Office of Legislative Counsel

Authentication of Primary Legal Materials and Pricing Options

December 2011

Primary Contributors:

Brad Chang, Xcential Group, and Dragomir Cosanici, Office of Legislative Counsel, with special thanks to Diane Boyer-Vine, Bill Behnk, Will Chan, Fred Messerer, and Mendora Servin, Office of Legislative Counsel. This project was funded by the Minnesota Historical Society through its grant from the Library of Congress' National Digital Information Infrastructure and Preservation Program (NDIIPP).

Authentication of Primary Legal Materials and Pricing Options

Abstract

The recent passage of the Uniform Electronic Legal Material Act (UELMA) has brought to the forefront the issue of costs of authenticating primary legal materials in electronic format. This white paper briefly reviews five methods of electronic authentication. These methods are based on trustworthiness, file types, effort to implement, and volume of electronic documents to be authenticated. Six sample solutions are described and their relative costs are compared. The white paper also frames the legal landscape and background of authentication for primary legal materials in electronic format, and provides context and points to applicable resources. The aim of this collective effort is to promote the understanding of costs related to authentication and invite further discussion on the issue.

DISCLAIMER:
This white paper is a brief study and review of relative costs associated with the authenticating primary legal materials in electronic format. It is not intended to offer legal advice. Please consult an attorney for assistance with specific concerns or advice.

Any comments, corrections, or recommendations may be sent to the OLC project team, in care of:

Dragomir Cosanici
Supervising Librarian
Office of Legislative Counsel
dragomir.cosanici@lc.ca.gov/ (916) 341-8030

Authentication of Primary Legal Materials and Pricing Options

Introduction and Brief Background

The State of California's Office of Legislative Counsel (OLC) and its partner the Minnesota Historical Society (MHS), as well as the Minnesota's Office of the Revisor of Statutes, have long been interested in enhancing the capacity to preserve primary legal materials in electronic format. One specific area of interest has been the process and cost of authenticating these materials in electronic format because the literature in this field is largely devoid of studies that examine the cost of such authentication efforts.

In late 2011, the MHS, through its grant from the Library of Congress' National Digital Information Infrastructure and Preservation Program (NDIIPP) agreed to fund the OLC's project to test various methods of authenticating primary legal materials in electronic format and assess their costs. The findings from the project are published in this white paper and made widely available, in the hope that other governmental entities weighing whether to authenticate electronic records will have a better idea of the costs involved.

The specific goal of this white paper is to test and compare five different methods of authentication of California's primary legislative documents in electronic format. The studied materials include the chaptered bills, resolutions, state constitution, and state codes of California. The white paper not only addresses the chosen methods of authentication, but also their cost and reviews various software options for authentication.

Traditionally, official (and hence authentic) versions of primary legal sources are found in print publications. The content of print work is consistent once printed, making the text easily verifiable and alterations readily detectible. Today's electronic versions of these primary legal materials lack legal authority because they are largely not authenticated.[1] For example, electronic legal materials could be changed over time as they move from format to format or from server to server. In addition, hackers may easily alter the content of these legal materials without raising much suspicion. Authors of primary legal materials have now recognized that primary sources can be published electronically without losing authoritativeness, often for lower cost than in print, provided that the documents are authenticated.[2] Before states can transition fully into the electronic legal information environment, new procedures must be developed to ensure the trustworthiness of the electronic legal information.

Another important reason for this project is the recent passage of the Uniform Electronic Legal Material Act (UELMA) by the National Conference of Commissioners on Uniform State Laws. Its drafting committee was established to draft a proposed

---

[1] Whiteman, *The Death of Twentieth-Century Authority* (2010) 58 UCLA L. REV. DISCOURSE 27, 38.
[2] *Ibid*.

uniform law that will effectively deal with the authentication and preservation of state electronic legal materials. Consequently, UELMA mandates authentication of electronic legal documents, including "… a method for a user to determine that the record received by the user from the publisher is unaltered from the official record …"[3] The OLC study sheds further light on such methods of authentication as well as their financial feasibility. In addition, the state adoption of UELMA would help address many concerns associated with an increase in the exclusive use of online legal authority.

> "Authentication merely presumes accuracy, and any party disputing the accuracy of legal material in an authenticated electronic record can offer proof as to its inaccuracy. The idea is to provide the same level of assurance of accuracy in an electronic record as it is already available in a printed book. Just as the reader of a book can look at it to determine if its contents have been altered, so should a user of electronic legal material through various authentication methods."[4]

The OLC's definition of authentication is modeled on the U.S. Government Printing Office's 2005 *Authentication* white paper, and later utilized by the American Law Library Association (AALL) in its 2007 state-by-state online primary legal resources survey:

> "An authentic text is one whose content has been verified by a government entity to be complete and unaltered when compared to the version approved or published by the content originator. Typically, an authentic text will bear a certificate or mark that conveys information as to its certification, the process associated with ensuring that the text is complete and unalterered when compared with that of the content originator. An authentic text is able to be authenticated which means that the particular text in question can be validated, ensuring that it is what it claims to be."[5]

Authentication of electronic legal documents is an issue of national importance. A few state governments and agencies across the United States have already begun authenticating electronic legal material and are developing best practices. For example, the State of Arkansas issues its appellate opinions in an authenticated electronic format, with the help of digital signatures.[6] The State of Delaware provides an authenticated electronic version of administrative rules also using a digital signature.[7] Moreover, the Indiana Administrative Code is exclusively published in an electronic

---

[3] <http://www.uniformlaws.org/Shared/Docs/AM2011_Prestyle%20Finals/UELMA_PreStyleFinal_Jul11.pdf> (last visited Dec. 5, 2011).
[4] *Ibid*.
[5] Richard J. Matthews & Mary Alice Baish (2007) Am. Ass'n of Law Libraries, State-By-State Report on Authentication of Online Legal Resources p. 8; <http://www.gpoaccess.gov/authentication/authenticationwhitepaperfinal.pdf> (last visited Dec. 5, 2011).
[6] <https://courts.arkansas.gov/court_opinions/sc/2009a/20090528/published/09-540.pdf> (last visited Dec. 5, 2011).
[7] <http://regulations.delaware.gov/AdminCode/> (last visited Dec. 5, 2011).

format with a "Certificate of Authenticity" to provide the user with a level of confidence in the document that existed with the previously published print version of the code.[8] Finally, the State of Utah authenticates its administrative code using hash values.[9]

This next section summarizes the legal status of electronic primary legal materials in California, and the already existing federal legislation and rules (Federal Rules of Evidence, Federal Rules of Civil Procedure) related to authentication of primary legal materials. This brief analysis is intended to provide additional understanding of the existing legal background for the authentication of primary legal materials.

State and Federal Laws, Legislative Acts and Interpretive Rules

Legal Status of Electronic Records in California

In California, there is no official version of the state's statutory codes, and there is no one state entity that acts as a digital clearinghouse for all electronic records. The Office of Legislative Counsel is required to make the California Codes available to the public in electronic form.[10] However, the Secretary of State is the custodian of all acts and resolutions passed by the Legislature,[11] but the Secretary of State does not maintain an official electronic version of California's laws.

In the State Records Management Act,[12] the Director of General Services is required to "establish and administer in the executive branch of state government a records management program, which will apply efficient and economical management methods to the creation, utilization, maintenance, retention, preservation, and disposal of state records."[13] Although the State Records Management Act applies to electronic records,[14] there is no provision in the act related to the authentication of electronic records.[15] Moreover, the Secretary of State, in consultation with the Department of General Services, is required to approve and adopt appropriate standards for the purpose of storing and recording permanent and nonpermanent documents in electronic media.[16] But those standards have not been finalized.[17]

Given the lack of a digital clearinghouse for electronic records in California, the state entity that has jurisdiction or responsibility over the repository of permanent physical records presumably has jurisdiction or responsibility over the repository of

---

[8] Ind. Code Ann. tit. 4, art. 22, ch. 8, subd. 5(c).
[9] <http://www.rules.utah.gov/publicat/codeudt.htm> (last visited Dec. 5, 2011).
[10] Cal. Gov. Code, §10248, subd. (a)(8).
[11] Cal. Gov. Code, §12160.
[12] Cal. Gov. Code, §14740 et seq.
[13] Cal. Gov. Code, §14745.
[14] Cal. Gov. Code, §14741.
[15] Cal. Gov. Code, §14746 & §14750.
[16] Cal. Gov. Code, §12168.7.
[17] <http://www.sos.ca.gov/archives/local-gov-program/> (last visited Dec. 5, 2011).

electronic records. The content of writing[18] may be proved by an otherwise admissible original.[19] Existing law permits a public employee to certify that a copy of a writing is a correct copy of the original writing.[20] Also, the official record of a writing is prima facie evidence of the existence and content of the original record.[21] A record of a writing is an official record if the "record is in fact a record of an office of a public entity" and a "statute authorized such a writing to be recorded in that office."[22]

As such, a state entity has jurisdiction or responsibility over the repository of permanent records, physical or electronic, if a statute authorizes the writing to be recorded with a specific state entity. However, other than the admissibility of records pursuant to the Evidence Code,[23] there are no statutory provisions in California that require the authentication of primary electronic legal materials.

