

INTERNET-DRAFT

<[draft-balenson-secure-email-00.txt](#)>

D. Balenson (TIS)

J. Cook (TIS)

R. Housley (SPYRUS)

September 30, 1996

Internet Secure Electronic Mail:
Algorithms, Modes, and Identifiers for FORTEZZA Cryptography

Status of This Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress''.

To learn the current status of any Internet-Draft, please check the ``[1id-abstracts.txt](#)'' listing contained in one of the Internet-Drafts Shadow Directories on [ftp.is.co.za](#) (Africa), [nic.nordu.net](#) (Europe), [munnari.oz.au](#) (Pacific Rim), [ds.internic.net](#) (US East Coast), or [ftp.isi.edu](#) (US West Coast).

Abstract

This document provides definitions, formats, references, and citations for cryptographic algorithms, usage modes, and associated identifiers and parameters used in support of the FORTEZZA suite of cryptographic algorithms with Privacy Enhanced Mail (PEM) [[4](#),[5](#)] and MIME Object Security Services (MOSS) [[10](#)]. This document is organized in the same manner as [RFC 1423](#) [[5](#)]. It is divided into four primary sections, dealing with message encryption algorithms, message integrity check algorithms, symmetric key management algorithms, and asymmetric key management algorithms (including both asymmetric encryption and asymmetric signature algorithms).

Acknowledgments

The authors would like to thank Mark Feldman, Jim Galvin, and Sandy Murphy from TIS for their contributions to this document.

Table of Contents

1.	Message Encryption Algorithms	2
1.1	SKIPJACK in CBC Mode (SKIPJACK-CBC)	2

2.	Message Integrity Check Algorithms	3
2.1	Secure Hash Algorithm (SHA-1)	3
3.	Symmetric Key Management Algorithms	4
4.	Asymmetric Key Management Algorithms	4
4.1	Asymmetric Keys	4
4.1.1	KEA-DSA Keys	4
4.2	Asymmetric Encryption Algorithms	5
4.2.1	Key Exchange Algorithm (KEA)	6
4.3	Asymmetric Signature Algorithms	7
4.3.1	Digital Signature Algorithm (DSA)	7
5.	Descriptive Grammar	8
	References	8
	Patent Statement	9
	Security Considerations	10
	Authors' Addresses	10

[1](#) Message Encryption Algorithms

This section identifies the alternative message encryption algorithms and modes that shall be used to encrypt message text. Character string identifiers are assigned and any parameters required by the message encryption algorithm are defined for incorporation in a "DEK-Info:" header field.

Only one alternative is currently defined in this category.

[1.1](#) SKIPJACK in CBC Mode (SKIPJACK-CBC)

Message text is encrypted using the SKIPJACK algorithm in the Cipher Block Chaining (CBC) mode of operation. SKIPJACK is a symmetric 64-bit block cipher designed to the requirements of the Escrowed Encryption Standard (EES) [[9](#)]. It uses an 80-bit cryptographic key.

The FORTEZZA PCMCIA crypto card incorporates a "Capstone" chip which implements the SKIPJACK algorithm. Further details of the SKIPJACK algorithm are beyond the scope of this document.

The SKIPJACK CBC mode of operation of is similar to that provided in ISO IS 8372 [1]. The character string "SKIPJACK-CBC" within the "DEK-Info:" header field indicates the use of this algorithm/mode combination.

The input to the SKIPJACK CBC encryption process shall be padded to a multiple of 8 octets, in the following manner. Let n be the length in octets of the input. Pad the input by appending $8-(n \bmod 8)$ octets to the end of the message, each having the value $8-(n \bmod 8)$, the number of octets being added. In hexadecimal, the possible

padding are: 01, 0202, 030303, 04040404, 0505050505, 060606060606, 07070707070707, and 0808080808080808. All input is padded with 1 to 8 octets to produce a multiple of 8 octets in length. The padding can be removed unambiguously after decryption.

The SKIPJACK CBC encryption process requires a 80-bit cryptographic key. A new, pseudorandom key is generated for each ENCRYPTED message.

