INTERNET-DRAFT
Intended Status: Informational
Obsoletes RFC 5485 (once approved)
Expires: 11 November 2016

R. Housley
Vigil Security

11 May 2016

Digital Signatures on RFC and Internet-Draft Documents
<draft-housley-rfc-and-id-signatures-02.txt>

Abstract

This document specifies the conventions for digital signatures on RFCs and Internet-Draft documents. For Internet-Drafts, the Cryptographic Message Syntax (CMS) is used to create a detached signature, which is stored in a separate companion file so that no existing utilities are impacted by the addition of the digital signature. For RFCs, an embedded digital signature is included in Portable Document Format (PDF) files types in addition to the detached signature in a separate companion file.

This document (once approved) obsoletes RFC 5485.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on11 November 2016.

1. Introduction

This document specifies the conventions for digital signatures on RFCs and Internet-Draft documents. For Internet-Drafts, the Cryptographic Message Syntax (CMS) [CMS] is used to create a detached signature, which is stored in a separate companion file so that no existing utilities are impacted by the addition of the digital signature. For RFCs, an embedded digital signature is included in Portable Document Format (PDF) [PDF] files types in addition to the detached signature in a separate companion file.

This document (once approved) obsoletes RFC 5485 [IDSIG], which contains the conventions that have been used by IETF Secretariat to digitally sign Internet-Drafts for the past few years.

The digital signature allows anyone to confirm that the contents of the RFC or Internet-Draft have not been altered since the time that the document was signed.

For RFCs, the RFC Production Center [RFCED] will generate the digital signature as the final step before passing the completed documents to the RFC Publisher.

For Internet-Drafts, the IETF Secretariat will generate the digital signature shortly after the Internet-Draft is posted in the repository.

The signature of the RFC Editor or the IETF Secretariat is intended to provide a straightforward way for anyone to determine whether a particular file contains the document that was made available by the RFC Editor or the IETF Secretariat. The signing-time associated with the signature provides the wall clock time at which the signature was generate; it is not intended to provide a trusted timestamp.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [STDWORDS].

1.2. ASN.1

The CMS uses Abstract Syntax Notation One (ASN.1) [X.680]. ASN.1 is a formal notation used for describing data protocols, regardless of the programming language used by the implementation. Encoding rules describe how the values defined in ASN.1 will be represented for transmission. The Basic Encoding Rules (BER) [X.690] are the most widely employed rule set, but they offer more than one way to represent data structures. For example, definite length encoding and indefinite length encoding are supported. This flexibility is not desirable when digital signatures are used. As a result, the Distinguished Encoding Rules (DER) [X.690] were invented. DER is a subset of BER that ensures a single way to represent a given value. For example, DER always employs definite length encoding.

2. Detached Signature Files

Detached digital signature files are created, and the name of the file directly identifies the RFC or Internet-Draft that is signed.

All RFC file names begin with "rfc". The next portion of the file name contains a unique integer assigned by the RFC Production Center. For example, rfc20.txt contains a document produced in October 1969. Some repositories contain this same document with a file name of rfc0020.txt.

All Internet-Draft file names begin with "draft-". The next portion of the file name depends on the source of the document. For example, documents from IETF working groups usually have "ietf-" followed by the working group abbreviation, and this is followed by a string that helps people figure out the subject of the document.

All Internet-Draft file names end with a hyphen followed by a two digit version number and a suffix. All RFC file names end with a suffix. The suffix indicates the type of file. For example, a plain text file will have a suffix of ".txt". Today, plain text files are the most common, but the RFC Editor has announced plans to make use of other formats [RFCSERIES]. Each file format employs a different suffix.

