PDF/A Competence Center

TechNote 0006: Digital Signatures in PDF/A-1

Digital signatures are primarily used to check the integrity of the signed part of the document. They also can be used to authenticate the signer's identity and determine the time of signing. The concept of digital signatures was introduced in PDF 1.3 and thus is part of the ISO 19005-1 standard.

PDF 1.5, 1.6, 1.7 documents can also be PDF/A-1 conformant if they meet the requirements of the standard. This applies in particular to digital signatures, where for example the appearance stream must obey the rules of the standard whereas the cryptographic message syntax may conform to newer versions. Therefore higher versions than 1.4 are mentioned in this TechNote where appropriate.

PDF Reference 1.4 defines how digital signatures are to be embedded into a document. There are aspects of the digital signature that are impacted by the PDF/A-1 standard, e.g. fonts and colors. However, the standard doesn't make any statements about the semantics, i.e. on how signatures are created and validated. The semantics of digital signatures is left up to the corresponding signature handlers which are uniquely identified by registered names. Furthermore, PDF/A-1 does not require that conforming readers be able to validate digital signatures.

The main purpose of this TechNote is to help manufacturers of PDF/A-1 conforming producers to correctly embed digital signatures. Discussions about the structure of the signature value (cryptographic message syntax) and the long term quality of specific signature techniques are beyond the scope of this TechNote.

In order to verify the statements made here, a test implementation of a signature handler has been used to create, embed and validate a PDF/A-1 conforming digital signature. The PDF/A-1 conformance has been tested using Acrobat 8.0 Preflight. The signature validation has been tested using the signature handler plug-ins of Acrobat 5.0, 6.0, 7.0 and 8.0.

1 Digital Signatures in PDF

1.1 Signature Types

A PDF document may contain the following defined types of signatures:

- One or more document signatures. These signatures appear in signature form fields.
- At most one MDP (modification detection and prevention) signature (PDF 1.5), also referred to as an author or certifying signature.
- At most two usage rights signatures (PDF 1.5). This type of signature is not embedded via a form field as described below. Furthermore usage rights signatures do not have an appearance.

1.2 Signature Embedding

A signature is embedded as a form field for which the field type **FT** is **Sig.** The form field value **V** is a signature dictionary containing the signature in its **Contents** entry. A signature field is described by a dictionary containing entries pertaining to a widget annotation as well as a form field. The annotation rectangle **Rect** in the dictionary gives the position of the form field on its page. Signature fields that are not intended to be visible should have an annotation rectangle that has zero height and width. The appearance dictionary **AP** of a signature field's annotation defines the field's visual appearance on the page.

1.3 Signature Handler

The specific form of creation and validation of a signature is implemented by a signature handler. The value of the **Filter** entry in the signature dictionary is the name of the preferred signature handler to validate the signature. Signatures are created by computing a digest of the data in a document, as specified by the **ByteRange** entry in the signature dictionary, and storing the encrypted digest in the **Contents** entry. To verify the signature, the digest is recomputed and compared with the one stored in the **Contents** entry. Differences in the digest values indicate that modifications have been made since the document was signed.

1.4 Digest Techniques

The PDF Reference 1.5 offers two techniques to compute a digest of the signed part of the document: A byte range digest and an object digest. Document signatures must use a byte range digest. The object digest technique was not de-

signed for document signatures. Object digests have been officially deprecated by Adobe, and are not included in the current draft for ISO 32000.

1.5 Incremental Update

Modifications to a document, after it has been signed, should be saved as incremental updates. This method preserves the consistency of the signed data with the digest and allows for recreating the state of the document as it existed at the time of signing. Be aware, however, that use of this method may cause a viewer to mark the signature as invalid, depending on the changes made.

1.6 Multiple Signatures

A document not only can contain more than one signature field but can also be signed again (after it has been saved) using the incremental update feature.

2 PDF/A Requirements

The following sections describe the minimum requirements to embed a PDF/A-1 conforming digital signature.

2.1 Signature Types

Only document signatures can be used in PDF/A-1 conforming documents. MDP and usage rights signatures contain either dictionary entries which are not defined in PDF Reference 1.4 or are referenced from undefined dictionary entries. A PDF/A-1 conformant viewer will treat them as private data and will not act on them.