The SKIPJACK CBC encryption process also requires a 192-bit Initialization Vector (IV). A new, pseudorandom IV shall be generated for each ENCRYPTED message. The IV is transmitted with the message within the "DEK-Info:" header field.

When this algorithm/mode combination is used for message text encryption, the "DEK-Info:" header field carries exactly two arguments. The first argument identifies the SKIPJACK CBC algorithm/mode using the character string defined above. The second argument contains the IV, represented as a contiguous string of 48 ASCII hexadecimal digits.

No symmetric key management is employed with this algorithm/mode.

This section identifies the alternative algorithms that shall be used to compute Message Integrity Check (MIC) values for messages. Character string identifiers and ASN.1 object identifiers are assigned for incorporation in "MIC-Info:" header fields to indicate the choice of MIC algorithm employed.

Only one alternative is defined in this category.

[2.1](#) Secure Hash Algorithm (SHA-1)

The Secure Hash Algorithm (SHA-1) message digest is computed using the algorithm defined in FIPS 180-1 [\[7\]](#). The character string "SHA-1" within a "MIC-Info:" header field indicates the use of this algorithm.

As specified in the SDNS Message Security Protocol (MSP) [\[2\]](#), the ASN.1 object identifier

```
id-mosaicUpdatedIntegrityAlgorithm OBJECT IDENTIFIER ::= {
    joint-iso-ccitt (2) country (16) us (840) organization (1)
    u.s.government (101) dod (2) 1 1 21)
```

}

identifies the SHA-1 algorithm. When this object identifier is used with the ASN.1 type AlgorithmIdentifier, the parameters component of that type is the ASN.1 type NULL.

The SHA-1 accepts as input a message of any length and produces as output a 20-octet (160-bit) quantity.

[3](#) Symmetric Key Management Algorithms

There are no symmetric key management algorithms for FORTEZZA.

[4](#) Asymmetric Key Management Algorithms

This section identifies the alternative asymmetric keys and the alternative asymmetric key management algorithms with which those keys shall be used, namely the asymmetric encryption algorithms with which DEKs and MICs are encrypted, and the asymmetric signature algorithms with which certificates and certificate revocation lists (CRLs) are signed.

[4.1](#) Asymmetric Keys

This section describes the asymmetric keys that shall be used with the asymmetric encryption algorithms and the signature algorithms described later. ASN.1 object identifiers are identified for incorporation in a public-key certificate to identify the algorithm(s) with which the accompanying public keys are to be employed.

[4.1.1](#) KEA-DSA Keys

A KEA-DSA set of asymmetric key pairs is comprised of two sets of matching public and private keys. The first set is a Key Exchange Algorithm (KEA) public/private key pair and the second set is a Digital Signature Algorithm (DSA) public/private key pair.

As specified in SDNS Message Security Protocol (MSP) [[2](#)], the following ASN.1 object identifier identifies KEA-DSA public key pairs:

```
id-mosaickMandUpdSigAlgorithms OBJECT IDENTIFIER ::= {
    joint-iso-ccitt (2) country (16) us (840) organization (1)
    u.s.government (101) dod (2) 1 1 20
}
```

When this object identifier is used with the ASN.1 type

AlgorithmIdentifier, the parameters component of that type is the ASN.1 type Kea-Dss-Parms, which is specified in [2] as:

```
Kea-Dss-Parms ::= CHOICE {
    [0]          Different-Parms,
    [1]          Common-Parms
}

Different-Parms ::= SEQUENCE {
    Kea-Parms,
    Dss-Parms
}

Kea-Parms ::= SEQUENCE {
    p          OCTET STRING,
    q          OCTET STRING,
    g          OCTET STRING
}

Dss-Parms ::= SEQUENCE {
    p          OCTET STRING,
    q          OCTET STRING,
    g          OCTET STRING
}

Common-Parms ::= SEQUENCE {
    p          OCTET STRING,
    q          OCTET STRING,
    g          OCTET STRING
}
```

The format of the KEA-DSA public key pair carried in a FORTEZZA public-key certificate is described in detail in the FORTEZZA Application Implementors Guide [3].

