



PDHonline Course G463 (6 PDH)

Introduction to Digital Signatures - Part Two

Instructor: Daryl S. Banks, PSM

2020

PDH Online | PDH Center

5272 Meadow Estates Drive
Fairfax, VA 22030-6658
Phone: 703-988-0088
www.PDHonline.com

An Approved Continuing Education Provider

TABLE OF CONTENTS

INTRODUCTION	6
Abstract	6
Assumptions.....	7
Prerequisite Documents	7
Audience and Document Conventions.....	8
About the Author	9
METHODS FOR SECURITY	9
Password Protect.....	10
Message Encryption.....	10
Digital Signatures.....	12
<i>Applications</i>	14
One-Way Hash.....	17
CREATING AND USING SELF SIGNED CERTIFICATES.....	19
Review of Certificates.....	19
Distinguished Names	20
What are Self-Signed Certificates.....	20
When to use Self-Signed Certificates	21
How to Create and use Self-Signed Certificates.....	22
CREATING A CUSTOMIZED DIGITIZED SIGNATURE FOR PDF.....	25
Setting up Your Digitized Signature.....	25
Add a Timestamp to Signatures	25
Configure a Timestamp Server	26
Set a Timestamp Server as the Default.....	26
Implementing the Signature.....	28
Setting up Digital Signature Validation.....	29
USING SELF-SIGNED CERTIFICATES FOR DIGITAL SIGNATURES	31
Self-Signed Certificates for Certified, Non-Encrypted Digital Signatures.....	32
Self-Signed Certificates for Certified, Encrypted Digital Signatures.....	33

Custom Certificate Definite Encoding Rules..... 35

Create a Primary and Secondary Self-Signed Certificate 35

Implementing the Digital Signature..... 42

Distributing the Certified Document 44

USING SELF-SIGNED CERTIFICATES FOR MESSAGE ENCRYPTION 45

 Create a Primary and Secondary Self-Signed Certificate 45

 Implement the Message Encryption..... 47

 Distribute the Encrypted Document..... 48

ENCRYPTING AND SIGNING WITH A THIRD PARTY DIGITAL ID 48

UNDERSTANDING HOW PDF CONVERTS TO CAD FORMATS..... 49

 Advances in Technology..... 49

 Raster to Vector 49

 Conversion Techniques..... 51

 PostScript 51

 Bézier Curves 51

 Vector File Formats 52

 Resolution Types 53

 Image Resolution 53

 Monitor Resolution 55

 Printer Resolution 55

HOW TO PREVENT PDF FILES FROM BEING REVERSE ENGINEERED 57

 Understanding PDF Print Resolution..... 57

 Understanding Text Fonts..... 58

 Microprinting 59

 QR Codes 60

 Flatten and Clean the PDF 61

 Find and Remove Hidden Content..... 62

 Remove Hidden Information Options..... 62

 Redact (Black Out and Remove) Sensitive Content 64

 Discard Objects Panel 64

Discard All Form Submission, Import and Reset Actions..... 64

Flatten Form Fields 64

Discard All JavaScript Actions 64

Discard All Alternate Images..... 64

Discard Embedded Page Thumbnails 64

Discard Document Tags..... 65

Convert Smooth Lines to Curves 65

Detect and Merge Image Fragments 65

Discard Embedded Print Settings 65

Discard Embedded Search Index 65

Discard Bookmarks..... 65

Discard User Data Panel 65

Discard All Comments, Forms and Multimedia 65

Discard Document Information and Metadata..... 65

Discard All Object Data..... 66

Discard File Attachments..... 66

Discard External Cross References..... 66

Discard Private Data of Other Applications 66

Discard Hidden Layer Content and Flatten Visible Layers 66

Clean Up panel..... 66

Object Compression Options 66

Use Flate to Encode Streams That Are Not Encoded 66

In Streams That Use LZW Encoding, Use Flate Instead 66

Discard Invalid Bookmarks 67

Discard Invalid Links..... 67

Discard Unreferenced Named Destinations..... 67

Optimize Page Content 67

Optimize the PDF for Fast Web View 67

Encrypt the Document 67

STRATEGIES FOR DOCUMENT MANAGEMENT 69

What Is a Roaming Policy?.....	69
Internal Document Management.....	70
External document management.....	72
Summary.....	75
Appendix A.....	76
Adobe Portable File	76
Certificate Security	76
Encrypt a PDF or PDF Portfolio with a Certificate	76
Change Encryption Settings.....	78
Remove Encryption Settings.....	78
Sharing Certificates with Others	79
Get Certificates from Other Users	79
Request a Certificate from Another User.....	79
Verify Information on a Certificate	81
Verify Your Own Certificate	82
Verify information on the Certificate of a Contact.....	82
Delete a Certificate from Trusted Identities.....	82
Create a Self-Signed Digital ID	83
Register a Digital ID	84
Specify the Default Digital ID	85
Change the Password and Timeout for a Digital ID.....	85
Delete your Digital ID	86
Protecting Digital IDs	86
How to Protect Your Digital IDs	86
What to do if a Digital ID is Lost or Stolen	87
Smart Cards and Hardware Tokens	87
Appendix B.....	89
Glossary	89
Data Encipherment.....	89
Decipher Only.....	89

Extensions 89

Encipher Only 89

Key Usage 89

Key Encipherment 89

Key Agreement 90

Layered security 90

Roundtrip Protection 90

Appendix C 91

 Florida Digital Signature Standards for Surveying and Mapping 91

Appendix D 93

Works Cited 93

INTRODUCTION

Abstract

Part One of this series of courses focused on CAD files. In Part Two, the intention is directed toward the most widely used online reader in the market, the Adobe PDF. With a “Digital Signature,” a Digital ID is attached to an Adobe PDF file so users can work together more easily with others on projects. Recipients of PDF drawings are provided with a trustworthy certified PDF about the individual who created a document, and whether the transmitted PDF was modified since it was digitally signed.

Digital Signatures provide the following benefits:

- Recipients of digitally signed drawings can be sure that the organizations or individuals who sent the drawings are trustworthy.
- A Digital Signature guarantees that a drawing’s content and properties have not changed since the drawing was digitally signed.
- With a third party Digital ID, a signed file cannot be rejected as invalid. The signer of a file cannot reject the file later by claiming the signature was copied.

A Digital Signature is not a digitized signature (an electronic scan of a signature). While a Digital Signature helps prove your identity and a drawing's authenticity, a digitized signature is nothing more than an electronic version of your own signature inserted or attached to a CAD drawing. It can be forged and copied, and has no tangible security value. Digital Signature is a widely used term that uses a Public Key Infrastructure (PKI) certificate known as a “Digital ID.” A PKI is inseparable from digital certificates. A PKI is responsible for issuing certificates, ensuring the distribution of these certificates through a directory, and validating certificates.

By the completion of this course, you will be able to electronically sign any Adobe PDF file with your own Digital ID and establish a drawing delivery policy with PDFs at your company to prevent reverse engineering of their documents from PDF to CAD vector format. As Part One of this series concentrated on CAD drawings, the second part focuses on the PDF delivery system since most drawings are distributed by PDF to clients at various points in the project lifecycle. The goal is to understand and meet government recommended guidelines for distributing electronic documents provided by institutions like the National Institutes of Standards and Testing (NIST) and the National Security Agency (NSA) along with the Florida Minimum Technical Standards (5J-17) as it applies to Digital Signatures shown in the Appendix C. Along with the support site aecsignature.com, there are additional videos showing the step-by-step procedures for many of the topics discussed.

Assumptions

Applying Digital Signatures with Digital IDs with PDFs are the focus of the seminar, but Digital IDs are laden with a few college semesters of topics like Encryption, Cryptography, and basic Computer Science. One goal of this course is to streamline these ideas and present material on a how to use Digital IDs, not how to design and build your own Digital ID, while condensing the material into a six-hour continuing education course.

The topics are presented with typical uses of Digital Signatures being discussed. Various constituent parts will be presented along the way to aid in understanding the next topic. Unlike most courses, many of you may have never used a Digital Signature to sign CAD or PDF drawings. If you have never used a Digital Signature or certificates to sign PDF drawings, spend 15 minutes viewing the videos on the support blog at aesignature.com (Banks & Banks Consulting, 2013). Click the tutorial link to understand the concept by seeing it in action. The videos will give you a basic layout of the signature process and demonstrate the straightforward nature of signing PDF files with a Digital ID.

Prerequisite Documents

Here is a link to the video on what product to purchase at VeriSign (Symantec Now) to one of the providers <http://youtu.be/NnEh0CXmobw>. You can find the link on the tutorial section at aesignature.com. There you will also find supplemental videos on these and more topics.

This video shows readers how to download and install the free 30-day certificate or Digital ID from VeriSign. Readers may use this Digital ID with their own CAD and PDF software. If they purchase a Digital ID, the cost is \$20-30 USD and expires one year from the date of purchase.

If you want to sign PDF documents with your Digital ID, you will need the *Adobe Acrobat Professional* version. You can download a 30-day trial located at Adobe.com and walk through the examples in Appendix A.

If you want to reverse engineer a PDF, you will need you can use the following programs:

- **Adobe Illustrator**. You can download a 30-day trial located at Adobe.com.
- **AutoDWG** You can download this at: <http://www.autodwg.com/pdf-to-dwg-converter/>
- **Able2Extract** You can download a 7 day full trial here:
http://www.investintech.com/prod_downloads2e.htm

Keytool Software Development Kit is part of the standard java distribution that can be downloaded from the following link on Oracle:

<http://www.oracle.com/technetwork/java/javaee/downloads/java-ee-sdk-6u3-jdk-7u1-downloads-523391.html>

OpenSSL can be downloaded from the following link:

<http://slproweb.com/products/Win32OpenSSL.html>

You might need the following files:

1. [Win32 OpenSSL v0.9.8y](#)
2. [Visual C++ 2008 Redistributables](#) or Windows 64-Bit [Visual C++ 2008 Redistributables \(x64\)](#)

KeyTool Explorer for Windows 4.1 can be downloaded at the following link:

<http://www.lazgosoftware.com/kse/>

Audience and Document Conventions

During this seminar, ideas from the AEC industry's viewpoint are discussed as much as possible. From time to time this might not be feasible. In that event, these departures will be noted. However, for most of the course, the term "files," for example, will be referred to drawings or Computer Aided Design (CAD) files. Moreover, no distinction are made between CAD and CADD being Computer Aided Drafting and Design.

Although a special emphasis was given to converting CAD files to PDF documents in the first course, in practice, these Digital Signature methods can be any files but not limited to the following list:

- ESRI Shape Files, shp
- Text Files ,txt
- ASCII Files, asc
- Word, docx
- Excel, xls
- Photos, jpg
- Images, gif
- Adobe File, pdf
- AutoCAD, dwg

Furthermore, if the file can be serialized into a stream of bytes (a computer data construct), then a hash value can be calculated. Knowing the limitations and practical uses of these algorithms will help streamline the drawing distribution process.

The material in this seminar is biased toward the Adobe® product line, but knowingly understand the parity between competing software companies. The standards found in one software package are shared with other software packages by industry standards set by the National Institutes for Standards and Testing (NIST) and the National Security Agency (NSA)

and the well-known Internet Engineering Task Force Agency (IETA). They set and promote standards for compliance in design and software relating to applications and a part of these standards are how public and private keys, certificates, and Digital IDs are handled over the Internet. With that stated, be assured the topics discussed herein are mature and have been used for at least a decade in one form or another. Autodesk® started using certificates after the 2000, 2000i versions of AutoCAD, and product design has not changed with certificate use since. Adobe Acrobat Reader® has been implementing security since version 1.3 released in 2000.


About the Author

Daryl Banks has been a Professional Surveyor and Mapper for 12 years. He is a Software Developer with eight years of AutoCAD.Net experience. He is currently an active licensed Autodesk® software developer and Autodesk Developer Network (ADN) member. He has over 20 years of geospatial experience with Land Development, and over 10 years of experience in software development. He holds a degree in Nuclear and Radiological Engineering from the University of Florida and a degree in Computer and Information Systems from the University of North Florida.

Over the past three years, Mr. Banks has developed proprietary applications using Autodesk's Software Development Kits (SDK) RealDWG. Currently, he has two software products developed using the Autodesk API that are both listed in the *Autodesk App Store* and commonly referred to as the *Exchange Store*.

Mr. Banks has worked with the ESRI GIS product line including ArcMap and ArcGIS Enterprise Server (past licensed developer), and has managed large-scale development projects using ESRI product and Autodesk Civil 3D. He has used Autodesk Map 3D and Mapguide (Web Platform) to implement Aerial mapping support on land development projects. He has helped support and integrate Map 3D and GIS databases for a large utility company in the Northwest. He has worked for Fortune 100 Engineering Companies with over 29,000 AEC Professionals, with national and international project management experience. He currently operates a geospatial consulting company specializing in custom geospatial application development and land surveying and mapping services.

METHODS FOR SECURITY

 *If you purchased the first seminar of this series, you may decide to skip parts of this section since it is provided as a review.*

In this first section, there is a brief review of the four security methods discussed in Part One of this series. Although the focus is on digital signatures and message encryption, it is a good review for you to understand the differences between all four methods of security listed below:

- Password Protect
- Message Encryption
- Digital Signatures
- One-Way Hash Functions

Password Protect

In order to protect PDF files, one function of PDF software systems is password protection. In many systems, password protection and file properties encryption are used together. In this course, the focus is on digital signatures with Digital IDs. However, a comprehensive picture of different types of security would not be complete without mentioning password protection. Password protection puts a lock on the file. It protects the file in transit to the end user. The end user must know the password to open the file. However, if an attacker breaks the password, they can change the drawing, save the drawing with the original password and send it back to you, as if nothing changed.

It is important to understand that Password Protect does not give a full roundtrip protection and detection of changes to a PDF file. As illustrated, a hacker can break the password and impersonate the user by changing the drawing without the sender knowing. Only a Digital Signature by a Digital ID protects the file for the complete roundtrip, but it does not secure the drawing like password protection with file encryption. As a result, Password Protect functionality and Digital Signature perform two distinct functions. Password Protect locks the file. A Digital Signature verifies who sent it, and Message Encryption hides the contents. In various applications, these approaches are used together. One part verifies the sender and the other method protects the file.

One method to help protect a password-protected document is to encrypt it inside a folder with stronger encryption mechanisms using software like “Bitser.” The software has a stronger level of encryption that can be used in coordination with a copyright or confidentiality agreement.



Video tutorial is available on the support site for this topic.

Message Encryption

Digital Signatures with Message Encryption used with certificates completes the security in a few PDF systems Digital Signature implementations, as the Digital IDs authenticate and the Message Encryption hides the contents. With Digital Signatures, password protection may be an option, but it is a separate operation out of context for this topic. By using the public and private keys from your Digital ID, the drawing may be signed with a Digital ID and content hidden by a function called “encryption.” The examples of this type of functionality are encrypting the drawing file properties, which do not hide the file content – only the metadata file properties.

With the purchase of a Digital ID, most email packages allow Message Encryption and Digital Signatures by importing the Digital ID. However, the Message Encryption requires the distribution of the public key to the recipient for a secure connection to be established as shown below in Figure 1. In order to encrypt files, you must trust the recipient to be an honest person. The recipient sends you their public key for the you to encrypt the file. Upon the encrypted file's return to the recipient, the recipient's own installed private key of the matching pair will decrypt the message. You will notice the Signing and Verifying with a Digital ID that requires the distribution of the public key, and your private key used to encrypt or sign the data. In comparison, a Digital ID with Message Encryption requires the public key of the recipient installed on your computer to encrypt the data. The processes are similar in that comparison, but they differ when compared to which key does the signing or encrypting (See Figures 2 and 3).

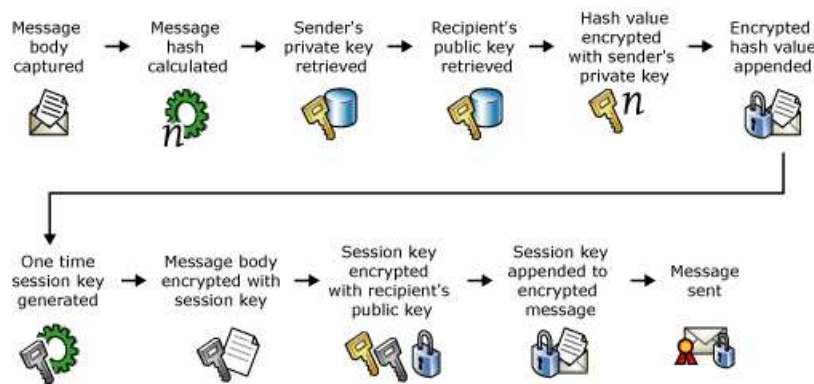




Figure 1 shows message encryption implemented by a Digital ID.

(How Digital Certificates are used for Digital Signatures and Message Encryption, 2005)

A good written policy can be established with certain clients, designs, or internal personnel. If the PDF software does not support Message Encryption directly, then using the Digital ID installed into the email client will encrypt the contents of a correspondence. Make sure the email client encrypts the attachments along with the message. If not, then compress and encrypt the PDF files into a zipped file that can be encrypted with a Password Protect option to decrypt the contents on the recipient's end.

 *Video tutorial is available on the support site for this topic.*

Digital Signatures

 *Digital Signatures require a Digital ID to sign documents. A Digital ID MUST have a public and private key pair. A user can export out the public key from the Digital ID to generate a certificate for distribution.*

Digital Signatures are frequently used to implement electronic signatures, a term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use Digital Signatures. The focus is the specific case with PKI Certificates.

Digital Signatures are a type of asymmetric cryptography (using a public and private key exchange). For files sent through a less secure method such as email, a properly implemented Digital Signature gives the receiver reason to believe the file was sent by the originating sender. Digital Signatures are comparable to traditional handwritten signatures in many respects, but properly implemented Digital Signatures are more difficult to forge than the handwritten type. Digital Signature schemes in the sense used here are cryptographically based and must be designed and implemented properly to be effective. Digital Signatures can also provide non-rejection, meaning you cannot successfully claim you did not sign a file while also claiming your private key remains secret. Furthermore, a few non-repudiation schemes offer a time stamp for the Digital Signature, so that even if the private key is exposed, the signature is valid. Digitally signed files may be anything representable as characters in a sentence. Examples include electronic mail, Word, Excel, and PDF Documents.

Digital certificates provide support to public key cryptography by providing a reliable means to distribute and access public keys. When a sender is signing a PDF file, the sender provides the public key that is associated with the private key available on the digital certificate. In turn, when the recipient is validating a Digital Signature on a file, the recipient is obtaining the public key to perform that operation from your digital certificate. Figure 2 shows the following sequence of signing with the addition of the supporting elements of digital certificates.

1. PDF file, the Message is captured.
2. Hash value of the message is calculated.
3. Sender's private key is retrieved from the sender's digital certificate.
4. Hash value is encrypted with the sender's private key.
5. Encrypted hash value is appended to the message as a Digital Signature.
6. Message is sent.

To verify the signature, the following steps are required as shown in Figure 3.

1. Signed CAD file, the Message is received.
2. Digital Signature containing encrypted hash value is retrieved from the message.

3. Message is retrieved.
4. Hash value of the message is calculated.
5. Sender's public key is retrieved from the sender's digital certificate.
6. Encrypted hash value is decrypted with the sender's public key.
7. Decrypted hash value is compared against the hash value produced on receipt.
8. If the values match, the message is valid.

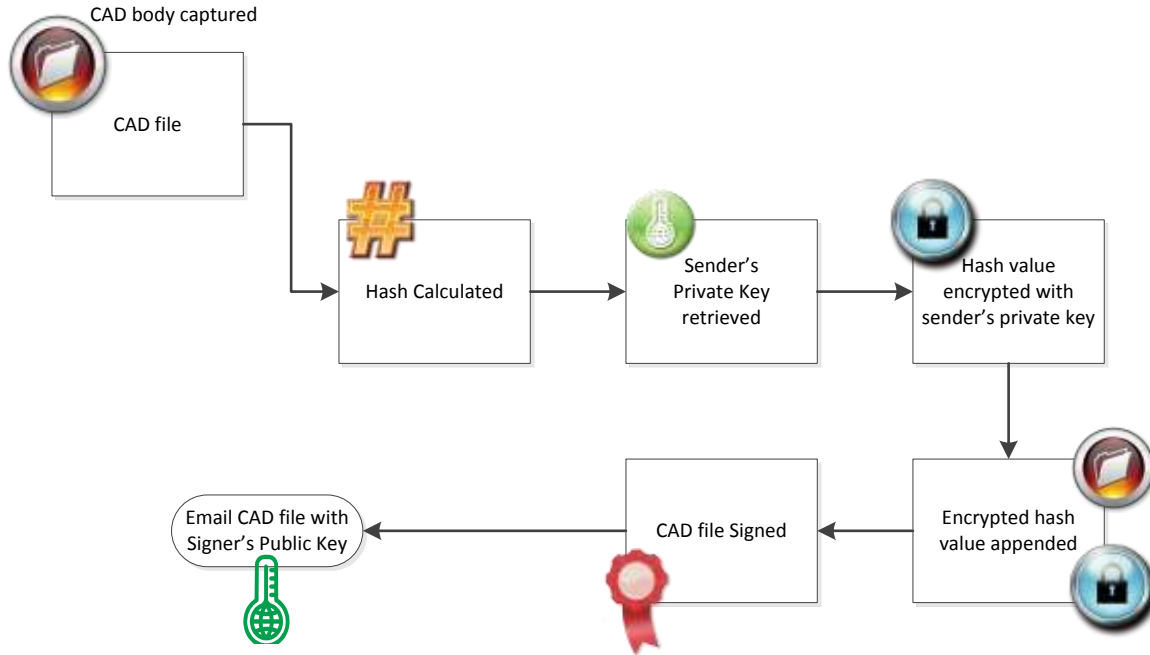


Figure 2 shows the PDF file being digitally signed by you.