Uniform and Federal Acts Related to Electronic Government Information

The need for authenticated electronic documents has also been explored by other state legislatures as well as the federal government. While the immediate purpose of each act pertaining to authenticated records may be unrelated, the underlying need for security, reliability, accessibility, and cost effectiveness is universal.

State legislative bodies began to address the advent of electronic communication and commerce in the late 1990s. The Uniform Electronic Transaction Act (UETA) was the first to ensure "that an electronic record of a commercial transaction is the equivalent of a paper record, and that an electronic signature will be given the same legal effect, whatever that might be, as a manual signature."[24] The National Conference of Commissioners on Uniform State Laws promulgated UETA and a little more than a decade later, 47 out of 50 states have adopted it.[25]

The most applicable portion of UETA related to the authentication of electronic legal materials is found in section 12. It states that "if a law requires that a record be retained, the requirement is satisfied by retaining an electronic record of the information in the record which accurately reflect the information set forth in the record after it was first generated in its final form as an electronic record or otherwise, and remains accessible for later reference."[26] Section 12 "assures that the information stored electronically will remain effective for all audit, evidentiary, archival and similar

---

[18] The definition of a writing, as specified in § 250 of the Evidence Code, is broad enough to include virtually every form of data recordation, including information stored electronically or in any other manner, and e-mail or facsimile transmissions (Cal. Civil Discovery Practice (Cont. Ed. Bar 4th ed. 2007), §8.10).
[19] Cal. Evid. Code, §1520.
[20] Cal. Evid. Code, §1530.
[21] Cal. Evid. Code, §1532.
[22] Cal. Evid. Code, §1532, subd. (a).
[23] Cal. Evid. Code, §1520, 1530, 1531, and 1532.
[24] <http://www.law.upenn.edu/bll/archives/ulc/fnact99/1990s/ueta99.pdf> (last visited Dec. 5, 2011).
[25] <http://www.ncsl.org/IssuesResearch/TelecommunicationsInformationTechnology/UniformElectronicTransactionsActs/tabid/13484/Default.aspx?tabid=13484> (last visited Dec. 5, 2011).
[26] See Cal. Civ. Code, §1633.12, subd. (a).

purposes" – provided that there is a "reliable assurance that the electronic record accurately reproduces the information" of the original.[27] Moreover, section 12 is consistent with the Uniform Rules of Evidence §1001(3).[28]

At the federal level, the E-Sign Act of 2000, formally known as the Electronic Signatures in Global and National Commerce Act,[29] is intended to facilitate the use of electronic records and signatures in interstate and foreign commerce.[30] Under the "general rule of validity," the act establishes two fundamental points ensuring the validity and legal effect of contracts entered into electronically:

> (1) that "a signature, contract, or other record relating to such transaction may not be denied legal effect, validity, or enforceability solely because it is in electronic form;"
>
> (2) that a "contract relating to such transaction may not be denied legal effect, validity, or enforceability solely because an electronic signature or electronic record was used in its formation."[31]

The Act defines an electronic signature as "an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record."[32]

Next is the E-Government Act of 2002. Its purpose is to improve the management and promotion of electronic government services and processes by establishing a Federal Chief Information Officer within the Office of Management and Budget, and by establishing a framework of measures that require using Internet-based information technology to improve citizen access to government information and services, as well as other purposes.[33] In that vein, the definition section of the act outlines that information security means "integrity … guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity."[34]

---

[27] U. Electronic Transactions Act (1999), com. to § 12, p. 42 <http://www.law.upenn.edu/bll/archives/ulc/fnact99/1990s/ueta99.pdf> (last visited Dec. 5, 2011).
[28] Com to § 5, p. 38.
[29] 15 U.S.C. §7000 et seq.
[30] <http://www.gpo.gov/fdsys/pkg/PLAW-106publ229/pdf/PLAW-106publ229.pdf> (last visited Dec. 5, 2011).
[31] 15 U.S.C. §7001a(1)&(2).
[32] 15 U.S.C. §7006(5).
[33] <http://www.gpo.gov/fdsys/pkg/PLAW-107publ347/pdf/PLAW-107publ347.pdf > (last visited Dec. 5, 2011).
[34] 44 U.S.C. §3542(b)(1)(A).

Federal Rules of Evidence and Federal Rules of Civil Procedure

The most pertinent provisions relating to the authentication of primary legal materials can be found in Article IX (Authentication and Identification), specifically Rules 901 and 902.

"**Rule 901. Requirement of Authentication or Identification**

"(a) General provision.—The requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims.[35]

"(b) Illustrations.—By way of illustration only, and not by way of limitation, the following are examples of authentication or identification conforming with the requirements of this rule:

* * *

"(7) Public records or reports.—Evidence that a writing authorized by law to be recorded or filed and in fact recorded or filed in a public office, or a purported public record, report, statement, or data compilation, in any form, is from the public office where items of this nature are kept.

* * *

"(9) Process or system.—Evidence describing a process or system used to produce a result and showing that the process or system produces an accurate result.

"(10) Methods provided by statute or rule.—Any method of authentication or identification provided by Act of Congress or by other rules prescribed by the Supreme Court pursuant to statutory authority."[36]

"Rule 902. Self-authentication

"Extrinsic evidence of authenticity as a condition precedent to admissibility is not required with respect to the following:

* * *

[35] <http://www.uscourts.gov/uscourts/RulesAndPolicies/rules/2010%20Rules/Evidence.pdf> (last visited Dec. 5, 2011).
[36] *Ibid*.

"(4) Certified copies of public records.—A copy of an official record or report or entry therein, or of a document authorized by law to be recorded or filed and actually recorded or filed in a public office, including data compilations in any form, certified as correct by the custodian or other person authorized to make the certification, by certificate complying with paragraph (1), (2), or (3) of this rule or complying with any Act of Congress or rule prescribed by the Supreme Court pursuant to statutory authority.

"(5) Official publications.—Books, pamphlets, or other publications purporting to be issued by public authority.[37]

* * *"

Rule 1002 establishes that an original is required as evidence. However, Rule 1003 establishes that a duplicate is admissible unless "(1) a genuine question is raised as to the authenticity of the original or (2) "in the circumstances it would be unfair to admit the duplicate in lieu of the original."[38]

"Rule 1005. Public Records

"The contents of an official record, or of a document authorized to be recorded or filed and actually recorded or filed, including data compilations in any form, if otherwise admissible, may be proved by copy, certified as correct in accordance with rule 902 or testified to be correct by a witness who has compared it with the original. If a copy which complies with the foregoing cannot be obtained by the exercise of reasonable diligence, then other evidence of the contents may be given."[39]

The Federal Rules of Civil Procedure, specifically rule 34(a)(1)(A) addresses "any designated documents or electronically stored information-including writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations-stored in any medium from which information can be obtained either directly or, if necessary, after translation by responding party into a reasonably usable form."[40]

[37] *Ibid.*
[38] *Ibid.*
[39] *Ibid.*
[40] <http://www.law.cornell.edu/rules/frcp/Rule34.htm> (last visited Dec. 5, 2011).

Authentication Methods and Cost Options

Authentication Requirements

Primary legislative documents include chaptered bills, chaptered resolutions, state constitution, and state codes. Primary legislative documents, as described above, are by nature public documents. Users could be any member of the public or any organization such as a government agency, corporation, or non-profit group. The documents are likely to be accessed anonymously, via a legislative Web site. They may also be delivered to specified entities such as legal publishers.

Best practices for security are described the National Institute of Standards and Technology (NIST).[41] Because of the public nature of primary legislative documents, some traditional security concerns, such as restrictive access, encryption, and end user identification are not required for this application. The two basic authentication requirements that do remain are that each document must be able to verify:

> **Authenticity of Origin**—verification that the document is actually from the source that it claims to come from *(e.g. the Office of Legislative Counsel).*
>
> **Document Integrity**—verification that the document has not been altered since it left its source.

A document's authenticity and integrity must be maintained through any intermediate storage, processing, or transmission. Consumers must be able to verify authenticity and integrity, no matter how they received the document, what chain of custody the document has had, or even if they do not know how the document came to them. For instance, a document may have been received as an attachment to an email from an unknown source. It must still be verifiable.[42]

PDF is expected to be the primary authenticated file format. Authenticated HTML and XML formats also need to be considered.

Some aspects of authentication go beyond the scope of this white paper. While Appendix A reviews prices for commercial software and services, Appendix B briefly mentions authentication-related topics that are not covered in this paper. A technical terms glossary is found in Appendix C.

---

[41] <http://tools.ietf.org/html/rfc2818> (last visited Dec. 23, 2011).