[4.2](#) Asymmetric Encryption Algorithms

This section identifies the alternative algorithms that shall be used when asymmetric key management is employed to encrypt DEKs and sign MICs. Character string identifiers are assigned for incorporation in "Key-Info:" and "DEK-Info:" header fields to indicate the choice of algorithm employed.

Only one alternative for each of asymmetric encryption and asymmetric signatures is presently defined in this category.

[4.2.1](#) Key Exchange Algorithm (KEA)

The asymmetric Key Exchange Algorithm (KEA) is used for DEK exchange. The character string "KEA-SJ" within a "Key-Info:" header field indicates the use of this algorithm.

The only type of DEK that undergoes KEA processing is a DEK generated for the SKIPJACK algorithm. The KEA process employs a 128-octet (1024-bit) random value, Ra, to create a token encrypting key (TEK). The TEK is then used to "wrap" (encrypt) the DEK.

When KEA is used, the second argument in a "Key-Info:" header field is represented using the "base 64" printable encoding technique defined in [Section 4.3.2.4 of RFC 1421](#) [4]. This second argument is a printably encoded ASN.1 sequence of three items: (a) the TEK-wrapped DEK (12 octets), (b) the random Ra value (128 octets), and (c) the KEA public key Y (128 octets) used to generate the TEK. In particular, the ASN.1 sequence is defined as:

```
SEQUENCE {
    wrappedDEK  OCTET STRING  -- 12 octets
    ra          OCTET STRING  -- 128 octets
    y           OCTET STRING  -- 128 octets
}
```

The Ra and Y values are needed by the recipient to generate the TEK needed to decrypt the DEK (the value of the "random" value Rb is always one).

[Section 4.2.1 of RFC 1423](#) [5] indicates that, for RSA Encryption [6], only the printably encoded RSA-encrypted DEK is included in the second argument of the "Key-Info:" header field. This would not be a viable solution for FORTEZZA, as the recipient must obtain the random value Ra and the KEA public key Y of the originator before the DEK can be decrypted (unwrapped). The KEA public key could be extracted from the originator's certificate if the message was signed, but for encrypted-only messages the originator is not identified and therefore the KEA public key of the originator must be provided to permit the recipient to decrypt the DEK.

The appearance of the KEA public key Y of the originator within a

"Key-Info:" header field associated with an encrypted MOSS body part is required for message decryption, but Y can also be used to infer the identity of the originator and thus can provide a form of authentication. The Y value is the minimum amount of originator

information required and thus provides the weakest implication possible as to the identity of the originator. This information shall not be used by MOSS to provide authentication, because confidentiality is the only security service provided by MOSS for encrypted body parts. Authentication is provided by MOSS for signed body parts only.

[4.3](#) Asymmetric Signature Algorithms

A description of the algorithms used to asymmetrically sign certificates and certificate revocation lists (CRLs) for FORTEZZA is outside the scope of this document.

This section identifies the alternative algorithms which shall be used to asymmetrically sign messages, public-key certificates and certificate revocation lists (CRLs). An ASN.1 object identifier is identified for incorporation in certificates and CRLs to indicate the choice of algorithm employed.

Only one alternative is presently defined in this category.

[4.3.1](#) Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) defined in FIPS 186 [\[8\]](#), is used for digital signatures. The character string "DSA" within a "MIC-Info:" header field indicates the use of this algorithm.

The only type of MIC value that undergoes DSA encryption is a 20-octet MIC generated by the SHA-1 algorithm. The result is a 40-octet signed MIC.

As specified in the SDNS Message Security Protocol (MSP) [2], the ASN.1 object identifier

```
id-mosaicUpdatedSigAlgorithm OBJECT IDENTIFIER ::= {
    joint-iso-ccitt (2) country (16) us (840) organization (1)
    u.s.government (101) dod (2) 1 1 19
}
```

identifies the combination of the DSA and SHA-1 algorithms. When this object identifier is used with the ASN.1 type AlgorithmIdentifier, the parameters component of that type is the ASN.1 type Dss-Parms, which is specified in [2] as:

```
Dss-Parms ::= SEQUENCE {
```

```
    p          OCTET STRING,
    q          OCTET STRING,
    g          OCTET STRING
}
```

When DSA encryption is used to sign a MIC, the third argument in a "MIC-Info:" header field, an asymmetrically signed MIC, is represented using the printable encoding technique defined in [Section 4.3.2.4 of RFC 1421](#)."

