Network Working Group
Request for Comments: 4073
Category: Standards Track

R. Housley
Vigil Security

May 2005

Protecting Multiple Contents with the Cryptographic Message Syntax (CMS)

Status of This Memo

Copyright Notice

Abstract

This document describes a convention for using the Cryptographic Message Syntax (CMS) to protect a content collection. If desired, attributes can be associated with the content.

1. Introduction

This document describes a convention for using the Cryptographic Message Syntax (CMS) [CMS] to protect a content collection. The content-collection content type is used to transfer one or more contents, each identified by a content type. If desired, the content-with-attributes content type can be used to associate arbitrary attributes with the content.

The convention described in this document is not needed when CMS is used with MIME [MSG]. MIME multipart [MIME] provides a straightforward and widely deployed mechanism for carrying more than one content item, each associated with a MIME type.

However, CMS is not always used with MIME. Sometimes CMS is used in an exclusively ASN.1 [ASN1] environment. In this case, the content-collection content type is used to gather more than one content item, each with an object identifier to specify the content type.

In this document, the key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL are to be interpreted as described in [STDWORDS].

1.1. Content Collection Example

This section provides one simple example to illustrate the need for the content-collection content type. Consider an art collector who wants to sell one of his pieces, an ancient Greek urn called an amphora. The collector wants to compose a digitally signed offer for sale. It includes three parts. The first part contains the owner's offer for sale, including the asking price. The second part contains a high-quality image of the amphora. The final part contains an appraisal from a well-respected ceramics expert. The final part is digitally signed by the expert. Figure 1 illustrates the structure, and the CMS SignedData content type is used for the two digital signatures.

