps
  - Legal Admissibility
  - Written Agreement
  - Statute of Limitations

John W. Wooten

Thu Aug 15 19:50:02 EDT 1996



Next: Basic Concepts **Up:** Electronic Signature Requirements **Previous:** Electronic Signature Requirements

Definition of Digital Signature

- Basic Concepts
- Checksums

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Basic Concepts

A **digital signature** is a private object that is owned by a person or process and that is used for noting the ownership, approval, or acceptance of another object such as a document or message. If *A* signs a message *M* and sends it to

recipient *B*, then *A*'s signature must satisfy the following requirements [5]:

B must be able to **validate** *A*'s signature on *M*. It must be impossible for anyone, including *B*, to **forge** *A*'s signature. In case *A* should disavow signing a message *M*, it must be possible for a judge or third party to **resolve** a dispute arising between *A* and *B*.

A digital signature establishes **sender authenticity**. It is analogous to an ordinary written signature that can be **validated** by a handwriting expert. The court testimony by expert witnesses that a written signature was produced by a given owner can be used to **resolve**

disputes over wills, contracts, and other legal instruments. Digital signatures must provide the same validation and authentication capabilities as written signatures for legal purposes.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Checksums

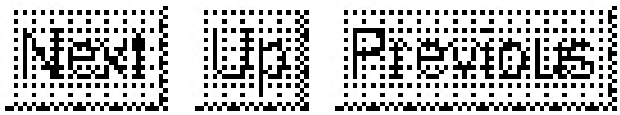
Methods known as public key (or RSA for Rivest, Shamir, and Adleman) encryption algorithms [5] can be used successfully to provide signature validation and dispute resolution, but generally require either full encryption

at all times, or else require the full message to be stored as clear text and also as a public key encrypted text for validation purposes. Since this has a large impact on computer performance or on data storage requirements, methods known as checksums or ``digests'' have been developed to characterize the message, which is then signed using Data Encryption Standard (DES) [5] or RSA encryption methods. The message is still stored as clear text with a small ``signature'' attached for signature validation and resolution.

Many banking systems use simple electronic signature methods to verify that transactions are conducted by the

proper owners of the respective funds. The present systems provide reasonable protection against outside threats, and internal controls are used to prevent abuse by inside agents. The methods are sufficient to provide legal protection against liabilities by providing assurances that fraud or negligence were not involved if disagreements arise over the authorization of transactions. External controls exist for tracking the movements of large amounts of funds, thus reducing the requirements for extreme robustness of the electronic signature method. The limitation of liability approach will probably not suffice for the present project.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: Legal Issues **Up:** Electronic Signature Requirements **Previous:** Checksums

Standards

In 1977 the National Bureau of Standards announced a Data Encryption Standard [6] to be used in unclassified U. S. Government applications. The algorithm was developed by IBM and enciphers 64-bit blocks of data with a 56-bit key. The algorithm is used for both encryption and decryption and has been implemented in both software and hardware. Hardware implementations achieve rates of several million bits per second.

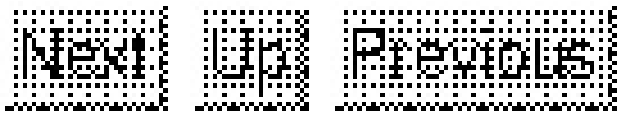
While DES or other methods could be used to encrypt complete messages for attachment to a signature, as was

explained in Sect. , present electronic signature methods typically use smaller checksums or ``message digests'' to characterize the message in order to save time and storage space. Present standards (ANSI X9.9- 1986 [7] specify the use of a 32-bit checksum code. This will probably be extended in the future to a 64-- bit checksum.

Several committees are currently reviewing drafts of proposed standards to address electronic mail to be used with the proposed OSI X.400 Message Handling System Model. Among these are proposals for enciphering electronic mail messages and authenticating their contents [8], and algorithms for

implementing the proposed standards [9].

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: Compliance Steps **Up:** Electronic Signature Requirements **Previous:** Standards

Legal Issues

- Compliance Steps
- Legal Admissibility
- Written Agreement
- Statute of Limitations

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Compliance Steps

There are several steps to be taken to assure compliance with all legal statutes. First, the following steps must be taken to provide for the creation and use of text documents [10]:

provide a method for all authorized users of the system to retrieve desired

documents, such as indexing or a text search system; provide an appropriate level of security to ensure integrity of the documents and the electronic processes; provide a standard interchange format when necessary to permit the exchange of documents on electronic media between agency components using different software/operating systems and the conversion or migration of documents on electronic media from one system to another; and provide for the disposition of the documents, including, when necessary, the requirements for transferring permanent records to the National Archives and Records Administration (NARA).

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Legal Admissibility

Secondly, steps must be taken to assure the legal admissibility of evidence in the event of a dispute. With proper documentation, electronic records are admissible in evidence to Federal courts for use in court proceedings [11]. Trustworthiness must be established by thoroughly documenting the record

storage system's operation and the controls imposed to protect the information. The following procedures should be implemented to enhance the legal admissibility of electronic records.

Document that the processes used to create, store, and retrieve the electronic records are the same for similar documents. Substantiate that security procedures prevent unauthorized modification or deletion of a record and provide for data integrity. Identify the electronic media on which records are stored through out their life cycle, the maximum time span that records remain on each storage medium, and the NARA-approved disposition of all records.

Coordinate all of the above with legal counsel and senior information records manager and the records management staff.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Written Agreement

Finally, in addition to the above procedures, an initial written agreement with each person authorized to use electronic signatures should define the procedures and protocols whereby the parties would conduct a series of future

transactions, together with an agreed means and procedure for recording the elements of such transactions. The following two features should be designed into this agreement .

As part of their initial written agreement, the parties must specify a particular jurisdiction under whose laws the agreement is to be governed and the forum of litigation of disputes that may arise out of transactions executed via the electronic communication system. The parties should stipulate as part of their written agreement that they will be bound by their digital signatures, that they agree to submit all disputes to a referee, and that they agree that the

concept of digital signatures is cryptographically sound. However, this agreement will not prevent one of the parties from later raising the claim that the indicated result lacks validity, that he did not understand the underlying scientific principle, or that he was forced to sign the stipulation as a condition of his transacting business with the other party.

It should be expected that disputes will arise and that, until the courts pass on the cryptographic strength of the algorithms used in the generation of signatures, the disputes in court will be over the validity of the technique, rather than a trial of the issues involved in the case

[12].

