CCSDS
The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

AUTHENTICATION/INTEGRITY ALGORITHM ISSUES SURVEY

INFORMATIONAL REPORT

CCSDS 350.3-G-1

GREEN BOOK
March 2008

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

AUTHENTICATION/INTEGRITY ALGORITHM ISSUES SURVEY

INFORMATIONAL REPORT

CCSDS 350.3-G-1

GREEN BOOK

March 2008

AUTHORITY

Issue:	Informational Report, Issue
Date:	March 2008
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Taiwan.
- Naval Center for Space Technology (NCST)/USA.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 350.3-G-1	Authentication/Integrity Algorithm Issues Survey, Informational Report, Issue	March 2008	Original issue

CONTENTS

Section Page

Table

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

This Report presents the results of a survey conducted by the CCSDS Security Working Group (SecWG), which has been actively engaged in developing a recommendation for a CCSDS standard algorithm for authentication and integrity. This algorithm may be used by all member agencies to provide command (uplink) authentication, command integrity, as well as telemetry (downlink) housekeeping or mission data authentication and integrity.

A single algorithm is used to provide both authentication and integrity; integrity essentially comes for 'free' as a side-affect of the authentication algorithm.

Authentication provides a service that allows a receiver of data to verify the source of the data and be assured that it came from the claimed source. Integrity provides a service that allows the receiver of data to be assured that what the sender transmitted has been received and that no unauthorized modification of the data (accidental or intentional) has occurred in transit.

The information contained in this report is not part of any of the CCSDS Recommended Standard. In the event of any conflict between any CCSDS Recommended Standard and the material presented herein, the CCSDS Recommended Standard shall prevail.

1.2 REFERENCES

The following documents are referenced in this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS documents.

[1] *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*. ANSI X9.31:1998. New York: ANSI, 1998.

[2] *Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)*. ANSI X9.62:2005. New York: ANSI, 2005.

[3] *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-2. Gaithersburg, Maryland: NIST, January 2000. <http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>

[4] H. Krawczyk, M. Bellar, and M. Bellar. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. Reston, Virginia: ISOC, February 1997.

[5] *The Keyed-Hash Message Authentication Code (HMAC)*. Federal Information Processing Standards Publication 198. Gaithersburg, Maryland: NIST, March 2002. <http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf>

[6] T. Krovetz, ed. *UMAC: Message Authentication Code using Universal Hashing*. RFC 4418. Reston, Virginia: ISOC, March 2006.

[7] P. Metzger and W. Simpson. *IP Authentication using Keyed MD5*. RFC 1828. Reston, Virginia: ISOC, August 1995.

[8] *Computer Data Authentication*. Federal Information Processing Standards Publication 113. Gaithersburg, Maryland: NIST, May 1985. <http://www.itl.nist.gov/fipspubs/fip113.htm>

[9] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*. National Institute of Standards and Technology Special Publication 800-38B (Draft). Gaithersburg, Maryland: NIST, March 9, 2005. <http://www.mirrors.wiretapped.net/security/info/reference/nist/special-publications/sp-800-38b-draft.pdf>

[10] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*. National Institute of Standards and Technology Special Publication 800-38C. Gaithersburg, Maryland: NIST, May 2004. <http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf>

[11] *Secure Hash Standard*. Federal Information Processing Standards Publication 180-2. Gaithersburg, Maryland: NIST, August 2002.

2 OVERVIEW

2.1 BACKGROUND

At the spring 2004 CCSDS meeting held in Montreal, Canada, a proposal was made to adopt the Digital Signature Standard (DSS) (FIPS 186-2) as the CCSDS authentication/integrity standard. However, there was no consensus on this proposal at that meeting. The same proposal was then discussed again and tabled at the fall 2005 CCSDS meeting held in Toulouse, France. It was discussed yet again at the spring 2005 meeting in Athens, Greece.

At the Athens meeting, it was decided that a trade survey should be performed in order to compare and contrast all of the algorithms available for adoption by CCSDS. This trade survey was analogous to the encryption algorithm trade survey also agreed upon to be performed.

While at first the authentication algorithm trade survey appeared to be straightforward (e.g., algorithm X versus algorithm Y versus algorithm Z), upon further analysis it appears that this is not the case and in fact, the trade survey is more complicated than first envisioned (at least by this author).