The companion signature file has exactly the same file name as the RFC or Internet-Draft, except that ".p7s" is added to the end. This file name suffix conforms to the conventions in [MSG]. Here are a few example names:

```
              RFC: rfc8765.txt
   Signature File: rfc8765.txt.p7s
```

RFC: rfc8765.xml
Signature File: rfc8765.xml.p7s

RFC: rfc8765.pdf
Signature File: rfc8765.pdf.p7s

RFC: rfc8765.html
Signature File: rfc8765.html.p7s

Internet-Draft: draft-ietf-example-widgets-03.txt
Signature File: draft-ietf-example-widgets-03.txt.p7s

Internet-Draft: draft-ietf-example-widgets-03.ps
Signature File: draft-ietf-example-widgets-03.ps.p7s

Internet-Draft: draft-housley-internet-draft-sig-file-00.txt
Signature File: draft-housley-internet-draft-sig-file-00.txt.p7s

2.1. Need for Canonicalization

In general, the content of the RFC or Internet-Draft is treated like a single octet string for the generation of the digital signature. Unfortunately, the plain text and HTML files require canonicalization to avoid signature validation problems. The primary concern is the manner in which different operating systems indicate the end of a line of text. Some systems use a single new-line character, other systems use the combination of the carriage-return character followed by a line-feed character, and other systems use fixed-length records padded with space characters. For the digital signature to validate properly, a single convention must be employed.

2.2. Plain Text and HTML Canonicalization

The canonicalization procedure follows the conventions used for text files in the File Transfer Protocol (FTP) [FTP]. Such files must be supported by FTP implementations, so code reuse seems likely.

The canonicalization procedure converts the data from its internal character representation to the standard 8-bit NVT-ASCII representation (see TELNET [TELNET]). In accordance with the NVT standard, the <CRLF> sequence MUST be used to denote the end of a line of text. Using the standard NVT-ASCII representation means that data MUST be interpreted as 8-bit bytes.

Trailing space characters MUST NOT appear on a line of text. That is, the space character must not be followed by the <CRLF> sequence. Thus, a blank line is represented solely by the <CRLF> sequence.

The form-feed nonprintable character (0x0C) is expected in RFCs and Internet-Drafts. Other nonprintable characters, such as tab and backspace, are not expected, but they do occur. For robustness, any nonprintable or non-ASCII characters (ones outside the range 0x20 to 0x7E) MUST NOT be changed in any way not covered by the rules for end-of-line handling in the previous paragraph.

Trailing blank lines MUST NOT appear at the end of the file. That is, the file must not end with multiple consecutive <CRLF> sequences.

Any end-of-file marker used by an operating system is not considered to be part of the file content. When present, such end-of-file markers MUST NOT be processed by the digital signature algorithm.

Note: This text file canonicalization procedure is consistent with the NVT-ASCII definition offered in Appendix B of RFC 5198 [UFNI].

2.3. XML File Canonicalization

Utilities that produce XML files are expected to follow the guidance provided by the World Wide Web Consortium (W3C) in Section 2.11 of [R20060816]. If this guidance is followed, no canonicalization is needed.

A robust signature generation process MAY perform canonicalization to ensure that the W3C guidance has been followed. This guidance says that a <LF> character MUST be used to denote the end of a line of text within a XML file. Therefore, any two-character <CRLF> sequence and any <CR> that is not followed by <LF> are to be translated to a single <LF> character.

2.4. No Canonicalization of Other File Formats

No canonicalization is needed for file formats currently used or planned for RFCs and Internet-Drafts other than plain text files and XML files. Other file formats are treated as a simple sequence of octets by the digital signature algorithm.

3. Signed PDF Files

PDF [PDF] has supported digital signatures since PDF 1.2. The embedded signature covers the document content and embedded content. The RFC Editor plans to use this feature to include the XML that was used to produce the PDF covered by the signature. Authors of Internet-Drafts might do this as well, but they are not required to do so.

The IETF Secretariat will generate detached signature files for

Internet-Drafts that are posted in PDF format. If an author has embedded a digital signature in the PDF file before posting it, then the author's signature will remain in the PDF file.

The RFC Production Center will embedded a digital signature in the PDF file and also generate a detached signature file for RFCs before passing them to the RFC Publisher for posting.