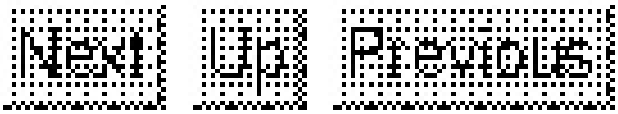
In order that a particular digital signature method have reasonable chance of achieving judical notice or being accepted as a recognized principle, the algorithm should be based on a strong encryption algorithm which itself has already been scientifically recognized and accepted. Several candidate algorithms are discussed in Sect. .

A mechanism must exist for each party to authenticate independently the nonsecret signature validation information which he holds. This could be done if each party were to record his own signature validation information at some

established registry with recognized and accepted integrity, or it could be established on the initial written agreement.

When the method and information to generate signatures is stored on a computing system, the burden is on the installation management to assure that this information is kept secret and that an adequate access control mechanism is in place so that signatures can only be created by authorized users. Whoever has access to a principal's secret signature information will be deemed to be the principal's authorized agent. Installation management must therefore implement sufficient security measures

in order to be alerted if this information becomes exposed. Failure of one of the principals to notify other parties that his digital signature has become compromised may be deemed his own negligence and might defeat any defenses he may later raise as to the authority of his agents.



Next: Statute of Limitations **Up:** Legal Issues **Previous:** Legal Admissibility

---

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Statute of Limitations

The statute of limitations defines the period of time within which a lawsuit must be commenced from the time a cause of action accrues. In disputes involving contracts, the period in most states is six years. It would appear necessary, therefore, that both parties to

a transaction (sender and receiver) retain all data relating to their initial written agreement and to each subsequent signed message for at least the period of the applicable statute of limitations.

There should exist a trusted mechanism for the recording of the time and date of a transaction. This could be included as part of the signature of the transaction. In addition, a method should exist to allow the authorized user to review a dated list of documents to which his signature has been affixed.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Approaches to E-signatures

Signatures provide a method of certifying the contents and the originator of a message. Validation and authentication can also be achieved by digital signatures. However, digital signatures differ from handwritten signatures in at least two key aspects:

- A handwritten signature can eventually be duplicated by a forger, while a digital signature is

by definition unable to be duplicated.

- A person's handwritten signature is constant or the same on every document, while the digital signature will be different for every message [13].

Thus, a digital signature is dependent on the message being signed and can be computed only by the sender based on secret information. The form of digital signatures varies from special bit patterns attached to the message to an integral part of the cryptographically transformed data.

Digital signatures must perform several

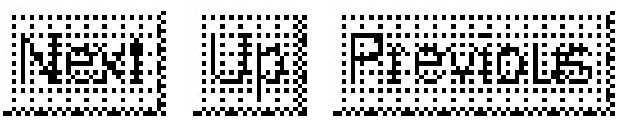
functions, as stated in Sect. ▨. To summarize, digital signatures provide authentication of messages because they are unforgeable. If person *A* signs message *M* it is impossible for anyone else to produce that combination; thus the receiver is confident that the message was sent by *A*, and *A* cannot disavow sending *M*. Since the signature is dependent on every bit of the message, validation is possible. The receiver is confident that the message was not changed during transmission, and the sender is confident that the receiver cannot alter the message.

There are several techniques for generating digital signatures. Generally,

digital signature schemes fall into one of two categories: *universal*, *general*, or *true* signatures and *arbitrated* signatures. *True* signature schemes are characterized by the fact that the sender directly transmits the signed message to the receiver, who verifies and authenticates the message, while in an *arbitrated* signature approach a trusted third party transmits the message between the parties and validates and authenticates the message. A significant number of papers have been written describing the implementation of digital signatures and encryption systems. A selected bibliography has been included at the end of the cited references to provide the interested reader with

additional information.

- True Signature Schemes
- Arbitrated Signature Schemes
- Reuse of Signatures
- Checksums
  - Properties of Checksums
  - Algorithms



Next: True Signature Schemes **Up:** No Title **Previous:** Statute of Limitations

---

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

True Signature Schemes

True signatures can be implemented using either private key or public key encryption techniques. In private key or conventional cryptographic systems, the encryption and decryption keys are identical or can easily be generated from one another. In public--key systems, a public key is used for encryption and a private key for decryption, thus knowledge of the public key does not compromise the private key [1].

Public--key systems provide a convenient method for implementation of digital signatures. Using this scheme, system users register their public keys in a directory. When user *A* wants to send a signed message *M* to another user *B*, *A* signs *M* by computing , where is the transformation private to *A*. *A* transmits the pair to *B*. *B* validates and authenticates the message by computing and comparing the message received with the value computed. Since is private to *A*, the recipient is assured of both sender and data authenticity. In addition, it is impossible for *A* to disavow the signed message. Finally, since is public, the receiver can validate the signature and

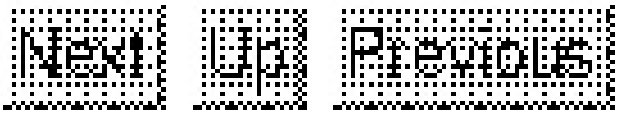
disputes can be resolved easily by a judge. The most well known public--key encryption system is RSA [14].

With private key encryption, the secret key provides data authenticity but does not prevent forgery or sender authenticity. Since the sender and the receiver share the same key, the receiver could create the sender's signature and a judge could not resolve the dispute. However, true signatures can be implemented using conventional or private--key encryption systems by applying more complex protocols in order to meet all the requirements. Once again the users of the system must share public validation information, but

additional secret and/or nonsecret information is sent with the message in order to provide proper validation. Digital signature schemes in this category are generally time consuming, costly, and wasteful[1,13].

Unfortunately, all true signature schemes share a common weakness. The approach relies on the fact that the sender is the only person who can compute the message/signature pair based on the private or secret information. This is true only if the secrecy of the key has been perfectly protected so that the key has not been compromised. This issue has already been addressed in the discussion of the

legal issues related to the use of digital signatures.



Next: Arbitrated Signature Schemes
Up: Approaches to E-signatures
Previous: Approaches to E-signatures

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Arbitrated Signature Schemes

Arbitrated signature schemes apply the same principles as witnesses in the paper world. Just as a witness's signature on a document protects against forgery and prevents the sender from disavowing the document, as well as proves the validity and authenticity of the document, an arbiter provides these same assurances in an electronic context. Using this scheme, every message from a sender *S* to a receiver *R* must first go to an arbiter *A*, who determines the origin

and verifies the contents of the document and provides verification to *R* that indeed the document is from *S*.

