NUMBER 6 — JULY 16, 1997

RSA Laboratories' Bulletin

News and advice on data security and cryptography

# Extensions and Revisions to PKCS #7

**Burton S. Kaliski Jr.**
*RSA Laboratoties*

**Kevin W. Kingdon**
*RSA Data Security, Inc.*

## Introduction

The Public-Key Cryptography Standards (PKCS) are offered by RSA Laboratories to promote the development of secure application and other standards based on public-key cryptography. First published in 1991, PKCS has become widely implemented and referenced, and a significant amount of experience is now available to assist the development of related formal standards, as well as for the improvement of PKCS itself.

PKCS #7, "Cryptographic Message Syntax Standard," provides a particular example of the PKCS process at work. PKCS #7, now at version 1.5, defines the syntax for several kinds of cryptographically protected messages, including encrypted messages and messages with digital signatures. Originally an outgrowth of Internet Privacy-Enhanced Mail [1], PKCS #7 has become the basis for the now widely implemented S/MIME secure electronic mail specification [2]. But its applications have not been limited to mail; PKCS #7 has also become a basis for message security in systems as diverse as the Secure Electronic Transaction (SET) specifications for bank card payments [3], the W3C Digital Signature Initiative [4], and PKCS #12, "Personal Information Exchange Syntax Standard" [5]. These and other applications have provided significant experience with PKCS #7, including implementation guidelines, profiles, and various extensions.

RSA Laboratories is now taking steps toward a second major version of PKCS #7, version 2.0, building on the experience just described. A variety of improvements are expected to be covered in the revision, representing experience gained with version 1.5, a proposed alignment with the next version of the Secure Data Network System Message Security Protocol [6], and general improvements in cryptography and key management over the last several years. The development of version 2.0, like that of version 1.5 and its precursors, will be done through workshops and other forms of public review; development is expected to continue through the rest of the year.

As there is benefit in documenting a few extensions to PKCS #7 in the meantime, RSA Laboratories also plans to publish a minor revision of PKCS #7, version 1.6. The primary application of the revision is to support the SET specifications. The revision is also intended to "close" the version 1.x series, providing a stable base during development of the version 2.0 specification and any applications based on it. We consider that the introduction of PKCS #7 version 1.6, rather than introducing further incompatibilities, will stabilize a number of potentially divergent alternatives and extensions, thereby increasing interoperability. There are other alternatives still to be aligned; these will be covered by version 2.0.

*Burt Kaliski is chief scientist at RSA Laboratories and Kevin Kingdon is a technical director at RSA Data Security, Inc. They can be reached at burt@rsa.com and kevin@rsa.com.*

RSA Laboratories
A Division of RSA Data Security

Version 1.5 and version 1.6 syntax are distinguished by a version number field, and the versions should be considered alternate forms of cryptographic message protection. Applications supporting either version may be considered to conform with PKCS #7. Existing applications based on version 1.5, such as S/MIME, need not be upgraded to version 1.6. Likewise, version 1.6 applications need not support version 1.5 syntax.

RSA Laboratories is also publishing an ASN.1 module in updated 1994 ASN.1 syntax that includes syntax for version 1.5 as well as version 1.6. The module is available from RSA Laboratories' PKCS Web page, `http://www.rsa.com/rsalabs/pubs/PKCS`.

The remainder of this note summarizes the changes that RSA Laboratories will publish in version 1.6 of PKCS #7, and gives an overview of potential revisions for version 2.0.

## Version 1.6

This section describes the differences between version 1.5 of the standard and version 1.6. The information in this section, together with the version 1.5 standard document and the ASN.1 module mentioned previously, provides the developer with enough information to implement version 1.6. RSA Laboratories plans to publish a consolidated document for version 1.6 of the standard subsequent to the release of this bulletin.

- *Exports all symbols*. Version 1.5 exported one type, `ContentInfo`, as well as the various object identifiers. Version 1.6 exports all symbols. This permits applications to make direct use of other data types without having to wrap them in an outer `ContentInfo`.

- *Permits additional content types*. Version 1.5 was unclear about whether the set of content types defined in the standard could be extended by the application. Version 1.6 definitely permits applications to extend the set of content types to include application-specific types, identified by application-specific object identifiers.

