Network Working Group
Request for Comments: 3058
Category:Informational

S. Teiwes
P. Hartmann
iT_Security AG (Ltd.)
D. Kuenzi
724 Solutions Inc.
February 2001

Use of the IDEA Encryption Algorithm in CMS

Status of this Memo

Copyright Notice

Abstract

This memo specifies how to incorporate International Data Encryption Algorithm (IDEA) into CMS or S/MIME as an additional strong algorithm for symmetric encryption. For organizations who make use of IDEA for data security purposes it is of high interest that IDEA is also available in S/MIME. The intention of this memo is to provide the OIDs and algorithms required that IDEA can be included in S/MIME for symmetric content and key encryption.

1. Introduction

This memo specifies how to incorporate International Data Encryption Algorithm (IDEA) [IDEA] into CMS or S/MIME [SMIME2, SMIME3] as an additional strong algorithm for symmetric encryption. For organizations who make use of IDEA for data security purposes it is of high interest that IDEA is also available in S/MIME. The intention of this memo is to provide the OIDs and algorithms required that IDEA can be included in S/MIME for symmetric content and key encryption.

The general functional capabilities and preferences of S/MIME are specified by the registered list of S/MIME object identifiers (OIDs). This list of OIDs is available from the Internet Mail Consortium at <http://www.imc.org/ietf-smime/oids.html>. The set of S/MIME functions provided by a client is expressed by the S/MIME capabilities attribute. This attribute contains a list of OIDs of supported cryptographic functions.

In this document, the terms MUST, MUST NOT, SHOULD, and SHOULD NOT are used in capital letters. This conforms to the definitions in [MUSTSHOULD].

2. Object Identifier for Content and Key Encryption

The Cryptographic Message Syntax [CMS], derived from PKCS#7 [PKCS7], is the framework for the implementation of cryptographic functions in S/MIME. It specifies data formats and encryption processes without naming the cryptographic algorithms. Each algorithm which is used for encryption purposes must be specified by a unique algorithm identifier. For example, in the special case of content encryption the ContentEncryptionAlgorithmIdentifier specifies the algorithm to be applied. However, according to [CMS] any symmetric encryption algorithm that a CMS implementation includes as a content-encryption algorithm must also be included as a key-encryption algorithm.

IDEA is added to the set of optional symmetric encryption algorithms in S/MIME by providing two unique object identifiers (OIDs). One OID defines content encryption and the other one key encryption. Thus an S/MIME agent can apply IDEA either for content or key encryption by selecting the corresponding object identifier, supplying the required parameter, and starting the program code.

For content encryption the use of IDEA in cipher block chaining (CBC) mode is recommended. The key length is fixed to 128 bits.

The IDEA content-encryption algorithm in CBC mode has the object identifier

```
IDEA-CBC OBJECT IDENTIFIER
  ::= { iso(1) identified-organization(3)
      usdod(6) oid(1) private(4) enterprises(1)
      ascom(188) systec(7) security(1) algorithms(1) 2 }
```

The identifier's parameters field contains the initialization vector (IV) as an optional parameter.

```
IDEA-CBCPar ::= SEQUENCE {
  iv  OCTET STRING OPTIONAL } -- exactly 8 octets
```

If IV is specified as above, it MUST be used as initial vector. In this case, the ciphertext MUST NOT include the initial vector. If IV is not specified, the first 64 bits of the ciphertext MUST be considered as the initial vector. However, this alternative of not including IV into "iv OCTET STRING" of IDEA-CBCPar SHOULD NOT be applied in CMS or S/MIME.

The key-wrap/unwrap algorithms used to encrypt/decrypt an IDEA content-encryption key with an IDEA key-encryption key are specified in the following section. Generation and distribution of IDEA key-encryption keys are beyond the scope of this document.

The IDEA key-encryption algorithm has the object identifier

```
id-alg-CMSIDEAwrap OBJECT IDENTIFIER
  ::= { iso(1) identified-organization(3)
      usdod(6) oid(1) private(4) enterprises(1)
      ascom(188) systec(7) security(1) algorithms(1) 6 }
```

The identifier's parameters field MUST be NULL.

3. Key-Wrapping and Unwrapping

In the following subsections IDEA key-wrap and key-unwrap algorithms are specified in conformance with [CMS], section 12.3.