4. CMS Profile

The CMS is used to construct the detached signatures for RFCs and Internet-Drafts. The CMS ContentInfo content type MUST always be present, and it MUST encapsulate the CMS SignedData content type. Since a detached signature is being created, the CMS SignedData content type MUST NOT encapsulate the RFC or Internet-Draft. The CMS detached signature is summarized by:

```
ContentInfo {
  contentType          id-signedData, -- (1.2.840.113549.1.7.2)
  content              SignedData
}

SignedData {
  version              CMSVersion, -- Always set to 3
  digestAlgorithms     DigestAlgorithmIdentifiers,
  encapContentInfo     EncapsulatedContentInfo,
  certificates         CertificateSet, -- Secretariat certificate(s)
  crls                 CertificateRevocationLists, -- Optional
  signerInfos          SET OF SignerInfo -- Only one signer
}

SignerInfo {
  version              CMSVersion, -- Always set to 3
  sid                  SignerIdentifier,
  digestAlgorithm      DigestAlgorithmIdentifier,
  signedAttrs          SignedAttributes, -- Always present
  signatureAlgorithm   SignatureAlgorithmIdentifier,
  signature            SignatureValue,
  unsignedAttrs        UnsignedAttributes -- Optional
}

EncapsulatedContentInfo {
  eContentType         id-ct-asciiTextWithCRLF,
                                  -- (1.2.840.113549.1.9.16.1.27)
  eContent             OCTET STRING  -- Always absent
}
```

4.1. ContentInfo

The CMS requires the outer-most encapsulation to be ContentInfo [CMS]. The fields of ContentInfo are used as follows:

contentType
: indicates the type of the associated content, and for the detached RFC or Internet-Draft signature file, the encapsulated type is always SignedData, so the id-signedData (1.2.840.113549.1.7.2) object identifier MUST be present in this field.

content
: holds the content, and for the detached RFC or Internet-Draft signature file, the content is always a SignedData content.

4.2. SignedData

The SignedData content type [CMS] contains the signature of the RFC or Internet-Draft and information to aid in the validation of that signature. The fields of SignedData are used as follows:

version
: is the syntax version number, and for this specification, the version number MUST be set to 3.

digestAlgorithms
: is a collection of one-way hash function identifiers. It MUST contain the identifier used by the RFC Production Center or the IETF Secretariat to generate the digital signature. See the discussion of digestAlgorithm in Section 4.2.1.

encapContentInfo
: is the signed content, including a content type identifier. Since a detached signature is being created, it does not encapsulate the RFC or Internet-Draft. The use of the EncapsulatedContentInfo type is discussed further in Section 4.2.2.

certificates
: is an optional collection of certificates. It SHOULD include the X.509 certificate needed to validate the digital signature value. Certification Authority (CA) certificates and end entity certificates MUST conform to the certificate profile specified in [PKIX1].

crls
: is an optional collection of certificate revocation lists (CRLs). It SHOULD NOT include any CRLs; however, any CRLs that are present MUST conform to the CRL profile specified in [PKIX1].

signerInfos
: is a collection of per-signer information, and for this specification, each item in the collection must represent the IETF Secretariat. More than one SignerInfo MAY appear to facilitate transitions between keys or algorithms. The use of the SignerInfo type is discussed further in Section 4.2.1.

4.2.1. SignerInfo

The RFC Editor or the IETF Secretariat is represented in the SignerInfo type. The fields of SignerInfo are used as follows:

version
: is the syntax version number. In this specification, the version MUST be set to 3.

sid
: identifies the public key of the RFC Editor or IETF Secretariat. In this specification, the subjectKeyIdentifier alternative is always used, which identifies the public key directly. This identifier MUST match the value included in the subjectKeyIdentifier certificate extension in the certificate of the RFC Editor or the IETF Secretariat.

digestAlgorithm
: identifies the one-way hash function, and any associated parameters, used by the RFC Production Center or the IETF Secretariat to generate the digital signature.

signedAttrs
: is an optional set of attributes that are signed along with the content. The signedAttrs are optional in the CMS, but signedAttrs is required by this specification. The SET OF Attribute must be encoded with the distinguished encoding rules (DER) [X.690]. Section 4.2.3 of this specification lists the signed attributes that MUST be included in the collection. Other signed attributes MAY also be included.

signatureAlgorithm
: identifies the digital signature algorithm, and any associated parameters, used by the RFC Production Center or the IETF Secretariat to generate the digital signature.

signature
: is the digital signature value generated by the RFC Production Center or the IETF Secretariat.

unsignedAttrs
: is an optional set of attributes that are not signed. Unsigned attributes are usually omitted; however, the unsigned attributes MAY hold a trusted timestamp generated in accordance with [TSP]. Section 2.2.4 of [TSP] provides more information about this unsigned attribute.

