Internet Draft
draft-ietf-smime-sigattr-01
July 2, 1998
Expires in six months

Jim Schaad
Microsoft

Signing Certificate Attribute Specification

Status of this memo

Abstract

A set of attacks on SignedData objects have been identified relating to the fact that the certificate to be used when verifying the signature in a SignerInfo is not cryptographically bound into the signature. This leads to a set of attacks where the certificate used to verify the signature is altered leading to possible incorrect results. This document describes these attacks and provides ways to deal with some of the attacks.

This draft is being discussed on the "ietf-smime" mailing list. To join the list, send a message to <ietf-smime-request@imc.org> with the single word "subscribe" in the body of the message. Also, there is a Web site for the mailing list at <http://www.imc.org/ietf-smime>.

1. Introduction

Concerns have been raised over the fact that the certificate which the signer of a [CMS] SignedData object desired to be bound into the verification process of the SignedData object is not cryptographically bound into the signature itself. This draft addresses this issue by creating a new attribute to be placed in the signedAttributes or authenticated attributes section of a SignerInfo object.

This document presents a description of a set of possible attacks dealing with the substitution of one certificate to verify the signature for the desired certificate. A set of ways for preventing

or addressing these attacks is presented to deal with the simplest of the attacks.

Attribute certificates can be used as part of a signature verification process. As [CMS] currently stands there is no way to include the list of attribute certificates to be used in the verification process. The set of attribute certificates used in the signature verification process needs to have the ability for the signer to restrict the set of certificates. This information needs to be encoded in a manner that is covered by the signature on the SignedData object. This document allows for the set of attribute certificates to be listed as part of the signing certificate attribute.

Throughout this draft, the terms MUST, MUST NOT, SHOULD, and SHOULD NOT are used in capital letters. This conforms to the definitions in [MUSTSHOULD]. [MUSTSHOULD] defines the use of these key words to help make the intent of standards track documents as clear as possible. The same key words are used in this document to help implementers achieve interoperability.

2. Attack Descriptions

I have identified 3 different attacks that can be launched against a possible signature verification process by changing the certificate(s) used in the signature verification process.

Substitution Attack

The first attack deals with simple substitution of one certificate for another certificate. In this attack, the issuer and serial number in the SignerInfo is modified to refer to a new certificate. This new certificate is used the signature verification process.

The first version of this attack is a simple denial of service attack where an invalid certificate is substituted for the valid certificate. This renders the message unverifiable, as the public key in the certificate no longer matches the private key used to sign the message

The second version is a substitution of one valid certificate for the original valid certificate where the public keys in the certificates match. This allows the signature to be validated under potentially different certificate constraints than the originator of the message used

Reissue of Certificate

The second attack deals with a Certificate Authority re-issuing the signing certificate (or potentially one of its certificates). This attack may start becoming more frequent as Certificate Authorities re-issue there own root certificates and change policies in the certificate while doing so. This problem also occurs when cross

certificates (with potentially different restrictions) are used in the process of verifying a signature.

Rogue Duplicate Certificate Authority

The third attack deals with a rogue entity setting up a certificate authority that attempts to duplicate the structure of an existing Certificate Authority. Specifically, the rogue entity issues a new certificate with the same public keys as the signer used, but signed by the rogue entity's private key.

3. Attack Responses

This document does not attempt to solve all of the above attacks, however a brief description of responses to each of the attacks is given in this section.

Substitution Attack

The denial of service attack cannot be prevented, once the certificate identifier has been modified in transit no verification of the signature is possible. There is no way to automatically identify the attack either, it is indistinguishable from a message corruption.

The substitution of a valid certificate can be responded to in two different manners. The first is to make a blanket statement that the use of the same public key in two different certificates is bad practice and should be avoided. In practice, there is no practical way to prevent users from doing this and we need to assume that they will. Section 4 provides a new attribute to be included in the SignerInfo signed attributes. This binds the correct certificate identifier into the signature. This will convert the attack from a potentially successful one to a denial of service attack.

Reissue of Certificate

A Certificate Authority should never reissue a certificate with different attributes. Certificate Authorities that do so are following incorrect practices and cannot be relied on. Using the hash of the certificate as the reference to the certificate prevents this attach for end-entity certificates.

Preventing the attack based on reissuing of CA certificates would require a substantial change the attribute presented in section 5. It would require that a sequence of certificate identifiers be included in the attribute. This presents problem under the circumstances where the relying party is using a cross certificate as part of its authentication process and this certificate does not appear on the list of certificates. The problems outside of a closed PKI make the addition of this information prone to error causing the rejection of valid chains.

Rogue Duplicate Certificate Authority

The best method of preventing this attack is to avoid trusting the rogue certificate authority. The use of the hash to identify certificates prevents the use of end-entity certificates from the rogue authority, however the only true way to prevent this attack is to never trust the rough CA.