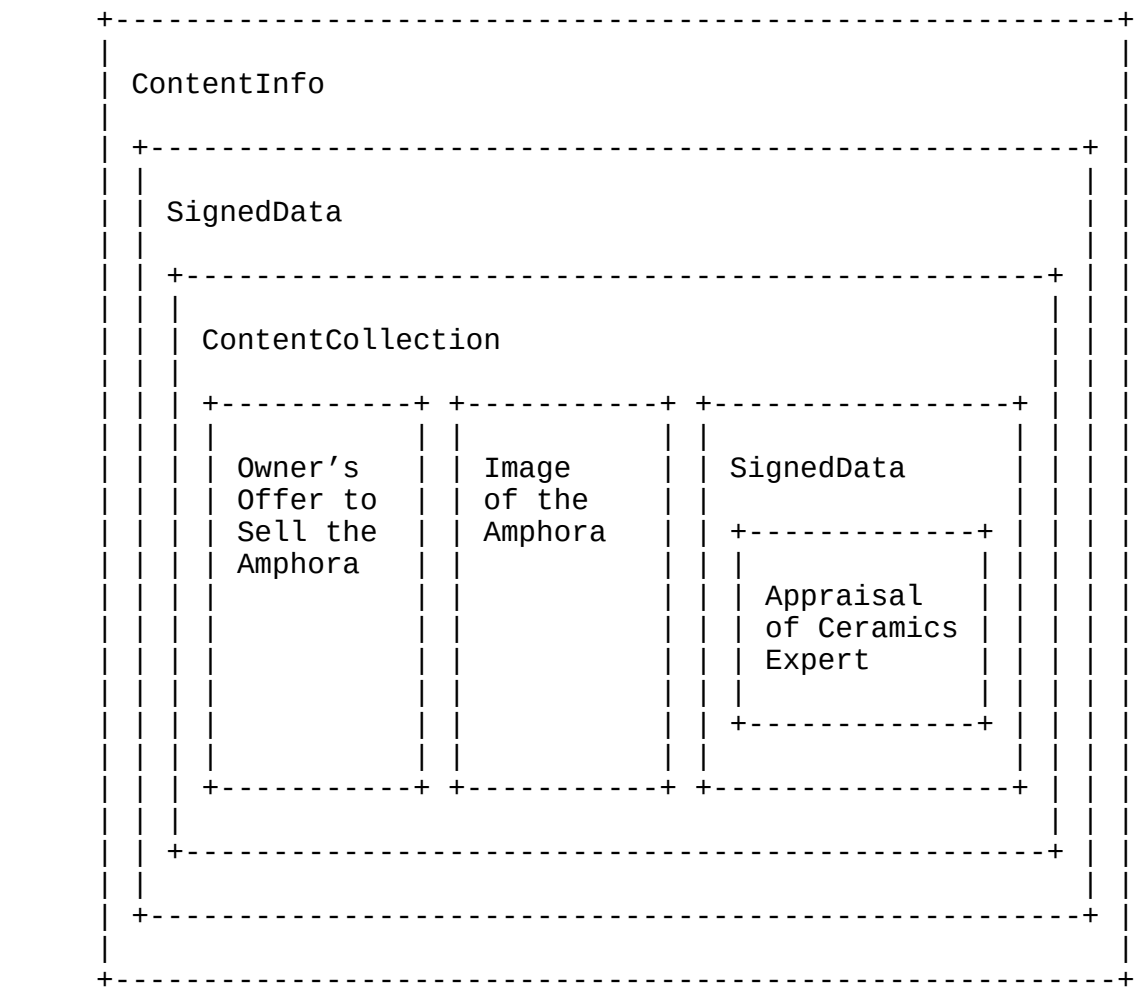

Figure 1. Sample use of the ContentCollection Content Type

1.2. Content with Attributes Example

This section provides one simple example to illustrate the need for the content-with-attributes content type. Consider the art collector from the previous example. Instead of providing a single image of the amphora, the collector provides several images. To aid potential buyers, the collector attaches several attributes to each image. The attributes provide information about the resolution of the image, the date the image was taken, the photographer, and so on. Figure 2 illustrates the collection of images, showing only two images, each with three attributes. This entire image content collection could be carried instead of the single image shown in Figure 1, allowing it to be covered by the collector's digital signature.