2.2 TRADE SURVEY SPACE

2.2.1 GENERAL

At past Security Working Group meetings, the authentication/integrity algorithm proposal had been based solely on digital signature technology. However, it turns out that this was terribly premature and incomplete as will be observed from the descriptions provided in the subsequent paragraphs. There are at least two major ways to implement authentication/integrity mechanisms and in one of the major methods, there are two sub-methods. All of this will be described in the subsequent paragraphs below.

2.2.2 DIGITAL SIGNATURE

Digital signature technology requires the use of public key cryptography, that is, the use of public and private key pairs. A sender would digitally sign a message by computing a hash (or checksum) over the data creating a check-word and then encrypt the check-word using its private key. The encrypted check-word would be sent to the receiver as an accompaniment to the data. The receiver would verify the authenticity of the message by recalculating the check-work, decrypting the transmitted check-word using the sender’s public key (which had been previously obtained and cached, or was obtained from a public key directory) and comparing the two check-words. If they match, then the message is authentic and came from the claimed sender.

2.2.3 MESSAGE AUTHENTICATION CODES

While digital signatures are appropriate for use as an authentication/integrity mechanism, another type of mechanism that has been used for this purpose is called the Message Authentication Code (MAC). While digital signatures use public/private key pairs, MACs use a shared secret key. And even more interesting, MACs can use a private key to provide authentication/integrity in several different ways. In one way, a check-word can be created over the data with an embedded secret key and in another way, the check-word is created by a hash algorithm which is then encrypted using the secret key.

2.2.4 HASH-BASED MESSAGE AUTHENTICATION CODES (HMAC)

One variety of MAC is called a hash-based message authentication code (HMAC). This type of MAC makes use of a 'strong' hash algorithm (e.g., MD5, SHA1, SHA256) to create a check-word over the data and an embedded key. For example, if the data consists of the string, 'Mary had a little lamb', and the secret key were '01234567890000' then the hash algorithm would create a check-word by hashing over the concatenated string of 'Mary had a little lamb 01234567890000'. Alternatively, the string could be constructed as '01234567890000 Mary had a little lamb'. Or it could be constructed as 'Mary had a 01234567890000 little lamb'. There are various HMAC algorithms that specify exactly how the data and the key are combined before hashing.

A receiver, who possessed the secret key, would re-generate the same check-word by performing the same hash function over the concatenated data and key. If the check-word received matches the one re-generated, then the authenticity and integrity of the received data is assured.

2.2.5 ENCRYPTION-BASED MESSAGE AUTHENTICATION CODES

The more 'traditional' MAC is based on the combination of hashing and encryption (typically cipher-block-chaining, or CBC). This type of MAC creates a check-word over the data using the hash algorithm. Then an encryption algorithm is used to encrypt the check-word using the secret key.

A receiver, who possessed the secret key, would re-generate the check-word and decrypt the sent check-word using the secret key. The re-generated check-word would be compared with the decrypted check-word and if they were identical the receiver would be assured of the authenticity and integrity of the data received.

3 POTENTIAL ALGORITHMS (BY CATEGORY) FOR CCSDS ADOPTION

3.1 GENERAL

Given the above descriptions, there are three categories of algorithms that could be adopted for use by CCSDS: digital signature algorithms, hash-based message authentication codes, and encryption-based message authentication codes. The following paragraphs describe algorithms of each type.

Table 3-1: Digital Signature Algorithms

Name	Type	Characteristics	Min. Key Size
Digital Signature Standard (DSS)	FIPS 186-2 digital signature	Digital signature based on SHA1 hash, un-encumbered (no patents, no licenses)	1024 bits
RSA Digital Signature	RSA digital signature (FIPS approved)	Previously patented digital signature (expired 2000)	1024 bits
Elliptic Curve Digital Signature (ECDSA)	Elliptic curve digital signature	Digital signature based on elliptic curve key technology which uses smaller keys than other public key technologies but may be encumbered by various Certicom intellectual property, licenses, and patents. Apparently, ECDSA is not covered by any Certicom patents and there are open-source ECC libraries; but Certicom does have over 300 patents on various aspects of ECC including ‘efficient implementations of ECC in hardware and software’, key agreements, etc.	160 bits

3.2 DIGITAL SIGNATURE ALGORITHMS

There are three digital signature algorithms that should be considered by CCSDS: Digital Signature Algorithm (DSA), Rivest-Shamir-Adleman (RSA) as specified in ANSI X9.31, and Elliptic Curve DSA (ECDSA) as specified in ANSI X9.62. All of these digital signature algorithms are captured in FIPS PUB 186-2 (with change notice 1 dated 5 October 2001).