Arbitrated signature schemes can be implemented with either conventional key systems or public key systems. In a conventional key approach, each member *X* of the system has registered a key with the arbiter. When *S* wants to send a signed message, *S* sends the message *M* and the message encrypted under to *A*. The arbiter decrypts the message using the registered key, , and verifies that the two messages agree. Then the arbiter sends *M*, *E(M*,), and a message from the arbiter indicating that the message was verified, all

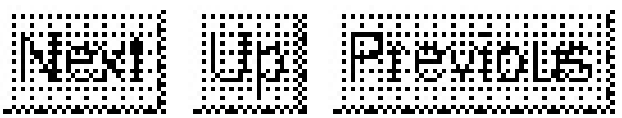
encrypted under . The receiver trusts the arbiter who confirmed that the message was sent from *S*; thus, the authentication requirement is satisfied. The no forgery requirement is also satisfied, since *R* cannot send a message to himself and claim it came from *S*, and *S* cannot disavow a message that went through *A*.

This approach suffers from two problems. A significant amount of trust must be placed in the arbiter, who could form an alliance with either the sender or the receiver. By aligning with the sender, the sender could deny a signed message. By aligning with the receiver, the receiver could forge the sender's

signature. In addition to this problem, the arbiter has access to whatever information is contained in the signed message. Hiding the contents of the message from the arbiter is fairly easily implemented; however, the first weakness is more difficult to resolve using a private key approach.

Both problems can be addressed by using a public--key encryption system. The disadvantages can be overcome by using two encryptions, one private and one public. The message is signed by using the private key of the sender. Applying a second encryption using the public key of the receiver ensures that the message's contents remain secret

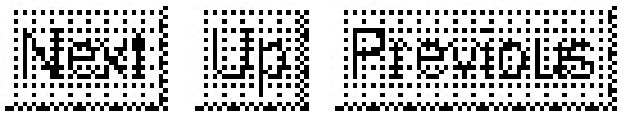
since only the receiver can decrypt the message. Once S has doubly encrypted the message, a new message to the arbiter is concatenated to the first message, thus providing the arbiter with information to authenticate and validate the message. As in the previous approach, the final packet is sent by the arbiter to the receiver. This solves all the problems of the first scheme. The information is invisible to the arbitrator, and alliances are prevented.



Next: Reuse of Signatures **Up:** Approaches to E-signatures **Previous:**

True Signature Schemes

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: Checksums **Up:** Approaches to E-signatures **Previous:** Arbitrated Signature Schemes

Reuse of Signatures

All of the schemes presented meet the digital signature requirements of authenticity and preventing forgery; however, it is also necessary to prevent the reuse of an old message. In each digital signature approach, the receiver cannot create new messages but can reuse the old ones. Thus, it is necessary to document the use of a message.

Reuse of messages can be prevented by making a time stamp an integral part of the signature. If the message includes the time and date sent, the receiver cannot

reuse the same message without detection of the forgery. It is not imperative that the message contain the actual date that it was sent; a sequence number would provide the same protection. In fact, a message sequence number not only prevents reuse of old messages but also prevents message deletion.

In true signature schemes, each message that is sent should contain at least two parts: the actual message to be transmitted and a submessage that includes both the sender's and receiver's names and a message sequence number. In arbitrated signature schemes, even if the original message contains a sequence

number, it is generally part of the arbiter's responsibilities to time and date receipt of the message. This information is then appended to the message in route to the receiver. It is also possible to prevent the breaking up of a message and the replay of a piece or a block of the message. The approach is to make each piece of the message dependent on the time stamp or the previous block, thus preventing reuse of a block of one message in another message.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Checksums

Manipulation detection codes are a class of checksums that can detect both accidental and intentional modification of a message without requiring the use of encryption [15]. Message digest algorithms or message authentication codes permit someone to determine with a fairly high degree of confidence whether the text has been accidentally or intentionally altered. Generally, the manipulation detection code is encrypted to prevent the substitution of both a new code and a corresponding message, and this encrypted code is appended to the

text to provide authentication.

The protocols described earlier which generate digital signatures may provide more security than the users require in certain systems. In all of the schemes discussed, the actual message is transmitted in an encrypted form, even if secrecy is not a requirement of the system. Thus, manipulation detection codes or checksums provide an alternative that do not require encryption of the entire message.

Manipulation detection codes have several advantages. First, manipulation detection codes avoid the time--consuming process of encrypting and

decrypting the entire message. If the sender appends a bit pattern, referred to as an authentication code, to the document, the recipient can check the message and determine if any modifications were made to the message. Since only the sender and the receiver know the algorithm, it is not possible for anyone else to generate a correct authentication code for a modified message. Second, the use of manipulation detection codes separates the functions of encryption and authentication. Given that the certifications and standards for encryption techniques are still evolving, the separation of these two functions is important from the digital signature

perspective. Separating these functions in the digital signature approach allows for major changes or swapping of encryption techniques without affecting the signature protocols. Finally, since some of these algorithms rely only on publicly known quantities, they do not introduce key management issues [16].

- Properties of Checksums
- Algorithms

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Properties of Checksums

In order to provide adequate security, it must be assumed that the set of all checksums is very nearly one to one with respect to the set of all message texts. If two messages A, and B, have checksums, then the checksum(A) and checksum(B) are identical if and only if the messages A and B are themselves

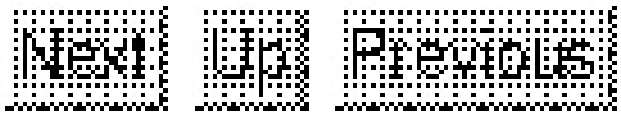
identical. With a good checksum algorithm, the chances that A and B are *not* identical given that checksum(A) equals checksum(B) should be , where k is the number of bits in the checksum and the probabilities are averaged over all possible messages.

Specifically, the algorithm should have the properties listed below [16].

If two different texts (of arbitrary length) are checksummed, the probability that the two checksums will be the same when the two documents are not identical should be a uniformly distributed random variable that is independent of the text, with an average

value over all possible texts of , where N is the number of bits in the checksum. The checksum must be sensitive to permutations, so that the message ABC will produce a different value than ACB, etc. The resulting checksum must be sufficiently long to resist a so-called ``birthday attack'' against the text itself. The name is based on the statistical number of people that must be in a room for there to be a good chance that two people will have the same birthday. For a 64-bit checksum, about 4.3 billion iterations produced by systematically varying 32 lines of text would suffice to determine what changes could be made to the document and still produce the same checksum. While the

calculation of this data would require under 2 CPU days on a 10--microsecond--per--iteration machine, this would require about 51.5 megabytes of data to be sorted and compared. If the data was stored on magnetic tape, approximately 340 reels of 6250 bpi tape would be required to be mounted and compared. All of the bits of the checksum must be an over-determined function of all of the bits of the text and all of the bits of the checksum of the previous block. This is required so that simple mappings between sections of the clear text and parts of the checksum cannot be determined from analysis of the checksums.