The following figure shows the sequence of verifying a PDF file with digital certificates.

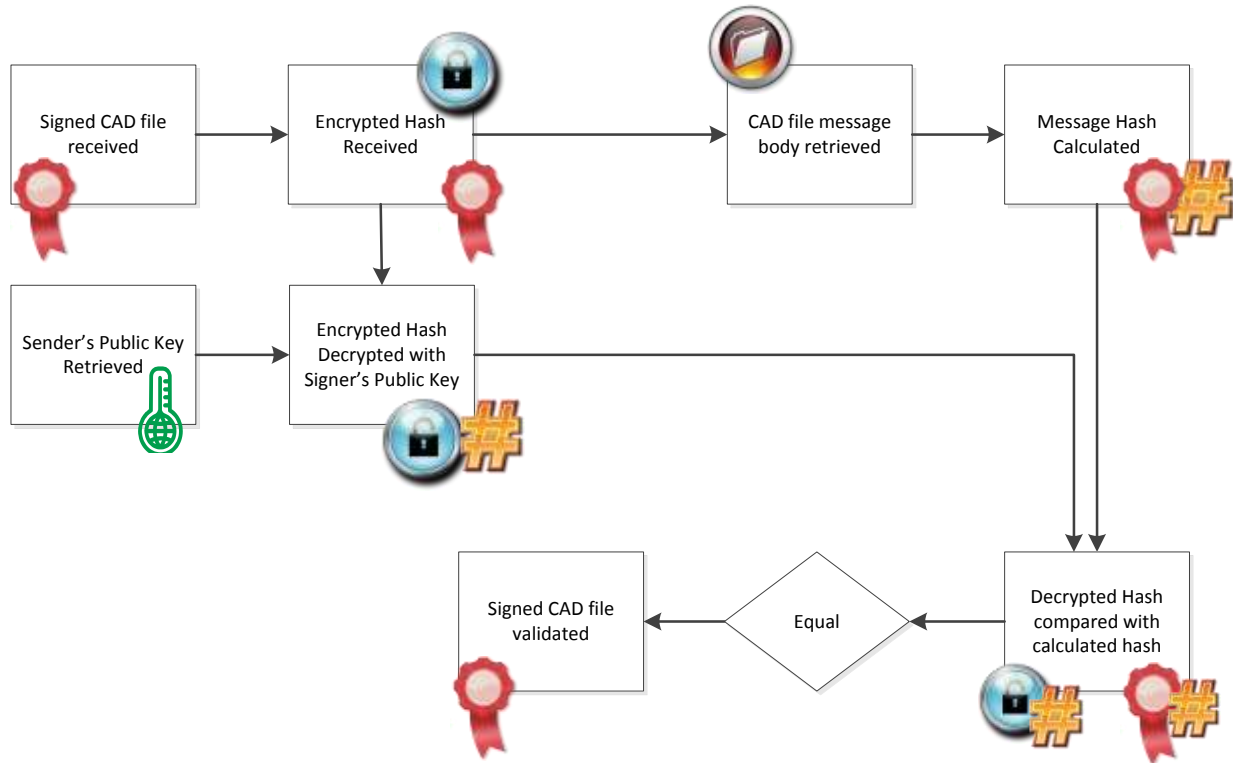


Figure 3 shows the verification process of a signed PDF file.

Applications

As AEC organizations move away from paper documents and blueprints with ink signatures or authenticity stamps, Digital Signatures can provide added evidence to the origin, identity, and content of electronic PDF files as well as acknowledging informed consent and approval by the undersigned. The United States Patent and Trademark Office (USPTO) and the United States Government Printing Office (GPO) publish electronic versions of the budget, public and private laws, and Congressional bills with Digital Signatures. Universities including Penn State, the University of Chicago, and Stanford are publishing electronic student transcripts with Digital Signatures. It is time for the AEC industries to start adopting the practice of signing all our PDF files and PDFs that are sent electronically to our clients with Digital IDs.

Authentication

Although PDF files may include information about the entity sending the file, that information may not be accurate. For example, receiving a PDF file from another employee is more common than receiving the file from the undersigned. Here are a few basic questions to ask the next time a PDF file is received:

- Who is the responsible party?
- Who either possesses the signed hard copy or digitally signed files?
- Did the sending party edit the drawing, and send it with full consent of the professional who signed the hard copies? If yes, where is the consent?
- Was the sender part of the project relating to the PDF files?
- Did the sender send the correct file version?

In an effort to not interject opinions here, let us suffice to say, everything is okay until there is a problem. The sender could eloquently answer the above questions, but if there is no assignment of responsibility with an approved method of authentication, the recipient will be in a vulnerable position.

Digital Signatures can be used to authenticate the source of the file. When ownership of a Digital Signature secret key is bound to you, a valid signature proves that you sent the message. The importance of high confidence in sender authenticity is especially obvious in critical design deliveries. For example, suppose an engineering branch office sends instructions to the central office requesting a change in the design. If the central office is not convinced that such a message is truly sent from an authorized source, acting on such a request could be a serious mistake.

Integrity

In many circumstances, you and the receiver of a PDF file may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a file, it may be possible to modify an encrypted message without understanding it. However, if a file is digitally signed, any change in the PDF file after signature will invalidate the signature. A Digital ID is not Message Encryption. A Digital ID only validates you – it does not Password Protect the drawing. Furthermore, there is no efficient way to modify a file and its signature to produce a new file with a valid signature, because this is still considered computationally infeasible by most cryptographic hash functions, as notated in a term called “collision resistance,” which is out of the scope of this seminar. However, it is described in the Glossary.

When a client downloads a file over the Internet, they need to be sure the file they receive is the one they wanted. The client needs to be assured of the file’s reliability and integrity. Many people make the following assumptions when they download a file over the Internet:

- The file is not a malicious program.
- The file has not been replaced, unbeknown to the server’s owners, by a malicious program.

- There is not another computer between the sender and the server, sending the user a different file than the one they want or modifying the file the originator sent. This is the “man-in-the-middle attack.”

These concepts would not have been an issue in the past, since electronic data was kept to a minimum. Now, most document content is electronic and stored on a server. As the movement to switch from paper deliveries to electronic deliveries continues to increase, professionals need to make the transition and keep in conformity with the professional standards as it pertains to Digital IDs and signed and sealed documents.

An important concept to discuss is the full roundtrip verification of electronic transmission of PDF files. You trust if they have the signed and sealed prints in your cabinet, then those prints are the originals. You are not concerned with what changes were made to the prints that were delivered to the client after that point in time. In the past, that was the situation for everyone. However, if you send those same PDF files by email, FTP, or they are downloaded from a server, the files sent electronically are the new “originals” because the nature of the electronic medium is different from the hard copy of signed and sealed prints. Moreover, if there were any changes to that drawing at any point along the electronic pathway, you have no tracking mechanism without using Digital IDs or manual hashes. Without full verification by a Digital ID, you are in a vulnerable position.

A concept has been used over the past decade to erase all traces of the company logo and signing block before sending the client a drawing. While this may give the illusion of anonymity, your digital fingerprints are everywhere. From the PDF file containing pertinent creation information to email correspondence, it would be very difficult to hide any trace of the file origin.

The term “Digital Fingerprint” may not have meaning to the average user, but electronic transactions can be traced. Likewise, the concept of electronically tracing files is certain to increase in the years to come. Undoubtedly, if Google keeps all the searches from its search engine from a decade ago and the US Government hoards all the emails that are ever sent, any anonymity was lost years ago. Digital IDs establish a company’s reputation for legitimacy and being technologically savvy by using ideas presented in this seminar. They will help mitigate these types of situations, while giving the final recipient (the client) a measure of file integrity from you.

Non-Abandonment

Non-Abandonment, or more specifically, non-rejection of the drawing origin, is an important aspect of Digital Signatures. By definition, an entity or licensed professional who has signed a CAD drawing cannot later deny having signed it. Similarly, access to your public key does not

enable a fraudulent party to fake a valid signature. This topic will be explained in more detail later.

One-Way Hash

A Cryptographic Hash Function is a hash function that is described as an algorithm that takes an arbitrary block of data and returns a fixed-size bit string.

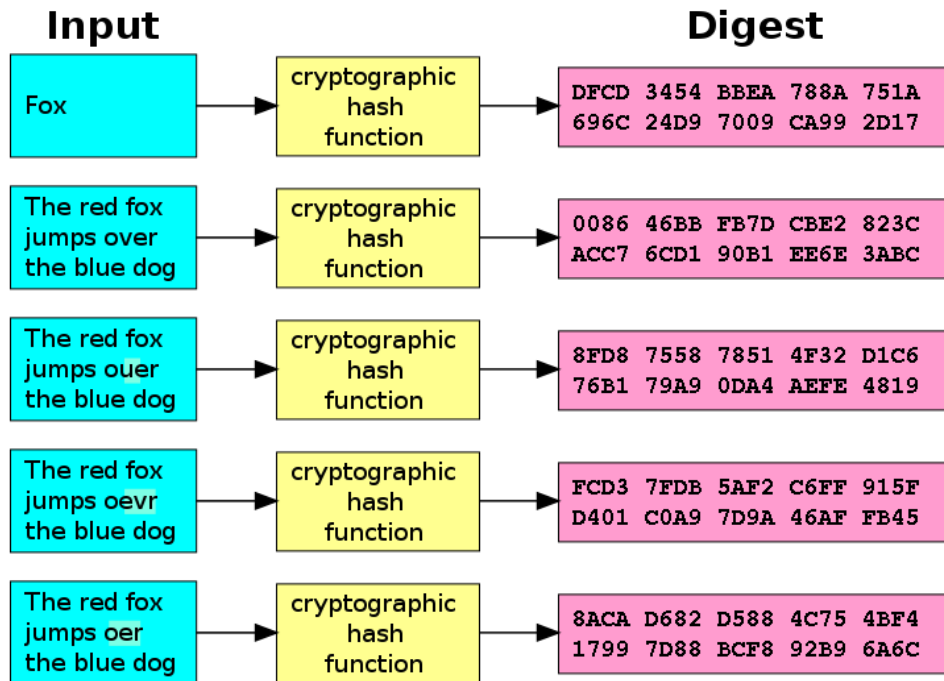


Figure 4 shows a typical one-way hash function converting text into ciphertext or digest.

A Cryptographic Hash Function such as the Secure Hash Algorithm (SHA-1) produces a hash code 160-bits in length. These hashes are unique output values in that even small changes in the source input (a change in the CAD file) drastically changes the resulting output hash, by the avalanche effect.

A Cryptographic Hash Function is a hash function that is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that any (accidental or intentional) change to the drawing will change the hash value. The data to be encoded are often called the "message," and the hash value is sometimes called the message digest or simply "digests" (How Certificates Work, 2003).

The ideal Cryptographic Hash Function has four main properties:

- It is easy to compute the hash value for any given file.
- It is infeasible to generate a message that has a given hash.
- It is infeasible to modify a file without changing the hash.
- It is infeasible to find two different files with the same hash.

There are several reasons to sign such a hash (or message digest) instead of the whole document or in this case a CAD file (Digital Signatures, 2011).

For efficiency: The signature will be much shorter and thus save time since hashing is generally much faster than signing in practice.

For compatibility: Messages are typically bit strings, but particular signature schemes operate on other domains or areas. A hash function can be used to convert an arbitrary input into the proper format.

For integrity: Without the hash function, the text "to be signed" may have to be split (separated) in blocks small enough for the signature scheme to act on them directly. However, the receiver of the signed blocks is not able to recognize whether all the blocks are present and in the appropriate order.

To end this review of security, it is good practice to set a company standard for PDF delivery to internal clients, external clients, and other professionals. When setting standards for file delivery, practical security methods should follow a layered approach. A layered approach to security means there is more than one security method applied for a typical delivery. For example, one approach may be to digitally sign and certify all PDFs internal clients, and external clients would receive either a password-protected document or an encrypted document. The company should set standards based on the minimum security required to prevent the document from being reverse engineered, which does not lessen the client's ability to use and disseminate files to third parties.

The second part to the layered approach is a security term called "security by obscurity." This approach to layered security adds the other security element to a PDF that integrates with the four types of security discussed above. These methods of security by obscurity will be discussed throughout this document, but as an example, one method of security by obscurity would be micro-printing. Micro-printing is the principal of hiding information in a document by decreasing the font size to an extent the font becomes unreadable if reproduced. Therefore, the sender knows if the document has been forged by viewing the document's micro-printing features along with one of the four security methods.

CREATING AND USING SELF SIGNED CERTIFICATES

Review of Certificates

In cryptographic terminology, a certificate associates an identity with a public key. The identity is called the “subject.” The identity that signs the certificate is the “signer.” The certificate contains information about the subject and the subject’s public key, plus information about you. The entire file is cryptographically signed, and the signature becomes part of the certificate. Because the certificate is signed, it can be freely distributed over insecure channels.

Simply, a certificate contains these elements:

- Information about the subject.
- The subject’s public key.
- Information about the issuer.
- The issuer electronic signature of the above information.

A “public key certificate” (also known as a “digital certificate” or “identity certificate”) is an electronic document that uses a Digital Signature to bind a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

In a typical PKI scheme, the signature will be of a Certificate Authority (CA). In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users (“endorsements”). In either case, the signatures on a certificate are confirmations by the certificate signer that the identity information and the public key belong together.

For provable security, this reliance on something external to the system has the consequence that any public key certification scheme has to rely on a specific special setup assumption, such as the existence of a certificate authority. Below is a list of definitions of attributes typically found in an X.509 version of a certificate:

- **Version.** The X.509 version number.
- **Serial number.** The unique serial number that the issuing CA assigns to the certificate. The serial number is unique for all certificates issued by a given CA. This number does not change unless it is updated by a CA when renewing. It should not change for a minimum of one year.
- **Signature algorithm.** The hash algorithm that the CA uses to digitally sign the certificate.
- **Issuer.** Information regarding the CA that issued the certificate.
- **Valid from.** The beginning date for the period in which the certificate is valid.
- **Valid to.** The final date for the period in which the certificate is valid.

- **Subject.** The name of the individual, computer, device, or CA to whom the certificate is issued. If the issuing CA exists on a domain member server in a user's enterprise, this will be a distinguished name within the enterprise. Otherwise, this may be a full name and e-mail name or other personal identifier.
- **Public key.** The public key type and length associated with the certificate.
- **Thumbprint algorithm.** The hash algorithm that generates a digest of data (or thumbprint) for Digital Signatures.
- **Thumbprint.** The digest (or thumbprint) of the certificate data.
- **Friendly name.** (Optional) A display name to use instead of the name in the Subject field.
- **Enhanced key usage.** (Optional) The purposes for which this certificate can be used.

Distinguished Names

Names in X.509 certificates are not encoded simply as *common names*, such as "AEC Signature," or "Certificate Authority XYZ," or "CAD Administrator." They are encoded as distinguished names, which are a comma-separated list of name-value pairs. For example, the following could be my distinguished name:

- O=Banks & Banks Consulting
- OU=Geospatial Department
- CN=AEC Signature (could be users name here)
- E=help@banksandbanksconsulting.com

Therefore, what do "O," "OU," and "CN" mean to Digital ID users. A distinguished name can have several different attributes. The most common are the following:

- **(O)** Organization
- **(OU)** Organizational Unit
- **(CN)** Common Name (the user's name, or software)
- **(C)** Country
- **(E)** Email Address

What are Self-Signed Certificates

Self-Signed Certificates are created when the issuing authority has the same name as the subject authority, as seen in Figure 5. As a third-party certificate, such as VeriSign, the issuing authority is VeriSign, and the subject is the recipient's name (Your Name).

These types of certificates are created by anyone who owns a PC. All it requires is the Software Development Kits to create these Certificates like OpenSSH and Keytool for Java. By using these tools versus the integrated tools with Adobe PDF and other software, the user gains a finer granularity on the expiration date of the certificate. There may be a few select packages out there that will give you the ability to expire in the number of days for the certificate. However, the author has not found any that perform such functions other than the ones listed in this paragraph. For example, if you use Adobe to create your certificate, it will automatically create a five-year self-signed certificate. This may be acceptable for most users; however, some users might want more flexibility with the expiration of their self-signed certificate. As a result, you will need to download both the open SSH and Keytool software development kits and install them on your local computer.



Figure 5 shows the Issuer to and by fields are the same for Self-Signed Certificates.




Figure 6 shows the Issuer to and by fields are different for third Party Certificates.

When to use Self-Signed Certificates

Self-signed certificates are acceptable for small to medium size businesses. It depends on the particular use and the clientele. If you were distributing documents to the state and federal organizations, then a self-signed certificate would not be the best choice. However, self-signed certificates are acceptable if you own an office and do work locally. The main benefit of this

example is the client has been to your office and at some point physically met the owner of the certificate. In cases like this, self-signed certificates are well suited for this purpose.

 *Self-Signed Certificates are not verified by another company stating you are a real person. Third Party Companies, like VeriSign, verify Class 1 Certificates by a valid email address only.*

Another use for self-signed certificates is providing a secondary certificate for distribution to the client, which was not used to sign the document, but if you use it together with a third party certificate in message encryption, it will enforce the notion of certificate redundancy for times when you wish to encrypt a document. By having a master certificate, you will never be locked out of your own PDF. Further discussion will explain and illustrate this concept in depth in a later chapter; however, Adobe Acrobat has a mechanism to trust other certificates. In this example, you could assign the self-signed certificate as a trusted user by using trusted identities, and sign the document with a second certificate purchased from the CA. Then, you could distribute the self-signed certificate with the private key attached (PFX file) for message encryption, or send the file without the private key attached as a CER file, if digital signatures were implemented. The recipient could install the self-signed certificate and open and verify the PDF file for digital signatures or decrypt an encrypted document without having the master certificate, as shown in Figures 20 and 21. Furthermore, the recipient could not remove the signature from the certified document, as they do not hold the private key from the CA from the master certificate. As a result, it is recommended you use the master certificate for digital signatures and self-signed certificates for message encryption techniques.

How to Create and use Self-Signed Certificates

Note: Building a self-signed certificate means you will build a PFX file, as described below.

Before the specifics are shown on how to create your own self-signed certificate, you need to understand a few file extensions or formats. By understanding what these files contain, you will continue to build on the knowledge from Part One of these courses.

CER, CRT are the public key certificates exported from the PFX file or full Digital ID. The client primarily uses them to verify Digital Signatures. The CER file does not contain the private key and is not password protected, but it is safe for distribution.


CSR. This is a Certificate Signing Request. Some applications can generate these for submission to certificate authorities. It includes many of the key details of the requested certificate such as subject, organization, state, and the public key of the certificate to be signed. The CA signs these and a certificate is returned. The returned certificate is the public certificate, which itself can be in several formats (PEM, PKCS12, PFX, and P12). Generally, this signing request is used for web servers for SSL Encryption (Banking, Credit Card Info).

PEM. Described in Request For Comments (RFC) 1421 through 1424, this is a container format that may include just the public certificate or may include an entire certificate chain including public key, private key, and root certificates. The name is from Privacy Enhanced Email, a failed method for secure email, but the container format continues to be used with certificates.

KEY. This is a PEM formatted file containing just the private-key of a specific certificate. In Apache installs, this frequently resides in /etc./ssl/private. The folder permissions on this directory and the certificates are very important, and some programs will refuse to load these certificates if they are set wrong.

PKCS12, PFX, P12 Originally described by RSA in the Public-Key Cryptography Standards, the "12" variant was enhanced by Microsoft. This password protected container format contains both public and private certificate pairs. Unlike PEM files, this container is fully encrypted. To separate out the key and certificate information into a KEY and PEM file, use the Openssl command like "openssl-su."

Below are the two representative software packages used to manually create self-signed certificates. Although Adobe Acrobat allows users to create certificates, the finer aspect of adjusting the expiration date of the certificate is not allowed. Keytool and OpenSSL SDKs allow for minute adjustment of expiration fields in the number of days. The basic process for Keytool is to create a certificate store (Java Key Store known by .jks). Then, generate a key pair (Public and Private Keys). Finally, export the certificate from the store as a PEM file and rename it to a PFX file. It is important to know that both packages do the same functions, but in general practice, you should use one package or the other. Furthermore, if the command line approach is not your style, then use the Windows version called "Keystore Explorer" to create the self-signed certificates. It is currently in version 4.1.1. Moreover, for the remainder of this course, you will be shown the commands via the Keystore Explorer screenshots. However, it was necessary to show the command line approach for creating self-signed certificates as a matter of stepwise procedure as shown below.

 *There are videos that show how to install and export Digital IDs and public keys for KeyTool, OpenSSL, and Keystore Explorer on the support site.*

Keytool

1. Create a Java Key Store: **keytool** -genkeypair -dname "cn=Daryl Banks, ou=Geospatial, o=Banks and Banks Consulting, c=US" -alias "geospatial" -keypass "bbc-6063" -keystore "keystore.jks" -storepass "password" -validity "180"
2. Generate a Key Pair for Certificate: **keytool** -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks -storepass password -validity 365 -keysize 2048

3. Export as a PEM file: **keytool** -exportcert -keystore keystore.jks -storepass password -alias business -file selfsignedfile.pem
4. Rename the PEM extension to PFX.

OpenSSL

1. Create a file containing key and self-signed certificate: **openssl** req -x509 -nodes -days 365 -newkey rsa:1024 -keyout selfsignedfile.pem
2. Export *selfsignedfile.pem* as PKCS#12 file, mycert.pfx: **openssl** pkcs12 -export -out mycert.pfx -in selfsignedfile.pem -name "My Certificate"
3. Rename the PEM extension to PFX.