3.1 IDEA Key Wrap

The IDEA key-wrap algorithm encrypts an IDEA content-encryption key with an IDEA key-encryption key. The IDEA key-wrap algorithm is defined by:

1. Let the content-encryption key (16 octets) be called CEK
2. Compute an 8 octet key checksum value on CEK as described in [CMS], section 12.6.1, call the result ICV.
3. Let CEKICV := CEK || ICV.
4. Generate 8 octets at random, call the result IV.
5. Encrypt CEKICV using IDEA in CBC mode and the key-encryption key. Use the random value generated in the previous step as the initialization vector (IV). Call the ciphertext TEMP1.
6. Let TEMP2 = IV || TEMP1.
7. Reverse the order of the octets in TEMP2. That is, the most significant (first) octet is swapped with the least significant (last) octet, and so on. Call the result TEMP3.
8. Encrypt TEMP3 using IDEA in CBC mode and the key-encryption key. Use an initialization vector (IV) of 0x4adda22c79e82105. The ciphertext is 32 octets long.

3.2 IDEA Key Unwrap

The IDEA key-unwrap algorithm decrypts an IDEA content-encryption key using an IDEA key-encryption key. The IDEA key-unwrap algorithm is defined by:

1. If the wrapped content-encryption key is not 32 octets, then error.
2. Decrypt the wrapped content-encryption key using IDEA in CBC mode with the key-encryption key. Use an initialization vector (IV) of 0x4adda22c79e82105. Call the output TEMP3.
3. Reverse the order of the octets in TEMP3. That is, the most significant (first) octet is swapped with the least significant (last) octet, and so on. Call the result TEMP2.
4. Decompose the TEMP2 into IV and TEMP1. IV is the most significant (first) 8 octets, and TEMP1 is the remaining (last) 24 octets.
5. Decrypt TEMP1 using IDEA in CBC mode with the key-encryption key. Use the IV value from the previous step as the initialization vector. Call the plaintext CEKICV.
6. Decompose the CEKICV into CEK and ICV. CEK is the most significant (first) 16 octets, and ICV is the least significant (last) 8 octets.
7. Compute an 8 octet key checksum value on CEK as described in [CMS], section 12.6.1. If the computed key checksum value does not match the decrypted key checksum value, ICV, then error.
8. Use CEK as the content-encryption key.

4. SMIMECapabilities Attribute

An S/MIME client can announce the set of cryptographic functions it supports by using the S/MIME capabilities attribute as specified in [SMIME3]. This attribute provides a partial list of OIDs of cryptographic functions and must be signed by the client. These OIDs should be logically separated in functional categories and MUST be ordered with respect to their preference. If an S/MIME client is required to support symmetric encryption and key wrapping based on IDEA, the capabilities attribute MUST contain the above specified OIDs in the category of symmetric algorithms and key encipherment algorithms. IDEA does not require additional OID parameters since it has a fixed key length of 128 bits.

The SMIMECapability SEQUENCE representing the IDEA symmetric encryption algorithm MUST include the IDEA-CBC OID in the capabilityID field and the parameters field MUST be absent. The SMIMECapability SEQUENCE for IDEA encryption SHOULD be included in the symmetric encryption algorithms portion of the SMIMECapabilities list. The SMIMECapability SEQUENCE representing IDEA MUST be DER-encoded as follows: 300D 060B 2B06 0104 0181 3C07 0101 02.

The SMIMECapability SEQUENCE representing the IDEA key wrapping algorithm MUST include the id-alg-CMSIDEAwrap OID in the capabilityID field and the parameters field of KeyWrapAlgorithm MUST be absent. The SMIMECapability SEQUENCE for IDEA key wrapping SHOULD be included

in the key encipherment algorithms portion of the SMIMECapabilities list. The SMIMECapability SEQUENCE representing IDEA key wrapping MUST be DER-encoded as follows: 300D 060B 2B06 0104 0181 3C07 0101 06.

The ASN.1 notation of the SMIMECapability SEQUENCE representing IDEA is

```
SMIMECapability ::= SEQUENCE {
   capabilityID OBJECT IDENTIFIER,
   parameters ANY DEFINED BY capabilityID OPTIONAL }
```

where capabilityID is IDEA-CBC (no parameters) for IDEA content encryption in CBC mode or capabilityID is id-alg-CMSIDEAwrap (no parameters) for IDEA key wrapping.