2.2 Dictionaries and required Entries

Since digital signatures in PDF are comprised of form fields and widget annotations, their describing dictionaries must conform to the following sections of ISO 19005-1:

- The annotation dictionary (which is the same as the signature field dictionary) must conform to section 6.5 and 6.6 and requires the entries **Subtype**, **Rect** and **AP**.
- The signature field dictionary must conform to sections 6.6.2, 6.9 and requires the entries **FT** and **V**.
- The appearance dictionary must conform to section 6.5.3 and requires the **N** entry.
- The signature dictionary requires the **Filter**, **Contents** and **ByteRange** entries.
- The interactive form dictionary must conform to section 6.9 and requires the **Fields** entry.
- The document catalog dictionary must conform to sections 6.6.2, 6.8.4, 6.7.2, 6.1.11, 6.1.13, 6.2.2, 6.8.3.3 and requires the **AcroForm** entry.

2.3 Signed Document Parts

The **ByteRange** entry contains an array of pairs of integers describing the parts of the document for which the digest is calculated. The **ByteRange** array must exclude the string value (including the string delimiters) of the **Contents** entry.

2.4 Visual Appearance

The annotation dictionary requires an appearance dictionary (**AP**), whose normal appearance (**N**) is a Form XObject, that describes the visual appearance of the signature. The visual appearance must also exist for an invisible signature but may be empty. The content stream of the Form XObject must conform to section 6.2.10 of ISO 19005-1. In addition, all requirements from 6.2.3 through 6.2.9 as well as 6.3 and 6.4 shall apply.

2.5 Optional Dictionary Entries

Signature handlers are free to use or omit those entries that are marked optional in PDF Reference 1.4 to 1.7, but are encouraged to use them in the described way if they are used at all.

2.6 Private Dictionary Entries

Specific signature handlers may add private entries of their own. To avoid name duplication, it is suggested that the keys for all such private entries be prefixed with the registered handler name followed by a period (.).

3 Recommendations for Signatures in PDF/A-1

The following recommendations apply to public key signature handlers only and are addressed to writers of vendor specific signature handlers. Although these recommendations are not part of the PDF/A-1 standard, implementers are encouraged to follow them in order to avoid security issues, obtain upward compatibility with PDF Reference 1.5–1.7 and maximize interoperability between security handlers from different vendors.

3.1 Signature Interoperability

The **SubFilter** entry in the signature dictionary specifies the encoding of the signature value and key information, and the **Filter** entry specifies the preferred handler to use to validate the signature. For maximum interoperability between signature handlers from different vendors, only the **SubFilter** values **adbe.pkcs7.detached** and **adbe.pkcs7.sha1**, as defined in PDF Reference 1.5, should be used.

3.2 Contents String

The **Contents** entry should contain the signature as a hex string, starting with **<** and ending with **>**.

3.3 Visual Appearance

It is highly recommended to structure the appearance streams according to Adobe's technical note *Digital Signature Appearances*, version for Acrobat 6 (see Bibliography). For security reasons only the layers **n0** and **n2** as well as the **DSz** entry in XObject dictionary of the **AcroForm**'s resources (**DR**) should be used. Specifically the **n1**, **n3** and **n4** layers as well as the **DSx** and **DSy** entries, as described in Adobe's earlier technical note *Digital Signature Appearances*, version for Acrobat 5, must not be used.

3.4 Cryptographic Message Syntax

Only a DER-encoded PKCS#7 binary data object, conforming to RFC 3852, should be used as the value of the signature dictionary's **Contents** entry thus making the PKCS#1 binary data object and the X.509 certificate chain stored in the **Cert** entry obsolete.

3.5 Digest Algorithm

Due to known security weaknesses the MD5 algorithm should not be used to calculate the digest of the signed part of the document. SHA-1 should be used instead.

3.6 Time stamp

The time of signing, if it is generated in a verifiable way from a secure time server, should be included in the signature value, i.e. embedded in the PKCS#7 binary data object. The time stamp token must conform to RFC 3161 and must be computed and embedded into the PKCS#7 object as described in Appendix A of RFC 3161. A normal unverified computer time should preferably be embedded into the PKCS#7 object or can alternatively be stored in the **M** entry of the signature dictionary.

3.7 Revocation Information

If needed, only Online Certificate Status Protocol (OCSP) responses, described in RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP)* should be included in the PKCS#7 object. Creators should refrain from embedding certificate revocation lists (CRLs) into the PKCS#7 object in order to keep file sizes small.

3.8 Trust Chain

The PKCS#7 object should contain the signer's certificate and all issuer certificates from the signer's trust chain. The signer's certificate should be the first certificate in the PKCS#7 object.