4.2.2. EncapsulatedContentInfo

The EncapsulatedContentInfo structure contains a content type identifier. Since a detached signature is being created, it does not encapsulate the RFC or Internet-Draft. The fields of EncapsulatedContentInfo are used as follows:

eContentType
: is an object identifier that uniquely specifies the content type. The content type associated with the plain text file MUST be id-ct-asciiTextWithCRLF. The appropriate content type for each format is discussed in Section 5 of this specification. Additional file formats can be added if the Internet community chooses.

eContent
: is optional. When an encapsulated signature is generated, the content to be signed is carried in this field. Since a detached signature is being created, eContent MUST be absent.

4.2.3. Signed Attributes

The RFC Production Center or IETF Secretariat MUST digitally sign a collection of attributes along with the RFC or Internet-Draft. Each attribute in the collection MUST be DER-encoded. The syntax for attributes is defined in [X.501], and the X.500 Directory provides a rich attribute syntax. A very simple subset of this syntax is used extensively in [CMS], where ATTRIBUTE.&Type and ATTRIBUTE.&id are the only parts of the ATTRIBUTE class that are employed.

Each of the attributes used with this CMS profile has a single attribute value. Even though the syntax is defined as a SET OF AttributeValue, there MUST be exactly one instance of AttributeValue present.

The SignedAttributes syntax within signerInfo is defined as a SET OF Attribute. The SignedAttributes MUST include only one instance of any particular attribute.

The RFC Production Center or the IETF Secretariat MUST include the content-type, message-digest, and signing-time attributes. The RFC Production Center or the IETF Secretariat MAY also include the binary-signing-time signed attribute as well as any other attribute that is deemed appropriate. The intent is to allow additional signed attributes to be included if a future need is identified. This does not cause an interoperability concern because unrecognized signed attributes are ignored at verification.

4.2.3.1. Content-Type Attribute

A content-type attribute is required to contain the same object identifier as the content type contained in the EncapsulatedContentInfo. The appropriate content type for each format is discussed in Section 5. The RFC Production Center or IETF Secretariat MUST include a content-type attribute containing the appropriate content type. Section 11.1 of [CMS] defines the content-type attribute.

4.2.3.2. Message-Digest Attribute

The RFC Production Center or IETF Secretariat MUST include a message-digest attribute, having as its value the output of a one-way hash function computed on the RFC or Internet-Draft that is being signed. Section 11.2 of [CMS] defines the message-digest attribute.

4.2.3.3. Signing-Time Attribute

The RFC Production Center or IETF Secretariat MUST include a signing-time attribute, specifying the time, based on the local system clock, at which the digital signature was applied to the RFC or Internet-Draft.

The IETF Secretariat may choose to perform signatures in batches, therefore the signing-time may be several hours or days after the time that the Internet-Draft was actually posted.

The RFC Production Center will generate the digital signature before passing the document to the RFC Publisher, therefore the signing-time will be shortly before the time that the RFC is made available in the repository.

Section 11.3 of [CMS] defines the content-type attribute.

4.2.3.4. Binary-Signing-Time Attribute

The RFC Production Center or IETF Secretariat MAY include a binary-signing-time attribute, specifying the time at which the digital signature was applied to the RFC or Internet-Draft. If present, the time that is represented MUST match the time represented in the signing-time attribute. The binary-signing-time attribute is defined in [BinTime].

4.2.3.5. Signing-Certificate-Version2 Attribute

The RFC Production Center or IETF Secretariat MAY include a signing-certificate-version2 attribute, specifying which certificate is to be used to validate the digital signature was applied to the RFC or Internet-Draft. If present, the certs field of the attribute MUST contain the list of certificates that are to be used in validating the RFC or Internet-Draft, and the optional policies field of the attribute MUST be absent. The first certificate identified in the the certs field of the attribute MUST be the certificate to be used to verify the signature on the the RFC or Internet-Draft. If more than one certificate identifier is present, the subsequent certificate identifiers MUST limit certificates that are acceptable during certification path validation. The signing-certificate-version2 attribute is defined in [ESSU].

4.2.4. Unsigned Attributes

Unsigned attributes are usually omitted. However, an unsigned attribute MAY hold a trusted timestamp generated in accordance with [TSP]. The idea is to time-stamp the RFC Production Center or the IETF Secretariat digital signature to prove that it was created before a given time. If the certificate of the RFC Editor or the IETF Secretariat is revoked the time stamp allows a verifier to know whether the signature was created before or after the revocation date. Appendix A of [TSP] defines the signature time-stamp attribute that can be used to time-stamp a digital signature.