Next: Algorithms **Up:** Checksums
Previous: Checksums

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Algorithms

Based on testing of checksum algorithms, the recommended length for cryptographic checksums should be on the order of 128 bits to prevent an individual from systematically changing

both the text and the manipulation detection code until a match is discovered. The 128--bit checksum is sufficient since sorting and searching becomes overwhelming as described in the requirements section. In addition, the 2 calculations required to complete the birthday attack would become computationally infeasible [15].

Several manipulation detection codes which meet this property are described in the current literature. Two have been defined by Ronald Rivest of RSA Data Security, Inc. They both accept messages of any length as input and provide a 128--bit quantity or message digest as the output. The first algorithm, called MD2,

is described in RFC--1115 in support of privacy enhanced electronic mail [9]. The MD2 algorithm is series of nonlinear byte substitution operations based on permutations constructed from the digits of pi. The second approach, referred to as MD4, [17] is a series of permutations based on the square root of two and the square root of three. The Rivest paper[17] states that the difficulty of deriving two messages with the same message digest is on the order of 2 operations and that the difficulty of deriving a message having a given message digest is on the order of 2 operations. Thus, the level of security should be sufficient for the implementation of digital signatures.

A third algorithm, QCMDCV4, [16] developed by Robert Jueneman of Computer Sciences Corporation, is based on his Quadratic Congruential Manipulation Detection Code (QCMDC). This algorithm now computes a 128--bit result using exclusive--ors and a history function to provide a function that is not invertible. The result is an over--determined function of 128 bits of the text and the 128--bit intermediate result of the previous text block, thus meeting all the requirements discussed in Sect. . Again, the level of security should be sufficient for implementing digital signatures where encryption will be provided separately.

Finally, cryptographic functions can also be used to implement cryptographic checksums [18]. By applying block chaining to the DES algorithm, DES can be used as a checksum. In block chaining, each block depends on the previous block. To apply block chaining to DES, each block would be combined with the exclusive--or of all the previous blocks. The last block of the chained DES encryption would serve as the checksum, since this block would depend on all the other blocks or the entire message.

All of these algorithms are currently being studied to determine any weaknesses. Several are being

considered as possible standards [8,9] . Currently, only the DES has been approved by the National Institute of Science and Technology (NIST) for use in unclassified environments [6].

Next Up Previous

Next: Approach to Implementation **Up:** Checksums **Previous:** Properties of Checksums

---

John W. Wooten

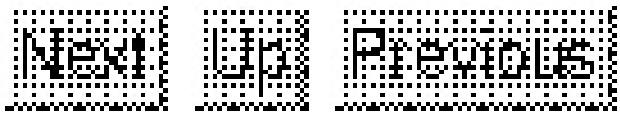
Thu Aug 15 19:50:02 EDT 1996

Approach to Implementation in PERM

- Introduction
- PERM Implementation Goals
- PERM E--Signature Design
- In Search of a Secure E--Signature Protocol
- The PERM E--Signature Protocol

    - E--Signature Authentication Level 1
    - E--Signature Authentication Level 2
    - E--Signature Authentication Level 3
- Summary

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: PERM Implementation Goals **Up:** Approach to Implementation **Previous:** Approach to Implementation

Introduction

The intention of this project is to develop a PERM system which has a pragmatic implementation of an Electronic--Signature (E-Signature) system. This E-Signature facility must be able to withstand judicial scrutiny implying that the implementation is based on sound security principles.

Since the PERM system will be an application code which will be executed by a user with ``ordinary'' application privileges, an ``extraordinary'' privileges mechanism must be incorporated into the E-Signature facility for an acceptable legal implementation.

As will be seen, this ``extraordinary'' privileges mechanism will be provided through trusted process arbiters or daemons to which the user's application communicates when E--Signature management is required.

It should be further noted that the PERM system is not a typical digital signature management system. The signed messages or documents are not ``ordinary'' messages or documents as presented in Sect. in the discussion of the theoretical development of digital signatures. PERM is designed such that general non--arbitrated access to documents and document signatures is not allowed. The documents will be

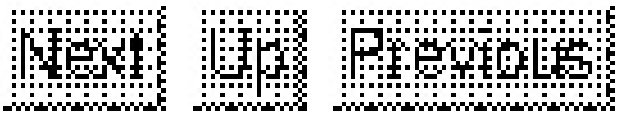
maintained, signed, and archived by the PERM application through which the user's signature will be authenticated. The electronic--document and E--Signature data bases maintained by the PERM application processes and arbiters will be isolated from ``normal'' user access. However, it is true that users with ``system'' privileges can access the data base through independent processes. The PERM application is designed to disallow access to the PERM data base if the user has ``system'' privileges, but PERM is a non--kernel application; therefore, it can not control external ``system'' access to PERM data bases. External ``system'' access must, perforce, be controlled

administratively. It is important that ``system'' privileges be minimized and well--controlled on the computing systems which contain the PERM data bases.

It is noteworthy that it is possible to apply automated, centralized system monitoring capabilities which oversee ``system'' access and changes. However, this capability is beyond the present design functions of PERM since this capability is philosophically of much more general concern to system management and control and not to document and signature management.

PERM is being designed to run in a POSIX operating system environment.

The initial development is being performed on Sun workstations using the UNIX SysV software base. The UNIX SysV base is required since this is the operating environment which will ultimately be the recipient of the PERM application. This environment consists of the Unisys 5000 implementation of UNIX SysV. The Amdahl UTX environment is also under consideration as a PERM system host.



Next: PERM Implementation Goals **Up:** Approach to Implementation **Previous:**

Approach to Implementation

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

PERM Implementation Goals

The ultimate goal of the PERM system is to provide a document and signature management system which exceeds that which is available in the world of handwritten documents and signatures. The document management design is presented in the PERM Document Design publication [19]. The requirements of E--Signature design are presented here.

Analysis of signature management and authentication centers essentially on the determination of ``who actually is

responsible for the signature affixed to a document in question." Informally, the individual questioning signature ownership can accept his own judgment that a given signature is valid. This is the predominant state of affairs with most signature validation. In the event that the individual who questions the authenticity of a signature does not accept his judgment, the questioning individual will require that an expert witness testify to the authenticity of a signature. This process is generally referred to as notarization.