- *Encodes lists as SEQUENCE OF instead of SET OF*. Some applications use DER encoding consistently throughout the application. In these applications, the overhead of sorting SET/SET OF contents (as required by DER) may be undesirable. Version 1.6 converts all occurrences of SET OF to SEQUENCE OF, eliminating the need for sorting in the encoding process.

- *Removes support for PKCS #6 extended certificates*. X.509 version 3 introduced an extension mechanism that effectively removes the need for a separate extended certificate standard. Version 1.6 permits the use of X.509 version 3 certificates, and precludes the use of PKCS #6 extended certificates.

- *Uses EXPLICIT tag for authenticated attributes list*. Version 1.5 used an `IMPLICIT` tag for the optional `authenticatedAttributes` field of `SignerInfo`. Version 1.6 uses an `EXPLICIT` tag so that an application can encode the underlying `SEQUENCE OF Attribute`, digest the encoding, and reuse the encoding as the contents octets of the explicitly-tagged field. Version 1.5 applications were required either to encode the attribute list twice, once with the implicit tag and once without, or to hack the results of the encoding.

- *Simplifies cryptographic processing of contents*. Version 1.5 required the application to 'dip under' the ASN.1 and deal directly with the BER/DER encoding. Message digests were done on the content octets of the DER encoding of the `content` field. Encryption was performed on the content octets of a definite-length BER encoding of the `content` field. This was done primarily for compatibility with PEM. Such BER/DER 'hacking' makes it difficult for users of ASN.1 compilers to generate encoding/decoding subroutines. Alternatives, such as wrapping the contents in an octet string, reduce the level of type-safety that can be specified in the ASN.1. Given the ascendancy of S/MIME over PEM, and the desirability of avoiding low-level 'hacking' of the BER/DER encoding, version 1.6 has modified the processing rules to operate on the entire BER/DER encoding of the `content` field.

This last change requires some additional explanation in order to be as unambiguous as possible. The affected paragraphs from version 1.5 are included here, modified to specify version 1.6's cryptographic processing rules:

### Section 7, Note 2:

When a `ContentInfo` value is the inner content of signed-data, signed-and-enveloped-data, or digested-data `content`, a message digest algorithm is applied to the octets of the entire DER encoding of the content field. When a `ContentInfo` value is the inner content of enveloped-data or signed-and-enveloped data content, a content-encryption algorithm is applied to the octets of the entire BER or DER encoding of the `content` field.

### Section 9.3 (in its entirety)

The message-digesting process computes a message digest on either the content being signed or the content together with the signer's authenticated attributes. In either case, the initial input to the message-digesting process is the entire content being signed. Specifically, the initial input is the octets of the entire DER encoding of the `content` field of the `contentInfo` value to which the signing process is applied. All of the octets of the DER encoding of that field are digested, including the identifier and length octets.

The result of the message-digesting process (which is called, informally, the "message digest") depends on whether the `authenticatedAttributes` field is present. When the field is absent, the result is just the message digest of the content. When the field is present, however, the result is the message digest of the complete DER encoding of the `SEQUENCE OF Attribute` contained in the `authenticatedAttributes` field. For clarity: The [2] `EXPLICIT` tag is not part of the `SEQUENCE OF Attribute` value. The `SEQUENCE OF Attribute` tag (DER value 30) is to be digested along with the length and contents octets of the encoded sequence. Since the `authenticatedAttributes` value, when the field is present, must contain as attributes the content type and the message digest of the content, those values are indirectly included in the result.

When the content being signed has content type `data` and the `authenticatedAttributes` field is absent, then just the DER-encoded `data` is digested. This implies that the length of the content being signed must be known in advance of the digesting process. This method is not compatible with Privacy-Enhanced Mail.

**Note.** The fact that the message digest is computed on the DER encoding of the content does not mean that DER is the required method of representing that part for data transfer. Indeed, it is expected that some implementations of this standard may store objects in other than their DER encodings, but such practices do not affect message-digest computation.

### Section 9.5 (in its entirety)

PKCS #7 version 1.6 breaks compatibility with PEM. Applications for which PEM compatibility is required may continue to use PKCS #7 version 1.5.

### Section 10.3 (replaces first three paragraphs)

The input to the content-encryption process is the entire `content` being enveloped. Specifically, the input is the octets of the entire BER encoding of the content field of the `ContentInfo` value to which the enveloping process is applied. All of the octets of the encoding are encrypted, including the identifier and length octets.