[42] < http://csrc.nist.gov/publications/nistpubs/800-33/sp800-33.pdf> (last visited Dec. 8, 2011).

Methods to Authenticate

In this section, we examine the major methods of authentication. These methods offer varying levels of security, require varying levels of action by the reader, and require varying levels of effort to implement. Currently available technologies support several methods of secure document delivery, including:

- Secure Web Sites
- Document Hashes (Digests)
- Digital Signatures
  - Self-Signed
  - Public Key Infrastructure (PKI) Certificates
- Proprietary Solutions
- Visual Signatures

We discuss each of these methods below. We recommend against some methods because they do not meet both of the minimum requirements described in the previous section. Following this, we discuss the influence of file-types (such as PDF or XML) on the authentication methods.

- **Secure Web Sites**

Secure Web sites are very common today. The check-out pages of most shopping Web sites are secure so that credit card information is not compromised. The methods and technologies to implement a secure Web site are well known[43], including HTTPS, TSL/SSL, and digital certificates. The technologies guarantee to the user that the Web site is truly the one it claims to be. The technologies also provide a secure communication channel to the Web site to ensure privacy.

A legislative body could use a secure Web site to ensure document authenticity and integrity to the Web site user. While the users are on the Web site, they can be assured that the documents are authentic. However, once a document leaves the Web site, it is outside the control of the security system and can no longer be authenticated. So, if a user saves a document to his or her PC, then later opens it or emails it, it is no longer authenticated.

For this reason, secure Web sites alone do not meet the minimum requirements for document authentication. However, a secure Web site can play a role in other authentication methods discussed below.

- **Document Hashes (Digests)**

Hashes are the foundation of nearly all document authentication methods. Hashes are cryptographic functions that, given a piece of data such as a document, compute a number, often referred to as a "hash code" or a "digest." The hash code is analogous to a thumbprint. The principal characteristic of a hash is that, if the document

[43] <http://tools.ietf.org/html/rfc2818> (last visited Dec. 23, 2011).

changes, the hash will change. Hashes can therefore be used to ensure document integrity. The most common hashing algorithms are:

MD5—Widely used, but recommended against due to proven security weakness.[44]

SHA-1—Widely used, but no longer recommended due to theoretical security weakness.[45]

SHA-2—Currently recommended.

SHA-3—Not yet available. In development by NIST[46].

The primary limitation of hashes is that, by themselves, they do not authenticate the origin of the document and are, therefore, not sufficient for document authentication.

However, hashes can be used in combination with a secure Web site to authenticate documents. For instance, the hash for a document can be posted on a secure Web site, and consumers of the document can verify that the hash from the Web site matches the hash computed directly from the document. See Figure 1. In this fashion, the hash guarantees document integrity and the secure Web site guarantees the authenticity of origin. The comparison of hash codes can be done on a user's machine using common tools. A legislature could also provide a document validation service on its Web site.

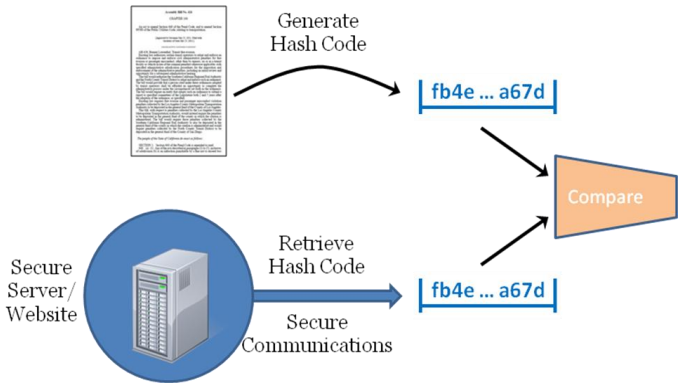


Figure 1. Validating a hash code

- **Digital Signatures**

Digital signatures build on document hashes, adding the identity of the signer in a secure manner. The addition of the signer's identity allows the document source to

[44] US-CERT; <http://www.kb.cert.org/vuls/id/836068> (last visited Dec. 8, 2011).
[45] <http://csrc.nist.gov/groups/ST/hash/policy.html> (last visited Dec. 8, 2011).
[46] <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html> (last visited Dec. 8, 2011).

be authenticated. The signer's identity is encapsulated in an X.509 standard digital certificate,[47] making the process open and standard.

Figure 2 illustrates the process of creating an X.509-based digital signature. The document hash is encrypted with a "private key" to which only the signer has access. It is *imperative* that the private key be managed in a secure manner. The certificate is also attached to the document. The certificate includes a "public key" which the document consumer can use to validate that the document did indeed come from the signer and that the document has not been altered since it was signed. In this fashion, a digitally signed document is a self-contained, authenticatable document. An external hash or Web site is not required.

A primary advantage of digital signatures approach over only hash codes is that the digital certificate only needs to be retrieved once to validate multiple documents, whereas the hash code would have to be retrieved for each document that is validated. For example, in case of hash codes, if 10 documents need to be validated, 10 hash codes would be retrieved. In case of digital signatures, a single signature could be used to validate 10 different documents. This is made possible because the digitally signed document contains both the unique hash code and the creator's certificate.

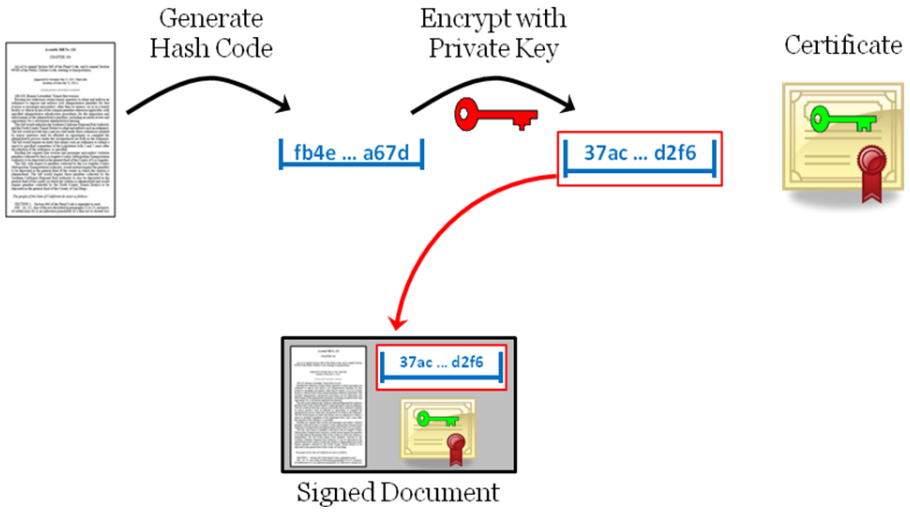


Figure 2. Creating a digitally signed document

With digitally signed documents, the question is who created the digital certificate and does the document consumer trust that issuing authority? For purposes of this white paper, there are two classes of certificates: self-signed and PKI.

[47] ITU, Public-key and attribute certificate frameworks, <http://www.itu.int/rec/T-REC-X.509-200811-I/en> (last visited Dec. 8, 2011).

- **Self-Signed Certificate**

Utilizing tools like Adobe Acrobat, any user may create a digital ID (a certificate and the corresponding private key),[48] which can then be used to digitally sign documents. These IDs are called *"self-signed"* because their authenticity is not guaranteed by anyone but the creator. An analogy would be a driver's license issued by the driver himself, in lieu of it being issued by a trusted government authority.

Most entities that sign documents find this self-signing limitation unacceptable, except for internal use. However, a state government or a state legislature has the standing and position that readers could reasonably trust a self-signed certificate.

In practice, document consumers are not likely to trust a self-signed certificate that is contained within the document, as the certificate may be easily forged. Consumers are, however, more likely to trust a self-signed certificate obtained from an official and secure state Web site.

- **Public Key Infrastructure (PKI) Certificates**

To avoid the need for users to download and trust certificates, a legislative body can purchase a certificate from one of a set of well-known Certification Authorities (CAs), such as VeriSign or Entrust. The certificate is issued in the name of the legislative body (e.g., Office of Legislative Counsel (OLC)). This certificate is backed by the Certification Authority. This process is known as "chaining." The highest level in a certificate chain is referred to as the "root." For instance, if OLC purchases a certificate from VeriSign, that certificate will chain up to the VeriSign root certificate. See Figure 3.