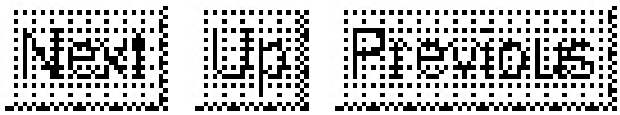
PERM is designed to provide signature validation facilities which will be acceptable under normal conditions.

However, PERM is also designed to provide a notary capability of comparable value to that for handwritten signatures.

Another important goal for the PERM signature capability is to be as nonintrusive as possible to the PERM user. This means that the user validation requirements will be as simple as can be and still provide assurances to the user that the user's signature can not be forged. As has been noted in the previous section on legal issues, the ease with which data can be duplicated and altered is a major objection to the use of E--Signatures. These objections are duly addressed in the PERM E--

Signature Design.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: In Search of **Up:** Approach to Implementation **Previous:** PERM Implementation Goals

PERM E--Signature Design

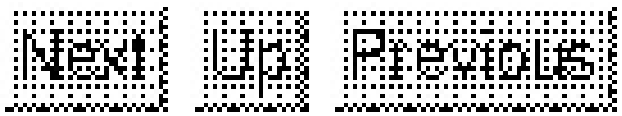
The various methods previously discussed in Sect. ▨ provide a mathematical basis for the authenticity of E--Signatures, but these methods are not exacting in how to conveniently and securely apply E--Signatures. Each method demonstrates its mathematical integrity but gives little consideration to human engineering issues. For example, concerns for encryption key management are not considered. Also, invasive possibilities due to the need to positively identify a user are of little apparent concern.

Each of the mathematical forms for digital signature methods is dependent on some form of encryption technique to generate and authenticate a signature. Public Key Encryption and private key encryption facilities are presented in Sect. . After having conversed with various individuals involved in the legal and mathematical issues associated with E--Signatures, it has been decided that private key management using DES encryption functionality is the best choice. Each user will possess a unique private key for signature validation. This private key will be maintained by a trusted E--Signature arbiter process rather than by requiring the user to provide a key which is difficult to

manage. This decision is based on the human engineering concept that a complex bit of data like that required for good encryption techniques is difficult to remember. This difficulty generally necessitates that the user will record the key in an easily accessible manner. This ease of accessibility reduces the chances of keeping the key value secret, thereby negating the effectiveness of the signature process. This consideration for key management places the security burden of the PERM system on positively identifying the user. This point will be addressed later in the design.

As has been formulated in the cited

works on digital or electronic signatures the process of generating and/or authenticating a signature attached to a document demands a secure--trusted arbiter or service. Since the process which requires this service must communicate to the trusted server a secure communication method is necessary.



Next: In Search of **Up:** Approach to Implementation **Previous:** PERM Implementation Goals

---

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

In Search of a Secure E--Signature Protocol

In the search for a secure communication method or protocol, a facility designed for networked authentication was identified and studied. This system, designed and programmed by Project Athena at the Massachusetts Institute of Technology (MIT), is called Kerberos. The following description of the Kerberos concept is offered by members of the Kerberos design team [20]:

> ``Most conventional time-sharing systems require a prospective user to identify him or herself and to

authenticate that identity before using its services. In an environment consisting of a network that connects prospective clients with services, a network service has a corresponding need to identify and authenticate its clients. When the client is a user of a time-sharing system, one approach is for the service to trust the authentication that was performed by the time-sharing system. For example, the network applications lpr and rcp provided with Berkeley 4.3 UNIX trust the user's time-sharing system to reliably authenticate its clients.

In contrast with the time-sharing system, in which a protection wall separates the operating system from its users, a workstation is under the complete control of its user, to the extent that the user can run a private version of the operating system, or even replace the machine itself. As a result, a network service cannot rely on the integrity of the workstation operating system when it (the network service) performs authentication.

This plan extends the conventional notions of authentication, authorization, and accounting to the network environment with untrusted

> workstations. It establishes a trusted third-party service named Kerberos that can perform authentication to the mutual satisfaction of both clients and services. The authentication approach allows for integration with authorization and accounting facilities. The resulting design is also applicable to a mixed time-sharing/network environment in which a network service is not willing to rely on the authentication performed by the client's time-sharing system."

This conception of Kerberos allows for the delineation of the following goals:

- **Authentication**---Authentication is not an end in itself, but rather a tool to support both integrity and authorization. Its basic purpose is to prevent fraudulent connection requests. The goal of Kerberos is to support both one-way and mutual authentication of principals, to the granularity of at least an individual user and specific service instance.

- **Authorization**---Authentication can imply a coarse-grained authorization; for example, some services may allow anyone who can be reliably authenticated by the local Kerberos to use the service. In cases where more selective

authorization is needed, the goal of Kerberos is to allow different services to implement different authorization models and to allow those authorization models to assume that authentication of user identities is reliable.

- **Accounting**---Given an authenticated client, the goal of accounting is to support either quotas charged against the client to limit consumption (e.g., disk quota), and/or charges based on consumption (e.g., $.01 per page printed). The goal of Kerberos is to permit modular attachment of an integrated, secure, reliable

accounting system.

Given this set of goals and given the nature of secure authentication, one is naturally led to the idea of applying the elegant Kerberos design to that of digital signatures.

However, close investigation of the Kerberos protocols reveals their heavy dependence on short time frames for the validity of user and workstation authentication. The Kerberos arbitration protocol generates a session tag when a user requires Kerberos authentication. This session tag is used on subsequent authentication requests so that the Kerberos arbiter can be utilized more efficiently. Since the validity of the

Kerberos authentication ticket or session tag is heavily dependent on a lifetime measured in at most tens of minutes, whereas the lifetimes of documents and their associated signatures vary from minutes and hours to indefinite, this protocol is not adequate for E--Signatures. However, the Kerberos session tag will be utilized for a computing session only when the user invokes the PERM application. Other devices must be sought to resolve the need for signature authenticity.

Applying the lessons learned from Kerberos combined with the offered concepts on the mathematical forms of digital signatures on arbitrated

signatures [1], an E--Signature method is proposed.

Next Up Previous

Next: The PERM E--Signature **Up:** Approach to Implementation **Previous:** PERM E--Signature Design

---

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

The PERM E--Signature Protocol

Figure illustrates the graphic design of the PERM E--Signature concept. Figure illustrates the graphic conception of PERM as presented at the onset of the PERM project. The PERM design contains this initial conception, but as expected, is more exacting.