OR

1. Create a PKCS#12 file: **openssl** pkcs12 -export -in *selfsignedfile.pfx* -out file.p12 -name "My Certificate"
2. Parse a PKCS#12 file and output it to a file: **openssl** pkcs12 -in file.p12 -out file.pem
3. Output only client certificates to a file: **openssl** pkcs12 -in file.p12 -clcerts -out file.pem
4. Do not encrypt the private key: **openssl** pkcs12 -in file.p12 -out file.pem -nodes
5. Print some info about a PKCS#12 file: **openssl** pkcs12 -in file.p12 -info -noout
6. Include some extra certificates: **openssl** pkcs12 -export -in *selfsignedfile.pem* -out file.p12 -name "My Certificate" -certfile othercerts.pem
7. Rename the PEM extension to PFX.



Figure 7 Shows the main screen of the Windows version of KeyTool Explorer.

CREATING A CUSTOMIZED DIGITIZED SIGNATURE FOR PDF

Setting up Your Digitized Signature

To set up your digitized signature within Adobe Acrobat, go into preferences and security to create a new signature image. You have the option to configure a graphic and text with the digital signature. This will determine how the digital signature will appear when placed on the PDF document. As seen below, there are a few examples of how you can change the appearance of your signature. Figure 8 shows the standard Adobe signature with all the text criteria such as name, date, logo, and timestamp. You have much more flexibility when used in coordination with a professionally designed signature and seal such as can be seen in the last figure. The timestamp and date may not need to be shown depending on your professional standards and state requirements as it pertains to digital signatures with PDFs.

Is important to understand that a timestamp by itself is a unique identifier, meaning it can be used as a serial number to identify the document. Since the value of the timestamp at any point in an interval, it is never the same. You may customize the timestamps with several different date time formats. To ensure your signed PDF has long verification, Adobe recommends implementing a timestamp into the digitally signed process. In the next section, you will learn how to setup your timestamp.

Add a Timestamp to Signatures

You can include the date and time you signed the document as part of your signature. Timestamps are easier to verify when they are associated with a trusted timestamp authority certificate. A timestamp helps to establish when you signed the document and reduces the chances of an invalid signature. You can obtain a timestamp from a third-party timestamp authority or the certificate authority that issued your Digital ID. (Setting Up Signing, N.D.)

Timestamps appear in the signature field and in the Signature Properties dialog box. If a timestamp server is configured, the timestamp appears in the Date/Time tab of the Signature Properties dialog box. If no timestamp server is configured, the signatures field displays the local time of the computer at the moment of signing. (Setting Up Signing, N.D.)

***Note:** If you did not embed a timestamp when you signed the document, you can add one later to your signature. A timestamp applied after signing a document uses the time provided by the timestamp server. Here is a server that is currently working “<https://timestamp.geotrust.com/>”*



Configure a Timestamp Server

To configure a timestamp server, you need the server name and the URL, which you can obtain from an administrator or a security settings file.

If you have a security settings file, install it and do not use the following instructions for configuring a server. Ensure that you obtained the security settings file from a trusted source. Do not install it without checking with your system administration or IT department.

1. Do one of the following:
 - In Acrobat, choose Tools > Sign & Certify > More Sign & Certify > Security Settings.
 - In Reader, choose Edit > Protection > Security Settings.

Note: If you do not see the Sign & Certify or Protection panel, see the instructions for adding panels at Task panes.


2. Select Time Stamp Servers on the left.
3. Do one of the following:
 - If you have an import/export methodology file with the timestamp server settings, click the Import button . Select the file, and click Open.
 - If you have a URL for the timestamp server, click the New button . Type a name, and then type the server URL. Specify whether the server requires a user name and password, and then click OK.

Set a Timestamp Server as the Default

To be able to use a timestamp server to timestamp signatures, set it as the default server.

1. Do one of the following:
 - In Acrobat, choose Tools > Sign & Certify > More Sign & Certify > Security Settings.
 - In Reader, choose Edit > Protection > Security Settings.

Note: If you do not see the Sign & Certify or Protection panel, see the instructions for adding panels at Task panes.

2. Select the timestamp server, and click the Set Default button .
3. Click "OK" to confirm your selection.

💡 Note all these examples below are valid digital signatures stamps. Although Figure 10 does not show a timestamp, it is embedded into the PDF's database dictionary. Not all information needs to appear for it to be considered a certified PDF.



Figure 8 Shows a standard text Adobe PDF signature block. Note the Digital ID contents and timestamp.



Figure 9 Shows a DocEdge Adobe PDF signature block. Note the cursive text and the timestamp.

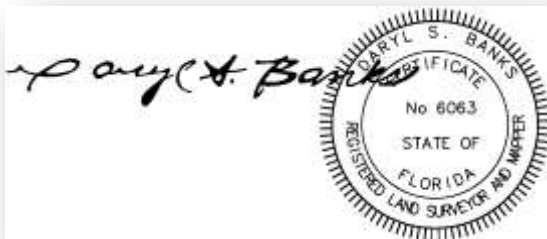


Figure 10 Shows an image Adobe PDF signature block. Note only the customized logo with a signature and seal is shown.

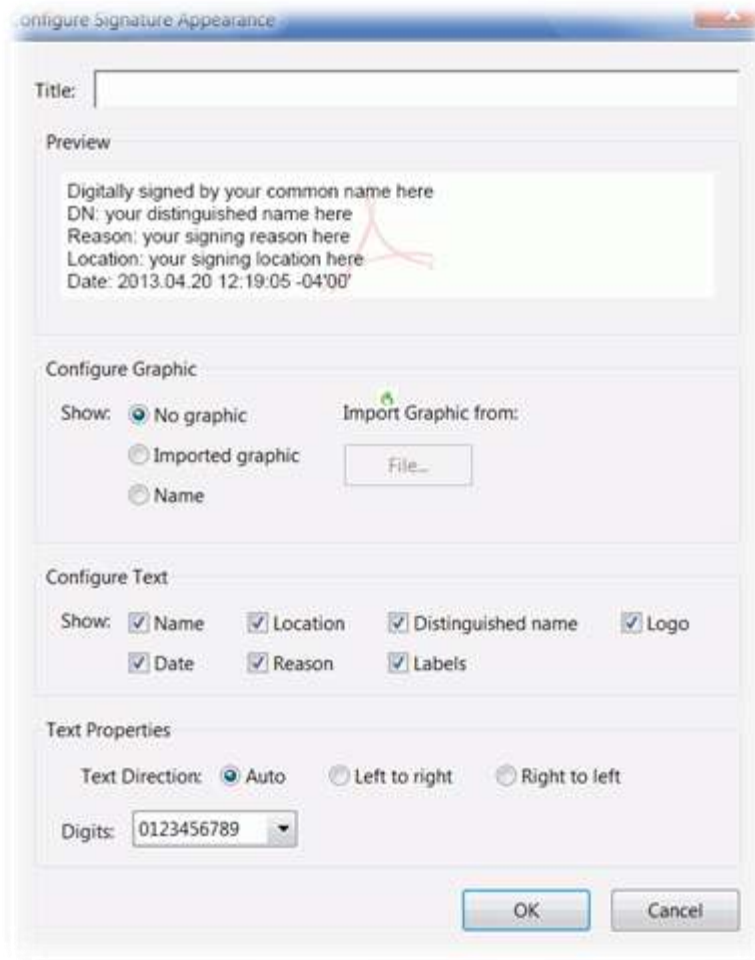



Figure 11 Shows the options to create a unique signature appearance.

Lastly, to improve appearance and give you more flexibility, you can elect to incorporate a watermark. A watermark is a recognizable image or pattern in paper that appears as various shades of whiteness or darkness when viewed by transmitted light. You can insert vector watermarks or bitmaps as raster watermarks. You also have the ability to rasterize or flatten (complete or partial rasterizing of a file) vector watermarks from Adobe Illustrator, and you have the capability to set the transparency level. Consequently, color watermarks are often used since the transparency levels can be adjusted.

Implementing the Signature

 *The Signature itself has intelligence stored in the PDF internal database.*

A standard PDF procedure for implementation of signatures has two primary components:

1. The signature dictionary, which stores the actual signature, includes attributes such as the your name, the signing time, the signed hash of the file, and your certificate (for example, embedded as a PKCS#7 object, discussed earlier does not contain the private key).
2. The signature appearance that is the visible representation of the signature. The signature appearance is specified in the signature annotation. A signature appearance is optional and standardizing the appearance is not necessary for the cryptographic purpose of verifying a signature. However, you may sign a PDF without a visible signature and still be a properly digitally signed.


In addition, to implement the digital signature on the current PDF, you may go to sign and certify. From there are several choices, two of the most common are, sign with the visible signature or sign without a visible signature. If you sign with the visible signature, it will ask you to draw a box to place your customized digital signature. Additionally, it will digitally sign the document without asking you to draw a box to place your customized logo. In either case, the drawing will be digitally signed and you will have a valid digital signature affixed to the document ready to send to a client.

Setting up Digital Signature Validation

When a client receives a digitally signed document, they may want to validate its signature(s) to verify your signature and the signed content. Depending on how the clients have configured their application, validation may occur when the file is initially opened.

A valid signature is determined by checking the authenticity of the signature's Digital ID certificate status and document integrity:

1. Authenticity verification confirms that your certificate or its parent certificates exist in the validator's list of trusted identities, also known as a "certificate chain of trust." It also confirms whether the signing certificate is valid based on your Acrobat or Reader configuration.
2. Document integrity verification confirms whether the signed content changed after it was signed. If content changes, the document integrity verification confirms whether the content changed in a manner permitted by you.

 *The settings described below are for the receiver or client side—not for you. These settings determine how Adobe handles certificates upon opening the signed PDF. Thus, you may distribute a policy with settings to optimize with your Digital ID.*

To set verification preferences, open the "preferences" dialog box and select "signatures." For Verification, and to make sure to check to automatically validate all signatures in a PDF when you open the document, select "Verify Signatures When The Document Is Opened." Also, to

check the validity of the long-term verification of a certificate, check the “Require Certificate Revocation Checking To Succeed Whenever Possible.” The revocation status is always checked for certifying signatures, but not every signature is certified.

You may elect to verify using the timestamp and verify the certificate is not expired. Select an option to specify how to check the digital signature for validity. By default, you can check the time based on when the signature was created. Alternatively, you can check based on the current time or the time set by a timestamp server when the document was signed. A Signature is timestamped, but the timestamp has expired. A message in this case is displayed if the timestamp signer's certificate expires before the current time. To let Acrobat or Reader accept an expired timestamp, select “Use Expired Timestamps” from the Digital Signature Advanced Preferences. By electing NOT to check this option, the client will be notified if the document’s certificate expiration date, which may be 1 day, 1 week, or 5 years, depending on the expiration date of the certificate, has expired.

In Acrobat or Reader, the signature of a certified or signed document is valid if you and the recipient have a transient trust relationship. An important concept is the certificate (CER) public key file that is embedded into the PDF. The trust level of the certificate indicates the actions for which the client trusts you and its embedded content to run on a client’s machine. The signer can change the trust settings of certificates to allow specific actions. For instance, you can change the settings to enable the dynamic content and embedded JavaScript within the certified document. However, signers will most often use the trust setting for establishing the root certificate. A root certificate is the originating authority in a chain of certificate authorities that issued the certificate. By trusting the root certificate, you trust all certificates issued by that certificate authority. An important concept with Self-Signed certificates is the client must manually promote the certificate to a trusted root certificate, or add the self-signed certificate to the trusted identities within Adobe for automatic “Blue Ribbon” validation.

There are different types of trust levels used with PDF documents:

- Signed Documents or Data
- Certified Documents
- Dynamic content
- Privileged System Operations

The recipient can trust documents in which the author has certified the document with a signature. They trust you for certifying documents, and accept actions that the certified document takes. The dynamic content is formats like movies, sound, and other dynamic elements that play in a certified document.

The Embedded High Privilege JavaScript option has more of a security risk, but allows privileged JavaScript embedded in PDF files to run. Finally, the privilege system operations allow Internet connections, domain scripting, silent printing, external-object references, and import/export methodology operations on certified documents.

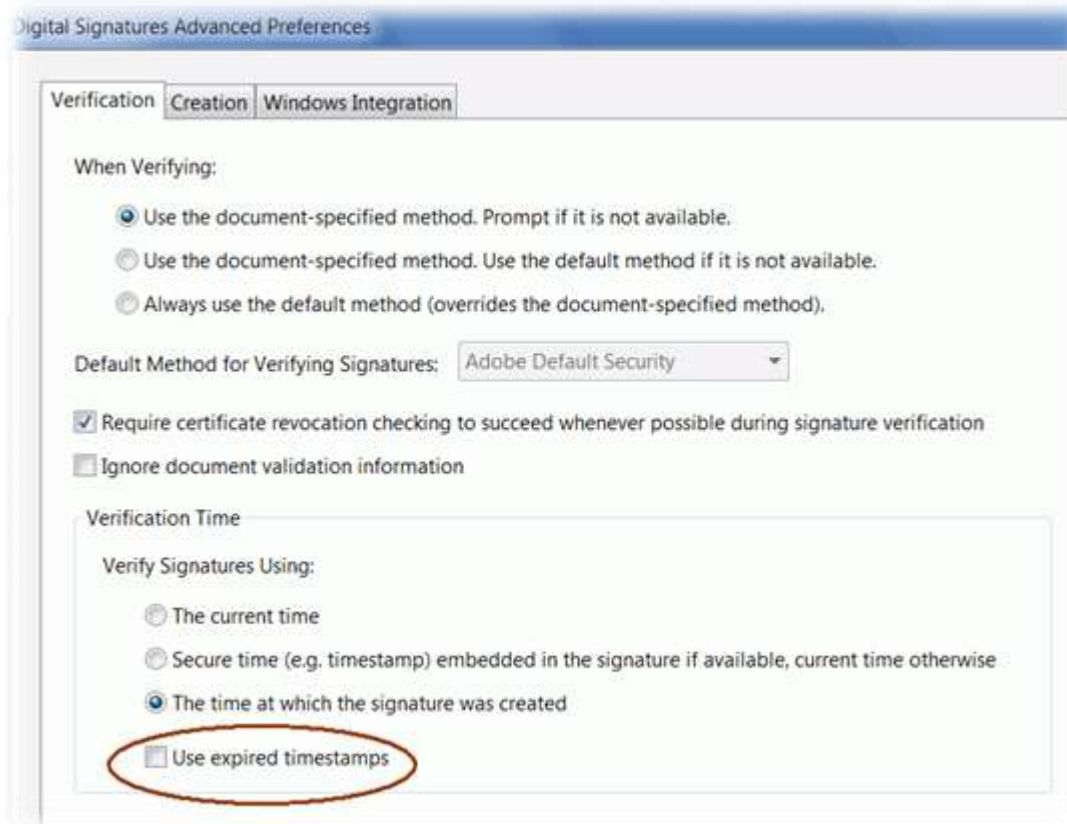



Figure 12 shows the Advanced Signature Preferences with the exclusion of expired timestamps.

USING SELF-SIGNED CERTIFICATES FOR DIGITAL SIGNATURES

PDFs and digital signatures are as robust as their CAD counterparts. They allow for all the same security measures as CAD software, but allow the PDF to have encryption of its contents and apply a certified digital signature to the PDF. This is a powerful tool to enforce policy and rules by the use of creating self-signed certificates with custom Distinguished Encoding Rules (DER).

 *With the use of encryption in a PDF, the encrypted document prohibits nearly all third party software packages from opening the document outside of Adobe Acrobat Reader. Digital Signatures do NOT provide this protection.*


You will purchase a third party Digital ID and create at least two self-signed Digital IDs to use for signing, certifying, and encrypting documents. You created the first, complete, self-signed Digital ID earlier. This may be used in lieu of purchasing a third party Digital ID for the seminar. For the purpose of this course, the third party Digital ID will be referred to as the “Administrator ID” or “Primary.” All the self-signed Digital IDs created will be referred to as the “Client” or “Secondary” Digital ID in the following three tasks:

Digital Signatures. Signing and Certifying a PDF by Digital Signature (Use Administrator or Client Digital ID). It is recommended to use the third party Digital ID when possible. However, it is acceptable to use a self-signed Digital ID. The recipient must add the certificate to the trusted identities in order to receive the visual indicators showing the PDF is valid.

Digital Signatures with Message Encryption. Signing, Certifying and Encrypting a PDF by Digital Signature (Use Administrator Digital ID) and Message Encryption (Use Client Digital ID). You use the Client Digital ID for Message Encryption, since you will need to distribute the private key, which compromises the integrity of your security. Yes, you will use both Digital IDs for this example.

Message Encryption. Message Encryption only should be limited to the Client Digital ID, since you will need to distribute the private key, which compromises the integrity of your security. When you distribute a Client Digital ID, you are preserving the integrity of the third party Digital ID. Once the recipient installs the Digital ID on their PC, they can sign and certify with your credentials. There are methods to limit the client from signing and certifying by using extensions. In reality, the client should start the encryption process by emailing you the public key certificate. Therefore, unless your clients work at the NSA, there is a good chance you will need to provide one for them to install—the client Digital ID. That will be the assumption made during these exercises.

Self-Signed Certificates for Certified, Non-Encrypted Digital Signatures

 *You only need one certificate or Digital ID, since digital signatures do not encrypt the PDF.*

If only a digital signature is applied, then you do not need to create a secondary certificate for distribution. This is only for encryption, since the recipient needs the installed private key to

complete the decryption (See Figure 1). Therefore, only the administrator's Digital ID is used to apply the digital signature to the PDF.


Steps for applying a digital signature with a self-signed certificate without encryption are as follows:

1. You can use the client or administrator Digital ID, which you will create in the next section. It is recommended you use third party Digital IDs for digital signatures.
2. Install the Digital ID in the Windows Certificate Store or imported into Adobe.
3. Under Tools, click certify with or without a signature.
4. Save the document with a new name. For example, append "DS" to the name for digital signature. "MyDigitalSigDoc-DS.pdf"
5. If you use the new version of Adobe Acrobat, then the certificate is embedded into the document and does not need to be sent along with the document.

Applying a Digital Signature to a PDF has the following features:

- Your document cannot be modified in another Adobe Product such as Illustrator.
- You can open the document with the option to "SaveAs" the PDF in another other third party PDF readers.
- Your document content is not protected from reverse engineering in programs like AutoDWG.

Self-Signed Certificates for Certified, Encrypted Digital Signatures

 *This is an advanced concept with much flexibility and strong security. It is recommended to use two certificates or Digital IDs since message encryption is used along with a digital signature—producing a certified, encrypted document with a high level of security. In addition, you do not want to give out your third party administrator's certificate private key.*

You do need the administrator's certificate in this example and described in the steps below. You want to encrypt the document with the recipient's public key, then apply a digital signature to the PDF. Next, you do need to create a secondary certificate for distribution if one was not provided by the client. The client requires the private key installed for the purpose of completing the decryption on their end. As a result, you must distribute the Client Digital ID that you will create in the next section.

The benefits of using different certificates are added granularity to control how the certificate can be modified. Since Adobe Acrobat will not allow the recipient to install an expired certificate, creating a short timespan like 90 days to expire will force the expiration of the certified, encrypted PDF after a period of time. If the client ever purchase a new computer, they could not install the certificate through Adobe. Furthermore, you may disable the extensions such as the

use of the private key after a period of time. Thus, even if you installed the certificate, the flag or extension (if set to timeout), would not allow the private key to be used. The expiration of the private key would not allow the decryption to succeed. Therefore, the PDF is rendered useless on the client's machine.

Steps for applying a digital signature with a self-signed certificate for encryption are as follows:

1. You must apply the encryption to the document before applying the digital signature.
2. Install the Administrator Digital ID and Client Digital ID to either the Windows Store or Adobe Software. See the Appendix A for the instructions for the latter. (ORC Inc)
3. Go to "Tools" and "Encrypt." You need to setup a policy for the encryption.
4. Under the policy settings choose the self-signed "client" Digital ID.
5. Choose the print and resolution settings you desire.
6. Make sure you insert the administrator or a third party certificate as the Trusted Identity. There is no limit to the number trusted identities.
7. Then, you can encrypt the document. It is a good idea for you to save the file with an appended "E" for encryption the end of the name. "MyEncryptedDoc-E.pdf."
8. Next, you apply a digital signature to the encrypted document.
9. You must use the client or administrator Digital ID created earlier. It is recommended you use third party Digital IDs for digital signatures.
10. Install the Digital ID in the Windows Certificate Store or imported into Adobe.
11. Under Tools, click certify with or without a signature.
12. Save the document with a new name. For example, append "DS" to the name for digital signature. Thus, the file will now read "MyEncryptedDoc-DSE.pdf."
13. If you use the new version of Adobe Acrobat, then the certificate is embedded into the document. Also, you must send the full, complete client Digital ID to the recipient to decrypt the document together with the password to the password protected file.

Applying a Digital Signature with Encryption to a PDF has the following features:


- Your document cannot be modified in any another Adobe Product.
- You cannot open the document with the option to "SaveAs" the PDF in another other third party PDF readers. Furthermore, you will not be able to open it in most third party programs.
- Your document content is protected from reverse engineering in programs like AutoDWG, since they cannot read the encrypted content.
- Vector data may still exist in the PDF if the clean-up option was not used.
- You must send the recipient the full Digital ID with both key pairs. However, only the private key will decrypt the document. There may be a possible DER rule to apply in this case.

- Your client must have the Adobe Acrobat or Reader Software installed. The majority of the third party readers do not support Adobes version of encryption at this time.