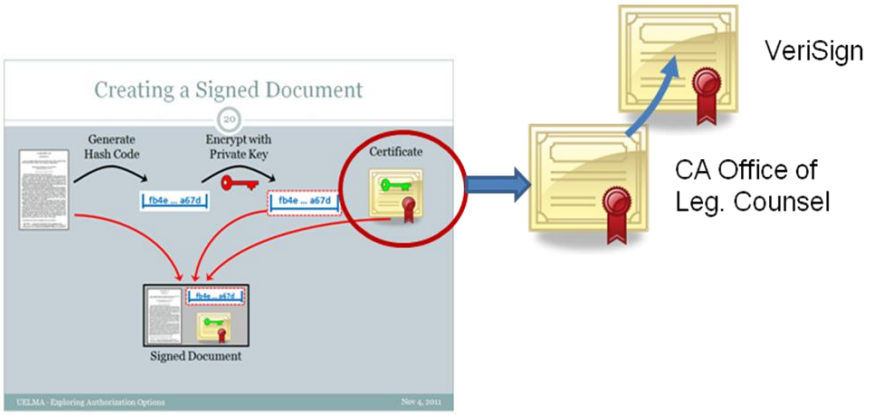


Figure 3. Certificate Chaining

[48] Adobe Acrobat X command: Edit>Protection>Security Settings>Add ID>new digital ID.

Windows and Mac both have a certificate storage system built into the operating system; some applications, such as Adobe Reader and some dedicated digital signature applications, also have a certificate store. These systems automatically maintain certain trusted root certificates, such as VeriSign or Entrust. In the example above, because OLC chains up to a root that is in the operating system store, the end user would not need to manually obtain the OLC certificate to trust it.

- **Proprietary Solutions**

  A few vendors provide a proprietary solution to document authentication.

  AbsoluteProof from Surety[49] provides a time-stamping service that guarantees that a file existed in its current form at a certain time.

  ProofMark by ProofSpace[50] is a transient key technology that renews keys to avoid problems of expiration.

  TruSeal from Tru Data Integrity[51] (United Kingdom) authenticates any document by creating a separate TruSeal file that contains a document hash, a time stamp, and signer identity information.

- **Visual Signatures**

Using a tool like Adobe Acrobat,[52] users may place a scanned signature, seal, or stamp on a PDF file. Indeed, any image may be placed on a PDF. Using other tools, the same may be done for other file types. Such an image gives the appearance of official certification, but it provides no guarantees or security whatsoever. In fact, if an organization uses a visual signature officially, it opens the opportunity for an unofficial source to use the same image on false documents. Because of this vulnerability, we recommend against the use of visual signatures alone.

File-Type Considerations

File types, such as PDF, XML, and HTML, have varying capabilities to store and process authentication information. We address these in the sections below. PDF has the most robust capabilities. XML has a standard for digital signatures, but current industry support is not particularly strong. Other file types such as HTML have very limited support and are best served by solution that keep the original file intact.

[49] <http://www.surety.com/> (last visited Dec. 8, 2011).
[50] http://www.proofspace.com/ (last visited Dec. 8, 2011).
[51] http://www.tru-dataintegrity.com (last visited Dec. 8, 2011).
[52] See Appendix 1 for list of products and services.

PDF Documents

PDF began as a proprietary file format. It is now an open standard published by the ISO.[53] PDF includes robust digital signature fields. In addition, the Adobe Reader product has inherent support for validating signed documents (see Figure 4) and for maintaining a list of trusted identities. These facilities make the authentication of PDF documents much easier for both the publishing entity and the document consumer.

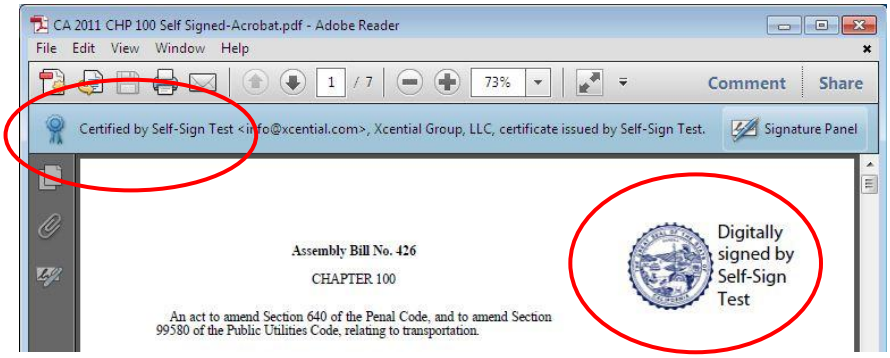


Figure 4. Verified Digital Signature in Adobe Reader

The sections below describe four methods of signing PDF documents using the technologies discussed above.

Method 1: Self-Signed PDF Digital Signature

If a PDF document is signed using a self-signed certificate, as described above, then when a user opens the document, Adobe Reader will indicate that the content has not been modified since it was certified, but that the signer's identity is unknown. See Figure 5.

Figure 5. Self Signed Certification

In order for this document to be authenticated, the user must tell Adobe Reader to trust the identity that signed it. See Figure 6. This process is tedious for the end user and is generally discouraged by the security community because the user cannot tell if the document was signed falsely.

To avoid the problem of trusting the document itself, a legislative body could post a signing certificate on its secured Web site. Users who wish to verify documents from that legislative body would download the certificate and import it into Adobe

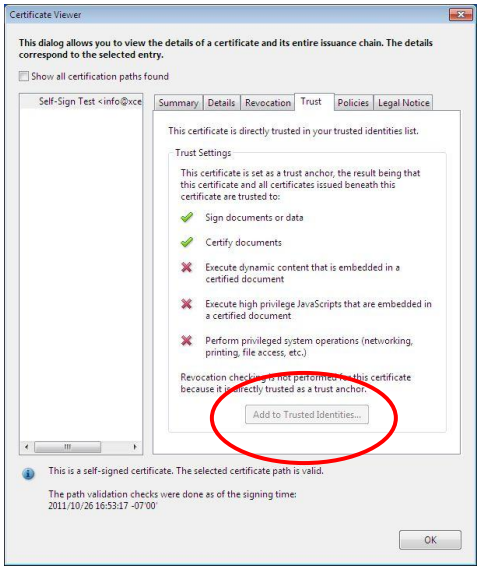


Figure 6. Trusting a Certificate

[53] International Organization for Standards, ISO 32000-1:2008.

Reader. After that, Adobe Reader will automatically validate all documents that were certified with that key. This is more secure than trusting the self-signed document itself, but it is a tedious process for the reader.

So, a self-signed PDF is a partial solution which is viable if the users can trust the source of the document.

Method 2: PDF Digital Signature - PKI Certificate

If a legislative body purchases a PKI certificate from well-known Certificate Authorities (CAs), the document consumer will not need to retrieve and trust the certificate from the legislature. Instead, the consumer only needs to configure Adobe reader to search the operating system's certificate store when verifying signatures. This will enable Adobe reader to trust the CA, which will enable trusting the legislature. By default, this option is turned off. So users will usually need to turn it on. See Figure 7.

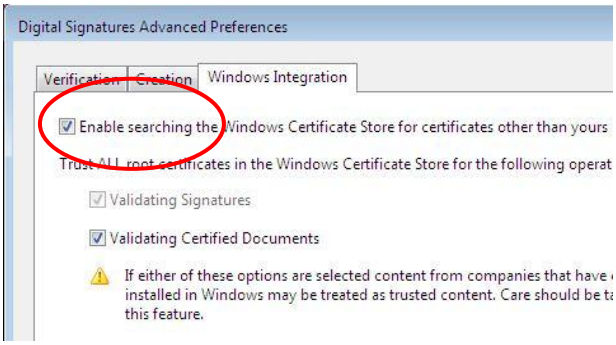


Figure 7. Windows Certificate Store

Method 3: PDF Digital Signature - Adobe CDS System Certificate

The Adobe Certified Document Services (CDS) system is the most convenient authenticating method for PDF files. In CDS, the signing certificate is "chained up" to the Adobe root. A certificate that chains up to the Adobe root is inherently trusted by Adobe Reader. So CDS certified documents are verified without any action required by the user. As small number of vendors are licensed to issue CDS certificates. Furthermore, a CDS certificate must be protected by a Hardware Security Module (HSM). The HSM may be as simple as a USB key or as complex as multiple rack mounted network devices for high volume batch signing. For example, the CDS system is used by the Government Printing Office (GPO) for authenticated documents.[54]

Method 4: PDF Digital Signature - Adobe AATL Certificate

Adobe Approved Trust List (AATL) is a certificate trust program introduced with Adobe Reader 9. AATL is similar in its goal to the CDS program. The main difference is that with CDS, the list of trusted root certificates is preset in Adobe Reader software and not changeable. With AATL, the list of root authorities is updated and refreshed every 30 or 90 days in Adobe Reader and Acrobat.

[54] < http://www.gpo.gov/pdfs/authentication/authenticationwhitepaper2011.pdf> (last visited Dec. 8, 2011).

XML Documents

Authenticating XML is more difficult than PDF because, unlike PDF, which is a single format standard, XML can have many different forms. In particular, we expect that states will have varying XML formats for their data. Thus, some states will have to extend their XML formats to provide a place for the signature and choose the details of the signature.