Figure: PERM E-Signature Diagram

Figure: E--Document Management System

Figure illustrates that PERM will consist of three levels of E--Signature authentication. These levels are described in Sects. 5.5.1 through 5.5.3

- E--Signature Authentication Level 1
- E--Signature Authentication Level 2
- E--Signature Authentication Level 3

John W. Wooten

Thu Aug 15 19:50:02 EDT 1996

E--Signature Authentication Level 1

The user will execute the PERM E--Document application as a typical application process. The PERM E--Document application will immediately

determine that the user is not one with ``system'' privileges masquerading as a user other than himself. This requires that all users must be registered individually with PERM and that the POSIX ``root'' user and the ``super user'' can not be PERM users.

The user will then begin using the PERM E--Document applications to create documents, sign documents, etc. See document K/DSRD--471 [19] for user capabilities in PERM. When the E--Document application determines that the user must be authenticated at a level for digital signature creation or verification, then a second level of authentication will be required.

Presently, the E--Document application is designed to require the user to ``login.'' In order to enhance the security of PERM, the E--Document application will then communicate via a secure remote procedure call (RPC) to invoke the E--Signature arbiter. This communication will consist of a secure message which is transmitted to the E--Signature arbiter. This message will consist of a two--way encrypted ordered pair, (user, user--password). The arbiter will authenticate the user and communicate the result back to the E--Document application. The E--Document application will then utilize the result to determine whether or not the user can continue with the E--Document process.

This authentication process results in the generation of a session tag in the manner of Kerberos (specific Kerberos protocols are not utilized at this point since this protocol is an overkill for this application and does not really fit within the philosophy of the application). This session tag is utilized between the E--Document application and the E--Signature arbiter which require E--Signature authentication security. Once a session is completed or a time interval has expired, the session tag is invalidated for further use.

E--Signature management needs to respond to but two types of requests, namely, signature creation and signature

verification. When the E--Document application requests that a signature be created, the E--Document application communicates such to the E--Signature arbiter. This secure communication between the E--Document application and the E--Signature arbiter is presented as a message in Table 1.

Table 1: Signature Creation 4-Tuple

The E--Signature arbiter will subsequently utilize the session tag associated with communications from the E--Document application to generate a signature which is unique to the user and the version of the document sent by

the E--Document application. This returned signature is documented in Table 2.

Table 2: Generated Signature 6-Tuple

The Encapsulating function above is an application of the DES encryption algorithm on the data stream as indicated. The encryption key to be utilized will be that which is uniquely associated with the user.

The message digest function which is applied to the data stream of the document to be signed is a digest or cyclic redundancy check (CRC) function

as described in Sect. . Presently, the digest functions contending for this process are the MD4 algorithm and the DES Block Chaining Algorithm.

Of course, the E--Document application will record the E--Signature message sent by the E--Signature arbiter so that subsequent signature verification can be performed. As a further means of providing sufficient responsiveness to judicial scrutiny, the E--Signature arbiter will log the information associated with the creation of each signature. This separation of signatures and creation log provides a higher probability of successful security.

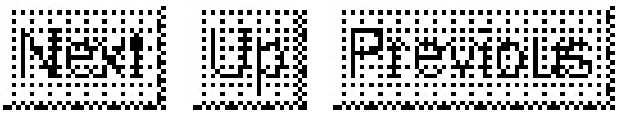
The E--Document Data Base (DDB)

and the E--Signature Data Base, (SDB) are designed as separate entities essentially for security purposes, but this separation of data as well as the separation of functions between the E--Document application and the E--Signature arbiter provide an opportunity for a totally distributed application of PERM.

Level 1 of Signature Authentication is designed to emulate the process of an individual signing a document, unnotarized. The major difference from the handwritten document world is that the document is well--controlled with limited access and maintenance of such. This adds considerably to a higher

probability of successful signature authentication.

However, as previously mentioned, notarized signatures have been devised in order that a signature can be successfully adjudicated. Since Level 1 does not provide a notarization facility, a second and third level of signature authentication is presented.



Next: E--Signature Authentication Level
Up: The PERM E--Signature **Previous:** The PERM E--Signature

---

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

E--Signature Authentication Level 2

Typically, signature notarization is performed in the presence of an individual who is warranted to be of acceptable judgment to authenticate that

a document was signed by an individual in question. Level 2 of the PERM E--Signature Authentication system is designed with this point in mind.

Typically, the notary affixes a seal to a signed document to ``prove'' the authenticity of the document and its signatory. This capability for PERM is accomplished by having the E--Doc/Sign Arbiter DAEMON (SDAD) scan both data bases, DDB and SDB, and generate a message digest signature of each of the data bases. This scan is to be accomplished as a continuous, low--priority, background process under normal conditions. This continuous scan provides a high degree of assurance that

no outside influence has been applied to the data base and has made unauthorized changes.

In actuality, SDAD will receive secure RPC information from the E--Signature arbiter in order to generate the message digest ``fingerprint'' of both DDB and SDB. This running summary will be compared to the scanned summary calculated by SDAD in order to notarize that neither DDB or SDB have been altered without authorization.

This method actually merely provides notary functionality to the E--Signature arbiter since the authentication of the user is still a function of the arbiter process.

Since DDB and SDB will be maintained on storage devices belonging to the standard file system so that ``real--time'' on--line access to the PERM documents is accomplished, these data can be altered. Level 3 of E--Signature Authentication provides a means of additional notary powers which addresses the problem of alterable documents.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

E--Signature Authentication Level 3

Another trusted application, the E--Doc/Sign Archiver (DSA), is introduced for Level 3 authentication. DSA behaves in a manner similar to the SDAD except

that DSA performs all of its data storage of completed and signed documents on write--once--read--many (WORM) optical disk storage cartridges. DSA receives a trusted message from the E--Document application stating that a document residing in DDB needs to be archived. The archiver requests from SDAD a notarization of the integrity of both DDB and SDB.

It must be noted that the implementation of Level 3 requires hardware, namely WORM optical disk storage, which is not designed for the Phase I implementation of PERM. The project is developing a plan for implementing this level of PERM since project research

through the Department of Treasury, the General Accounting Office (GAO), and the NARA places considerable importance on the Level 3 concept of notarization.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Next: Conclusions **Up:** Approach to Implementation **Previous:** E--Signature Authentication Level

Summary

The design of the E--Signature system for PERM is unique. It derives some of its functionality from suggestions offered by the digital signature and cryptological expertise from the NIST, the NARA, and the GAO. The thrust of the suggestions from these groups is that no accepted standards yet exist for an E--Signature system such as that being designed for the MICOM PERM facility. The choice for message digest function is leaning toward the MD4 algorithm strictly because the only negative comment about it at this point is that it is not yet the digest standard. However, it is

expected to become the standard.