This process implies that the length of the content being encrypted must be known in advance of the encryption process. This method is not compatible with Privacy-Enhanced Mail.

### Section 11.3 (in its entirety)

Version 1.6 of this standard does not attempt to maintain compatibility with PEM.

## Version 2.0

The following are a few of the potential revisions that will be considered in PKCS #7 version 2.0:

- *Algorithm independence*. While version 1.5 was intended to support arbitrary cryptographic algorithms, a number of technical issues make such support somewhat difficult. An example of issue is the "DigestInfo" construction during signature generation, which appends an algorithm identifier to a message digest prior to a signature operation with a private key. It presents a difficulty for signature schemes that do not have a comparable step, such as the Digital Signature Standard [7]. Version 2.0 is intended to be more naturally algorithm-independent.

- *More flexible key identification*. PKCS #7 version 1.5 follows the key management model of Privacy-Enhanced Mail, where a public key is identified by the issuer and serial number of a public-key certificate. There are certainly other ways of identifying public keys, such as ones based on the key owner's name, and these will be considered in version 2.0. Version 1.5, like PEM, is also based on X.509 certificates, although unlike PEM it does not assume a particular certificate hierarchy. Alternative certificate systems may also be considered in version 2.0.

Related to this, it may be appropriate to consider more flexible forms of key management for both symmetric and public keys in version 2.0. In version 1.5, the distribution of both types of keys occurs “in-line,” being transmitted along with the message. It may be worthwhile to consider a more general approach, where the key management syntax is separate from the message processing syntax, so that, for instance, there is provision for distributing keys that are not associated with a particular message.

- *Additional cryptographic transformations*. PKCS #7 currently provides syntax for digitally signed messages, encrypted messages with public-key-based key management, and messages to which a message digest has been appended. For version 2.0, encrypted messages with symmetric-key-based key management and messages with symmetric-key authentication codes will also be considered, as each has important applications in secure communications. (The appropriateness of further symmetric-key-based techniques in a “public-key cryptography standard” is a worthwhile topic to discuss. One motivation is that all kinds of cryptographic techniques have a role in a security system, even one based to a large extent on public-key concepts, and it is helpful to describe them all in a common document.)

Other suggestions for improvement to PKCS #7 are most welcome. The development of PKCS #7 version 2.0 was launched by RSA Laboratories at a PKCS workshop in June and is expected to continue throughout the year. Further information on the development of version 2.0 has been posted on RSA Laboratories’ PKCS Web page and sent to the `<pkcs-tng@rsa.com>` mailing list.

## Conclusion

A significant amount of experience is now available to assist the improvement of PKCS #7, “Cryptographic Message Syntax Standard.” RSA Laboratories is taking steps toward a second major version, and is also publishing a minor revision, version 1.6. Other PKCS documents will benefit from a similar process, and further information on other revisions will be announced on RSA Laboratories’ PKCS Web page. For more information, please contact `<pkcs-editor@rsa.com>`.

## Acknowledgements

The authors gratefully acknowledge the contributions of Phil Griffin and Terry Hayes to version 1.6 of this standard.

## References

[1] J. Linn. *RFC 1421: Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*. February 1993.

[2] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, L. Repka. *S/MIME Message Specification*. March 1997.

[3] MasterCard and Visa. *Secure Electronic Transactions (SET) Specification DRAFT for Testing*. June 1996.

[4] World Wide Web Consortium. W3C Digital Signature Initiative. `http://www.w3.org/pub/WWW/Security/DSig`.

[5] RSA Laboratories. *PKCS #12: Personal Information Exchange Syntax Standard Version 1.0 DRAFT.* April 1997.

[6] National Security Agency (NSA) MISSI Program Office. *SDN. 701: Secure Data Network System Message Security Protocol 4.0*. February 1996.

[7] National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard (DSS)*. May 1994.

For more information on this and other recent security developments, contact RSA Laboratories at one of the addresses below.

**RSA Laboratories**
100 Marine Parkway, Suite 500
Redwood City, CA  94065 USA
415/595-7703
415/595-4126 (fax)
*rsa-labs@rsa.com*
*http://www.rsa.com/rsalabs/*