There is a well-accepted international standard for signing XML documents: XML Signature,[55] commonly referred to as "XML DSig." The standard provides guidance on how to add a signature to an XML document and how to process a signature. At the current time, there are no packaged solutions for creating and validating XML DSig. Organizations utilizing this route will need to use lower level libraries and invest in custom implementations. For instance, the organization would have to provide for validation in one form or another, for example, special software to be downloaded by recipients, or a Web site providing validation services to visitors uploading documents to it.[56]

The validation problem could be simplified if XML validation by the general public is determined to be unnecessary. Large document consumers that desire authenticated XML documents could be required to implement their own validation solutions. This might not be an unreasonable burden for large organizations because the relevant standards are well established and are supported by software libraries.

HTML Documents

The methods to authenticate HTML are not as well developed as the methods for PDF or even for XML. There is no standard way to place a signature in an HTML document. There is no standard software that will embed a signature. And there is no standard software to validate signed HTML documents.

This does not mean, however, that HTML is impossible to sign. Using XML Signature for instance, it is possible to generate a signature for an HTML document. The signature would be detached, not embedded, in a separate file that does not necessarily travel with the HTML document. A user wishing to authenticate the document must find the signature. Solutions similar to the ones outlined in the hash code section could be applied.

Another way to sign HTML documents is to use signed envelopes. In addition to embedded signatures and detached signatures, it is possible to sign any type of document or set of documents by placing them in a signed envelope. This is analogous to placing documents in a "zip" file and signing the file. Signed envelopes are widely used in signed email exchange. The sender's mailer wraps the message (often HTML) in an

---

[55] <http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/ > (last visited Dec. 8, 2011).

[56] Such a service has been implemented by, for instance, the Austrian state; <http://www.digitales.oesterreich.gv.at/site/6793/default.aspx> (last visited Dec. 8, 2011). The service can validate PDF documents signed by the state, as well as XML documents like proof of residency, identity link of the Austrian citizen card, electronic mandate, or electronic invoice.

envelope, signs it, and sends it up. The recipient's mailer opens the envelope and displays the message together with signature validation results.

Although there is commercial software available for wrapping, signing, unwrapping, and signature validation,[57] the general public typically does not install it, making enveloped signatures inconvenient.

This inconvenience can be circumvented by using a PDF document as the envelope. A PDF file can contain arbitrary file attachments. When a PDF file is signed, the file attachments are signed as part of the same file. This can be done as follows: 1) create a PDF document; 2) attach other documents (such as HTML or XML) to it as file attachments; and 3) sign the PDF and publish it. These steps can be accomplished either manually or they may be automated with software. Adobe Reader will validate this type of PDF as usual, certifying the origin and integrity of both the PDF contents and the attachments. This approach, however, is not supported in the long-term storage format PDF/A.

System Components

In this section we examine the system components that a state might use to implement authentication. We describe these components and identify leading vendors for each. Detailed product listings and pricing are given in Appendix A.

The four main components of a document authentication system are:

- **Signing Certificate**
- **Hardware Security Modules (HSM)**
- **Signing Software**
- **Verification Tools**

Below are the leading options available for each of these components. Each of the lists below gives representative options. The lists are not exhaustive.

- **Signing Certificate**: the electronic codes uniquely tied to a person or an organization, and which are used for signing. Various types of certificates are available as is outlined below.

Several types of digital certificates are available, including:

Self-Generated Certificate. This type of certificate can be generated in Adobe Acrobat or Java software libraries. Other than the cost of the software, there is no cost of this type of certificate. The certificate is self-rooted.

[57] <http://www.aloaha.com/wi-software-en/aloaha-sign.php> (last visited Dec. 8, 2011).

Purchased Personal Certificate. Commercial vendors will issue certificates in an individual's name. Modest checking is done before issuing the certificate. The certificate root will be either the vendor or Adobe (CDS).

Purchased Group Certificate. Commercial vendors will issue certificates in the name of a group. Substantial checking is done before issuing the certificate. The certificate root will be either the vendor or Adobe (CDS).

Purchased Enterprise CDS Certificate. Commercial vendors will issue an Adobe CDS certificate in the name of an enterprise. Thorough checking is done before issuing the certificate. Adobe (CDS) will be the root. A Hardware Security Module (HSM) is required to store the certificate.

Managed PKI. If an organization needs multiple certificates, a few vendors provide certificate management services to allow the organization to create and manage its own certificates. We do not envision that this is required for a typical legislative body.

The leading certificate vendors are:

Entrust - Offers Group and Enterprise Certificates
GlobalSign - Offers Group Certificates
Symantec/VeriSign - Offers Managed PKI and Organizational Certificates

Please note that SSL certificates (for secure Web sites) and code signing certificates (for software) are not appropriate for document signing.

- **Hardware Security Modules (HSM)**: a secure physical storage location for a certificate. An HSM is required for CDS-level certificates. It is optional for other types of certificates.

HSMs are available as three different physical forms:

USB Key. For personal and group CDS certificates, the certificate may be stored on a specialized USB device.

Add-on Devices. For enterprise use, the CDS certificate may be stored on a PC Card. This allows access to the certificate on that computer.

Network Appliances. For enterprise use, the CDS certificate may be stored on a dedicated network device. This allows access to the certificate from multiple computers on the intranet.

There is one primary HSM vendor, SafeNet. SafeNet devices are resold by certificate vendors for certificate storage.

- **Signing Software**: the software application that attaches the signing certificate to a document. Most signing software can apply any type of certificate to a document as long as it has access to the certificate.

The leading vendors and applications for signing documents are:

Adobe Acrobat. Adobe Acrobat (Standard and Pro versions) are desktop applications that can manually sign PDF documents using any type of key. Acrobat, by itself, does not provide for batch or automatic signing.

iText. iText is a Java library for creating and manipulating PDF files. It is available in free and licensed versions. It supports batch automatic signing of PDF files.

Aloaha. Aloaha PDF Signator is a desktop application for Windows that can sign PDF documents, using either standard software certificates or smart cards. Signing is entirely manual.

Adobe LiveCycle. Adobe LiveCycle Digital Signatures ES2 is a module in the Adobe LiveCycle line of server products. This module applies digital signatures to PDF files in batch mode, supporting very high volume throughput.

Java Libraries. Java defines a set of APIs spanning major security areas, including cryptography, public key infrastructure, authentication, hashing, secure communication, and access control. These APIs allow developers to integrate security mechanisms into their application code. Since Java 1.6, XML Digital Signatures are also supported.
Other software providers such as Cryptolog (http://web.cryptolog.com) and Crypto toolkit (http://jce.iaik.tugraz.at/) supply their own commercial libraries and products to allow for signing of documents. They supplement the security functionality of the default JDK and support JDK older than 1.6.

- **Verification Tools**: the software tool that the end user uses to validate the signature of a document. Verification tools are often specific to the type of document (e.g., Adobe Reader for PDF).

These are the leading vendors and applications for verifying signatures:

Adobe Reader. Adobe Reader is the most widely used PDF display tool. Adobe Reader X supports digital signature validation and the extracting of attached files. "X" indicates version 10.

Web Browser. Web browsers contain software implementing the HTTPS protocol, which uses underlying cryptographic (either SSL or its successor, TLS) to establish a secure web connection. When a connection is attempted, the web server presents to the browser an SSL certificate. This is essentially the same kind of certificate as those used for signing documents, but issued for a different purpose (secure Web connections instead of signing) and tied to a domain name instead of to a person's or

organization's name. The browser validates the certificate, verifying that it chains up to one of the trusted roots maintained by the browser and that the domain name in the certificate matches that to which the connection is attempted. If the certificate is deemed valid, a secure, encrypted connection is established guaranteeing that the accessed site is what it claims to be (and not a forged site made up to gather credit card numbers or banking passwords) and the integrity and confidentiality of any data transmitted over the connection is preserved.

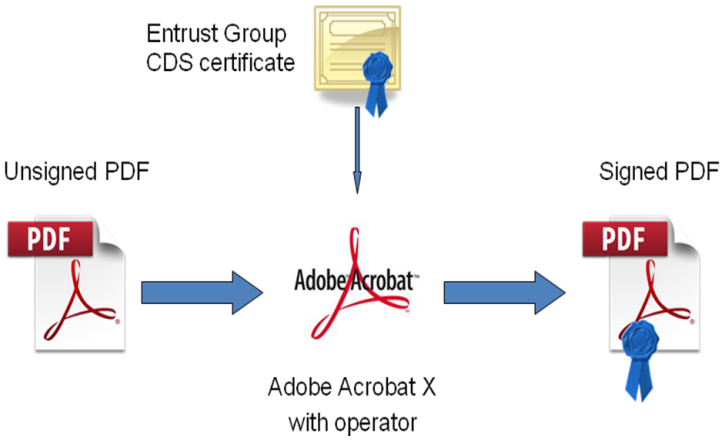
Aloaha Sign! Sign! is a free desktop application for Windows that can display various types of documents and verify the signatures within.

Java Libraries. The same libraries (JDK) and products that allow developers to sign documents (Cryptolog, Crytpto toolkit) also allow developers to develop solutions which will validate certificates.