5. Content Types

This section lists the content types that are used in this specification. The eContentType field as described in Section 4.2.2 contains a content type identifier, and the same value appears in the content-type attribute as described in Section 4.2.3.1.

The following table lists the file formats and the associated content type.

File Format	Content Type
Plain text	id-ct-asciiTextWithCRLF
Extensible Markup Language (XML)	id-ct-xml
Portable Document Format (PDF)	id-ct-pdf
PostScript	id-ct-postscript
HyperText Markup Language (HTML)	id-ct-htmlWithCRLF

The object identifiers associated with the content types listed in the above table are:

```
id-ct  OBJECT IDENTIFIER  ::= { iso(1) member-body(2)
     us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) 1 }

id-ct-asciiTextWithCRLF  OBJECT IDENTIFIER  ::= { id-ct 27 }

id-ct-xml  OBJECT IDENTIFIER  ::= { id-ct 28 }

id-ct-pdf  OBJECT IDENTIFIER  ::= { id-ct 29 }

id-ct-postscript  OBJECT IDENTIFIER  ::= { id-ct 30 }

id-ct-htmlWithCRLF  OBJECT IDENTIFIER  ::= { id-ct <TBD1> }
```

6. IANA Considerations

Please assign and object identifier for id-ct-htmlWithCRLF in the SMI Security for S/MIME CMS Content Type registry.

7. Security Considerations

The RFC Production Center and the IETF Secretariat MUST protect their private keys. The use of a hardware security module (HSM) is RECOMMENDED because compromise of these private keys permits masquerade.

The RFC Production Center currently maintains staff at a more than one location. This situation requires an HSM at each location where signatures will be generated. However, the HSMs do not need to use the same signing key. Each HSM can have a different signing key, as long as each one has their own certificate.

The IETF Secretariat currently maintain servers at a primary location and a backup location. This configuration requires two HSMs, one at each location. However, the two HSMs do not need to use the same

signing key. Each HSM can have a different signing key, as long as each one has their own certificate.

The generation of a public/private key pair for signature operations relies on random number generation. The use of an inadequate pseudo-random number generator (PRNG) can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the key pair, searching the resulting small set of possibilities, rather than brute force searching the whole private key space. The generation of quality random numbers is difficult, but [RANDOM] offers important guidance in this area.

The RFC Series Editor and the IETF Secretariat should be aware that cryptographic algorithms become weaker with time. As new cryptanalysis techniques are developed and computing performance improves, the work factor to break a particular digital signature algorithm or one-way hash function will be reduced. Therefore, it SHOULD be possible to migrate these algorithms. That is, the RFC Series Editor and the IETF Secretariat SHOULD be prepared for the supported algorithms to change over time.

The IETF Secretariat must take care to use the correct time in signing-time and binary-signing-time attributes. The inclusion of a date within the Internet-Draft by the authors that is shortly before the signing time attributes supplied by the IETF Secretariat provide confidence about the date that the Internet-Draft was posted to the repository. However, the IETF Secretariat may choose to perform signatures in batches, and the signing-time may be several hours or days after the time that the Internet-Draft was actually posted.

The RFC Production Center may choose to sign RFCs in small batches just before the documents are passed to the RFC Publisher. This allows a single HSM to be used at one location, even if the documents are edited at different locations, and it allows the HSM to be off-line except when signatures are being generated. Further, this allows the RFC Production Center to include manual steps, such as entering a HSM passphrase or inserting a smartcard, as part of the signing procedure to improve operations security.

The IETF Secretariat may choose to sign Internet-Drafts in batches. This allows a single HSM to be used if multiple servers are located in one geographic location, and it allows the HSM to be off-line except when signatures are being generated. Further, this allows the IETF Secretariat to include manual steps, such as entering a HSM passphrase or inserting a smartcard, as part of the signing procedure to improve operations security.