[5](#) Descriptive Grammar

; Addendum to PEM BNF representation, using [RFC 822](#) notation
; Provides specification for FORTEZZA cryptographic algorithms,
; modes, identifiers and formats.

; Imports <hexchar> and <encbin> from [RFC 1421](#)

```
<dekalgid> ::= "SKIPJACK-CBC"
<ikalgid>  ::= "KEA-SJ"
<sigalgid> ::= "DSA"
<micalgid> ::= "SHA-1"
```

```
<dekparameters> ::= <SkipjackCBCparameters>
```

<SkipjackCBCparameters> ::= <IV>

<IV> ::= <hexchar48>

<asymsignmic> ::= <encbin>

<asymencdek> ::= <encbin>

<hexchar48> ::= 48*48<hexchar>

References:

- [1] ISO 8372, Information Processing Systems: Data Encipherment: Modes of Operation of a 64-bit Block Cipher.
- [2] "SDNS Message Security Protocol (MSP)", Specification SDN.701, Revision 4.0, 1996-01-16.
- [3] "FORTEZZA Application Implementors Guide for the FORTEZZA Crypto Card (Production Version)", Document #PD4002103-1.01, SPYRUS, 1995.
- [4] Linn, J., "Privacy Enhancement for Internet Electronic Mail:

Part I: Message Encryption and Authentication Procedures", [RFC 1421](#), February, 1993.

- [5] Balenson, D., "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers", [RFC 1423](#), February 1993.
- [6] PKCS #1: RSA Encryption Standard, Version 1.4, RSA Data Security, Inc., June 3, 1991.
- [7] Federal Information Processing Standard (FIPS) 180-1, Secure Hash Standard, National Institute of Standards and Technology, May 31, 1994.
- [8] Federal Information Processing Standard (FIPS) 186, Digital Signature Standard, National Institute of Standards and

Technology, May 19, 1994.

- [9] Federal Information Processing Standard (FIPS) 185, Escrowed Encryption Standard, National Institute of Standards and Technology, February 9, 1994.
- [10] S. Crocker, N. Freed, J. Galvin, and S. Murphy, MIME Object Security Services, [RFC 1848](#), October 1995.

Patent Statement

FORTEZZA cryptography relies on the use of patented public key technology, namely DSA. The Internet Standards Process as defined in [RFC 1310](#) requires a written statement from the Patent holder that a license will be made available to applicants under reasonable terms and conditions prior to approving a specification as a Proposed, Draft or Internet Standard.

A patent statement for DSA follows. This statement has been supplied by the patent holder, not the authors of this profile.

Digital Signature Algorithm (DSA)

The U.S. Government holds patent 5,231,668 on the Digital Signature Algorithm (DSA), which has been incorporated into Federal Information Processing Standard (FIPS) 186. The patent was issued on July 27, 1993.

The National Institute of Standards and Technology (NIST) has a long tradition of supplying U.S. Government-developed techniques committees and working groups for inclusion into standards on a royalty-free basis. NIST has made the DSA patent available royalty-free to users worldwide.

Regarding patent infringement, FIPS 186 summarizes our position; the Department of Commerce is not aware of any patents that would be infringed by the DSA. Questions regarding this matter may be directed to the Deputy Chief Counsel for NIST.

Security Considerations

This documents defines the use of FORTEZZA cryptographic algorithms with secure Internet electronic mail.

Authors' Addresses:

David Balenson
Trusted Information Systems
3060 Washington Road
Glenwood, Maryland 21738 USA
EMail: balenson@tis.com

Jeff Cook
Trusted Information Systems
11340 W. Olympic Blvd.
Los Angeles, California 90064 USA
EMail: jvc@tis.com

Russell Housley
SPYRUS
PO Box 1198
Herndon, VA 22070 USA
EMail: housley@spyrus.com