Custom Certificate Definite Encoding Rules

For certificates since X.509, Version 3 is the most recent (1996) and supports the notion of extensions, whereby anyone can accept an extension and include it in the certificate. A few common extensions in use today for example are KeyUsage (limits the use of the keys to particular purposes such as "signing-only") and AlternativeNames (allows other identities to also be associated with this public key, e.g., DNS names, email addresses, IP addresses). Extensions can be marked "critical" to indicate that the extension should be checked and enforced. For example, if a certificate has the KeyUsage extension marked critical and set to "keyCertSign," and if this certificate is then presented during SSL communication, it should be rejected, as the certificate extension indicates the associated private key should only be used for signing certificates and not for SSL use. (Oracle)

Figures 14 and 15 show the extension of an X.509 Certificate being set to expire one day after created. It is important to mark the extensions as being "critical" to enforce these rules. In this case, the RFC-3280 documentation states, "Private Key Usage Period" is a non-critical extension that does not require the critical indicator to be flagged. (Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile) A primary benefit for you is limiting a client usage of an expired certificate by cancelling its keys, as a secondary precaution. In that way, if the expired certificate is already installed, the keys are useless, and the certificate is inoperable.

 *The Private Key Usage Period has since been deprecated; however, with various software packages, it is still recognized and shown herein to illustrate how to apply extensions to certificates.*


Create a Primary and Secondary Self-Signed Certificate

 *If you are in a large corporation with LDAP servers, then public key certificates (CER or CRT files) may be available to you by setting up the remote LDAP server directory within Adobe. If your digital signature is not embedded into the PDF for long-term validation, you provide a certificate (self-signed) for your client to install and open the certified document. Edit Directory Server is found in the Documents preferences using the Preferences dialog box.*

One approach discussed throughout this document is creating two certificates or Digital IDs for one certificate to handle the signing and the other certificate for distribution to the client for decryption. To reiterate the concept, the two Digital IDs perform different roles. The Administrator's third party Digital ID is used for digital signatures only, and the multiple Client

Digital IDs are used for message encryption or digital signatures, depending on your preferences. With this approach, the primary goal is to enforce inserting a third party Digital ID as a trusted user for all digital signatures, so you will never will be locked out of your own document. The only time you could be locked out is when a document is encrypted by message encryption. If you only use Digital Signatures without message encryption, then this example does not apply to you.

In this next example, you would like the flexibility to send the recipient a client Digital ID for verifying digital signatures and starting an encryption conversation. Moreover, this is the reason you create a master Digital ID that will be called the “administrator” Digital ID, and the second certificate will be called the “client” Digital ID that is added to our trusted identities as a recipient. Access your administrator Digital ID to apply the digital signature; however, the client Digital ID is a trusted identity and can validate the digital signature, since it will be embedded into the document. With the newer versions of Adobe, if the certificate is embedded, the client can automatically validate the PDF by the embedded administrator public key. The distribution of the client Digital ID is not needed, since only a digital signature was applied.

 *Remember, you may have one or more client Digital IDs installed on your machine used for message encryption, and one third Party Administrator Digital ID used for Digital Signatures. By following this method, you share the client Digital ID (PFX file) with the recipient to install for the decryption. In addition, you may apply a digital signature to the encrypted document. Use the Administrator Digital ID.*

If you would like additional security, then send a certified, encrypted digital signature. There are several advantages to using this approach. For example, you can adjust a DER extension and can implement a timeout period to limit the use of the private key to 90 days. By implementing certificate extensions, you may impose limitations on the distributed certificate to only be used for the purposes you allocate in the DER settings. In this case, you set the Message Encryption flag since the recipient will only need to access their matching private key. By setting the “Key Encipherment” value, you will enable the Digital ID to be used for Encryption. Furthermore, you may decide to disable the Digital Signatures function in this Digital ID by removing the settings for Digital Signatures.

How the certificate hierarchy relates to Digital IDs is through a process called “certificate chaining.” Adobe calls this process “trust identity.” The Adobe software wants you to select certificates to sign the document and any other certificates from recipients are included in the trusted identities. For digital signatures, you are using your private key to sign the document and distribute their public key to clients. If you are in a network environment, you may have access to all the recipients’ public keys certificates. This is the point to add all those contacts’ certificates to be trusted identities. Be sure to always include the “Administrator” Digital ID so you may be able to open your own document on another computer. Self-signed certificates may

not be recoverable if lost. As a result, any document signed with a self-signed certificate should have a third party certificate included as one of the trusted identities.

The client Digital ID may serve several purposes depending on the workflow. In this example, you have included the client's Digital ID in the trusted identities section; therefore, they may sign with the administrator Digital ID and distribute the client Digital ID with its private key to the client to install on their machine. Moreover, this client does not have to be a one-to-one correlation. That means one Digital ID per email address. It may be more useful to assign all clients one Digital ID. Then you may assume the all of your clients have the same Digital ID, since, in this case, they are going to only verify the digital signature.

At this point, let us walk through the procedures to create the first Digital ID, which will be self-signed and you will enforce an expiration date on the Digital ID. The software you will use is keytool explorer 4.1.1 as of this writing. The first step is to open or create a new key store. Once this is completed, you may follow instructions in Figure 12, which is to create a key pair.

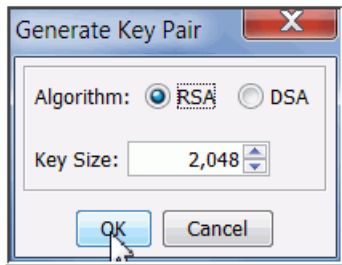


Figure 12 shows how to create a 2048-Bit RSA key pair.

Then you have a choice to either create a key pair from an RSA or DSA algorithm. The default selection is RSA. In the next step, you have the option of the signature algorithm version validity and custom serial number. Finally, name the file "Admin.PFX."

The signature algorithms are standard choices, but are out of the scope of this seminar. The one you discussed up to now is the SHA-1 with RSA. The validity period is something that has not been discussed since CA providers automatically expire a Digital ID one year from the date created. Here you have the ability to expire the Digital ID whenever you would like such as one day, one month, or one year. By forcing an expiration date on a Digital ID, you may impose a policy with third-party PDF readers, since most third-party software does not allow installation of expired certificates or Digital IDs. The Windows operating system will allow you to install expired certificates but third party software may not allow such operations depending on their design.

By creating a custom Digital ID, another feature that you may implement is extensions. Extensions are features of the certificate that will allow acceptable control by implementing policies. Finally, after all the extension information is entered and the distinguished names are filled out, you may press create to generate their certificate. The last step is exporting the key pair to a personal information file or a PFX file. This is done by right clicking on the certificate and selecting export key pair and selecting a file destination and rename PFX as the file extension name. Then, you may choose to install the Digital ID on your own computer by double clicking the file.

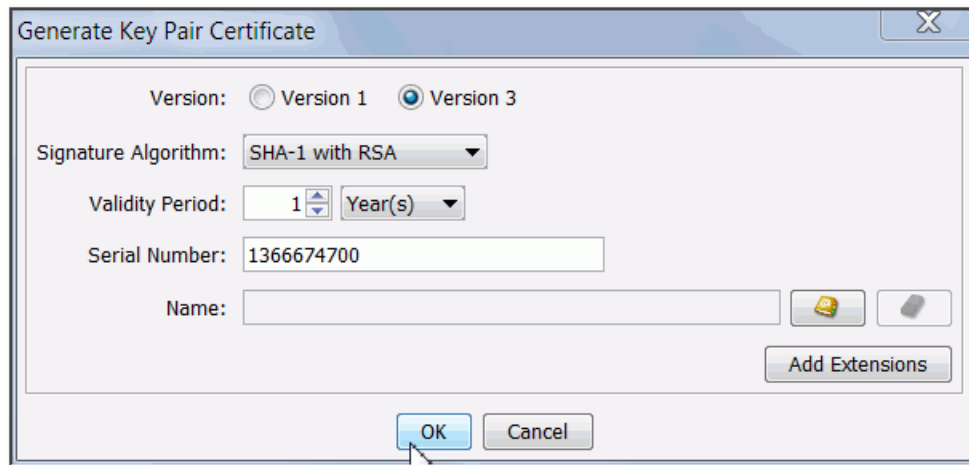


Figure 13 shows that SHA-1 with RSA is selected as the RSASHA1 discussed in the first course.

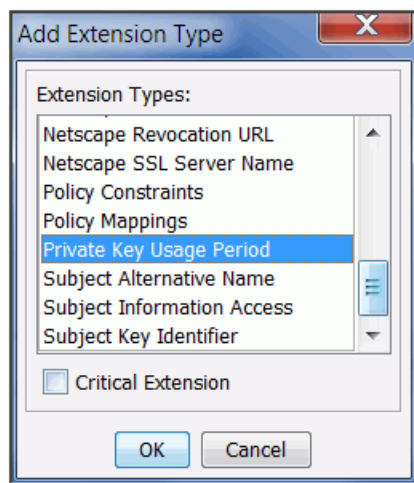



Figure 14 shows an extension to limit the private key usage period.

 Notice the Critical Extension is not enforced (checked). The RFC-3280 states the Private Key Usage Period is a non-critical extension, which means it will always be enforced.

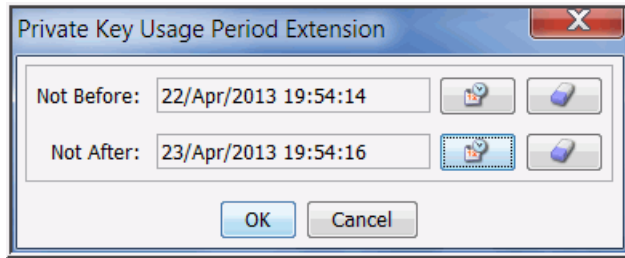


Figure 15 shows the period extension.

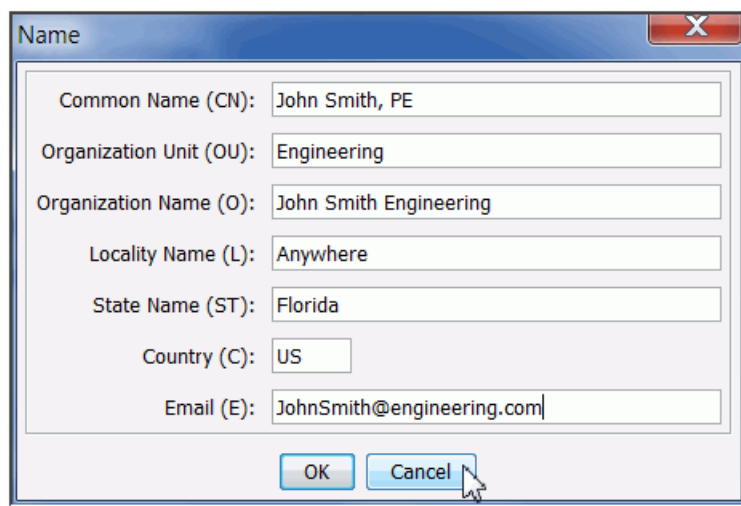


Figure 16 shows the Distinguished Names of the certificate.

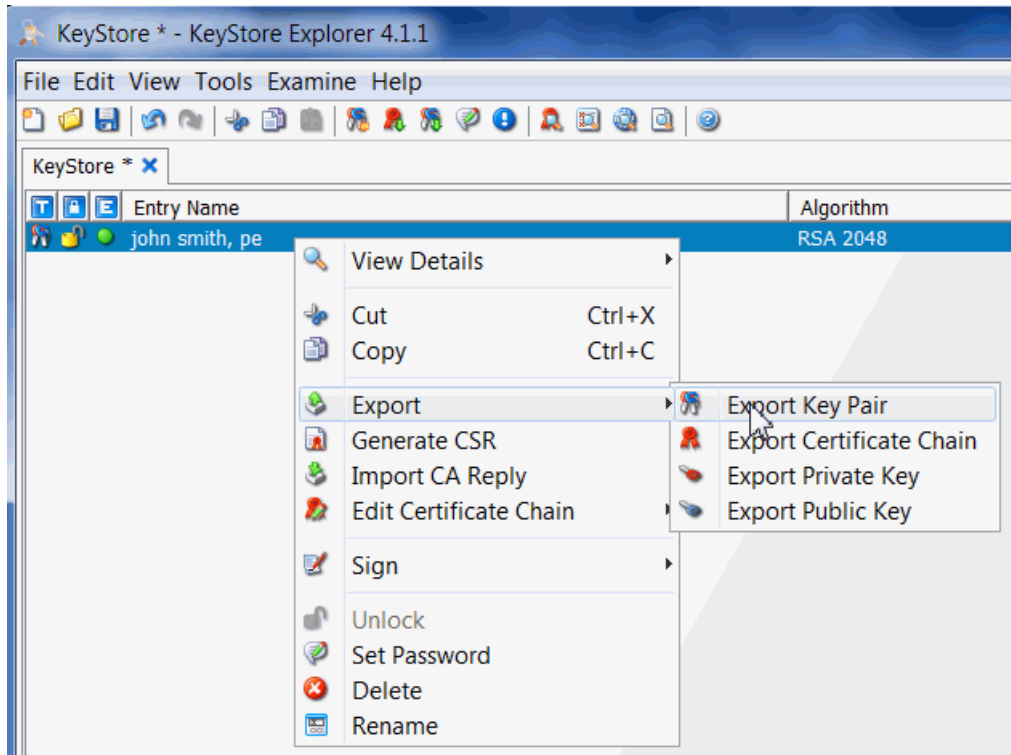


Figure 17 shows the final step of exporting the key pair to a pfx format.

To continue with a more advanced approach by using DER extensions that have not been deprecated, you will create two self-signed Digital IDs. The first Digital ID will only support Digital Signatures, and the second Digital ID will only support Message Encryption. Both of these Digital IDs will result in creating two PFX files with the goal of showing how to customize a Digital ID with DER extensions.

Follow these steps for creating Client Digital ID and only enable Message Encryption:

1. Generate a 2,048 RSA Key Pair and press OK.
2. Select SHA-1 with RSA as your Signature Algorithm and press OK.
3. Make sure you have Version 3 selected and press OK.
4. Select your Validity Period for the Certificate 1 year is the default value.
5. Add your Distinguished Names.
6. Click the Extensions Button, and click on "Add Extension."
7. Select "Key Usage" and mark it "Critical", and press OK.
8. Select "Key Encipherment" and "Data Encipherment" and press OK.
9. Verify "Digital Signatures" is not checked.
10. Press OK and Generate the Digital ID.

11. Right Click file in KeyTool Explorer and select Export→Export Key Pair.
12. Select the Directory to where you want the file to reside.
13. Name or Rename the file to “Client ID (Encryption).pfx.
14. Right Click and Select Install to Install the Certificate.

Follow these steps for creating Client Digital ID and only enable Digital Signatures:

1. Generate a 2,048 RSA Key Pair and press OK.
2. Select SHA-1 with RSA as your Signature Algorithm and press OK.
3. Make sure you have Version 3 selected and press OK.
4. Select your Validity Period for the Certificate 1 year is the default value.
5. Add your Distinguished Names.
6. Click the Extensions Button, and click on “Add Extension.”
7. Select “Key Usage” and mark it “Critical”, and press OK.
8. Select “Digital Signatures” and press OK.
9. Verify “Key Encipherment” is not checked.
10. Press OK and Generate the Digital ID.
11. Right Click file in KeyTool Explorer and select Export→Export Key Pair.
12. Select the Directory to where you want the file to reside.
13. Name or Rename the file to “Client ID (Signing).pfx.
14. Right Click and Select Install to Install the Certificate.

At this point, you have created three Digital IDs. One “Admin.PFX” and two “Client.PFX” files. Moreover, you have separated the functionality of the two client Digital IDs to signing and encrypting. What can you do with these certificates? You can perform the following combinations with these certificates:

- Admin.PFX (Sign and Encrypt)
- Client(Sign).PFX–only apply Digital Signatures
- Client (Encrypt).PFX–only apply Encryption
- Admin and Client (Encrypt) –to apply Sign and Encrypt (recommended approach)
- Admin and Client (Sign) –to apply Sign and Encrypt (not recommended)

Consequently, if you want to only encrypt your document, you would use either Admin or Client (Encrypt) certificates. Otherwise, if you want to Certify and Encrypt, you would use Admin with Client (Sign), which is not recommended since you would email your third party Admin certificate. A simpler approach would be to use only the Admin to sign and encrypt the document. However, for the decryption to succeed on the recipient’s end, the client must have the corresponding private key installed. If you signed with only the Admin, then you would compromise this certificate security since you are providing the Admin Digital ID by email.

The key point here is to have the Admin certificate as part of the combination to prevent you from being locked out of an encrypted document. Otherwise, import the Admin as a trusted identity in Adobe Acrobat. Then, you can use only the self-signed Client certificates in any combination with any proper DER extension customization.

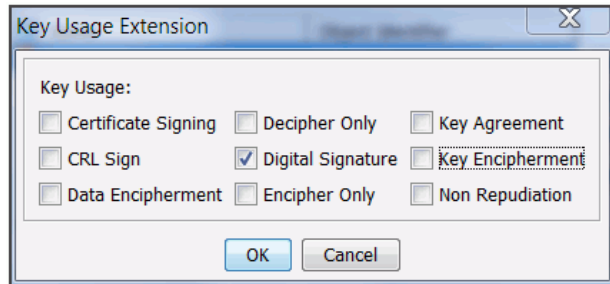


Figure 18 shows the enabling of only Digital Signatures.

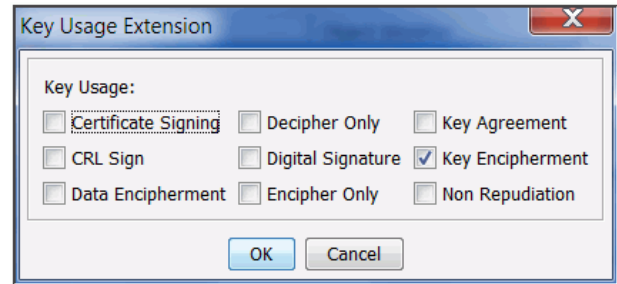


Figure 19 shows the enabling of only Message Encryption.

Implementing the Digital Signature

After both Digital IDs are created (administrator and client), the next step would be to implement the digital signature on an Adobe PDF. Therefore, install both certificates on your PC. Once they are installed on your PC, they can be managed, and added to the managed trusted identities. By adding both these certificates to the managed trusted identities, any of the trusted identities will be able to open the document. However, only you the owner of the digital signature will be able to remove a signature once the document is signed.

Digital IDs from the client's point of view only need the public key certificate to verify a signature. In the latest version of Adobe Acrobat Version 11, the public key certificate is actually embedded into the signed document. Consequently, by adding each individual certificate to the trusted identities you are embedding the public key information into the current signed document. This does not pose a security risk since it is proper protocol to distribute only the public key.

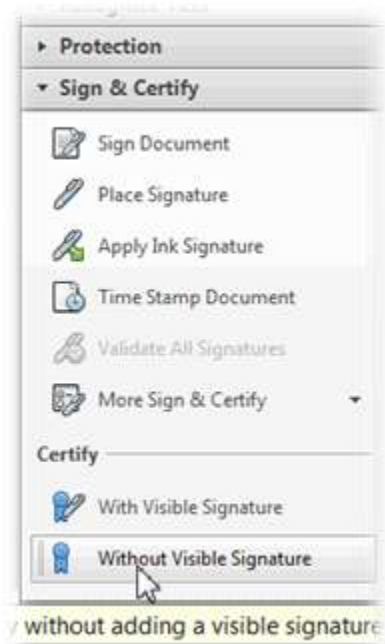



Figure 20 shows the last step of attaching a digital signature to the document without a visible signature.

When you certify a PDF, you indicate that they approve of its contents. You also specify the types of changes permitted for the document to remain certified. For example, if the recipient removes pages or adds comments, the document does not retain its certified status. Figure 20 shows the signing of a document without a visible signature.

You apply a certifying signature only if the PDF does not already contain any other signatures. Certifying signatures can be visible or invisible. A blue ribbon icon in the signatures panel indicates a valid certifying signature. A Digital ID is required to add the certifying digital signature. Remove content that may compromise document security, such as JavaScripts, actions, or embedded media.

1. Choose Sign > Work With Certificates to open the panel.
2. Click one of the following options:
 - Certify (Visible)

 Places a certified signature in either an existing digital signature field (if available) or in the location you designate.

- Certify (Not Visible)

 Certifies the document, but your signature appears only in the Signatures panel.

3. Follow the onscreen instructions to place the signature (if applicable), specify a Digital ID, and set an option for Permitted Actions After Certifying.
4. Save the PDF using a different filename than the original file, and then close the document without making additional changes. It is a good idea to save it as a different file so that you can retain the original unsigned document. There are settings that will allow you to create another file automatically. (Adobe)

Distributing the Certified Document

As was stated earlier, the newest version of Adobe Acrobat (Version 11) embeds the public key certificates of the trusted identities, so there will be no need to send a separate public key file to the client. The only file needed is the digitally signed document. However, if you are using older versions of Adobe Acrobat, they will need to export the public key certificate and produce a CER file to send each client for complete verification of the digital signature. If the signature is embedded into the document, the client can check the signature and certificate by clicking on the signature box and navigating to the certificate. Once at the certificate tree node, they can add that particular sender to the trusted identities. By performing this action, they will receive the blue certification ribbon at the top left of their Adobe document that gives them visual cues to the state of the document.