3.9 Document XMP Metadata

PDF/A conformant signature tools should record the signing process as an action in the **xmpMM:History** entry in the document's XMP metadata. The **softwareAgent** field should be specified, and the action field should be specified as **signed**.

4 Summary

The table below summarizes the required or recommended values of the relevant dictionary entries if one or more digital signatures are embedded.

Dictionary	Entry	PDF/A-1 conformance	Recommendation
Catalog	AcroForm	required	
AcroForm	Fields	required	
AcroForm	SigFlags	optional	3 (integer)
AcroForm/DR/XObject	DSx	private	Don't use
AcroForm/DR/XObject	DSy	private	Don't use
AcroForm/DR/XObject	DSz	private	Can be used
Annotation	Subtype	required	
Annotation	Rect	required	
Annotation	AP	required	
Annotation	AS	optional	Don't use
Field	FT	required	
Field	Parent	required if a field hierarchy is used	Don't use
Field	Kids	required if a field hierarchy is used	Don't use
Field	V	required	
Field	DV	optional	Don't use
Signature	Filter	required	Name of the signature handler to use to validate the signature
Signature	SubFilter	optional	adbe.pkcs7.sha1 adbe.pkcs7.detached
Signature	Contents	required	DER-encoded PKCS#7 hex string; don't use MD5
Signature	Cert	private	Don't use
Signature	ByteRange	required	
Signature	Reference	private	Don't use
Signature	Name	optional	Use the subject name of the PKCS#7 object
Signature	M	optional	Unverified time stamp if not embedded in the PKCS#7 object
Appearance	N	required	

Bibliography

[1] ISO 19005-1: Document management — Electronic document file format for long-term preservation — Part 1: Use of PDF 1.4 (PDF/A-1)
www.iso.ch

[2] Adobe Systems Incorporated: PDF Reference third edition, Adobe Portable Document Format Version 1.4
www.adobe.com/devnet/pdf/pdfs/PDFReference.pdf

[3] Adobe Systems Incorporated: PDF Reference fourth edition, Adobe Portable Document Format Version 1.5
www.adobe.com/devnet/pdf/pdfs/PDFReference15_v6.pdf

[4] Adobe Systems Incorporated: PDF Reference fifth edition, Adobe Portable Document Format Version 1.6
www.adobe.com/devnet/pdf/pdfs/PDFReference16.pdf

[5] Adobe Systems Incorporated: PDF Reference sixth edition, Adobe Portable Document Format Version 1.7
www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf

[6] Adobe Systems Incorporated (October 2006): Digital Signature Appearances
www.adobe.com/devnet/acrobat/pdfs/PPKAppearances.pdf

[7] Adobe Systems Incorporated (March 2006): Digital Signatures in the PDF Language
www.adobe.com/devnet/acrobat/pdfs/DigitalSignaturesInPDF.pdf

[8] ITU-T X.690 (07/2002): SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS, OSI networking and system aspects – Abstract Syntax Notation One (ASN.1)
www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

[9] ISO/IEC 8825-1 (2002): Information technology - ASN.1 encoding rules: specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
www.iso.ch

[10] IETF RFC 2315: PKCS #7: Cryptographic Message Syntax, Version 1.5
www.ietf.org/rfc/rfc2315.txt

[11] IETF RFC 2630: Cryptographic Message Syntax
www.ietf.org/rfc/rfc2630.txt

[12] IETF RFC 3852: Cryptographic Message Syntax (CMS)
www.ietf.org/rfc/rfc3852.txt

[13] IETF RFC 3039: Internet X.509 Public Key Infrastructure, Qualified Certificates Profile
www.ietf.org/rfc/rfc3039.txt

[14] IETF RFC 3280: Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile
www.ietf.org/rfc/rfc3280.txt

[15] IETF RFC 2560: Internet X.509 Public Key Infrastructure: Online Certificate Status Protocol - OCSP
www.ietf.org/rfc/rfc2560.txt

[16] IETF RFC 3161: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)
www.ietf.org/rfc/rfc3161.txt

[17] IETF RFC 1321: The MD5 Message-Digest Algorithm
www.ietf.org/rfc/rfc1321.txt

[18] IETF RFC 3174: US Secure Hash Algorithm 1 (SHA1)
www.ietf.org/rfc/rfc3174.txt

Copyright and Usage

Status of this Document

2007-09-20 First released version

2008-03-14 Update:

- Updated formatting and added references to the entries in the bibliography