Implementation Options

There are a large number of combinations of the software components, certificate levels, and vendors that a legislature could choose to use to create a document authentication solution. Below we describe six example implementations. Many more configurations are possible. Estimated costs are included for each sample implementation.

Example #1: Manual CDS Signing with Adobe Acrobat



Description: In low volume environments, a human process may be used to sign each document. In this example, the organization purchases an Entrust CDS group certificate with a USB token. A person in the legislature opens each document in Adobe Acrobat, uses the certificate to sign it, and puts the document back into the work process.

Components: Entrust Group CDS certificate, Adobe Acrobat X Standard

Initial cost: $1,049 (certificate, Acrobat) plus PC and labor

Ongoing cost: $618/year (certificate renewal)

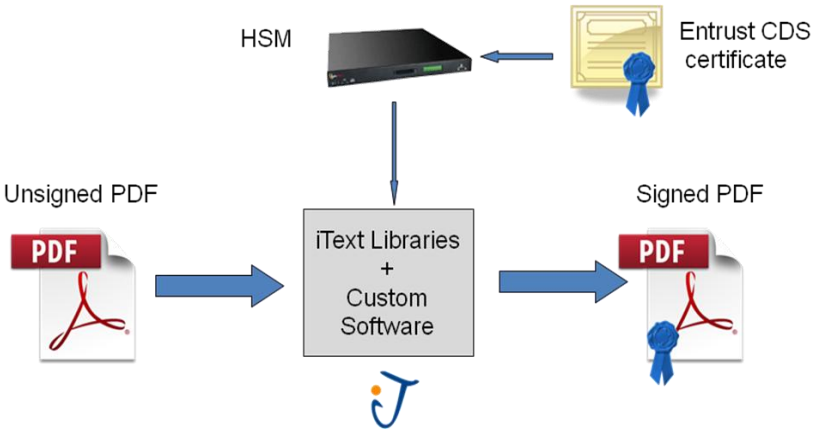
Advantages

- Low Initial Cost (for low volume)
- CDS certificate
- Simple process

Disadvantages

- Labor intensive – impractical for high volume
- Error prone

Example #2: Mass Signing with iText



Description: The organization purchases an Entrust Enterprise CDS certificate. Custom software using iText libraries is used to automatically sign PDF files, and inject them back into the work process.

Components: Entrust Enterprise CDS certificate, SafeNet Luna SA HSM, iText libraries, custom software

Initial cost: $22,100 (certificate, iText, HSM) and custom software

Ongoing cost: $9,670/year (certificate renewal, iText maintenance, HSM maintenance)

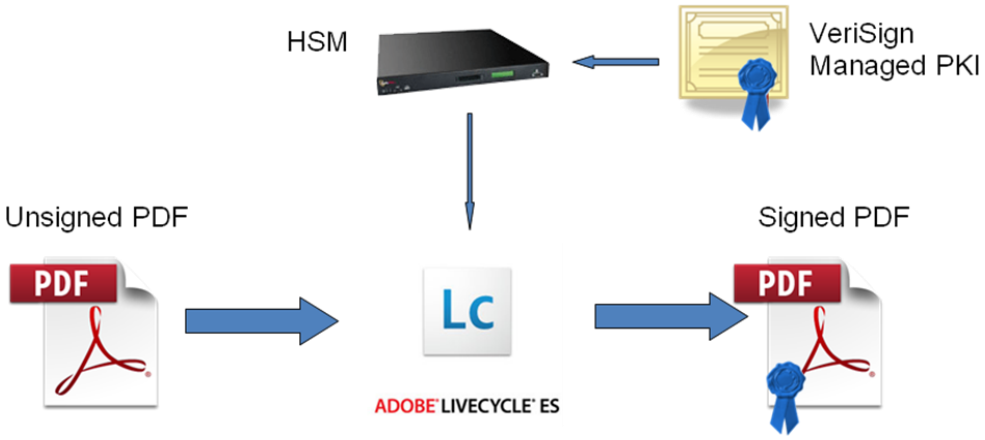
Advantages

- Automated
- High volume
- PKI/CDS certificate
- Moderate cost

Disadvantages

- Custom software needs to be developed

Example #3: Adobe LiveCycle Digital Signatures



Description: The organization purchases a VeriSign certificate and managed PKI service and an HSM. Adobe LiveCycle is used to automatically sign PDF files, and put them back into the work process.

Components: VeriSign Managed PKI Service, SafeNet Luna SA HSM, Adobe LiveCycle Digital Signature module

Initial cost: $178,100 plus system integration (includes one 2-CPU production server license and one development server license)

Ongoing cost: $44,620/year (MPKI, LiveCycle maintenance, HSM maintenance)

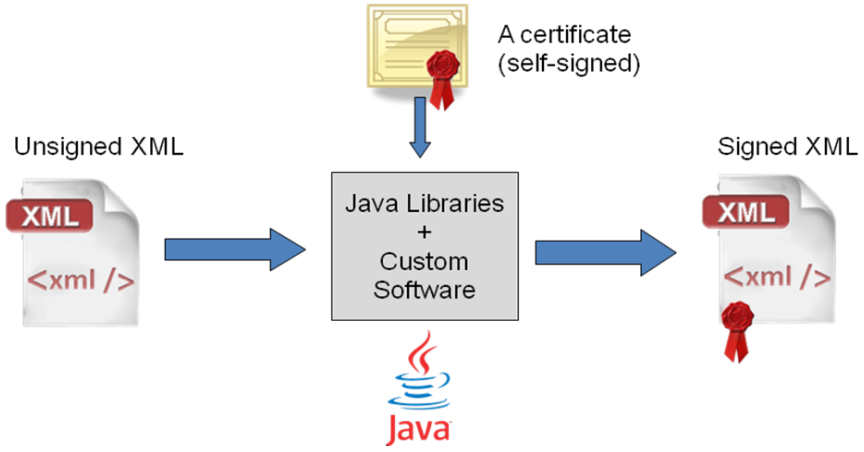
Advantages

- High volume
- PKI/CDS certificate
- Integrated in a larger document management framework: Workflow, Forms, PDF generation from MS Office documents

Disadvantages

- Expensive
- Cost increases for multi-CPU setups

Example #4: Automated XML Signing with Java Libraries



Description: XML files are passed to a signing module that adds a signature conforming to the XML DSig standard. The file is then introduced back into the process workflow and may, for instance, be forwarded to an external subscriber of such XML files (e.g., a legal publisher). This subscriber will use the signature element and the state's public key to validate the document. The state will have to inform its subscribers of the specifics of the signature element and provide the public key (most likely in a certificate).

Components: Java JDK 1.6, self-signed certificate and custom software

Initial cost: Custom software

Ongoing cost: Custom software maintenance

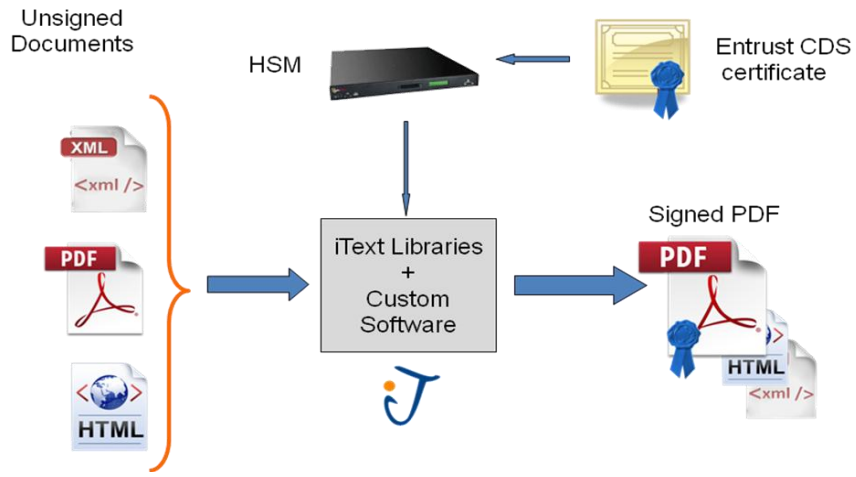
Advantages

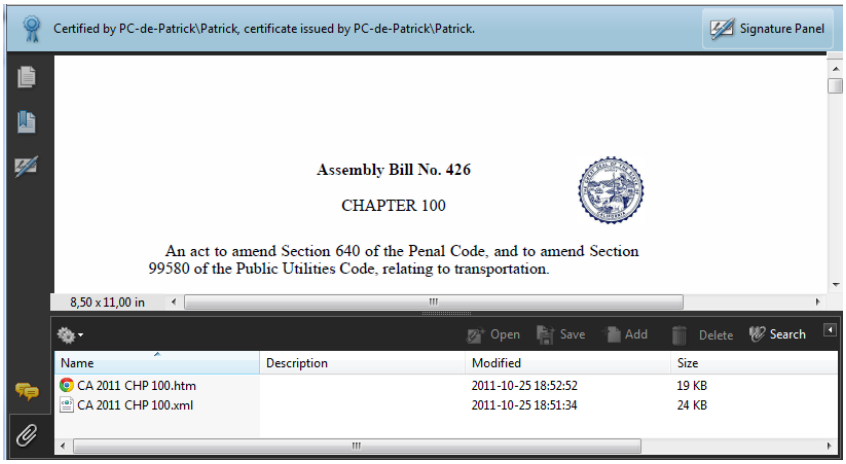
- Inexpensive
- Standardized authentication (XML DSig)

Disadvantages

- Validation needs additional software

Example #5: Automated Signing of PDF with HTML and XML Attachments





PDF with Attachments in Adobe Reader

Description: The organization purchases an Entrust PKI certificate. HTML and XML files may be certified by inclusion into a PDF document, which is then itself signed with its attachments. iText is used to attach the XML and HTML to the PDF, automatically sign the PDF, and place it back into the work process. As with other solutions, the user will see a ribbon on top of the screen indicating that the PDF file and its attachments are validated. The attachments may then be extracted.