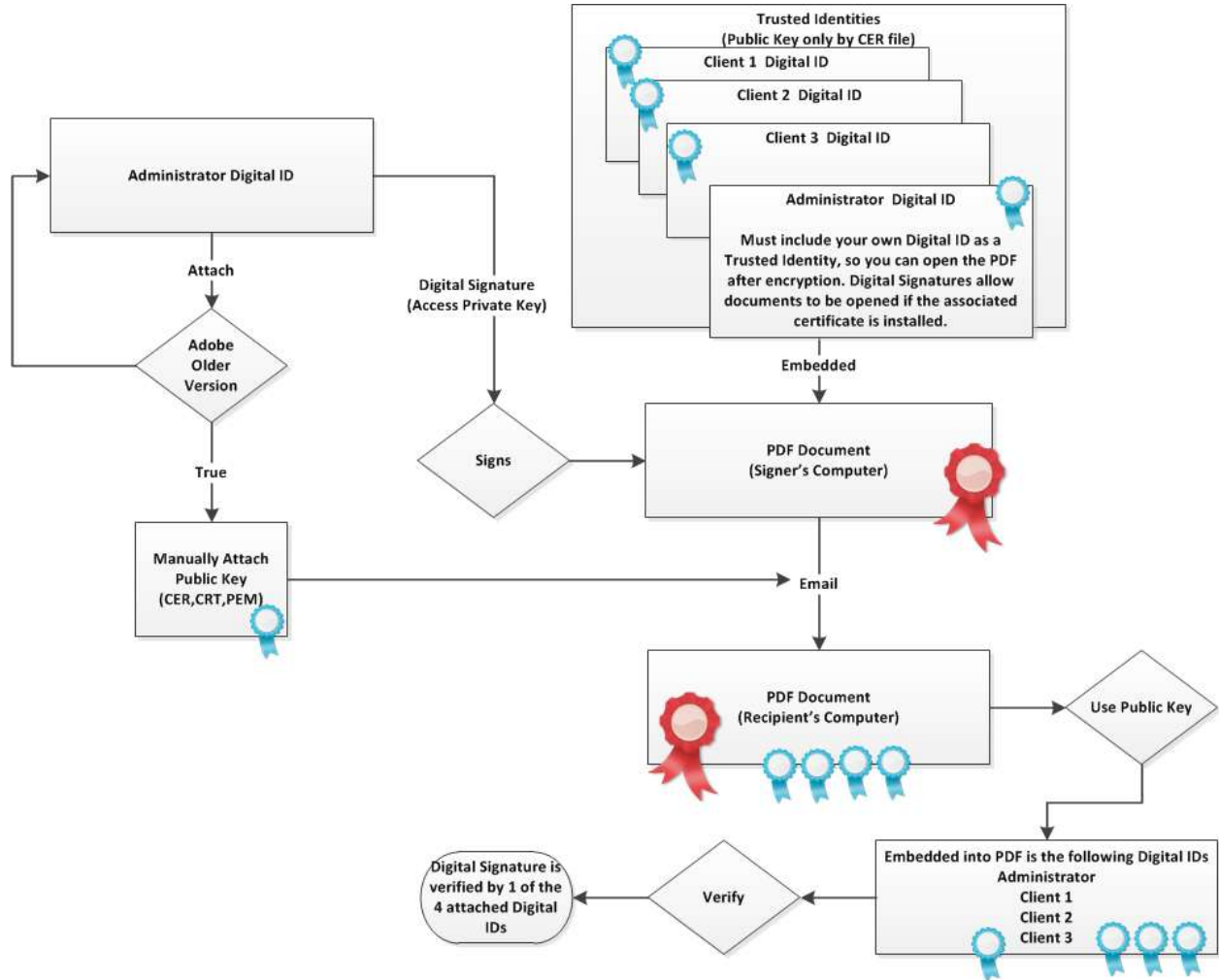


Figure 21 shows the trusted identities attached to the digitally signed and certified PDF that is sent to the client.

USING SELF-SIGNED CERTIFICATES FOR MESSAGE ENCRYPTION

Create a Primary and Secondary Self-Signed Certificate

If you are in a large corporation with LDAP servers, then certificates may be available to you by setting up the remote LDAP server directory within Adobe. However, the increased usage of certificates by a particular company may not have a LDAP server setup for general usage. In this case, you provide a certificate (self-signed) for your client to install and open the certified, encrypted document. Edit Directory Server is found in the Documents preferences using the Preferences dialog box.

Using self-signed certificates for message encryption is very similar to self-signed certificates with digital signatures. However, the one major difference is message encryption uses your public key distributed by the recipient to you by email. In other words, the recipient has a Digital ID installed on their computer and they are requesting an encrypted conversation from you, so they include or attach their public key certificate in an email, which you install and use to encrypt the PDF. Then, you email the encrypted document back to the intended recipient. Finally, the recipient decrypts the encrypted PDF with the private key.

The challenge with using public and private keys is a small portion of the population understands and actually uses Digital IDs and certificates on a daily basis. At this point in time, it's safe to assume that your client will not understand how to use a Digital ID much less have one installed on their system that is designated for signing PDFs. Consequently, an approach the author has implemented is to create a self-signed "client" Digital ID specifically for decrypting PDFs. The signer must attach the client Digital ID along with the PDF in the email for the recipient to install for complete trust to be actualized. As in the previous example with message encryption, the administrator Digital ID signed the document and the distributable client Digital ID, which was emailed in a trusted identity for the client to install. In this example, when you create your client Digital ID, you do not want to expire your private key too soon, since the private key decrypts the PDF on the client side.

It is important to understand that the client Digital ID is a complete Digital ID. Although it is self-signed and created by you, the client knows you and has visited your office. The only difference with this example is you are providing the client a Digital ID instead of the client providing you the public key CER file. Ten years from now the situation will be different. The client will provide you with their public key to encrypt the PDF and email, but, until that time where every user knows their Digital ID mechanics, you have to be proactive and set up a process that is safe and effective for both parties.

The final point to make concerning the client Digital ID is that it is separate from the administrator ID. The administrator ID is what will actually decrypt the document in the situation of a lost client Digital ID, since it is installed on your machine, and the client and administrator Digital ID are listed as a trusted identity under managed trusted identities settings. This trusted identity feature allows the encryption by one certificate like the administrator's Digital ID to be successful on the client machine, and the decryption to be completed with the client Digital ID on another machine.

Follow these steps for creating Client Digital ID without DER features:

1. Generate a 2,048 RSA Key Pair and press OK.
2. Select SHA-1 with RSA as your Signature Algorithm and press OK.
3. Make sure you have Version 3 selected and press OK.

4. Select your Validity Period for the Certificate 1 year is the default value.
5. Add your Distinguished Names.
6. Press OK and Generate the Digital ID.
7. Right Click file in KeyTool Explorer and select Export → Export Key Pair.
8. Select the Directory to where you want the file to reside.
9. Name or Rename the file to “Client ID (DSE).pfx”.
10. Right Click and Select Install to Install the Certificate.

Implement the Message Encryption

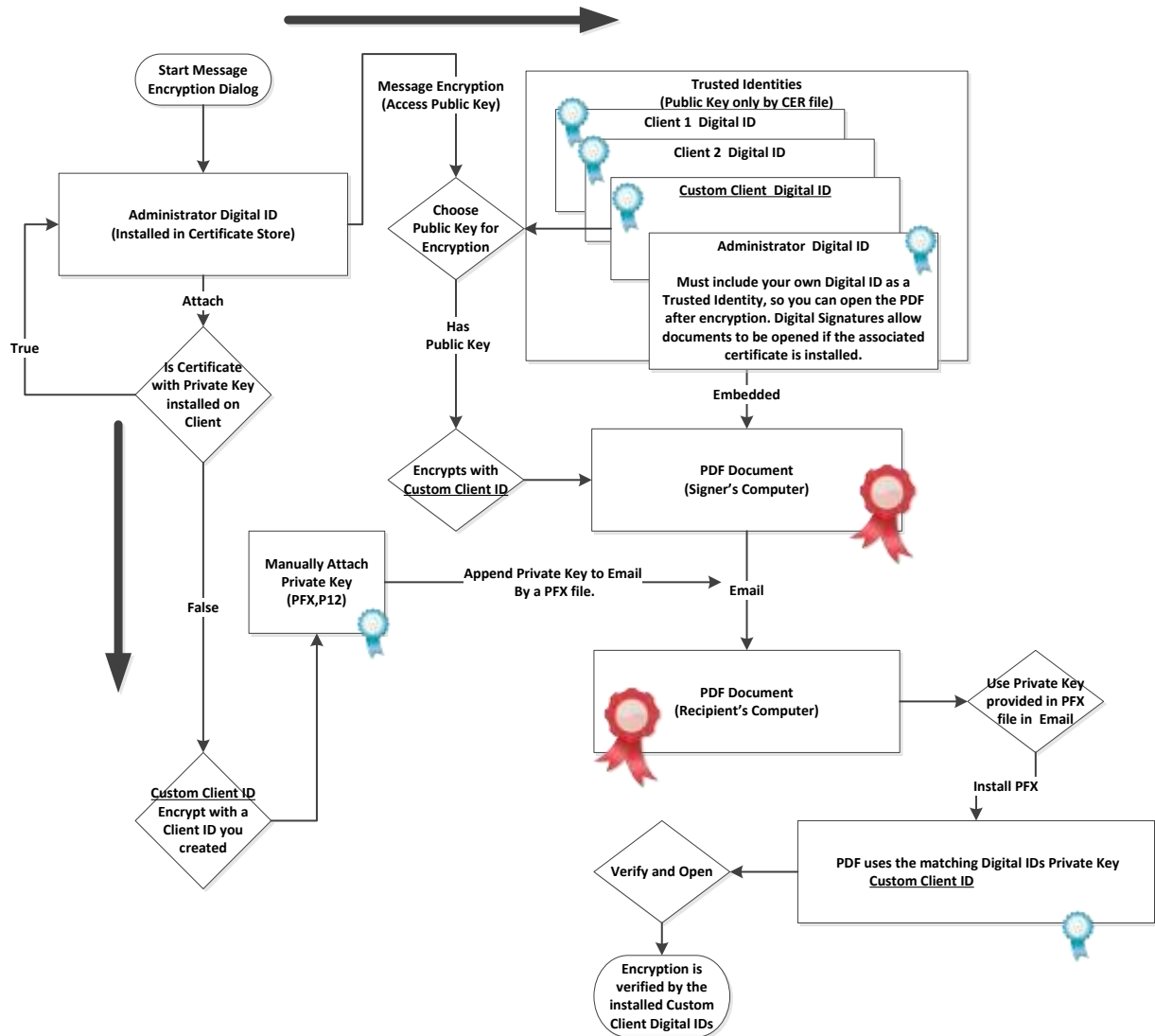



Figure 22 shows the custom client Digital ID for Message Encryption.

To implement message encryption with the administrator and client Digital ID, it is important to reiterate the point that you are providing a complete Digital ID for the client to install on his or her machine. Consequently, you are performing the process out of order, since the client should be emailing the public key certificate to you. You will then encrypt the PDF and email it to the client (they must have the original Digital ID installed with a private key attached) where they have the private key with the corresponding certificate to decrypt it.

Figure 22 illustrates how to create a custom client ID in which you use the public key to encrypt the document. Then, you attach the full Digital ID for the custom client Digital ID in an email along with the encrypted PDF. Once the client receives the encrypted PDF and the corresponding Digital ID that was used to encrypt it, the recipient must install the complete Digital ID before he or she can decrypt the PDF.

 *Important! The Digital ID is a password-protected certificate (PFX). The password is NOT the private key used as part of the key-pair. These are separate items.*

Distribute the Encrypted Document

As shown in Figure 22, once the document is encrypted with the client Digital ID, the sender must distribute the complete Digital ID with the private key of the client. This is assuming the sender created the self-signed Digital ID for the client to install. Otherwise, all the sender would have to do is email the PDF to the client and the client would then open it up since they have the Digital ID installed.

Note: Since encryption requires public and private keys and the private key resides on the client side, you use a client Digital ID. As a result, you do not have to deliver your private administrator Digital ID, which may leave you vulnerable to impersonation.

Also, a client Digital ID may be a one-to-one correlation or one-to-many. This means you may create one client Digital ID for all your clients, or create one Digital ID for each client. The latter example is used more for a high security facility.

ENCRYPTING AND SIGNING WITH A THIRD PARTY DIGITAL ID

Signing with a third party, Digital ID has one major advantage over self-signed certificates. This advantage being the ability to automatically trust all the way up to the originating CA root certificate. For example, VeriSign issues a certificate to Company X. Company X then uses that Digital ID to digitally sign a PDF with a digital signature, and email it to their client. Company X's Digital ID is automatically trusted by Adobe Reader to verify up to the root certificate—that being VeriSign, as seen in Figure 6. The certificate is verified by Internet via a web page link of the issuer's serial number. This verifies whether the Digital ID is in good standing with the issuer or has been revoked by the CA.

If you and the recipient use a self-signed certificate, you both have to add it to the trusted root certificate store to gain the same level of trust by third party applications. By moving it to the trusted root certificate store, no applications will do external checks to verify the certificate. It will be automatically trusted for signing and encrypting or the certificate's intended purpose. In comparison with the two types of certificates, there is a manual intervention by the client for self-signed certificates to receive the same level of trust as third party Digital IDs.

UNDERSTANDING HOW PDF CONVERTS TO CAD FORMATS

You are expected to have a basic understanding of basic image processing and vector and raster file formats. To keep within the scope of the course, you will need to explore these topics on your own for a complete overview of the latest uses and technology like Bézier Curves.

Advances in Technology

With the advances in software development kits and the proliferation of the number of software developers, the number of devices and high-level software products on the market has exploded over the past decade. For example, once thought impractical, voice to text translation is commonplace and will only improve in the years to come. With this exponential growth in software development, you need to be informed of the ability for average users to possess professional tools. The consideration of the impractical to commonplace by an average user should be a concern for the AEC industry. Before the Internet, the hard copy signed and sealed drawings were limited by the paper medium. Now, the electronic medium with third party plugins to adapt and enhance software baseline packages has the ability to easily change and print the drawing outside of a CAD package. The basics will be discussed with simple solutions to mitigate theft and impersonation of your content.

Raster to Vector



Adobe PDFs can be either raster or vector, and vector PDFs may have raster images embedded into the document.

Vector graphics is the use of geometrical primitives such as points, lines, curves, and shapes or polygon, which are all based on mathematical expressions, to represent images in computer graphics.

Raster graphics is a bitmap that corresponds bit-for-bit with an image displayed on a screen, generally in the same format used for storage in the display's video memory, or maybe as a device-independent bitmap. A bitmap is technically characterized by the width and height of the image in pixels and by the number of bits per pixel (a color depth, which determines the number of colors it can represent). Raster images are more commonly called *bitmap* images.

A bitmap image uses a grid of individual pixels where each pixel can be a different color or shade. Bitmaps are composed of pixels. Vector graphics use mathematical relationships between points and the paths connecting them to describe an image. Vector graphics are composed of paths. The image to the left below is representative of a bitmap and the image to the right is representative of a vector graphic. The images shown below are magnified to exaggerate the fact that the edges of a bitmap become jagged as it is scaled up.

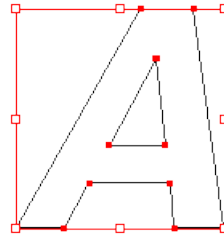


Figure 23 shows a jagged bitmap of raster text. Figure 24 shows the smooth mathematical vector text.

There are several conversion tools over the market specifically for PDF to CAD conversion. They are listed as follows:

- Adobe Illustrator
- AutoDWG
- Able2Extract

Adobe Illustrator is the most obvious choice. If a conversion with software products exists within its product line, it is most likely the best choice to stay with that company. These all will convert a PDF file (raster or vector PDF) to vector CAD format. You may convert a raster PDF that has been flattened (completely rasterized) to vector, but there is little intelligence gained such as lines, text, and arcs brought into the newly converted vector drawing. As the software continues to improve, the “smart objects” will eventually convert nearly 100% of the raster objects to CAD native vector objects (text, line, polyline, etc.). Several factors will increase the chances of a successful conversion from PDF to CAD. These factors are listed below for the source file:

- Image Resolution
- File type is vector format
- Use of the Bézier Curve or other Mathematical Approximations

Conversion Techniques

There are a few software packages that can read a 100% raster image and convert each object to vector text and line work. Autodesk Raster Design is a software package that can convert raster bitmap images to vector formats. However, the software is not 100% automated; there is an enormous amount of manual conversion with the “smart tools” to convert individual objects. As a result, it is time consuming and may not be cost effective to convert each line to a vector format. However, if a project is cost effective enough to convert each line and text, then it may be a point of consideration for your company as a point of security concern.

Adobe Illustrator is a more mainstream approach the author has used to convert CAD drawings. This software is the primary focus of this paper and the author will spend much of his time convincing AEC professionals the need to secure their deliverables.

Adobe Acrobat and Illustrator used a programming language called postscript to render raster and vector objects to printers. Laser printers combine the best features of both printers and plotters. Like plotters, laser printers offer high quality line art, and like dot-matrix printers, they are able to generate pages of text and raster graphics. Unlike either printers or plotters, however, a laser printer makes it possible to position high-quality graphics and text on the same page. PostScript made it possible to fully utilize these characteristics by offering a single control language that could be used on any brand of printer.

PostScript

PostScript is a complete programming language of its own. Many applications can transform a document into a PostScript program whose execution will result in the original document. This program can be sent to an interpreter in a printer, which results in a printed document, or to one inside another application, which will display the document on-screen.

PostScript is worth mentioning for implementing on-the-fly rasterizing; everything, even text, is specified in terms of straight lines and cubic Bézier curves (previously found only in CAD applications), which allows arbitrary scaling, rotating and other transformations. When the PostScript program is interpreted, the interpreter converts these instructions into the dots needed to form the output. For this reason, PostScript interpreters are occasionally called PostScript Raster Image Processors, or RIPs. (Adobe Corporation)

Bézier Curves

Bézier curves, mentioned above, are widely used in computer graphics to model smooth curves. As the curve is completely contained in the convex (upside down u-shape) hull of its control points, the points can be graphically displayed and used to manipulate the curve intuitively. Geometric transformation that maps points and parallel lines to points and parallel lines such as

translation and rotation can be applied on the curve by applying the respective transform on the control points of the curve.

Quadratic and cubic Bézier curves are the most common. Higher degree curves are more computationally expensive to evaluate. When more complex shapes are needed, low order Bézier curves are patched together, producing a Bézier spline. A Bézier spline is commonly referred to as a "path" in vector graphics standards (like SVG) and vector graphics programs (like Adobe Illustrator, CorelDraw, and Inkscape). To guarantee smoothness, the control point at which the two curves meet must be on the line between the two control points on either side.

The simplest method for scan converting (rasterizing) a Bézier curve is to evaluate it at many closely spaced points and scan convert the approximating sequence of line segments. However, this does not guarantee that the rasterized output looks sufficiently smooth because the points may be spaced too far apart. Conversely, it may generate too many points in areas where the curve is close to linear.



To avoid too much math, investigate the animation of the Bézier curves found here http://en.wikipedia.org/wiki/B%C3%A9zier_curve#Linear_curves.

Vector File Formats



Along with PDF formats, a few other well-known vector formats are shown below:

EPS. If the graphics application you are using cannot open native vector files, the next option would be to save them as EPS (Encapsulated PostScript) files. These are self-contained PostScript files, which contain the same mathematical descriptions as the source vector files. Even bitmaps can be saved in the EPS file format. EPS files are supported by most all graphics applications. It is the most portable format for this reason. It is best to use EPS files for all line art and illustrations because they can be reproduced at any size or resolution and still display exactly as they were drawn. Use them wherever native vector files cannot be used.

It is important to understand the different formats such as AI, PNG, EPS, PDF, BMP, GIF, and TIFF. Of those examples, Adobe Illustrator (AI) and these EPS, PDF software contain vector data.

SVG. Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics that has support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999. SVG has been used in Internet home design applications and websites.

AI. Adobe Illustrator is the companion product of Adobe Photoshop. Photoshop is primarily geared toward digital photo manipulation and photorealistic styles of computer illustration, while

Illustrator provides results in the typesetting and logographic areas of design. Early magazine advertisements (featured in graphic design trade magazines such as Communication Arts) referred to the product as "the Adobe Illustrator."

A few last words regarding file format. The vector PDF is much more likely to contain high level or smart objects such as fonts, lines, curves, and text blocks that will be readily reproducible into a CAD format. If you receive an EPS, AI, SVG, PDF (Vector) file, then it can be scaled to print any size, since it contains vector objects. (Adobe Corporation)

Resolution Types

Before PDFs are reverse engineered, the source PDF is may be scaled by a small percent when sending a print to printer. If a PDF's scale is reduced by 2%, then a 100-foot line can become a 98-foot line. Along with the possible text overrides, this can make for a dangerous rubber sheeting exercise that you would like to avoid. If you strategically reduce the quality of a print in an attempt to mitigate reverse engineering, then you are adhering to the layered approach to securing your documents.

A short introduction into different types of resolution can help you determine what the resolution of a file is by determining a few key properties of the file. If you can deduce the image resolution, then you have a straightforward, measurable approach to reducing the quality of a file. As image files are conveniently described by their dimension in pixels (1200 ppi x 800 ppi), you will learn how to quickly deduce the resolution of the file and resample the original file to either a larger or smaller sheet size.

Image Resolution

Pixel dimensions measure the total number of pixels along an image's width and height. Resolution is the quality of detail in a bitmap image and is measured in pixels per inch (ppi). The more pixels per inch, the greater the resolution. An image with a higher resolution produces a high quality printed image.

If an image is resampled, the amount of image data remains constant as you change either the print dimensions or resolution. If you change the resolution of a file, its width and height change accordingly to maintain the same amount of image data. Downsampling reduces the number of pixels per inch, while Upsampling increases the number of pixels per inch and the file size accordingly.

Looking at the table below, the resolution from a digital camera (in megapixels MP) shows the print size condition of a photo as it is scaled up to 20 inches by 30 inches. In current mobile phones, it is common to have 8 MP resolution camera. By analyzing this table, if a client used the camera on the phone to take a photo and reproduce the drawing, the results are indicating the

quality of the photo will be maintained up to the 20 inch by 30 inch print size. Consequently, a client could take a photo or scan of a drawing producing a high quality print.