Public key cryptography for the signature sealing function is rejected on the grounds of speed and required key size. The DES encryption function is becoming a de facto private key encryption standard and has the added benefit of speed along with compactness. For these reasons, DES with key management performed by the E--Signature arbiter has been selected as the means of sealing the individual signature with the document digest.

The technique for notarization seems to be mathematically sound and is unique in its features. The selection of WORM optical disk technology to provide the

Level 3 notarization adds another unique, ``leading edge'' feature to the PERM implementation.

It is important to note that there is a significant dependence on user authentication for the Level 1 signature generation. The Phase I method of requiring a secondary ``login'' to the E--Document application and having the E--Signature arbiter validate the user is not the most sound means of securing individual identity, but it is the most practical means for the Phase I time frame. Subsequent work on PERM should delve into the interfacing of some efficient and cost--effective form of smart--card technology for user

identification.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Conclusions

A discussion of electronic signatures and a design for their implementation and use in an electronic records management environment have been presented. In an effort to adhere to currently accepted practices and standards and to determine the legal precedents for electronic signatures, several federal agencies were consulted. The agencies consulted included:

- NARA;

- NIST;
- GAO; and
- the Department of the Treasury.

While certification efforts are currently underway, at this time there are no applicable standards for electronic signatures. However:

1. the Center for Electronic Records at NARA is preparing a draft standard that addresses the admissibility of electronic records as court evidence;

2. NIST is currently working to develop a standard for public key

encryption;

3. NIST is also studying message digest algorithms to determine their appropriateness for certification--- at this time, no message digest algorithms have been accepted as standards; and

4. the Department of the Treasury has a system that replaces written signatures on disbursements that has been accepted by the GAO. This system uses DES to calculate a message digest based on a binary key.

In the absence of defined standards, an attempt has been made to design a

system that will meet the proposed legal criteria. The system is flexible enough in its use of both encryption and message digest algorithms to facilitate adherence to standards that will be defined in the near future in these two areas. Due to their involvement in this area, both NARA and the GAO have expressed an interest in the approach described in this document.

- Checklist

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

Next: References **Up:** Conclusions
Previous: Conclusions

Checklist

In order to provide reasonable certainty of the legal acceptance of records in the event of a dispute, the following steps should be verified as having been completed.

Review the security of the computer system to verify that documents and programs are adequately protected against unauthorized access.

Review the procedures for archiving documents with NARA.

Verify that the documentation for the electronic signature system shows that the same processes are used to create, store, and retrieve similar documents.

Review the system with legal counsel and senior information records management.

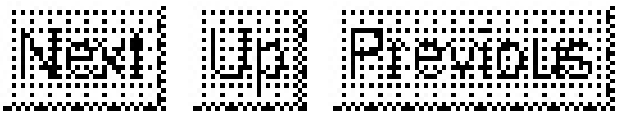
Seek GAO certification of the electronic signature system.

Provide an initial written agreement for each person authorized to use electronic signatures that defines the procedures and protocols for the use of electronic signatures. See Sect. ▨ for specifics about this agreement.

As can be seen from the legal issues, research issues, and design issues, an electronic--signature system is quite complex if it is to be effective. It is imperative to understand that to carry this complexity to fruition that a mutiphased approach is necessary. Further research with the GAO, NARA, and NIST are required in order that the decisions that the PERM design team will be valid for an indefinite and lengthy time period. The choice of message digest algorithm and the choice of signature sealing function are crucial to the mathematical vigor and security of the signature process. The approval of the PERM choices by these agencies provide a considerable durability to the

lifetime of PERM. These processes of approval take time. For this reason PERM is being designed such that a choice of algorithm today will not be heavily impacted by the need to change an algorithm tomorrow.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Next: About this document **Up:** No Title
Previous: Checklist

References

1

C. H. Meyer and S. H. Matyas, *Cryptography: A New Dimension in Computer Data Security -- A Guide for the Design and Implementation of Secure Systems*, John Wiley and Sons, New York, 1982.

2

R. N. Corley and W. J. Robert, *Dillavou and Howard's Principles of Business Law*, 9th ed., Prentice-Hall, Englewood Cliffs, N.J., 1971.

3

``Restatement of the Law,'' Agency(2d), Sec. 1 (1).

4

``Uniform Commercial Code,'' 1972 Official Text with Comments, American Law Institute and National Conference of Commission on Uniform State Laws.

5

D. E. Denning, *Cryptography and Data Security*, Addison--Wesley Publishing Co., Reading, Mass., 1982.

6

National Bureau of Standards, *Data Encryption Standard*, 1977, FIPS PUB 46, Washington, D. C.

7

Financial Institution Message Authentication (Wholesale) X9.9-1986 (Approved August 15, 1986), X9 Secretariat, American Bankers Association, 1120 Connecticut Avenue, Washington, D.C. 20036, 1986.

8

IAB Privacy Task Force, *RFC 1113, Message Encipherment and Authentication Procedures*, 1989.

9

IAB Privacy Task Force, *RFC 1115, Algorithms, Modes, and Identifiers*, 1989.

10

``Legal Issues in Records Organization," National Archives and Records Administration, 1988, CFR Pt. 1234 , RIN: 3095-AA29, Draft.

11

Federal Rules of Evidence 803(6).

12

``State vs. Cary," 99 N.J. Sup. 323, 239 A.2d 680, aff'd, 56 N.J. 16, 264 A.2nd, p. 209, 1970.

13

S. G. Akl, ``Digital Signatures: A Tutorial Survey,'' *IEEE Computer*, 15--24 (1983).

14

D. E. Denning, ``Digital Signatures with RSA and Other Public--Key Cryptosystems,'' *Communications of the ACM*, **27**(4),388--92 (1984).

15

R. R. Jueneman, ``Electronic Document Authentication,'' *IEEE Network Magazine*, **1**(2),17--23 (1987).

16

R. R. Jueneman, ``A High Speed Manipulation Detection Code," p. 329 in *Advances in Cryptology: Proc. of Crypto86*, Springer-Verlag, Berlin, 1987.