8. Deployment and Operational Considerations

The private keys used to generate the RFC Production Center and the IETF Secretariat signatures ought to be stored in a HSM to provide protection from unauthorized disclosure. While the HSMs will be operated by the RFC Production Center and IETF Secretariat, they ought to be owned by the IETF Trust. Accordingly, the Trustees of the IETF Trust should designate an appropriate certification authority to issue a certificate to the RFC Editor and the IETF Secretariat, and they should approve any procedures used by the RFC Production Center and the IETF Secretariat for signing documents consistent with this specification.

9. Design Rationale

A detached signature is used for all file formats. In addition, RFCs in PDF format are also signed with an embedded signature.

PDF has a widely deployed way of handling digital signatures, and the tools for verifying the embedded PDF digital signatures are freely available.

Other file formats do not have widely deployed file-format-specific ways of handling digital signatures. Use of the detached signature provides a single way to sign RFCs and Internet-Drafts that is easy to implement using freely available tools. In addition, if an Internet-Draft author includes a signature using a file-format-specific approach, the IETF Secretariat signature does not harm it in any way.

File names provide a straightforward linkage between the document and the detached signature file. A CMS signed attribute could have been specified to include another form of linkage, and this could be added in the future. At this point in time, it is important to support signature validation of expired Internet-Drafts regardless of the way that they are obtained. Therefore, the appropriate value for such a signed attribute is unclear. This specification allows an Internet-Draft and companion signature file to be stored anywhere without hindering signature validation.

10. Normative References

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[PKIX1] Cooper, D., Santesson, s., Farrell, S., Boeyen, s., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[PDF] ISO, "Portable document format -- Part 1: PDF 1.7", ISO 32000-1, 2008.

[STDWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[X.680] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.

[X.690] ITU-T Recommendation X.690 (2002) | 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).

11. Informative References

[BinTime] Housley, R., "BinaryTime: An Alternate Format for Representing Date and Time in ASN.1", RFC 4049, April 2005.

[ESSU] Schaad, J., "Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility", RFC 5035, August 2007.

[FTP] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[IDSIG] Housley, R., "Digital Signatures on Internet-Draft Documents", RFC 5485, March 2009.

[MSG] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[OpenSSL] http://www.openssl.org/

[R20060816] Bray, T., J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, 16 August 2006. http://www.w3.org/TR/2006/REC-xml-20060816.

[RANDOM] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Recommendations for Security", BCP 106, RFC 4086, June 2005.

[RFCED] Kolkman, O., and J. Halpern, "RFC Editor Model (Version 2)", RFC 6635, June 2012.

[RFCSERIES] Flanagan, H., and N. Brownlee, "RFC Series Format Requirements and Future Development", RFC 6949, May 2013.

[TELNET] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983.

[TSP] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

[UFNI] J. Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[X.501] ITU-T Recommendation X.501: Information Technology - Open Systems Interconnection - The Directory: Models, 1993.

12. Acknowledgements

The idea for the Internet-Draft signature file came from a discussion with Scott Bradner at IETF 69 in Chicago, IL, USA. Many helpful suggestions came from Jim Schaad, Pasi Eronen, and Chris Newman in the creation of [IDSIG]. Glen Barney played a vital role in implementing Internet-Draft signatures as specified in [IDSIG].

The IETF Secretariat has been generating digital signatures for many years. Recently, the RFC Series Editor, Heather Flanagan, decided that the RFC Production Center should sign RFCs before they are posted by the RFC Publisher. In addition, as part of the format changes that are underway [RFCED], the decision was made to take advantage of the native digital signature capabilities available in PDF.

Many thanks for Joe Hildebrand, Stefan Santesson, and Robert Sparks for their insightful suggestions on this document.

Appendix: A

OpenSSL 0.9.9 (and later versions) [OpenSSL] includes an implementation of CMS. The following command line can be used to verify a detached signature on a RFC or Internet-Draft:

```
openssl cms -verify -CAfile <cert-file> -content <signed-doc> /
     -inform DER -in <p7s-file> -out /dev/null
```

The arguments need to be provided as follows:

<cert-file>
: the name of the file containing the trust anchor, which is typically the self-signed certificate of the certification authority that issued a certificate to the RFC Editor or the IETF Secretariat.

<signed-doc>
: the name of the file containing the RFC or Internet-Draft after canonicalization.

<p7s-file>
: the name of the file containing the detached signature that was generated in accordance with this specification.

Author's Address

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: housley@vigilsec.com