	Print Size (inches)						
	2x3"	4x5"/4x6"	5x7"	8x10"	11x14"	16x20"	20x30"
Resolution(ppi)							
320x240	Good	OK	Poor	Poor	Poor	Poor	Poor
640x480 0.3 MP	Excellent	Good	Poor	Poor	Poor	Poor	Poor
800x600	Photo Quality	Very Good	Fair	Poor	Poor	Poor	Poor
1024x768	Photo Quality	Excellent	Good	OK	Poor	Poor	Poor
1280x960 1 MP	Photo Quality	Photo Quality	Very Good	Good	Poor	Poor	Poor
1536x1180	Photo Quality	Photo Quality	Excellent	Very Good	OK	Poor	Poor
1600x1200 2 Megapixel	Photo Quality	Photo Quality	Photo Quality	Very Good	OK	OK	Poor
2048x1536 3 Megapixel	Photo Quality	Photo Quality	Photo Quality	Excellent	Good	OK	OK
2240x1680 4 Megapixel	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Very Good	Good	OK
2560x1920 5 Megapixel	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Excellent	Very Good	Very Good
3032x2008 6 Megapixel	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Very Good	Very Good
3072x2304 7 Megapixel	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Photo Quality	Very Good	Very Good

Table 1 shows a subjective interpretation of a digital camera resolution versus print quality.

Monitor Resolution

The monitor's resolution is depicted in pixel dimensions. For example, if the monitor resolution and a photo's pixel dimensions are the same size like 640 x 480, the photo will fill the screen when viewed at 100%. How large an image appears on-screen depends on a combination of factors, the pixel dimensions of the image, the monitor size, and the monitor resolution setting. In Photoshop, you can change the image magnification on-screen, so you can easily work with images of any pixel dimensions.

Printer Resolution

Printer resolution is measured in ink dots per inch, also known as dpi. Generally, the more dots per inch, the more acceptable printed output you will receive. Most inkjet printers have a resolution of approximately 720 to 2880 dpi. Printer resolution is different from, but related to, image resolution. To print a high quality photo on an inkjet printer, an image resolution of at least 220 ppi should provide good results. Adobe PDF has a default setting of 75 or 150dpi for low-resolution print setting. Its high level is double that at 300dpi, which is most common for AEC Industries. However, if the 150dpi is sufficient many times for bitonal (black and white) drawings. Try to limit the resolution to only allow you enough information to perform your work efficiently.

For example, a bitmap image that measures 1,000 × 1,000 pixels has a resolution of 1 megapixels. If it is labeled as 250 PPI, that is an instruction to the printer to print it at a size of 4 × 4 inches. Changing the PPI to 100 in an image-editing program would tell the printer to print it at a size of 10×10 inches. However, changing the PPI value would not change the size of the image in pixels that would still be 1,000 × 1,000.

Suppose your image is 1200 pixels wide. Contemplating printing it, you see that you could print this image at several different sizes, simply by changing the scaled resolution: (Fulton)

- 1200 pixels / 11 inches = 109 dpi
- 1200 pixels / 10 inches = 120 dpi
- 1200 pixels / 9 inches = 133 dpi
- 1200 pixels / 8 inches = 150 dpi
- 1200 pixels / 6 inches = 200 dpi
- 1200 pixels / 4 inches = 300 dpi
- 1200 pixels / 3 inches = 400 dpi
- 1200 pixels / 2 inches = 600 dpi

For another example, the input settings in a monochrome \$100 scanner are set at 300 dots per inch (dpi). This preset is designed for optimum OCR results when scanning black text on white

paper, or when scanning line art. Below is an example of a legal size (8.5 x 14 inches) scan at 300dpi, and the corresponding print size that would be expected from the typical scanned image.

Scanning 8.5 x 14 inch legal size - Computing print short side

Scanning 14.00 x 8.50 inches (355.6 x 215.9 mm) at 300 dpi (118 pixels/cm)

Will print 13.18 x 8.00 inches (334.7 x 203.2 mm) at 319 dpi (125 pixels/cm).

This is 94% Size (Reduction to 0.941x, 8.00/8.50 inch, or 300/319 dpi)

Here is how it works:

Input (scanning parameters)

(14.000 inches x 300 dpi) x (8.500 inches x 300 dpi) = 4200 x 2550 pixels

Output (printing parameters)

(13.176 inches x 319 dpi) x (8.000 inches x 319 dpi) = 4200 x 2550 pixels

It is the same pixels either way. This subject is scaling.

Sufficient pixels to print at 240 to 300 dpi is reasonable quality for photo-quality.

This 4200 x 2550-pixel image size is 10.71 million pixels, with Aspect Ratio 1.65:1

The image size in memory is:

- 61.283 MB if 48 bit RGB
- 40.855 MB if 32 bit CMYK
- 30.642 MB if 24 bit RGB (most common)
- 20.428 MB if 16 bit GrayScale

- 10.214 MB if 8 bit GrayScale
- 1.277 MB if 1 bit Line Art

These examples illustrate the dependency of paper size and pixel dimensions as shown in the input and output parameters shown in the Table 1 above. Furthermore, the scanned file or a rasterized PDF may be scaled up to a larger print size depending on the resolution. As shown in Table 1, a 1200 ppi document would be expect to have a maximum print size of 8 inches by 10 inches for a professional quality print.

A final example shows an 11 inch by 17 inch scan at 300 dpi to be 16.8 MP. Reviewing the data in Table 1, you see the minimum MPs to have a professional 20 inch by 30 inch scaled print to need a minimum of 3 MPs to scale an 11 x 17 to 20 x 30 inch print. You can infer the minimum dpi from this as example as equals 127 dpi to 71dpi. To verify this result, you see the following calculations:

$$(11 \times 127\text{dpi}) \times (17 \times 127\text{dpi}) = 3.0 \text{ MP and } (20 \times 71\text{dpi}) \times (17 \times 71\text{dpi}) = 3.0 \text{ MP}$$

A final word of caution the Table 1 is entirely subjective or opinion based and may not hold true for every case. The estimate chosen at 3MP was a lower quality estimate. It should be more like 6MP for professional quality. However, you may develop a similar table of your own and use the examples presented here.

HOW TO PREVENT PDF FILES FROM BEING REVERSE ENGINEERED

In order to mitigate unwanted changes to your PDF files, you need to understand how to implement additional security features to augment the layered security approach to document management. These techniques are currently being using in the marketplace and based on well-established recommendations from the software vendors who design and create the software. Thus, you will be using the following measures to strengthen your documents:

- PDF Resolution
- Embedded Fonts
- Microprinting
- QR Codes
- Clean the PDF
- Document Encryption

Understanding PDF Print Resolution

Below in Table 2 are typical printer resolutions that correspond to the maximum image resolution for that particular printer. If the client is using a 600dpi laser printer, then there is no need to send them a 300ppi image of a PDF. Giving too much information by a high image

resolution can lead to issues with changing a printed drawing and scanning it back into a TIFF format at 1200-2400dpi. By setting the image resolution to the lowest standard (170ppi), the drawing will be sure to lose quality when scanned after printed, and microprinting items will be destroyed as well. Thus, the sender could positively confirm the drawing was scanned into another format. (Adobe Corporation)

Printer Resolution	Line Resolution	Image Resolution
300 dpi (laser printer)	60 lpi	120 ppi
600 dpi (laser printer)	85 lpi	170 ppi
1200 dpi (imagesetter)	120 lpi	240 ppi
2400 dpi (imagesetter)	150 lpi	300 ppi

Table 2 Shows the recommended PDF print resolution for a specific dpi printer.

Understanding Text Fonts

Adobe Acrobat embeds the fonts used to create or convert the PDF from Microsoft Word for example. This allows the PDF to have the exact appearance of the original file. Therefore, the PDF should always have the fonts embedded to ensure the appearance you desire.

Embedded fonts are vector graphics. Although a PDF may have a mixture of bitmap and vector fonts, when you rasterize the document, the fonts are deleted. After the PDF is flattened, the postscript language constructs recognize and approximate the curves of the rasterized text objects. Since the objects are rasterized, the approximation results in a loss of quality in the final print and most likely a large file size.

To ensure an exact match to the source document, it is a good idea to embed all fonts used in the document. If you do not need an exact match and you prefer a smaller file, you can choose not to embed fonts for roman text and East Asian text (Traditional Chinese, Simplified Chinese, Korean, and Japanese). Text in these languages is replaced with a substitution font when viewed on a system that does not have the original fonts. The Fonts panel of the PDF Optimizer contains two lists for fonts, fonts that are available for unembedding and fonts to unembed. Certain fonts are not available for unembedding and do not appear in the Fonts panel. To unembed fonts in a document, select one or more fonts in the Embedded Fonts list, and click the “Unembed” button. If you do not want to embed subsets of the embedded fonts, deselect “Subset All Embedded

Fonts.” To prevent unembedding for all fonts in the document, select “Do Not unembed Any Font.”

One lesser-known misunderstanding about PDF documents is that they should behave like any other document that contains images and text, letting you freely move or edit items on a page. A PDF is like a Polaroid of your original document. You can perform minor touch-ups, but if your PDF requires substantial revision, it is easier to make changes to the source document and regenerate the PDF. This is a nice feature to help resist your documents to unwanted changes. You would unembed all fonts, which theoretically would render the text immutable (Adobe Corporation, N.D.). In Adobe software, this may hold true; however, you should have doubts whether third party PDF readers in the future will have a problem reading these characters by Optical Character Recognition (OCR). By removing the fonts in your document, you are strengthening the document security by rendering the text unchangeable with other Adobe software, and you are increasing security by added to the layered approach to security. It would make an additional step for an individual to change the bitmap text.

Microprinting

One of the items from the past that is probably the easiest to implement and most forgotten methods of security is the term known as microprinting. Microprint, a microscopic printed character, cannot be readily copied on standard office photocopiers or multifunction printers (MFPs) widely used throughout the world. The characters can be read under magnification to verify document authenticity. If photocopied, the microprint characters appear as a solid line or are noticeably degraded and illegible.

The microprint appears as a solid “signature line” or other graphical element, but is actually a series of printed characters or words. The letters “MP” often accompany this application of microprint, providing an indicator that microprint is present and where document examiners should look for it during inspection. (Troy Corporation, 2009)

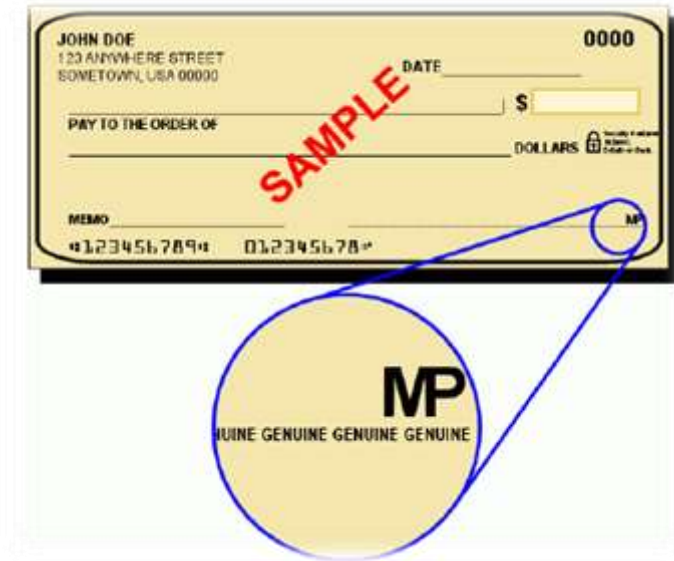


Figure 25 Shows the example of microprinting on a check from a financial institution. (Troy Corporation, 2009)

QR Codes

The Quick Response (QR) Code system has become popular outside the automotive industry due to its fast readability and greater storage capacity up to 7,000 characters (BaselineCorp) compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, general marketing, and much more. A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a smart phone's camera) and processed using Reed-Solomon error correction until the image can be appropriately interpreted; data is then extracted from patterns present in both horizontal and vertical components of the image as seen below.



Figure 26 Shows the example of a QR Code.

As mobile devices continue to improve, innovative developers are increasingly using security implementations that are more sophisticated. You can embed information with the QR Code into your documents to record various information like email addresses, website URLs, and company addresses. You can include the QR Code image into the final output either directly in the CAD drawing or as a stamp in the PDF.

An approach, you could develop a colored stamp that complements the microprinting into your drawings that would sufficiently degrade after one hard copy output. Thus, the QR Code would be the nonprofessionals' check if a document was returned. You could scan the QR Code, and if the code was unreadable, there would be reason to suspect the drawing as being copied more than once and void the document. As discussed, the QR Code is the preferred method to embed custom information into a bitonal image, but how do you handle unwanted information that resides in your PDF? Next, you will learn how to clean and remove these items from your documents.


Flatten and Clean the PDF

Before you distribute a PDF, you may want to examine the document for sensitive content or private information that can trace the document to you. Such information can be hidden or not immediately apparent. For example, if you created the PDF, the document metadata normally lists your name as the author. You may also want to remove content that can inadvertently change and modify the document's appearance. JavaScript, actions, and form fields are types of content that are subject to change.

Use the Remove Hidden Information feature to find and remove hidden content from a PDF. Use the Black Out & Remove Content tools to remove sensitive images and text that are visible in a PDF. (Adobe Corporation)

Find and Remove Hidden Content

Use the Remove Hidden Information feature to find and remove content from a document that you do not want, such as hidden text, metadata, comments, and attachments. When you remove items, additional items are automatically removed from the document. Items that are removed include digital signatures, document information added by third party plug-ins and applications, and special features that enable you to review, sign, and fill PDF documents.

 *To examine every PDF for hidden content before you close it or send it in e-mail, specify that option in the Documents preferences using the Preferences dialog box.*

1. Choose Tools > Protection > Remove Hidden Information.

If items are found, they are listed in the Remove Hidden Information panel with a selected check box beside each item.

2. Make sure that the check boxes are selected only for the items that you want to remove from the document.
3. Click Remove to delete selected items from the file, and click OK.
4. Choose File > Save, and specify a filename and location. If you do not want to overwrite the original file, save the file to a different name, location, or both.

The selected content is permanently removed when you save the file. If you close the file without saving it, repeat this process, making sure to save the file.

Remove Hidden Information Options

Metadata

Metadata includes information about the document and its contents, such as the author's name, keywords, and copyright information, used by search utilities. To view metadata, choose File > Properties.

File Attachments

Files of any format can be attached to the PDF as an attachment. To view attachments, choose View > Show/Hide > Navigation Panes > Attachments.

Bookmarks

Bookmarks are links with representational text that open specific pages in the PDF. To view bookmarks, choose View > Show/Hide > Navigation Panes > Bookmarks.

Comments and Markups

This item includes all comments that were added to the PDF using the comment and markup tools, including files attached as comments. To view comments, choose the Comments pane.

Form Fields

This item includes form fields (including signature fields) and all actions and calculations associated with form fields. If you remove this item, all form fields are flattened and can no longer be filled out, edited, or signed.

Hidden Text

This item indicates text in the PDF that is either transparent, covered up by other content, or the same color as the background.

Hidden Layers

PDFs can contain multiple layers that can be shown or hidden. Removing hidden layers removes these layers from the PDF and flattens remaining layers into a single layer. To view layers, choose View > Show/Hide > Navigation Panes > Layers.

Embedded Search Index

An embedded search index speeds up searches in the file. To determine if the PDF contains a search index, choose View > Tools > Document Processing > Manage Embedded Index. Removing indexes decreases file size but increases search time for the PDF.

Deleted or Cropped Content

PDFs sometimes retain content that has been removed and no longer visible, such as cropped or deleted pages, or deleted images.

Links, Actions and JavaScripts

This item includes web links, actions added by the Actions wizard, and JavaScripts throughout the document.

Overlapping Objects

This item includes objects that overlap one another. The objects can be images (composed of pixels), vector graphics (composed of paths), gradients, or patterns.

Redact (Black Out and Remove) Sensitive Content

Redaction is the process of permanently removing visible text and graphics from a document. You use the Black Out & Remove Content tools (also called redaction tools) to remove content. In place of the removed items, you can have redaction marks that appear as colored boxes, or you can leave the area blank. You can specify custom text or redaction codes to appear over the redaction marks.

Discard Objects Panel

The Discard Objects panel lets you specify objects to remove from the PDF and lets you optimize curved lines in CAD drawings. You can discard objects created in Acrobat and in other applications. Selecting an object removes all occurrences of that object within the PDF.

In the Discard Objects area, you can select from these and other options:

Discard All Form Submission, Import and Reset Actions

Disables all actions related to submitting or importing form data, and resets form fields. This option retains form objects to which actions are linked.

Flatten Form Fields

Makes form fields unusable with no change to their appearance. Form data is merged with the page to become page content.

Discard All JavaScript Actions

Removes any actions in the PDF that use JavaScript.

Discard All Alternate Images

Removes all versions of an image except the one destined for on-screen viewing. Some PDFs include multiple versions of the same image for different purposes, such as low-resolution on-screen viewing and high-resolution printing.

Discard Embedded Page Thumbnails

Removes embedded page thumbnails. This is useful for large documents, which can take a long time to draw page thumbnails after you click the Page Thumbnails button.

Discard Document Tags

Removes tags from the document, which also removes the accessibility and reflow capabilities for the text.

Convert Smooth Lines to Curves

Reduces the number of control points used to build curves in CAD drawings, which results in smaller PDF files and faster on-screen rendering.

Detect and Merge Image Fragments

Looks for images or masks that are fragmented into thin slices and tries to merge the slices into a single image or mask.

Discard Embedded Print Settings

Removes embedded print settings, such as page scaling and duplex mode, from the document.

Discard Embedded Search Index

Removes embedded search indexes, which reduces file size.

Discard Bookmarks

Removes all bookmarks from the document.

Discard User Data Panel

Use the Discard User Data panel to remove any personal information that you do not want to distribute or share with others. If you are unable to find personal information, it may be hidden. You can locate hidden text and user-related information by using the Examine Document command (Tools > Protection > Remove Hidden Information).

Discard All Comments, Forms and Multimedia

Removes all comments, forms, form fields, and multimedia from the PDF.

Discard Document Information and Metadata

Removes information in the document information dictionary and all metadata streams. (Use the Save As command to restore metadata streams to a copy of the PDF.)

Discard All Object Data

Removes all objects from the PDF.

Discard File Attachments

Removes all file attachments, including attachments added to the PDF as comments. (PDF Optimizer does not optimize attached files.)

Discard External Cross References

Removes links to other documents. Links that jump to other locations within the PDF are not removed.

Discard Private Data of Other Applications

Strips information from a PDF document that is useful only to the application that created the document. This does not affect the functionality of the PDF, but it does decrease the file size.

Discard Hidden Layer Content and Flatten Visible Layers

Decreases file size. The optimized document looks like the original PDF but does not contain layer information.

Clean Up panel

The options in the Clean Up panel of the PDF Optimizer remove useless items from the document. These items include elements that are obsolete or unnecessary for your intended use of the document. Removing certain elements can seriously affect the functionality of the PDF. By default, only elements that do not affect functionality are selected. If you are unsure of the implications of removing other options, use the default selections.

Object Compression Options

Specifies how to apply Flate compression in the file.

Use Flate to Encode Streams That Are Not Encoded

Applies Flate compression to all streams that are not encoded.

In Streams That Use LZW Encoding, Use Flate Instead

Applies Flate compression to all content streams and images that use LZW encoding.

Discard Invalid Bookmarks

Removes bookmarks that point to pages in the document that have been deleted.

Discard Invalid Links

Removes links that jump to invalid destinations.

Discard Unreferenced Named Destinations

Removes named destinations that are not being referenced internally from within the PDF document. Because this option does not check for links from other PDF files or websites, it does not fit in some workflows.

Optimize Page Content

Converts all end-of-line characters to space characters, which improves Flate compression.

Optimize the PDF for Fast Web View

Restructures a PDF document for page-at-a-time downloading (byte-serving) from web servers.

Encrypt the Document



A quick recap may be needed to differentiate the need for administrator and client Digital IDs. Below shows the difference in how they are implemented. In reality, the administrator and client certificates are named for convenience, show permissions, and level of authority.

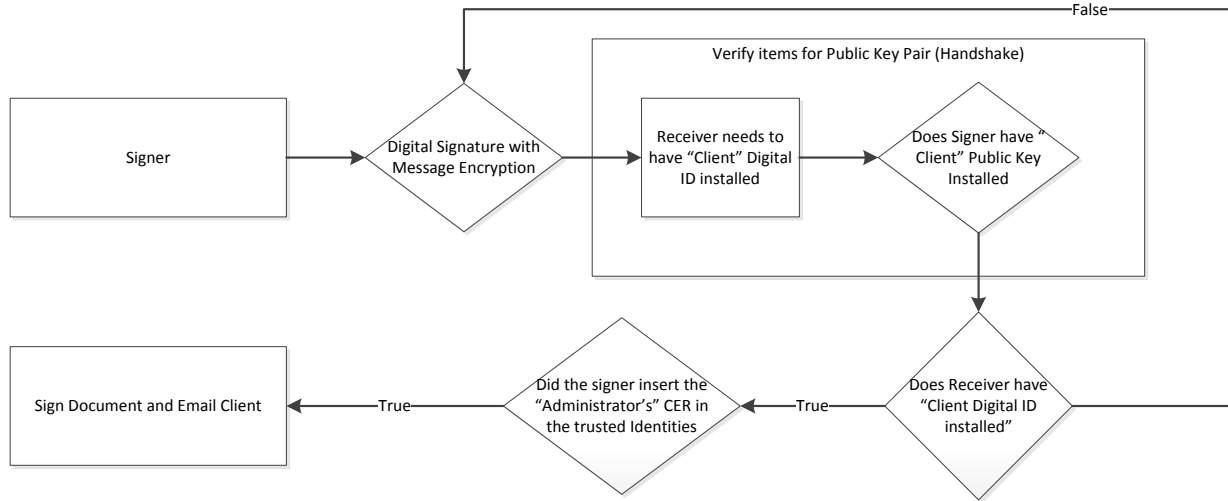


Figure 27 shows the items required for Encryption with Digital Signatures.