4. Attribute Certificates

Attribute certificates are required to do validation of signatures for some applications. This requires that the application be able to find the correct attribute certificates to perform the verification process and there is currently no list of attribute certificates in a SignerInfo object. The sender has the ability to include a set of attribute certificates in a SignedData object. The receiver has the ability to retrieve attribute certificates from a directory service. There are some circumstances where the signer may wish to limit the set of attribute certificates that may be used in verifying a signature. It would be useful to be able to list the set of attribute certificates the signer wants used in validating the signature.

Given that this attribute is dealing with the certificate used in verifying the signature on a SignerInfo object, it makes sense that it should also address the issue of limiting the set of attribute certificates as well.

5. Certificate Identification

The best way to identify certificates is an often discussed issued. [CMS] has imposed a restriction for SignedData objects that the issuer DN must be present in all signing certificates. The issuer/serial number pair is therefore sufficient to identify the correct signing certificate. This information is already present, as part of the SignerInfo object, and duplication of this information would be unfortunate. A hash of the entire certificate serves the same function (allowing the receiver to very the same certificate is being used), is smaller and permits a detection of the simple substitution attacks.

Attribute certificates do not have an issuer/serial number pair represented anywhere in a SignerInfo object. When the certificate is not included in the SignedData object, it becomes much more difficult to get the correct set of certificates based only on a hash of the certificate. For this reasons attribute certificates are identified by the issuer/serial number pair.

This document defines a certificate identifier as:

```
CertID ::=  SEQUENCE {
```

```
        certHash                 Hash,
        issuerAndSerialNumber    IssuerAndSerialNumber OPTIONAL
    }

    Hash ::= OCTET STRING -- SHA1 hash of entire certificate
```

When creating a CertID, the certHash is computed over the entire DER encoded certificate including the signature. The issuerAndSerialNumber would normally be present unless the value can be inferred from other information.

6. Signing Certificate Attribute

The signing certificate attribute is designed to prevent the simple substitution and re-issue attacks and to allow for a restricted set of attribute certificates to be used in verifying a signature.

The following object identifier identifies the encrypted-data content type:

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

The definition of SigningCertificate is

```
     SigningCertificate ::=  SEQUENCE OF CertID
```

The first certificate in the sequence of certificates MUST be the certificate used to verify the signature. The encoding of the CertID for this certificate SHOULD NOT include the issuerAndSerialNumber. (The issuerAndSerialNumber is already present in the SignerInfo.) This certificate is used during the signature verification process. If the hash of the certificate does not match the certificate used to decode the signature, the signature MUST be considered invalid.

If more than one certificate is present in the sequence of CertIDs, the certificates after the first one limit the set of attribute certificates that are used during signature validation. The issuerAndSerialNumber SHOULD be present, unless the validator is expected to have easy access to all certificates required. If only the signing certificate is present (a single item in the sequence) there are no restrictions on the set of attribute certificates used in validating the signature.

If present, the SigningCertificate attribute MUST be an authenticated attribute; it MUST NOT be an unauthenticated attribute. CMS defines authenticatedAttributes as a SET OF AuthAttribute. A SignerInfo MUST NOT include multiple instances of the SigningCertificate attribute. CMS defines the ASN.1 syntax for the authenticated attributes to include attrValues SET OF AttributeValue. A SigningCertificate

attribute MUST only include a single instance of AttributeValue.
There MUST NOT be zero or multiple instances of AttributeValue present
in the attrValues SET OF AttributeValue.

A. ASN.1 Module

```
SigningCertificate
     { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
       smime(16) modules(0) sca(?) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

EXPORTS
     SigningCertificate

IMPORTS
    -- Cryptographic Message Syntax
        IssuerAndSerialNumber
        FROM CryptographicMessageSyntax { iso(1) member-body(2)
             us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16)
             modules(0) cms(1) };

CertID ::=  SEQUENCE {
   certHash              Hash,
     issuerAndSerialNumber    IssuerAndSerialNumber OPTIONAL
}

Hash ::= OCTET STRING -- SHA1 hash of entire certificate

SigningCertificate ::=  SEQUENCE OF CertID

END
```

B. Open Issues

Should CertID be a CHOICE rather than a SEQUENCE? This makes the
encoded bytes smaller, but does mean that you can't have both the hash
and the issuer/serial number together. My personal feelings is that
the ability to have both outweighs the small overhead of the sequence.

C. Changes

On request from Russ, I have added the ability to refer to attribute
certificates from this attribute.
Allow for using hashes as well as issuer/serial number for referring
to certificates.
Added the ASN module to Appendix A

References

CMS "Cryptographic Message Syntax", Internet Draft ietf-draft-smime-cms

MUSTSHOULD "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119

Security Considerations

To be supplied
Must keep a complete copy or equivalent of the certificate in the trusted root database, issuer serial number is insufficient.
Private key material must be protected.

Author's Address

Jim Schaad
Microsoft
One Microsoft Way
Redmond, WA 98XXX
Jimsch@microsoft.com