Components: Entrust Group certificate, iText libraries, and custom software

Initial cost: $22,100 (Cert, iText, HSM) and custom software

Ongoing cost: $9,670/year (Cert renewal, iText maintenance, HSM maintenance)

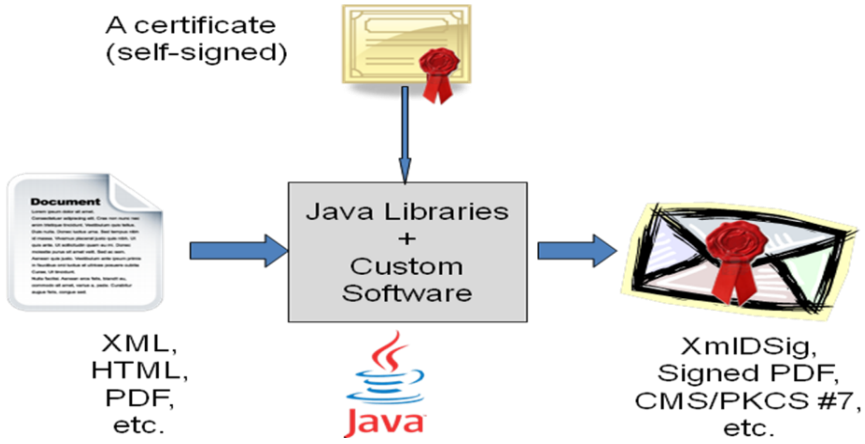
Advantages

- Automated, high volume
- PKI/CDS certificate
- Validation of PDF, XML and HTML in one step
- Moderate cost

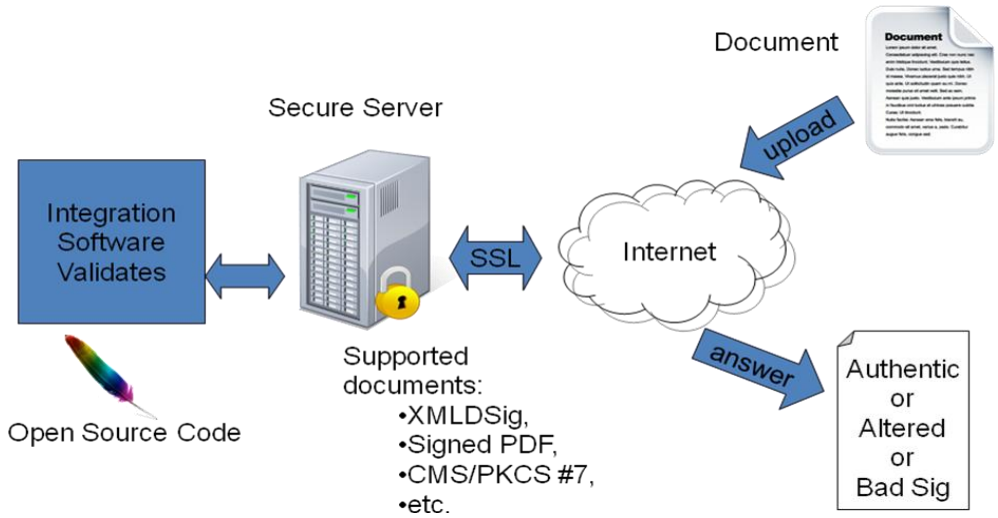
Disadvantages

- Custom software needs to be developed
- Requires Adobe Reader or Acrobat to extract data
- Automated processes need a PDF library to extract data
- Not compatible with PDF/A (archival PDF)

Example #6: Multi-Doc Signing with Validation



Signing Process



Validation Process

Description: The organization creates a self-signed certificate. Each file type is authenticated in its own manner (for example, signature for PDF, DSig for XML, PKCS #7 for HTML). Java libraries and customer software are used. Authentication validation is provided as a service on the legislative Web site.

Components: Self-signed certificate, Java libraries, custom software, web server

Initial cost: Custom software, web server

Ongoing cost: Custom software maintenance, web server maintenance

Advantages

- Numerous types of document may be verified
- Low cost

Disadvantages

- Requires online access to verify document's authenticity
- Validation service must be maintained

Sample Summary

Sample	Security	Volume	Doc Types	Initial Cost	Annual Cost
Manual PDF	High	Low	PDF	$1,049	$618
PDF with iText	High	High	PDF	$22,100	$9,670
PDF with LiveCycle	High	High	PDF	$178,100	$44,620
XML with Java	Medium	High	XML	Dev. Only (Moderate)	TBD
PDF with XML & HTML	High	High	XML, HTML, PDF	$22,100	$9,670
Multi-Doc Type	High	High	Any	Dev. Only (Higher)	TBD (Higher)

Note: These configurations are samples only. Many other configurations are possible.
Note: Costs do not include development, integration costs, or standard hardware platforms (servers, network, etc,)
Note: The security level is dependent on the certificate used, not the configuration.

APPENDIX A - Commercial Software and Services*

***Prices valid as of December, 2011**

Adobe Acrobat X

Description: Adobe Acrobat is a desktop PDF editing and creation tool. "X" indicates version 10. Acrobat X is available in Standard and Pro editions. Both editions can apply digital signatures to PDF files. Both editions can add file attachments to PDF files.

Pricing:	Acrobat X Standard	$299 list
	Acrobat Pro	$499 list

Web site: http://www.adobe.com/products/acrobat.html

Adobe Reader

Description: Adobe Reader is a PDF display tool. "X" indicates version 10. Adobe Reader X includes digital signature validation and extracting of attached files. Adobe Reader is the application of choice for validating signed PDFs. It is free, multi-platform, and can provide a seamless experience to the user. It is also already used by a large number of users as their default PDF Viewer.

When opening a PDF document, Reader looks for signatures and validates them if any are found. It then displays a ribbon at the top of the viewing window summarizing the status of signatures. A side panel can be invoked to obtain more detailed signature information.

Whether or not a signature is displayed as valid depends on the signer's certificate being chained to a trusted root. By default, Reader versions 6 and higher automatically trust the Adobe Root certificate. Any signer's certificate that chains to that root (called a CDS certificate) will be trusted and the signature displayed as valid. In addition, Reader versions 9 and higher trust a list of root certificates maintained by Adobe called the AATL. Any signer's certificate that chains to any of those roots will be trusted. Finally, a user can configure Reader to trust certificates found in the operating system's certificate store; signer's certificates that chain to one of those will also be trusted, but the user experience is not seamless anymore; only users that have changed the configuration of Reader to trust the OS certificate store will see a valid signature.

Pricing: Free

Web site: http://www.adobe.com/products/reader.html

Adobe LiveCycle Digital Signatures ES2

Description: LiveCycle ES2 is a product suite that enables developers to build visual interfaces to complex applications and to automate business processes through a workflow mechanism. LiveCycle ES2 is a J2EE-compatible product running on industry-standard operating systems and Java application servers. ES2 includes numerous modules that range from connectors to enterprise content management systems and data services, to PDF generation and digital signing.

Not all pieces of LiveCycle need to be implemented or deployed; the Digital Signatures component can be implemented separately. But it will need to be deployed on a Java application server like JBoss, Weblogic, or Websphere. It will also use a database such as MySQL or Microsoft SQL Server for configuration purposes.

Pricing: $60,000 per production CPU. $30,000 per development server (Digital Signatures module only)
20% annual maintenance

Web site: http://www.adobe.com/products/livecycle/digitalsignatures/

Aloaha PDF Signator

Description: Aloaha PDF Signator is a desktop application for Windows that can sign PDF documents, using either standard software certificates or smart cards. Signing is entirely manual.

Pricing: $72

Web site: http://www.aloaha.com/wi-software-en/aloaha-signator.php

Aloaha Sign!

Description: Aloaha sign! is a free desktop application for Windows that can display various types of documents and verify the signatures within. The user selects a document to open, the application opens it, searches for signatures and attempts to display the document within its own window, possibly using an external viewer (e.g. a PDF viewer). Signatures, if any were found, are displayed beside the viewing window.