17

R. L. Rivest, *The MD4 Message Digest Algorithm*, Technical report, RSA Data Security, Inc., Redwood City, Calif., 1990.

18

C. P. Pfleeger, *Security in Computing*, Prentice--Hall, Inc., 1988.

19

M. F. Theofanos, Y. H. Etheridge,

G. A. Miller, and D. V. Skyberg, *Prototype Electronic Record Management Software Design Document*, K/DSRD-- 471, Martin Marietta Energy Systems, Oak Ridge, TN, 1990.

20

J. G. Steiner, C. Neuman, and J. Schiller, *Kerberos: An Authentication Service for Open Network Systems*, Project Athena, Massachusetts Institute of Technology, Cambridge, Mass., 1988.

21

T. Athanasiou, ``Encryption

Technology, Privacy, and National Security," *Technology Review*, 57--64 (1986).

22

W. Bollinger and D. Ellingern, ``Evolving United States Information Policy and Its Effect on International Access to Online Technical Databases," presented at the Eleventh International Online Information Meeting, London, 1987.

23

B. Bryant, *Designing an Authentication System: a Dialogue in Four Scenes*, Project Athena,

Massachusetts Institute of Technology, Cambridge, Mass., 1988.

24

P. Christoffersson, ``Message Authentication and Encryption Combined," *Computers & Security*, **7**(1),65--71 (1988).

25

F. Cohen, ``A Cryptographic Checksum for Integrity Protection," *Computers & Security*, **6**,505--10 (1987).

26

D. W. Davies, ``Public Key

Ciphers and Signatures," *Information Age*, **6**(1),25--29 (1984).

27

D. W. Davies and W. L. Price, ``Digital Signatures--An Update," *Proceedings of the Seventh International Conference on Computer Communication*, Sydney, Australia, 1984.

28

W. Diffie, ``The First Ten Years of Public Key Cryptography," *Proceedings of the IEEE, May 1988*, 1988.

29

U.S. Department of Commerce, *Guidelines On Evaluation of Techniques For Automated Personal Identification, FIPS PUB 48*, 1977.

30

U.S. Department of Commerce, *Guidelines On User Authentication Techniques For Computer Network Access Control, FIPS PUB 83*, 1980.

31

U.S. Department of Commerce, *Computer Data Authentication, FIPS PUB 113*, 1985.

32

V. Fernandez, C., J. A. Troya, and J. Sanchez, ``Automatic Generation of Digital Signatures for Validating and Authenticating Messages," pp. 231--234 in *ISMM International Symposium on Mini And Micro--Computers and their Applications*, Montreal, Canada, 1985.

33

R. M. Andersen et al., *Electronic Records: Legal and Policy Considerations. A Report of the Electronic Records Committee*, Technical report, National Science Foundation, Washington, D. C. 20550, 1987.

34

``Part V: National Archives and Records Administration, General Services Administration," 1990, 36 CFR Pt.1234, 41 CFR Pt. 201-45, Electronic Records Management; Final Rules.

35

G. S. Kondos, ``Admissibility of Electronically Filed Federal Records as Evidence: A Guideline for Federal Records Managers or Custodians-DRAFT," United States Department of Justice, Justice Management Division, Office of Information Technology, Systems Policy Staff, 1986.

36

C. Meadows and D. Mutchler, ``Matching Secrets in the Absence of a Continuously Available Trusted Authority," *IEEE Transactions on Software Engineering, SE--13*, **2**,289--92 (1987).

37

S. Miller, B. Neuman, J. Schiller, and J. Saltzer, *Kerberos Authentication and Authorization System*, Project Athena Technical Plan, Massachusetts Institute of Technology, Cambridge, Mass., 1987.

38

L. Neuwirth, ``A Comparison of Four Key Distribution Methods,'' *Telecommunications* (1986).

39

D. B. Newman, J. K. Omura, and R. L. Pickholtz, ``Public Key Management For Network Security,'' *IEEE Network Magazine*, **1**(2),11--16 (1987).

40

J. K. Omura, ``A Computer Dial Access System Based on Public--Key Techniques,'' *IEEE Communications Magazine*, **25**(7),73--79 (1987).

41

G. Pluimakers and J. van Leeuwen, ``Authentication: A Concise Survey,'' *Computers & Security*, **5**,243--50 (1986).

42

A. Salomaa, ``A Public--Key Cryptosystem Based on Language Theory,'' *Computers & Security*, **7**(1),83--87 (1988).

43

P. Seipel, ``Pitfalls of the Electronic Revolution,'' *Information Age*, **8**(4),215--19 (1986).

44

G. J. Simmons, ``A Cartesian Product Construction for Authentication Codes that Permit Arbitration," *Journal of Cryptology* (1987).

45

G. J. Simmons, was to be published in Communications of the ACM, 1987.

46

M. Smid, E. Barker, D. Balenson, and M. Haykin, ``Message Authentication Code (MAC) Validation System: Requirements and Procedures," National Bureau

of Standards Special Publication 500-156, 1988.

47

V. L. Voydock and S. T. Kent, ``Security Mechanisms in High--Level Network Protocols," *Computing Surveys*, **15**(2),135--71 (1983).

48

P. Zimmermann, ``A Proposed Standard Format for RSA Cryptosystems," *IEEE Computer*, **7**,21--34 (1986).

49

M. Smid, Technology Section,

National Institute of Science and Technology (NIST), Personal Communication, 09:23 Tue. 19 June 1990.

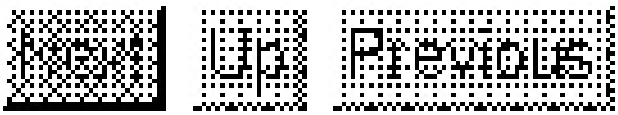
50

C. Martin, Treasury Electronic Certification System, Personal Communication, 11:13 Wed. 20 June 1990.

51

J. Powell, Financial Mgmt. Service, Department of Treasury, Personal Communication, 14:39 20 June 1990.

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996



Up: No Title **Previous:** References

About this document ...

This document was generated using the **LaTeX2HTML** translator Version 95 (Thu Jan 19 1995) Copyright © 1993, 1994, Nikos Drakos, Computer Based Learning Unit, University of Leeds.

The command line arguments were:
latex2html `Esig.tex`.

The translation was initiated by John W. Wooten on Thu Aug 15 19:50:02 EDT 1996

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996

...writing

See Reference [4]: Sec. 1-201 (39)

...signature

See Reference [4]: Sec. 3-401 (2)

John W. Wooten
Thu Aug 15 19:50:02 EDT 1996