Elliptic Curve Cryptography (ECC) is by far the most efficient algorithm with respect to key size; however many aspects of elliptic curve technology are patented by Certicom (www.certicom.com), and therefore licenses may have to be obtained. Certicom claims over 300 patents (and patents pending) on elliptic curve technologies. However, it appears from the literature that the basic ECDSA is free and open. Certicom does hold patents on various ‘efficient implementations of ECC’ in both hardware and software. It also holds patents on ECC key agreements, etc. There are open-source elliptic curve libraries that are available for use royalty- and license-free (e.g., *libecc*, a C++ open source ECC crypto library available at http://libecc.sourceforge.net/).

RSA had been patented by RSA but the patents have expired. DSA was developed to be fully open and is available for use with no license or patent restrictions.

3.3 HASH-BASED MESSAGE AUTHENTICATION CODES (HMAC)

In the hash-based MAC space there are several algorithms available for CCSDS consideration. However, there are two categories contained in this space: the HMAC algorithm specifications and the actual hash algorithms. Both are discussed in this section since the HMAC algorithm relies on the use of a hash algorithm and a hash algorithm relies on an HMAC specification to be used as a message authentication code.

The most notable HMAC algorithm is the IETF standard HMAC (RFC 2104) which is also a U.S. Federal Information Processing Standards (FIPS) Publication 198. HMAC can be used with either the MD5 or the SHA1 (or other variants of SHA) hashing algorithms. Likewise, there are other hash algorithms that could be used with FIPS PUB 198, such as RIPEMD-160 (RACE Integrity Principals Evaluation Message Digest) and TIGER.

An emerging hash-based algorithm is known as UMAC (Universal Message Authentication Code) (RFC 4418) whose design criteria was to be the fastest hash-based algorithm available. While it is fast, it is generally not considered to be as strong as other hash-based algorithms.

There is also a simple hash-based MAC algorithm known as *keyed MD5* (analogous to the ‘Mary had a little lamb’ example above) and is described in IETF RFC 1828. This simple keyed MD5 is based upon the following usage:

```
MD5 {key, keyfill, entire IP datagram, key, MD5fill}.
```

Each of the fills are used to pad out the message to a 512-bit boundary.

Table 3-2: Hash-Based Message Authentication Codes and Hashes

Name	Type	Characteristics	Output Hash Size
Secure Hash Algorithm 1 (SHA1)	Hash algorithm	FIPS approved; other versions (SHA256, SHA384, SHA512) provide longer outputs	160 bits
Message Digest 5 (MD5)	Hash algorithm	Potential weaknesses: can be used as a keyed hash	128 bits
Universal Message Authentication Code (UMAC)	Hash-based MAC	Designed to be the fastest hash-based algorithm ever	32, 64, or 96 bits (64 bits recommended)
RACE Integrity Primitives Evaluation Message Digest 160 (RIPEMD-160)	Hash Algorithm	Developed as part of the EC's Research and Development in Advanced Communications Technologies in Europe (RACE)	160 bits
TIGER	Hash Algorithm	Designed for efficient operation on 64-bit platforms	192 bits
HMAC-SHA1-96	Hash-based MAC	Uses SHA-1 for hash	96 bits; truncates SHA1 160 bit output
HMAC-MD5-96	Hash-based MAC	Uses MD5 for hash	96 bits; truncates MD5 128 bit output

3.4 ENCRYPTION-BASED MESSAGE AUTHENTICATION CODES

The last segment of possible MAC algorithms for CCSDS consideration is encryption-based MACs.

In this space are algorithms based on symmetric key block cipher algorithms (e.g., DES, AES, CAST, etc.). Such a MAC might be used instead of a hash-based MAC because a system might have an approved symmetric block cipher algorithm and not an approved hash algorithm.

There are several primary algorithms that should be considered by CCSDS: DES-CBC-MAC and Cipher-based MAC (CMAC) and Counter with Cipher Block Chaining MAC (CCM). DES-CBC-MAC is described in FIPS PUB 113, CMAC in NIST Special Publication 800-38B, and CCM in NIST Special Publication 800-38C.

While DES is notably weak as an encryption algorithm given the processing power of today's computers (and their inherent ability to perform brute force attacks), a DES-based MAC algorithm might be sufficient for use under some circumstances. However, with the almost universal deprecation of DES, it will not be in wide use anymore and will not be a convenient algorithm to use for a MAC. If a system already contained DES for other purposes, it would make sense to simply piggy-back and use the same algorithm for authentication. However, with new systems this will not be the case.