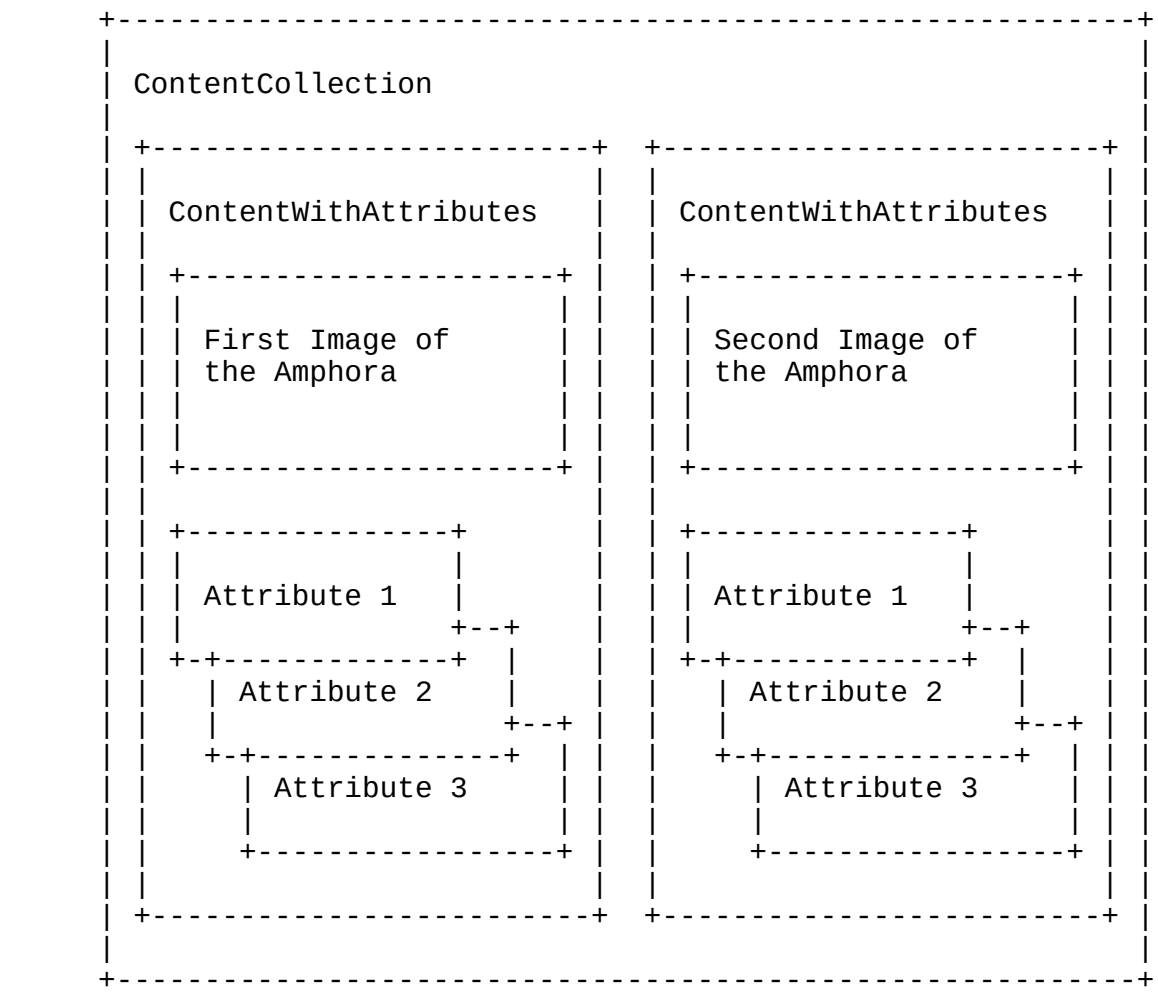

Figure 2. Sample use of the ContentWithAttributes Content Type

2. Content Collection Content Type

The content-collection content type is used to transfer a collection of content items, each identified by a content type. The syntax accommodates contents with varying levels of protection. For example, a content collection could include CMS protection content types as well as unprotected content types. A content collection is expected to be encapsulated in one or more CMS protecting content types, but this is not required by this specification.

The following object identifier names the content collection content type:

```
id-ct-contentCollection OBJECT IDENTIFIER ::= {
        iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) ct(1) 19 }
```

The content-collection content has the following syntax:

```
ContentCollection ::= SEQUENCE SIZE (1..MAX) OF ContentInfo
```

The ContentCollection contains a sequence of ContentInfo, one for each content in the collection. The ContentInfo structure is defined in CMS. The contentType object identifier within the ContentInfo indicates the type of the associated content. Implementations of this specification SHOULD be prepared to handle object identifiers for the SignedData, EncryptedData, EnvelopedData, and AuthenticatedData content types, as specified in [CMS]. Implementations of this specification SHOULD also be prepared to handle the object identifier for the CompressedData content type as specified in [COMPRESS].

3. Content-with-Attributes Content Type

The content-with-attributes content type is used to transfer a single content, which is identified by a content type, and a collection of attributes associated with that content. The syntax accommodates an arbitrary number of attributes; however, there must be at least one attribute.

The following object identifier names the content-with-attributes content type:

```
id-ct-contentWithAttrs OBJECT IDENTIFIER ::= {
        iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) ct(1) 20 }
```

The content-with-attributes content has the following syntax:

```
ContentWithAttributes ::= SEQUENCE {
    content     ContentInfo,
    attrs       SEQUENCE SIZE (1..MAX) OF Attribute }
```

The ContentWithAttributes contains a sequence of a single ContentInfo item followed by a sequence of attributes. The ContentInfo structure is defined in CMS. The contentType object identifier within the ContentInfo indicates the type of the content. The Attribute structure was originally defined in X.501 [X501], and the definition is repeated in CMS.

4. Security Considerations

The content-collection content type is used to transfer one or more contents, each identified by a content type. The syntax accommodates contents with varying levels of protection. For example, a content collection could include CMS protection content types as well as unprotected content types. A content collection is expected to be encapsulated in one or more CMS protecting content types, but this is not required by this specification. As a result, implementations MUST be prepared to handle multiple levels of encapsulation.

The security considerations discussed in [CMS] are relevant when CMS is used to protect more than one content by making use of the content collection content type or content with attributes content type.

5. References

5.1. Normative References

[ASN1] CCITT. Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1). 1988.

[COMPRESS] Gutmann, P., "Compressed Data Content Type for Cryptographic Message Syntax (CMS)", RFC 3274, June 2002.

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

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

5.2. Informative References

[MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

[X501] CCITT. Recommendation X.501: The Directory -- Models. 1988.

Appendix A: ASN.1 Module

The ASN.1 module contained in this appendix defines the structures that are needed to implement this specification. It is expected to be used in conjunction with the ASN.1 modules in [CMS] and [COMPRESS].

```
ContentCollectionModule
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0) 26 }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
  Attribute, ContentInfo
    FROM CryptographicMessageSyntax2004 -- [CMS]
      { iso(1) member-body(2) us(840) rsadsi(113549)
        pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2001(14) };


-- Content Collection Content Type and Object Identifier

id-ct-contentCollection OBJECT IDENTIFIER ::= {
        iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) ct(1) 19 }

ContentCollection ::= SEQUENCE SIZE (1..MAX) OF ContentInfo

-- Content With Attributes Content Type and Object Identifier

id-ct-contentWithAttrs OBJECT IDENTIFIER ::= {
        iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) ct(1) 20 }

ContentWithAttributes ::= SEQUENCE {
    content     ContentInfo,
    attrs       SEQUENCE SIZE (1..MAX) OF Attribute }

END
```

Author's Address

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

EMail: housley@vigilsec.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.