Pricing: Free

Web site: http://www.aloaha.com/wi-software-en/aloaha-sign.php

Entrust Enterprise CDS Certificate

Description: Entrust Enterprise Lite and Pro are organizational CDS certificates. The certificates run in a server environment, supporting high volume batch processing. The certificates are rooted in Adobe, must be installed on an HSM (not included), and include online key management service. The issuance of these certificates is validated by Entrust.

Pricing:	Lite 50,000 signatures/yr	$7,000 first year, $6,650 annual renewal
	Lite 100,000 signatures/yr	$12,000 first year, $11,400 annual renewal
	Pro unlimited signatures	$25,000 first year, $23,750 annual renewal

Web site: http://www.entrust.net/ssl-cert-comparisons.htm#tabs-2

Entrust Group CDS Certificate

Description: Entrust Group Certificates are issued in the name of a group. The certificates run in a desktop environment, supporting interactive signing, but not batch processing. The certificates are rooted in Adobe, are delivered on a USB hardware token, and include online key management service. The issuance of these certificates is validated by Entrust.

Pricing: $650 first year, $618 annual renewal

Web site: http://www.entrust.net/adobe-cds-certificates.htm

GlobalSign Group CDS Certificate

Description: GlobalSign Group Certificates are issued in the name of a group. The certificates run in a desktop environment, supporting interactive signing. The certificates are rooted in Adobe, are delivered on a USB hardware token, and include online key management service. The issuance of these certificates is validated by GlobalSign.

Pricing: $595 per year, up to 2000 signature per year
$949 per year, up to 5000 signatures per year

Web site: http://www.globalsign.com/document-security-compliance/adobe-cds/departmentsign-usb-adobe-cds.html

iText

Description: iText is a Java library for creating and manipulating PDF files. iText is free for versions prior to 5.0. As of version 5.0, it is distributed under the Affero General Public License. Projects may need to or may choose to purchase a commercial license to avoid licensing issues. iText can be used to add digital PKI signatures to preexisting PDF files, and add attachments to a PDF file.

Pricing: **$**2,000 per production server
$1,000 per development server
TBD annual maintenance

Web site: http://itextpdf.com/

SafeNet Luna SA 4

Description: SafeNet Luna SA is a rack-mounted, network attached, FIPS 140-2 compliant Hardware Security Module (HSM) for storing digital certificates. Luna SA can be accessed from multiple servers.

Pricing: **$**13,100
20% annual maintenance

Web site: http://www.chrysalis-its.com/products/pki/lunaSA.asp

SafeNet Luna PCI

Description: SafeNet Luna PCI is a PCI Card, FIPS 140-2 compliant Hardware Security Module (HSM) for storing digital certificates. Luna PDI can be only be accessed from server on which it is installed.

Pricing: **$**5,000
20% annual maintenance

Web site: http://www.chrysalis-its.com/products/pki/lunaPCI.asp

VeriSign Organizational Certificates

Description: Organizational Certificate issued in the name of the organization.

Pricing: **$**10,000 initial cost
Maintenance cost not provided by the vendor.

Web site: http://www.verisign.com/

Verisign Managed PKI

Description: Service to manage PKI certificates.

Pricing: $5,000 for Managed PKI account setup, remote hosting
$6,000/year PKI managed service fee
$13,000/year for 500 PKI co-branded seats
$5,000/year for PKI support and maintenance

Web site: http://www.symantec.com/business/verisign/managed-pki-service/?tid=gnps

Appendix B - Related Topics

The following topics are related to document authentication, but were not studied for this paper.

Archiving formats (e.g. PDF/A).

To ensure that PDF can be read over a long period – whether they are digitally signed or not – they should be created as PDF/A rather than regular PDF files. PDF/A differs from PDF by omitting features ill-suited to long-term archiving, such as font linking (as opposed to font embedding) which could present problems should the linked fonts be moved or deleted.

Long Term Certification

Over long periods of time, digital signatures can become vulnerable to such issues as certificate expiry, revocation and weakening of the underlying cryptographic algorithms. Standards exist,[58] beyond basic digital signature standards, which aim to protect digital signatures against these threats and allow long-term, archival authentication. We did not investigate this area further in this document, beyond noting that the longevity of digital signatures can be challenged and that there are proposed solutions.

Controlling Document Usage

The first signer of a PDF document can control what further operations are allowed on the document, such as copying or field fill-in.

Time Stamping Issues

When a document is signed, a time stamp may be applied. The time for the stamp could come from clock of the signing machine or from a network time stamping service. The differences were not investigated.

Key Management

We do not address the ongoing management issues with managing signing keys, such as renewing, replacing, or revoking them.

[58] Notable are CAdES, XAdES and PAdES (targeted respectively at CMS, XML and PDF) from ETSI, the European Telecommunications Standards Institute <http://www.etsi.org> (last visited Dec. 8, 2011); CAdES is technically equivalent to RFC 5126, the other two are transpositions of the same ideas to XML and PDF. All rely on cryptographic time stamps (RFC 3161).

APPENDIX C – Lexicon

AATL	Adobe Approved Trust List. Both Acrobat and Reader have been programmed to reach out to a web page to periodically download a list of trusted "root" digital certificates. Any digital signature created with a credential that can trace a relationship ("chain") back to the high-assurance, trustworthy certificates on this list is trusted by Acrobat and Reader 9 and later.
Authentication	A method for a user to determine that the record received by the user from the publisher is unaltered from the official record.
CA	See Certification authority.
CDS	A broad implementation of document validation technology based on public key infrastructures (PKIs) in Adobe products. Adobe Reader and Acrobat have built-in trust in CDS signatures, eliminating any additional software or configuration to validate or certify the PDF files certified with such signatures.
Certification authority	In a public key infrastructure (PKI), an entity that issues and verifies digital certificates that contain an encryption key and attest to the authenticity of the transaction party. See also authentication, digital certificate, and PKI.
Certification revocation list	A list of certificates that have been revoked and, therefore, should not be relied upon.
Chain of trust.	The Chain of Trust of a Certificate Chain is an ordered list of certificates, containing an end-user subscriber certificate and intermediate certificates (that represents the intermediate CA), that enables the receiver to verify that the sender and all intermediates certificates are trustworthy.
CRL	See Certification revocation list.
Digest	See Hash Function.
Digital Certificate	An encrypted and digitally signed attachment that authenticates a user on the Internet or an intranet. A digital certificate is issued by a certificate authority (CA), and attests to the legitimacy of an online transfer of information, funds, or other sensitive materials through the use of encryption. A digital certificate includes the sender's name, a serial number, expiration dates, a copy of the certificate holder's public key, and the digital signature of the issuing CA. A digital certificate holder has both a private key and a public key. The private key is held only by the user and is for signing outgoing messages and decrypting incoming messages. The public key is available to anyone for encrypting data to send to the holder of that public key, who then uses the private key to decrypt the message. Many digital certificates conform to the X.509 standard. Digital certificates are verified using a chain of trust. The trust anchor for the digital certificate

	is the Root Certificate Authority (CA).
Hash Function	A cryptographic hash function is a deterministic procedure that takes an arbitrary amount of data and returns a fixed-size binary string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The hash value is also called the message digest or simply the digest. Common hash functions are MD4, MD5, and the SHA series.
MD4, MD5	See Hash Functions.
PAdES	PDF Advanced Electronic Signatures is a set of restrictions and extensions to PDF and ISO 32000-1 making it suitable for advanced electronic signature. One important benefit from PAdES is that electronically signed documents can remain valid for long periods, even if underlying cryptographic algorithms are broken.
PKI	A formal structure that enables the user of an inherently insecure public network, such as the Internet, to transfer electronically information, funds, and other sensitive materials through the use of encryption key pairs obtained from and shared through a trusted entity. A certificate authority (CA) issues and verifies digital certificates that contain an encryption key and attest to the authenticity of the transaction party. A registration authority (RA) verifies the CA prior to the issuance of a digital certificate to the requesting party.
PKS	Public Key System - See PKI.
Private key	The unpublished key in a public key cryptographic system, which uses a two-part key: one private and one public. The private key is kept secret and never transmitted over a network. Contrast with "public key," which can be published on a Web site or sent in an ordinary e-mail message.
Public key cryptography	An encryption method that uses a two-part key: a public key and a private key. To send an encrypted message to someone, a person uses the recipient's public key, which can be sent to a person via regular e-mail or made available on any public Web site or venue. To decrypt the message, the recipient uses the private key, which he or she keeps secret. Contrast with "secret key cryptography," which uses the same key to encrypt and decrypt.
SHA	See Hash Function.
X.509	An ITU-T Recommendation (1988) for a public key infrastructure (PKI). X.509 establishes a hierarchical structure of certificate authorities (CAs) that issue digital certificates, which are electronic credentials that authenticate the identity of users on the Internet and intranets.