5. Activation of IDEA in S/MIME Clients

When a sending agent creates an encrypted message, it has to decide which type of encryption algorithm to use. In general the decision process involves information obtained from the capabilities lists included in messages received from the recipient, as well as other information such as private agreements, user preferences, legal restrictions, etc. If users require IDEA for symmetric encryption, it must be supported by the S/MIME clients on both the sending and receiving side, and it must be set in the user preferences.

A. References

[IDEA] X. Lai, "On the design and security of block ciphers", ETH Series in Information Processing, J.L. Massey (editor), vol. 1, Hartung-Gorre Verlag Konstanz, Technische Hochschule (Zurich), 1992. A. J. Menezes, P.C. v. Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press New York, 1997, p. 265. B. Schneier, "Applied Cryptography," 2nd ed., John Wiley & Sons Inc. New York, 1996, pp. 319-325. IPR: see the "IETF Page of Intellectual Property Rights Notices", http://www.ietf.org/ipr.html

[SMIME2] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, l. and L. Repka, "S/MIME Version 2 Message Specification", RFC 2311, March 1998.

[SMIME2] Dusse, S., Hoffman, P., Ramsdell, B. and J. Weinstein, "S/MIME Version 2 Certificate Handling", RFC 2312, March 1998.

[SMIME3] Dusse, S., Hoffman, P., Ramsdell, B. and J. Weinstein, "S/MIME Version 3 Certificate Handling", RFC 2632, March 1998.

[SMIME3] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

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

[CMS] Housley, R., "Cryptographic Message Syntax", RFC 2630, June 1999.

[PKCS7] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March 1998.

B. Comments on IDEA Security and Standards

The IDEA algorithm was developed in a joint project involving the Swiss Federal Institute of Technology in Zurich (Dr. X. Lai and Prof. J.L. Massey) and Ascom Ltd. The aim of the project was to develop a strong encryption algorithm that could replace the DES algorithm.

IDEA uses 128-bit secret keys and encrypts one 64-bit block at a time [IDEA]. It was particularly strengthened to protect against differential cryptoanalysis attacks. For the full 8-round IDEA there is no attack known which is better than exhaustive search on the total 128-bit key space.

IDEA permits the implementation of standard Electronic Data Interchange applications. It has been entered in the ISO/IEC register for encryption algorithms and incorporated in the "SECURITY GUIDE LINES" code list by the UNI/EDIFACT "SECURITY JOINT WORKING GROUP".

C. Intellectual Property Rights Notice

Ascom Ltd. holds the patent to IDEA. In accordance with the intellectual property rights procedures of the IETF standards process, Ascom offers a non-exclusive license under reasonable and non-discriminatory terms and conditions.

IDEA(TM) is protected by international copyright law and in addition has been patented in several countries. Because Ascom wants to make this highly secure algorithm widely available, the non-commercial use of this algorithm is free.

Any party wishing to know more about IDEA or to request a license should visit the web sites <http://www.media-crypt.com/>, <http://www.it-sec.com/> or send an e-mail to info@media-crypt.com or Idea@it-sec.com.

D. Acknowledgements

We would like to thank Russ Housley, Jim Schaad and Francois Zeller for their contributions to this document.

E. Authors' Addresses

Stephan Teiwes
iT_Security AG (Ltd.)
Badenerstrasse 530
CH-8048 Zurich, Switzerland

Phone: +41 1 404 8200
Fax : +41 1 404 8201
EMail: stephan.teiwes@it-sec.com

Peter Hartmann
iT_Security AG (Ltd.)
Badenerstrasse 530
CH-8048 Zurich, Switzerland

Phone: +41 1 404 8200
Fax : +41 1 404 8201
EMail: peter.hartmann@it-sec.com

Diego Kuenzi
724 Solutions Inc.
Bahnhofstrasse 16
CH-5600 Lenzburg, Switzerland

Phone: +41 62 888 3070
Fax: +41 62 888 3071
EMail: dkuenzi@724.com

F. Full Copyright Statement

Acknowledgement

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