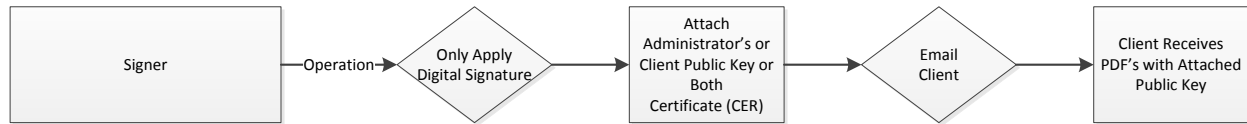


Figure 28 shows the items required for Digital Signatures.

One of the easiest methods to mitigate the loss of information from a PDF is to encrypt the document. Adobe uses the Certified Document Services (CDS) program, which automatically trusts new Digital IDs that are chained to (part of the family of) the Adobe Root certificate embedded in Adobe products. Simply stated, if you have purchased a third party Digital ID from a known CA, then your signed and encrypted documents will show up as a trusted signer (marked with a blue ribbon on Version 10). This is a visual indicator for the client that everything is trustworthy, as compared to yellow and red indicators from untrusted certificates.

By implementing encryption, you are forcing the client to become a transitive trust user, which means they must have given you a public key to install, or you must have provided a full digital self-signed client ID for them to install. Either method works; however, you must include the recipients in the trusted identities before encrypting the document as shown in Figure 27.

Once encrypted, you may elect to apply a digital signature. This is an advanced feature. PDF supports the Federal Information Processing Standard (FIPS) certified AES 256 algorithm and provides a number of advanced capabilities. Once the encrypted document has a digital signature attached, make certain the recipient has the private key certificate or Digital ID installed on their

machine. Otherwise, he or she may not be able to open the document. Moreover, the benefit of using encryption with a digital signature is it prevents third party software from opening the document. If you attempted to open the document with a software like Illustrator, it will not be allowed to open since it requires enhanced security to handle the request.

Another benefit to encryption with digital signatures is the ability to set policies. With policies, you may do the following tasks:

- Reduce the document resolution to a low-resolution output of 150dpi.
- Eliminate you from making any changes to the document.
- Force document expiration.

An increased security is achieved by reducing the image resolution and eliminating changes, but an overlooked aspect of encryption is expiring certificates. If you have the PDF stored on a file server along with the complete Digital ID (pfx file), the client will not be able to install the certificate after the expiration date on the certificate. As a result, the document is unable to be opened or viewed by the client, and the client must ask the sender for another copy of the document. As compared to signing with a digital signature, the client will always be able to open the document, but any changes to the document by the client will be automatically tracked.

With digital signatures, a third party software may allow the document to be “Saved As” another name, which opens the document to be changed. If opened in Adobe, however, the client does not have this capability. Thus, encryption forces the client to actually hack the document to get around the enhanced security. The only drawback the author has experienced using this method is with installing the full client Digital ID. If you have a “low tech” client, maybe the digital signature is enough security for safe distribution.

STRATEGIES FOR DOCUMENT MANAGEMENT

What Is a Roaming Policy?

When you secure a PDF using a certificate, you specify the recipients and establish the file access level for each recipient or group. For example, you can allow one group to sign and fill forms and another to edit text or remove pages. Users can choose certificates from their list of trusted identities, files on disk, LDAP server, or the Windows certificate store (Windows only). Always include your certificate in the recipient list so that you can open the document later.

There are two methods that will be discussed regarding policy and management to access public key certificates, Internal Access and External Access.

Internal Access with a Lightweight Directory Access Protocol (LDAP) server allows signers to add users directly from an internal server directory. This is usually seen with very large companies with email lists. Users can search their company directory for that recipient's email address. In a similar manner, you can search a specific LDAP directory for recipients' certificates, as seen in Figure 29.

External Access is implemented with roaming clients outside a corporate network structure. This means a server and software web service must manage the validity of the certificate and track other details from clients.

Internal Document Management

 *LDAP servers may be used internally and externally within an organization. Hence, the web addresses known as a URL as "directory.verisign.com" is used for external access over the Internet.*

Directory servers are commonly used as centralized repositories of identities within an organization. The server acts as an ideal location to store your certificates in enterprises that use certificate encryption. Directory servers let you locate certificates from network servers, including Lightweight Directory Access Protocol (LDAP) servers. After you locate a certificate, you can add it to your list of trusted identities so that you do not have to look it up again. By developing a storage area for trusted certificates, you or a member of your workgroup can facilitate the use of encryption in the workgroup.

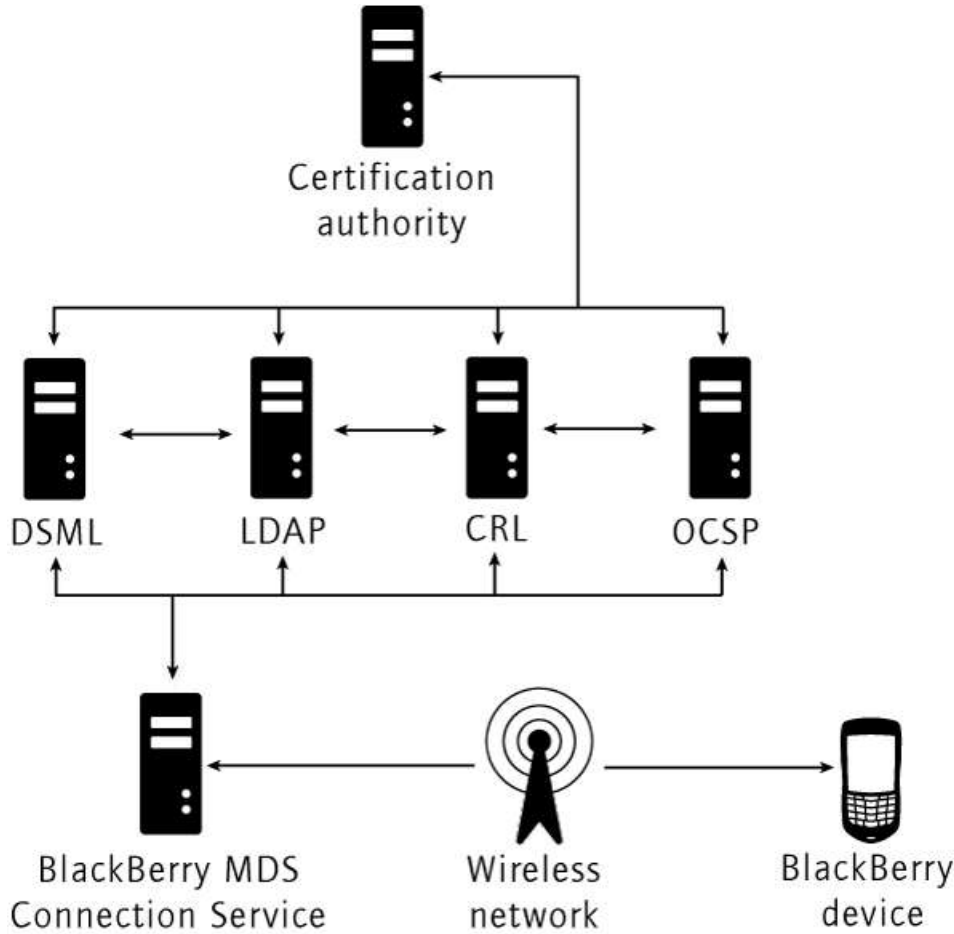


Figure 29 shows the LDAP server configuration downstream from a CA, and pushes content to external mobile devices. (Blackberry)

Below in Figure 30 is the LDAP server for our third party Digital ID provider VeriSign. LDAP may be used internally or externally as shown by the web URL “directory.verisign.com.” This is the address Adobe uses to verify the certificate up to the trusted root certificate.

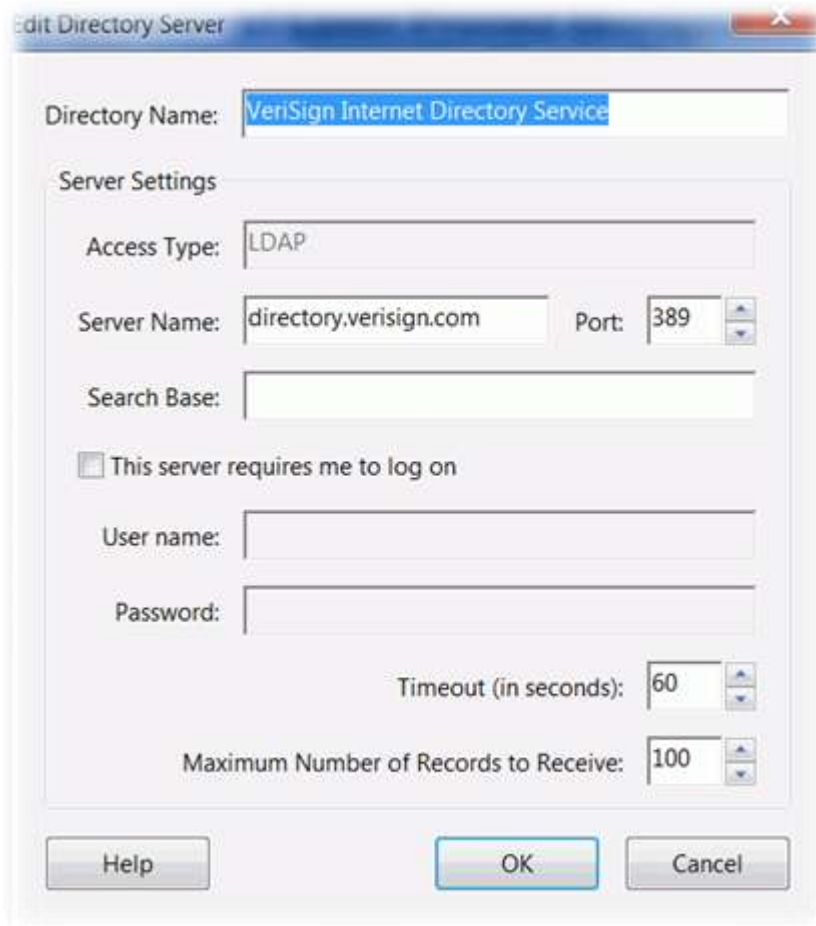


Figure 30 shows the LDAP server edit page.

The process to setup an LDAP server is the IT Administrator’s responsibility. They setup a server to serve files. In this case, it is an LDAP service. They specify a location for the server, port number and specify the permissions for the files.

External document management

Several companies offer affordable solutions for roaming external digital signature management referred to as document management. A few management workflows also provide the subscriber (you) the Digital ID for installation like ARX CoSign. The main point here is a document management may NOT include message encryption, so be cautious when subscribing to a service if message encryption is a required feature for your company.

As clients move from paper-based to electronic documents, electronic software workflows increase productivity and reduce costs for organizations, but can also increase the risk of

document forgery or tampering. By using a cloud-based solution, organizations receive a solution that offers their electronic documents the same assurance of origin and integrity as a wet ink signature provides a physical document. There are several cloud-based document management solutions on the market. A few notables are listed and discussed below:

- Adobe LifeCycle Management
- Adobe Certified Document Services (CDS)
- DocuSign
- ARX CoSign
- GlobalSign
- LockLizard

PDF Security dynamically protects PDF documents inside and outside the network, online and offline, with strong encryption, document expiration, and access rights, to provide persistent end-to-end protection throughout a protected PDF document's lifecycle.

The above solutions implement security with different approaches. Although the standards regarding the security algorithms may be similar, the workflows may be entirely different and need to be discussed and evaluated before moving forward with a company policy.

Adobe LifeCycle Management tools is an enterprise document and form platform that helps you capture and process information, deliver personalized communications, and protect and track sensitive information. LiveCycle ES4 extends business processes to your mobile workforce and clients, increasing productivity while broadening service access to any user equipped with a desktop, smartphone, or tablet. (Adobe Corporation)

While digital signature technology is not new, Adobe is working with security partners to provide an Adobe Certified Document Services solution that is easy to use for you and recipients on the Adobe PDF platform. Document recipients using the free Adobe Reader on supported platforms will have the ability to automatically validate a certified document without additional software or configuration. Adobe Systems Incorporated has contracted with one or more third parties to provide CA services including all Registration Authority (RA) functionality. If you are interested in creating certified documents, you will register with one of these authorized third parties and have your identification information verified. Only then you are then provided with a certificate used in Adobe Acrobat Standard or Professional to certify documents. All third parties providing CA services for Certified Document Services will be governed by the following policies.

- This Certificate Policy (CP or Policy) is called the CDS Certificate Policy.
- The Attribute Object Identifier (OID) for this Policy is: 1.2.840.113583.1.2.1.
- The extended key usage OID for the CDS PKI is: 1.2.840.113583.1.1.

GlobalSign is a provider of the Adobe CDS certificate. You may purchase this certificate from them for a few hundred dollars. This allows your clients to have automatic blue ribbon validation on the signing panel in Adobe. It is a nice feature that gives the recipient immediate confirmation without any hassle of installing certificates and trusting self-signed certificates. (GlobalSign)

DocuSign uses a “Software as a Service” (SaaS) option whereas you upload the PDF to be signed and then sign the PDF via the web page, which is then managed through the web portal. The PDF is protected by the security mechanisms of the provider. (DocuSign)

Arx CoSign issues a client software and certificate (Digital ID) for you to install and use with Adobe Acrobat. You can sign with the client software or the custom digital signature within Adobe. Prices range \$15-100 per month. (Arx CoSign)

Finally, the last of the document management solutions is LockLizard. LockLizard implements a proprietary solution that hosts the certificate in the cloud and the document as well. Additional control over the PDF is managed via the web portal. These controls allow you to add and remove users’ access to the PDF, and expire the PDF at a custom date. The client must download a software application to interface with the cloud data. (LockLizard Limited) Here is a list of functions LockLizard provides:

- Stop copying & unauthorized distribution.
- Prevent document sharing & piracy
- Stop printing, or control the number of prints
- Stop screen grabbing
- Instantly revoke access to information
- Expire access & control usage
- Apply dynamic watermarks with individual user identification
- Audit document use & identify leaks

There are several document management providers for small and large enterprises. One notable basic permissions based provider is “Google Docs.” Currently, there is no management of a Digital ID; however, Google is an alternative for an affordable solution for document management solutions without the Digital ID management feature. Therefore, sign the document with an embedded signature before uploading.

These companies provide different solutions depending on the specific need of the client. In many financial institutions, there are mandated document signing workflows that documents are

signed before proceeding to the next person within that organization. You may not need this level of security; however, these are the current possibilities provided in the marketplace. You will note that further investigation into each solution is needed before determining the best solution for your company. The good aspect of technology is there are free time trials to use the product.

SUMMARY

This course was designed to challenge you with integrated security implementations with PDFs that are forward-thinking by design, and differs from the plain Adobe Help Files. Moreover, you were reminded that producing a PDF from a CAD software does not necessarily remove any of the vector content. Although PDFs are great for distributing files over the Internet, it is important to understand the third party software that can manipulate and reverse engineer the PDF into a CAD file. As an operator of this technology, you should use these techniques yourself if a CAD file becomes corrupt. A quick export from Illustrator will give you a virtually complete CAD file again.

This course encourages you to not only apply a digital signature to a PDF, but also encrypt it to prevent third parties from reverse engineering the PDF. Although nothing is 100 percent, you can eliminate the advanced technical user from reverse engineering the document by encrypting your document. Implementing the recommended layered approach to security by adding microprinting and QR Codes to a drawing, you can certainly mitigate the chance of an undetectable change in a drawing such as rubber sheeting an image file.

Self-Signed certificates were implemented to apply the message encryption. Any Digital ID could perform message encryption, but to develop a concrete workflow, you were instructed to purchase a third party administrator's Digital ID for applying digital signatures and use custom self-signed certificates for message encryption. By following this approach, you can avoid ever being locked out of a PDF. Further, this approach also encourages safe distribution of the client certificate to decrypt the document.

Together with a Digital Signature, an understanding of the clean-up tools to remove and redact meta data and several other textual objects could significantly reduce the chances of reverse engineering the PDF to a CAD file. At a minimum, it would increase the amount of work on the hacker to re-vector all the raster objects. As a result, it will discourage malicious activities like reverse engineering. Moreover, if all else is not possible, send a raster image file of the drawing or PDF to completely remove the intelligent components from the file.

Along with the support site, the examples and software shown herein inform professionals who have not seen the capabilities of third party applications to exploit CAD and PDF files. With a

little knowledge and several videos showing each step discussed herein, you will have little difficulty developing your own standards for competent, secure delivery of PDF files.

APPENDIX A

Adobe Portable File

Certificate Security

The following items are directly from the help file located in the Adobe Portable File Software. Included are the relevant items relating to Digital IDs. Furthermore, to sign with a Digital Certificate with PDFs, you will need the Professional Version of Adobe Acrobat to sign with a certificate.

Once you obtain the software, you may use a certificate obtained by purchasing a Digital ID from VeriSign to encrypt documents and to verify a Digital Signature of an output CAD file into PDF. A Digital Signature assures recipients that the document came from you. Encryption ensures that only the intended recipient can view the contents. A certificate stores the public key component of a Digital ID. When you secure a PDF using a certificate, you specify the recipients and define the file access level for each recipient or group. For example, you can allow one client to sign with their certificate and fill forms and another to edit text or remove pages. You can choose certificates from your list of trusted identities, files on disk, or the Windows certificate store, which is native to Windows Operating Systems. Always include your certificate in the recipient list so that you can open the document later.

Adobe makes a disclaimer about encrypting documents using certificates from third party Digital IDs. If the certificate is lost or stolen, the issuing authority can replace it. If a self-signed Digital ID is deleted, all PDFs that were encrypted using the certificate from that ID are inaccessible forever so, be cautious using self-signed certificates. You do not want to be unable to open the document three years down the road.

Encrypt a PDF or PDF Portfolio with a Certificate

To encrypt many PDFs, use “Action Wizard” (File > Action Wizard) to apply a predefined sequence. Alternatively, select under Security Method Certificate Security. You can also save your certificate settings as a security policy and reuse it to encrypt PDFs.



Figure 1. Shows the certificate security method chosen from the drop down box.

1. For a single PDF or a component PDF in a PDF Portfolio, open the PDF. For a PDF Portfolio, open the PDF Portfolio and choose View > Portfolio > Cover Sheet.
2. Choose “Tools > Protection > Encrypt > Encrypt with Certificate.” If you do not see the Protection panel, see the instructions for adding panels at Task Panes.
3. At the prompt, click “Yes.”
4. In the Certificate Security Settings dialog box, select the document components to encrypt.
5. From the Encryption Algorithm menu, choose a rate of encryption, and then click “Next.”
6. The encryption algorithm and key size are version-specific. Recipients must have the corresponding version (or later) of Acrobat or Reader to decrypt and read the document.
7. If you select 128-bit AES, recipients must have Acrobat 7 or later or Reader 7 or later to open the document.
8. If you select 256-bit AES, Adobe Acrobat 9 or later or Adobe Reader 9 or later is required to open the document.
9. Create a recipient list for the encrypted PDF. Always include your own certificate in the recipient list so that you are able to open the document later.
10. Click “Search” to locate identities in a directory server or in your list of trusted identities.
11. Click “Browse” to locate the file that contains certificates of trusted identities.
12. To set printing and editing restrictions for the document, select “Recipients” from the list, and then click “Permissions.”
13. Click “Next” to review your settings, and then click “Finish.”

When a recipient opens the PDF or PDF Portfolio, the security settings you specified for that person are used.



Figure 2 Shows the installed Digital IDs within Adobe Acrobat.

Change Encryption Settings

1. Do one of the following:
 - o For a single PDF or a component PDF in a PDF Portfolio, open the PDF.
 - o For a PDF Portfolio, open the PDF Portfolio and choose “View > Portfolio > Cover Sheet.”
2. Select “Tools > Protection > More Protection > Security Properties.”
3. Click “Change Settings.”
4. Do any of the following then click “Next.”
 - o To encrypt different document components, select that option.
 - o To change the encryption algorithm, choose it from the menu.
5. Do any of the following:
 - o To check a trusted identity, select the recipient, and then click “Details.”
 - o To remove recipients, select one or more recipients, and then click “Remove.” Do not remove your own certificate unless you do not want access to the file using that certificate.
 - o To change permissions of recipients, select one or more recipients, and then click “Permissions.”
6. Click “Next” then click “Finish.” Click “OK” to close the Document Properties dialog box, and save the document to apply your changes.

Remove Encryption Settings


1. Do one of the following:
 - o For a single PDF or a component PDF in a PDF Portfolio, open the PDF.

- For a PDF Portfolio, open the PDF Portfolio and choose “View > Portfolio > Cover Sheet.”
2. Select “Tools > Protection > Encrypt > Remove.”
3. If prompted, type the permissions password. If you do not know the permissions password, contact the author of the PDF.

Sharing Certificates with Others

Businesses that use certificates for secure workflows often store certificates on a directory server that participants can search to expand their list of trusted identities.

When you receive a certificate from someone, you can add it to your list of trusted identities. You can set your trust settings to trust all Digital Signatures and certified documents created with a specific certificate. You can also import certificates from a certificate store, such as the Windows certificate store. A certificate store often contains numerous certificates issued by different certification authorities.

-  **Note:** third party security providers usually validate identities by using proprietary methods or they integrate their validation methods with Acrobat. If you use a third party security provider, see the documentation for the third party provider.