CMAC is based on the use of a symmetric key based algorithm in a cipher block chaining (CBC) mode (analogous to DES-CBC) to create a MAC. However, CMAC (in Special Publication 800-38B) does not specify a mandatory symmetric key block cipher algorithm for use. Rather it provides examples of how algorithms such as AES (128, 192, 256) as well as the Triple Data Encryption Algorithm (TDEA) (otherwise known as Triple DES or 3DES) can be used.

CCM combines cipher block chaining to create a MAC with counter mode encryption to create a data entity which is has both a MAC and is encrypted (both authentication and confidentiality). In generation-encryption, cipher block chaining is applied to the payload, the associated data, and the nonce to generate a message authentication code (MAC); then, counter mode encryption is applied to the MAC and the payload to transform them into an unreadable form, called the ciphertext. Thus, CCM generation-encryption expands the size of the payload by the size of the MAC. In decryption-verification, counter mode decryption is applied to the ciphertext to recover the MAC and the corresponding payload; then, cipher block chaining is applied to the payload, the received associated data, and the received nonce to verify the correctness of the MAC. Successful verification provides assurance that the payload and the associated data originated from a source with access to the key. CCM is only specified for an algorithm using 128-bit keys or larger so it is not compatible with TDEA/3DES which only uses a 64-bit key (multiple keys, each 64-bits in length).

Table 3-3: Encryption-based Message Authentication Codes

Name	Type	Characteristics	Key Size
DES-CBC-MAC	Cryptographic MAC	DES-based (FIPS PUB 113 dated 30 May 1985	64-bits
CMAC	Cryptographic MAC	Encrypted-based MAC using any symmetric key block cipher algorithm	64, 128, 192, 256 (depending on block cipher algorithm used)
CCM	Cryptographic MAC	Uses cipher-block-chaining (CBC) with counter mode encryption to provide both authentication and confidentiality using a block cipher algorithm with 128-bit key or greater)	128, 192, 256

4 SUMMARY AND CONCLUSIONS

This Report has attempted to provide information regarding the need for authentication/integrity for CCSDS, the three types of Message Authentication Codes (MAC) that could be adopted within CCSDS, and details regarding several MACs in each of the three types.

While digital signatures are ‘the modern’ technology and therefore would seem appropriate to adopt within CCSDS, they require the use of public/private key pairs. Therefore, in order to use digital signature based authentication, the ability to generate public/private key pairs and to distribute public keys to the affected parties must be in place. This could entail the construction of a Public Key Infrastructure (PKI) providing a means for key generation and a secure key server where authenticated public keys can be retrieved. Or it might entail a non-PKI-based generation and non-directory distribution of public keys (e.g., pre-loaded and cached public keys, etc.). Also, typical public keys (not including elliptic curve keys) are much larger than symmetric keys. All of this may end up being problematic for use with missions hoping to use CCSDS Recommended Standards.

For this reason, it is proposed that CCSDS adopt multiple authentication standards in order to cover more than one aspect of message authentication. A symmetric key based MAC uses smaller keys and does not require the generation and distribution of public/private keys and therefore no PKI (or moral equivalent) is required. However, there is still a need for the secure generation, distribution, and management of symmetric shared keys. These shared keys are only a fraction of the size of public keys (e.g., 128-bits vs. 1024-bits, although these may be larger depending on the algorithmic strength required). And, if an on-line Public Key Infrastructure is not to be used, then the public-key based digital signature solution and the symmetric key based solutions both have the need to distribute keys to end systems. The major difference is that the distribution medium does not have to be secure for public key distribution whereas it does have to be secure for symmetric key (or the keys themselves must be cryptographically ‘wrapped’ to protect them while in transit).

As a result of gathering the information and knowledge to produce this Report, the recommendation is to adopt the **Digital Signature Algorithm** (DSA) as the CCSDS digital signature standard and also adopt the **HMAC algorithm** as the hash-based MAC algorithm. DSA is specified in FIPS PUB 186-2 and the Secure Hash Standard (SHS) algorithm is specified in FIPS PUB 180-2. The Secure Hash Algorithm should, at a minimum, utilize SHA-1. HMAC is specified in FIPS PUB 198.

In this way, both public key and symmetric type of algorithms are recommended for use by CCSDS depending on the mission needs and the supporting infrastructure available.