S/MIME Working Group
Internet Draft
Document: draft-ietf-smime-pss-03.txt
Category: Standards

J Schaad
Soaring Hawk Consulting

December 2003

# Use of the RSA PSS Signature Algorithm in CMS

## Status of this Memo



## Abstract

This document specifies the conventions for using the RSA Probabilistic Signature Scheme (RSASSA-PSS) digital signature algorithm with the Cryptographic Message Syntax (CMS).

## Conventions used in this document

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. Overview

This document specifies the conventions for using the RSASSA-PSS (RSA

Signature Scheme with Appendix - Probabilistic Signature Scheme) [P1v2.1] digital signature algorithm with the Cryptographic Message Syntax [CMS] signed-data content type.



CMS values are generated using ASN.1 [X.208-88], using the Basic Encoding Rules (BER) [X.209-88] and the Distinguished Encoding Rules (DER) [X.509-88].

This document is written to be used in conjunction with RFC XXX [RSA-ALGS]. All of the ASN.1 structures referenced in this document are defined in RFC XXX.

## 1.1 PSS Algorithm

Although there are no known defects with the PKCS #1 v1.5 [P1v1.5] signature algorithm, RSASSA-PSS [P1v2.1] was developed in an effort to have more mathematically provable security. PKCS #1 v1.5 signatures were developed in an ad hoc manner, RSASSA-PSS was developed based on mathematical foundations.

# 2. Algorithm Identifiers and Parameters

## 2.1 Certificate Identifiers

The RSASSA-PSS signature algorithm is defined in RFC 3447 [P1v2.1]. Conventions for encoding the public key are defined in RFC XXX [RSA-ALGS].

Two algorithm identifiers for RSA subject public keys in certificates are used. These are:

```
rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1 }
```

and

```
id-RSASSA-PSS OBJECT IDENTIFIER ::= { pkcs-1 10 }
```

When the rsaEncryption algorithm identifier is used for a public key, the AlgorithmIdentifier parameters field MUST contain NULL. Complete details can be found in [RSA-ALGS].

When the id-RSASSA-PSS algorithm identifier is used for a public key, the AlgorithmIdentifier parameters field MUST either be absent or contain RSASSA-PSS-params. Again, complete details can be found in [RSA-ALGS].

In both cases, the RSA public key, which is composed of a modulus and a public exponent, MUST be encoded using the RSAPublicKey type. The output of this encoding is carried in the certificate subject public key.

```
RSAPublicKey ::= SEQUENCE {
   modulus INTEGER, -- n
   publicExponent INTEGER } -- e
```

## 2.2 Signature Identifiers



The algorithm identifier for RSASAA-PSS signatures is:

```
id-RSASSA-PSS OBJECT IDENTIFER ::= {pkcs-1 10 }
```

When the id-RSASSA-PSS algorithm identifier is used for a signature, the AlgorithmIdentifier parameters field MUST contain RSASSA-PSS-params. Information about RSASSA-PSS-params can be found in [RSA-ALGS].

When signing, the RSA algorithm generates a single value, and that value is used directly as the signature value.

# 3. Signed-data Conventions

digestAlgorithms SHOULD contain the one-way hash function used to compute the message digest on the eContent value.

The same one-way hash function SHOULD be used for computing the message digest on both the eContent and the signedAttributes value if signedAttributes exist.

The same one-way hash function MUST be used for computing the message digest on the signedAttributes and as the hashAlgorithm in the RSA-PSS-params structure.

signatureAlgorithm MUST contain id-RSASSA-PSS. The algorithm parameters field MUST contain RSASSA-PSS-params.

signature contains the single value resulting from the signing operation.

If the subjectPublicKeyInfo algorithm identifier for the public key in the certificate is id-RSASSA-PSS and the parameters field is

present, the following additional steps MUST be done as part of signature validation:

1. The hashAlgorithm field in the certificate subjectPublicKey.algorithm parameters and the signatureAlgorithm parameters MUST be the same.
2. The maskGenAlgorithm field in the certificate subjectPublicKey.algorithm parameters and the signatureAlgorithm parameters MUST be the same.
3. The saltLength in the signatureAlgorithm parameters MUST be greater or equal to the saltLength in the certificate subjectPublicKey.algorithm parameters.
4. The trailerField in the certificate subjectPublicKey.algorithm parameters and signatureAlgorithm parameters MUST be the same.

In doing the above comparisons, default values are considered to be the same as extant values. If any of the above four steps is not true, the signature checking algorithm MUST fail validation.



## 4. Security Considerations

Implementations must protect the RSA private key. Compromise of the RSA private key may result in the ability to forge signatures.

The generation of RSA private key relies on random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate these values can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult. RFC 1750 [RANDOM] offers important guidance in this area.

Using the same private key for different algorithms has the potential

of allowing an attacker to get extra information about the key. It is strongly suggested that the same key not be used for both the PKCS

#1 v1.5 and RSASSA-PSS signature algorithms.

When computing signatures, the same hash function should be used for all operations. This reduces the number of failure points in the signature process.

The parameter checking procedures outlined in section 3 are of

special importance. It is possible to forge signatures by changing (especially to weaker values) these parameter values. Signers using this algorithm should take care that only one set of parameter values

is used as this decreases the possibility of leaking information.

## 5. Normative References


CMS Housley, R, "Cryptographic Message Syntax", RFC 3369, August 2002.

P1v2.1 Jonsson, J., and B. Kaliski, "PKCS #1: RSA Cryptography Specification Version 2.1", RFC 3447, February 2003.

RSA-ALGS Schaad, J., B. Kaliski and R Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", draft-ietf-pkix-rsa-pkalgs-01.txt, November 2003.

STDWORDS S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

X.208-88 CCITT Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1), 1998.





X.209-88 CCITT Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), 1988.

X.509-88 CCITT Recommendation X.509: The Directory Authentication Framework, 1988.


## 6. Informational References


P1v1.5 Kaliski, B. and J. Staddon, "PKCS #1: RSA Encryption, Version 2.0, RFC 2437, October 1998.

PKALGS Polk, W, R Housley, L. Bassham, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

RANDOM  Eastlake, D., S. Crocker and J. Schiller
"Randomness Recommendations for Security",
RFC 1750, December 1994.

## 7. Author's Address

Jim Schaad
Soaring Hawk Consulting
PO Box 675
Gold Bar, WA 98251

Email: jimsch@exmsft.com

## Full Copyright Statement