Get Certificates from Other Users

Certificates that you receive from others are stored in a list of trusted identities. This list resembles an address book and enables you to validate the signatures of these users on any documents you receive from them. Once you receive a certificate from a trusted client, usually the only step needed to install is to double click the file.

Request a Certificate from Another User

1. Do one of the following:
 - a. In Acrobat, choose “Tools > Sign & Certify > More Sign & Certify > Manage Trusted Identities.”
 - b. In Reader, choose “Edit > Protection > Manage Trusted Identities.”



Note: *If you do not see the “Sign & Certify” or “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Click “Request Contact.”
3. Type your name, e-mail address, and contact information.
4. To allow other users to add your certificate to their list of trusted identities, select “Include My Certificates.”
5. Select either “Email Request” or “Save Request As A File.” Then click “Next.”

6. Select the Digital ID file to use, and then click “Select.”
7. Do one of the following:
 - a. If the “Compose Email” dialog box appears, type the e-mail address of the person you are requesting a certificate from, and click “Email.” Send the e-mail message that appears, with the attached certificate, in the default e-mail application.
 - b. If the “Export Data As” dialog box appears, specify a name and location for the file, click “Save,” and then click “OK.”

Add a Certificate from e-Mail

When a contact sends a certificate to you in e-mail, it is displayed as an import/export methodology file attachment.

1. Double-click the e-mail attachment, and then click “Set Contact Trust” in the dialog box that appears.
2. On the “Trust” tab of the “Import Contact Settings” dialog box, select “Trust Options.”
 - o Select “Use This Certificate as a Trusted Root” only if it is required to validate a Digital Signature. Once you make a certificate a trust anchor, you prevent revocation checking on it (or any certificate in the chain).
 - o To allow actions that can be a security risk, click “Certified Documents” then select the options you want to allow:
 - a. Dynamic Content.
 - b. Includes FLV and SWF files as well as external links.
 - c. Embedded High Privilege JavaScript.
 - d. Trusts embedded scripts.
 - e. Privileged System Operations.
 - f. Includes networking, printing, and file access.
3. Click “OK” to view the import details, and then click “OK” again.

Add a Certificate from a Digital Signature in a PDF

You can safely add a certificate to your trusted identities from a signed PDF by first verifying the thumbprint with the originator or the certificate.

1. Open the PDF containing the signature.
2. Open the signature panel, and select the signature in the “Signatures” panel.
3. On the Options menu, click “Show Signature Properties,” and then click “Show Certificate.”

4. If the certificate is self-signed, contact the originator of the certificate to confirm that the thumbprint values on the “Details” tab are correct. Trust the certificate only if the values match the values of the originator.
5. Click the “Trust” tab, click “Add to Trusted Identities,” and click “OK.”
6. In the “Import Contact Settings” dialog box, specify trust options, and click “OK.”

Import Certificates using the Windows Certificate Wizard (Windows only)

If you use the Windows certificate store to organize your certificates, you can import certificates using a wizard in Windows Explorer. To import certificates, identify the file that contains the certificates and determine the file location.

1. In Windows Explorer, right-click the “Certificate” file and choose “Install PFX.”
2. Follow the onscreen instructions to add the certificate to the Windows certificate store.
3. If you are prompted to validate the certificate before installing it, note the MD5 digest and SHA1 digest values (thumbprint). Contact the originator of the certificate to confirm that the values are correct before you trust the certificate. Click “OK.”

Associate a Certificate with a Contact

If you have a contact that is not associated with a certificate, or you want to change the certificate associated with a contact, follow these steps. A contact must have at least one valid certificate to exchange encrypted PDFs.

1. Do one of the following:
 - In Acrobat, choose “Tools > Sign & Certify > More Sign & Certify > Manage Trusted Identities.”
 - In Reader, choose “Edit > Protection > Manage Trusted Identities.”



Note: If you do not see the “Sign & Certify” or “Protection” panel, see the instructions for adding panels at Task Panes.

2. Select the contact, and click “Details.”
3. Click “Associate Certificate.”
4. Select a certificate, and click “OK.” Click “OK” again.

Verify Information on a Certificate

The Certificate Viewer dialog box provides user attributes and other information about a certificate. When others import your certificate, they often want to check your “thumbprint” information against the information they receive with the certificate. (The thumbprint refers to

the MD5 digest and SHA1 digest values.) You can check certificate information for your Digital ID files or the ID files that you import.

The “Certificate Viewer” dialog box provides the following information:

- Certificate validation period.
- Intended use of the certificate.
- Certificate data, such as the serial number and public key method.

You can also check if the certificate authority has revoked the certificate. Certificates are typically revoked when an employee leaves the company or when security is compromised in some way.

Verify Your Own Certificate

1. Do one of the following:
 - In Acrobat, choose “Tools > Protection > More Protection > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Select your Digital ID, and then click “Certificate Details.”

Verify information on the Certificate of a Contact

1. Do one of the following:
 - In Acrobat, choose “Tools > Sign & Certify > More Sign & Certify > Manage Trusted Identities.”
 - In Reader, choose “Edit > Protection > Manage Trusted Identities.”

Note: If you do not see the “Sign & Certify” or “Protection” panel, see the instructions for adding panels at Task Panes.

2. Select the contact, and click “Details.”
3. Select the certificate name, and click “Show Certificate.”

Delete a Certificate from Trusted Identities

1. Do one of the following:
 - In Acrobat, choose “Tools > Sign & Certify > More Sign & Certify > Manage Trusted Identities.”

- In Reader, choose “Edit > Protection > Manage Trusted Identities.”



Note: *If you do not see the “Sign & Certify” or “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Choose “Certificates” from the “Display” menu.
3. Select the certificate, and click “Delete.”

Create a Self-Signed Digital ID

Sensitive transactions between businesses generally require an ID from a certificate authority rather than a self-signed one.

1. Do one of the following:
 - In Acrobat, choose “Tools > Sign & Certify > More Sign & Certify > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the “Sign & Certify” or “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Select “Digital IDs” on the left, and then click the “Add ID” button.
3. Select the option “A New Digital ID I Want To Create Now” then click “Next.”
4. Specify where to store the Digital ID and click “Next.”

New PKCS#12 Digital ID File

Stores the Digital ID information in a file that has the extension .pfx in Windows and .p12 in Mac OS. You can use the files interchangeably between operating systems. If you move a file from one operating system to another, Acrobat still recognizes it.

Windows Certificate Store (Windows only)

Stores the Digital ID to a common location from where other Windows applications can also retrieve it.

5. Type a name, email address, and other personal information for your Digital ID. When you certify or sign a document, the name appears in the “Signatures” panel and in the “Signature” field.
6. (Optional) To use Unicode values for extended characters, select “Enable Unicode Support” then specify “Unicode” values in the appropriate boxes.
7. Choose an option from the “Key Algorithm” menu. The 2048-bit RSA option offers more security than 1024-bit RSA, but 1024-bit RSA is more universally compatible.

8. From the “Use Digital ID For” menu, choose whether you want to use the Digital ID for signatures, data encryption, or both.
9. Type a password for the Digital ID file. For each keystroke, the password strength meter evaluates your password and indicates the password strength using color patterns. Reconfirm your password.

You can export and send your certificate file to contacts that can use it to validate your signature.

Note: Make a backup copy of your Digital ID file. If your Digital ID file is lost or corrupted, or if you forget your password, you cannot use that profile to add signatures.

Register a Digital ID

To use your Digital ID, register your ID with Acrobat or Reader.

1. Do one of the following:
 - In Acrobat, choose “Tools > Protection > More Protection > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Select “Digital IDs” on the left.
3. Click the “Add ID” button.
4. Select “My Existing Digital ID From” and choose one of the following options:

A File

Select this option if you obtained a Digital ID as an electronic file. Follow the prompts to select the Digital ID file, type your password, and add the Digital ID to the list.

A Roaming Digital ID Stored On a Server

Select this option to use a Digital ID that is stored on a signing server. When prompted, type the server name and URL where the roaming ID is located.

A Device Connected To This Computer

Select this option if you have a security token or hardware token connected to your computer.

5. Click “Next” follow the onscreen instructions to register your Digital ID.

Specify the Default Digital ID

To avoid being prompted to select a Digital ID each time you sign or certify a PDF, you can select a default Digital ID.

1. Do one of the following:
 - In Acrobat, choose “Tools > Protection > More Protection > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Click “Digital IDs” on the left, and then select the Digital ID you want to use as the default.
3. Click the “Usage Options” button, and choose a task for which you want the Digital ID as the default. To specify the Digital ID as the default for two tasks, click the “Usage Options” button again and select a second option.

A check mark appears next to selected options. If you select only the signing option, the “Sign” icon appears next to the Digital ID. If you select only the encryption option, the “Lock” icon appears. If you select only the certifying option, or if you select the signing and certifying options, the “Blue Ribbon” icon appears.

To clear a default Digital ID, repeat these steps, and deselect the usage options you selected.

Change the Password and Timeout for a Digital ID

Passwords and timeouts can be set for PKCS #12 IDs. If the PKCS #12 ID contains multiple IDs, configure the password and timeout at the file level.



Note: Self-signed Digital IDs expire in five years. After the expiration date, you can use the ID to open, but not sign or encrypt, a document.

1. Do one of the following:
 - In Acrobat, choose “Tools > Protection > More Protection > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the Protection panel, see the instructions for adding panels at Task Panes.*

2. Expand “Digital IDs” on the left, select “Digital ID Files,” and then select a “Digital ID” on the right.
3. Click the “Change Password” button. Type the old password and a new password. For each keystroke, the password strength meter evaluates your password and indicates the password strength using color patterns. Confirm the new password, and then click “OK.”
4. With the ID still selected, click the “Password Timeout” button.
5. Type the password, and click “OK.”

Be sure to back up your password in a secure place. If you lose your password, either create a new self-signed Digital ID and delete the old one, or purchase one from a third-party provider.

Delete your Digital ID

When you delete a Digital ID in Acrobat, you delete the actual PKCS #12 file that contains both the private key and the certificate. Before you delete your Digital ID, ensure that it is not in use by other programs or required by any documents for decrypting.



Note: You can only delete self-signed Digital IDs that you created in Acrobat. A Digital ID obtained from another provider cannot be deleted.

1. Do one of the following:
 - In Acrobat, choose “Tools > Protection > More Protection > Security Settings.”
 - In Reader, choose “Edit > Protection > Security Settings.”



Note: *If you do not see the “Protection” panel, see the instructions for adding panels at Task Panes.*

2. Select “Digital IDs” on the left, and then select the “Digital ID” to remove.
3. Click “Remove ID” then click “OK.”

Protecting Digital IDs

By protecting your Digital IDs, you can prevent unauthorized use of your private keys for signing or decrypting confidential documents. Ensure that you have a procedure in place in the event your Digital ID is lost or stolen.

How to Protect Your Digital IDs

When private keys are stored on hardware tokens, smart cards, and other hardware devices that are password- or PIN-protected, use a strong password or PIN. Never divulge your password to others. If you must, write down your password and store it in a secure location. Contact your

system administrator for guidelines on choosing a strong password. Keep your password strong by following these rules:

- Use eight or more characters.
- Mix uppercase and lowercase letters with numbers and special characters.
- Choose a password that is difficult to guess or hack, but that you can remember without having to write it down.
- Do not use a correctly spelled word in any language, as they are subject to “dictionary attacks” that can crack these passwords in minutes.
- Change your password on a regular basis.
- Contact your system administrator for guidelines on choosing a strong password.

To protect private keys stored in P12/PFX files, use a strong password and set your password timeout options appropriately. If using a P12 file to store private keys that you use for signing, use the default setting for password timeout option. This setting ensures that your password is always required. If you are using your P12 file to store private keys that are used to decrypt documents, make a backup copy of your private key or P12 file. You can use the backed up private key of P12 file to open encrypted documents if you lose your keys.

The mechanisms used to protect private keys stored in the Windows certificate store vary depending on the company that has provided the storage. Contact the provider to determine how to back up and protect these keys from unauthorized access. In general, use the strongest authentication mechanism available and create a strong password or PIN when possible.

What to do if a Digital ID is Lost or Stolen

If your Digital ID was issued by a certificate authority, immediately notify the certificate authority and request the revocation of your certificate. In addition, you should not use your private key.

If your Digital ID was self-issued, destroy the private key and notify anyone to whom you sent the corresponding public key (certificate).

Smart Cards and Hardware Tokens

A “smart card” looks like a credit card and stores your Digital ID on an embedded microprocessor chip. Use the Digital ID on a smart card to sign and decrypt documents on computers that can be connected to a smart card reader. Some smart card readers include a keypad for typing a Personal Identification Number (PIN).

Similarly, a “security hardware token” is a small, keychain-sized device that you can use to store Digital IDs and authentication data. You can access your Digital ID by connecting the token to a USB port on your computer or mobile device.

If you store your Digital ID on a smart card or hardware token, connect it to your device to use it for signing documents.

APPENDIX B

Glossary

Data Encipherment

The data encipherment bit is asserted when the subject public key is used for enciphering user data, other than cryptographic keys.

Decipher Only

The meaning of the decipher only bit is undefined in the absence of the key agreement bit. When the decipher only bit is asserted and the key agreement bit is also set, the subject public key may be used only for deciphering data while performing key agreement.

Extensions

The extensions defined for X.509 v3 certificates provide methods for associating additional attributes with users or public keys and for managing a certification hierarchy. The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities. Each extension in a certificate is designated as either critical or non-critical. A certificate using system **MUST** reject the certificate if it encounters a critical extension it does not recognize; however, a non-critical extension **MAY** be ignored if it is not recognized.

Encipher Only

The meaning of the encipher only bit is undefined in the absence of the key agreement bit. When the encipher only bit is asserted and the key agreement bit is also set, the subject public key may be used only for enciphering data while performing key agreement.

Key Usage

An DER extension to enforce the types of uses the key pair may be implemented. Keys may be modified to limit use for signing, encryption, and digital signatures.

Key Encipherment

The key encipherment bit is asserted when the subject public key is used for key transport. For example, when an RSA key is to be used for key management, then this bit is set.

Key Agreement

The key agreement bit is asserted when the subject public key is used for key agreement. For example, when a Diffie-Hellman key is to be used for key management, then this bit is set. The Diffie-Hellman Key is usually a symmetric key exchange.

Layered security

Also known as layered defense, describes the practice of combining multiple mitigating security controls to protect resources and data.

Roundtrip Protection

A term used to define the electronic path of a file is defined as the complete path from the sender to the receiver and back to the original sender.

APPENDIX C

Florida Digital Signature Standards for Surveying and Mapping

5J-17.062 Procedures for Signing and Sealing Electronically Transmitted Plans, Specifications, Reports or Other Documents. (BPR, 2010).

(1) Information stored in electronic files representing plans, specifications, plats, reports, or other documents that must be sealed under the provisions of Chapter 472, F.S., shall be signed, dated, and sealed by the professional surveyor and mapper in responsible charge.

(2) A license holder may use a computer generated representation of his or her seal on electronically conveyed work; however,

The final hard copy documents of such surveying or mapping work must contain an original signature and raised seal of the license holder and date or the documents must be accompanied by an electronic signature as described in this section. A scanned image of an original signature shall not be used in lieu of an original signature and raised seal or electronic signature. Surveying or mapping work that contains a computer generated seal shall be accompanied by the following text or similar wording: “The seal appearing on this document was authorized by [Example: Leslie H. Doe, P.E. 0112 on (date)]” unless accompanied by an electronic signature as described in this section.

(3) An electronic signature is a digital authentication process attached to or logically associated with an electronic document and

shall carry the same weight, authority, and effect as an original signature and raised seal. The electronic signature, which can be generated by using either public key infrastructure or signature dynamics technology, must be as follows:

(a) Unique to the person using it;

(b) Capable of verification;

(c) Under the sole control of the person using it;

(d) Linked to a document in such manner that the electronic signature is invalidated if any data in the document are changed.

(4) Alternatively, electronic files may be signed and sealed by creating a “signature” file that contains the surveyor and mapper’s name and PSM number, a brief overall description of the surveying and mapping documents, and a list of the electronic files to be sealed. Each file in the list shall be identified by its file name utilizing relative Uniform Resource Locators (URL) syntax described in the Internet Architecture Board’s Request for Comments (RFC) 1738,

December 1994, which is hereby adopted and incorporated by reference by the Board and can be obtained from the Internet Website: <ftp://ftp.isi.edu/in-notes/rfc1738.txt>. Each file shall have an authentication code defined as an SHA-1 message digest described in Federal Information Processing Standard Publication 180-1 “Secure Hash Standard,” 1995 April 17, which is hereby adopted and incorporated by reference by the Board and can be obtained from the Internet Website: <http://www.itl.nist.gov/fipspubs/fip180-1.htm>. A report shall be created that contains the surveyor and mapper’s name and PSM number, a brief overall description of the surveyor and mapper documents in question and the authentication code of the signature file. This report shall be printed and manually signed, dated, and sealed by the professional surveyor and mapper in responsible charge. The signature file is defined as sealed if its authentication code matches the authentication code on the printed, manually signed, dated and sealed report. Each electronic file listed in a sealed signature file is defined as sealed if the listed authentication code matches the file’s computed authentication code.

APPENDIX D

WORKS CITED

How Certificates Work. (2003, March 28). Retrieved Feb 25, 2013, from Microsoft Technet:
[http://technet.microsoft.com/en-us/library/cc776447\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc776447(v=ws.10).aspx)

How Digital Certificates are used for Digital Signatures and Message Encryption. (2005, 5 19).
Retrieved 2 23, 2013, from Microsoft Technet: [http://technet.microsoft.com/en-us/library/bb123848\(v=exchg.65\).aspx](http://technet.microsoft.com/en-us/library/bb123848(v=exchg.65).aspx)

Adobe Corporation. (n.d.). *Adobe LifeCycle Management.* Retrieved June 5, 2013, from Product lifecycle management: <http://www.adobe.com/manufacturing/plm.html>

Adobe Corporation. (n.d.). *Flatten and Clean Document.* Retrieved May 30, 2013, from Adobe Help Files:
http://help.adobe.com/en_US/acrobat/X/pro/using/WS58a04a822e3e50102bd615109794195ff-7c84.w.html

Adobe Corporation. (n.d.). *Photoshop Printing.* Retrieved May 10, 2013, from Adobe Help:
http://help.adobe.com/en_US/photoshop/cs/using/WSfd1234e1c4b69f30ea53e41001031ab64-7945a.html

Adobe Corporation. (n.d.). *Postscript Overview.* Retrieved May 20, 2013, from Adobe Postscript 3: <http://www.adobe.com/products/postscript/overview.html>

Adobe Corporation. (n.d.). *Saving files in graphics formats.* Retrieved June 2, 2013, from
http://help.adobe.com/en_US/photoshop/cs/using/WSEC964A47-477C-4487-8CF4-332F92636117a.html

Adobe Corporation. (n.d.). *Text and Objects.* Retrieved June 11, 2013, from Adobe Help Files :
http://help.adobe.com/en_US/acrobat/X/pro/using/WS58a04a822e3e50102bd615109794195ff-7c7d.w.html

Adobe. (n.d.). *Setting Up Signing.* Retrieved June 10, 2013, from Adobe:
http://help.adobe.com/en_US/acrobat/X/standard/using/WS396794562021d52e4605066e12b3464c4db-8000.html#WS58a04a822e3e50102bd615109794195ff-7d44.w

Arx CoSign. (n.d.). *Digital Signatures.* Retrieved June 2, 2013, from Arx.com:
www.arx.com/digital-signature

BaselineCorp. (n.d.). *Will QR Codes Replace Paper Building Permits.* Retrieved June 1, 2013, from From the Baseline: <http://www.baselinecorp.com/will-qr-codes-replace-paper-building-permits/>

- Blackberry. (n.d.). *Configure MDSCS to Connect to LDAP*. Retrieved May 12, 2013, from blackberry.com:
http://docs.blackberry.com/en/admin/deliverables/14437/Config_MDSCS_to_connect_LDAP_for_keys_922522_11.jsp
- BPR. (2010, Dec 21). *5J-17.051 Minimum Technical Standards: General Survey- Florida*. Retrieved Feb 28, 2013, from Minimum Technical Standards:
www.800helpfla.com/psm/pdfs/5J-17051.pdf
- Cray, S. R. (2013, 1). *Parity Bit*. Retrieved 2 21, 2013, from Wikipedia:
http://en.wikipedia.org/wiki/Parity_bit
- Cryptographic Hash Function*. (n.d.). Retrieved Feb 28, 2013, from Wikipedia:
http://en.wikipedia.org/wiki/Cryptographic_hash_function
- DocuSign. (n.d.). *Products*. Retrieved June 2, 2013, from DocuSign.com:
<http://www.docusign.com/>
- Fulton, W. (n.d.). *Scaling to print a different size*. Retrieved May 2, 2013, from Scantips:
<http://www.scantips.com/basics2c.html>
- GlobalSign. (n.d.). *SSL & Digital Certificates*. Retrieved June 8, 2013, from Globalsign.com:
<https://www.globalsign.com/>
- Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. (n.d.). Retrieved June 10, 2013, from rfc-base: <http://www.rfc-base.org/txt/rfc-3280.txt>
- LockLizard Limited. (n.d.). *Products*. Retrieved June 3, 2013, from locklizard.com:
<http://www.locklizard.com/>
- Oracle. (n.d.). *keytool - Key and Certificate Management Tool*. Retrieved May 4, 2013, from Oracle: <http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html>
- ORC Inc. (n.d.). *Importing Certificate from Backup File*. Retrieved June 2, 2013, from ECA Documents: http://eca.orc.com/wp-content/uploads/ECA_Docs/IE_Instructions/Importing_Cert_from_Backup_file.pdf
- Troy Corporation. (2009). *TROY_Microprint_Whitepaper_011609.pdf*. Whelington, WV: Troy Group, Inc.

