The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

ENCRYPTION ALGORITHM TRADE SURVEY

INFORMATIONAL REPORT

CCSDS 350.2-G-1

GREEN BOOK

March 2008

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

ENCRYPTION ALGORITHM TRADE SURVEY

INFORMATIONAL REPORT

CCSDS 350.2-G-1

GREEN BOOK

March 2008

AUTHORITY

Issue:	Informational Report, Issue 1
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 350.2-G-1	Encryption Algorithm Trade Survey, Informational Report, Issue 1	March 2008	Original issue

CONTENTS

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

This Report presents the results of a survey conducted by the CCSDS Security Working Group, which has been actively engaged in developing security recommendations for CCSDS.

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

[1] *Report on the Development of the Advanced Encryption Standard (AES).* Gaithersburg, Maryland: NIST, 2002.

[2] *Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General*. Release 5. Third Generation Partnership Project, 3GPP TS 35.205 V5.0.0 (2002-06). Sophia-Antipolis: ETSI, 2002.

[3] *Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification*. Release 6. Third Generation Partnership Project, 3GPP TS 35.202 V6.0.0 (2004-12). Sophia-Antipolis: ETSI, 2004.

[4] *Comparison of Airlink Encryptions*. 1xEV-DO Web Paper. N.p.: Qualcomm, 2003. <http://www.wheresmyload.com/cdma/1xEV/media/web_papers/Airlink_Encryption.pdf>

[5] H. J. Lee, et al. *The SEED Encryption Algorithm*. RFC 4269. Reston, Virginia: ISOC, December 2005.

[6] *Advanced Encryption Standard (AES).* Federal Information Processing Standards Special Publication 197. Gaithersburg, Maryland: NIST, 2001.

2 OVERVIEW

While at first security was thought of as not required for civilian space missions by CCSDS, more recently it has become a much more desirable area of investigation. To that extent, the Security Working Group has been investigating the specification of a CCSDS standard for encryption. At a minimum, this standard would take the form of a CCSDS Recommended Standard for an encryption algorithm. Particulars regarding how the algorithm is actually employed (e.g., network layer, speeds, relationship to FEC, etc.) would not necessarily be standardized upon, at least not at the outset.

The first order of business would be to agree upon an encryption algorithm, its mode of operation, and its key length. The algorithm could be adopted or adapted from an existing one, or a new algorithm could be developed. The most expedient and safest way forward would be to adopt an existing algorithm that has been thoroughly crypt-analyzed by the cryptologic community to ensure its security.

At the spring 2004 CCSDS meeting held in Montreal, Canada, a proposal was made to adopt the Rijndael algorithm, written in Belgium and chosen as the United States's Advanced Encryption Standard (AES), as the CCSDS encryption algorithm standard.

At the Montreal meeting, those in attendance were given actions to investigate their nationalistic rules and laws concerning the use of encryption and whether or not it would be a national issue to adopt a CCSDS encryption algorithm standard. The results were discussed on the email list as well as at the fall 2004 meeting held in Toulouse, France. The outcome was that while there were many nationalistic issues regarding the use of encryption at one time, most of those issues have gone away with the increased use of commercial encryption. In essence, the issue had become a non-issue.

At the Spring 2005 CCSDS meeting held in Athens, Greece, the topic of selecting a CCSDS encryption algorithm was again discussed. Once again, the proposal to adopt Rijndael was made. However, at this meeting the suggestion of those in attendance at the Working Group meeting was first to perform an algorithm trade survey to examine other potential candidates before selecting one.

As a result, an action was taken to perform the research and write such an encryption algorithm tradeoff survey, contained herein.

3 CANDIDATE ALGORITHMS FOR CCSDS

3.1 GENERAL

A number of candidate algorithms are available for use by CCSDS. Among the notable set are the five finalist symmetric algorithms evaluated by the United States's National Institute of Standards and Technology (NIST) for the Advanced Encryption Standard (AES) selection. These algorithms are: Rijndael, Twofish, RC6, Serpent, and Mars.

In addition, there are other algorithms, such as the old U.S. Data Encryption Standard (DES), a variation of DES called Triple DES using multiple keys and multiple DES encrypt/decrypt actions; Blowfish, which is a predecessor of Twofish; IDEA, which has seen widespread usage in PGP; SEED, which is a standard in South Korea; KASUMI, which is used in GSM 3G UMTS; and TEA, which is an extremely small and simple algorithm using negligible amounts of memory or processor. In the asymmetric arena, the most well-known and utilized is RSA. However, for CCSDS use, RSA is not a viable candidate because of its key size and requirements for multiple round-trips.

A table of candidate algorithms and technical characteristics of the algorithms can be found in table 3-1. Examples of algorithm speeds on specialized security co-processor hardware can be found in table 3-2.

DES, undoubtedly the algorithm that has had the greatest impact on non-military cryptography, has long outlived its usefulness given its short, 56-bit key length. Given today's powerful desktop processors and the ability to perform parallel calculations via the Internet, performing a brute force attack on a 56-bit key is no longer a major or time-consuming challenge as has been shown time and again in recent history.

Triple DES (3DES), while stronger than DES by virtue of its use of multiple keys and multiple encrypt/decrypts, is excruciating slow in software and uses a tremendous amount of CPU power (in effect running DES three times to produce a single output data block). Both DES and 3DES should be eliminated from serious consideration by CCSDS because of their age and their consumption of resources.

An asymmetric algorithm such as **RSA**, while inviting because of its asymmetric keying, requires too much CPU (especially if asymmetric encryption is used) and too many round-trips to establish a traffic key (if symmetric encryption is used for actual traffic). RSA asymmetric should be eliminated because of the overhead involved and the typical lack of space-based resources.

There are several other groups of algorithms left to analyze: the AES finalists, the GSM/UMTS algorithm, and a few assorted independent algorithms. These are examined in the following pages.

Table 3-1: Potential CCSDS Encryption Algorithm Candidates

Algorithm Name	Algorithm Origin	Characteristics (block size/ (a)symmetric)	Key Sizes (bits)	Speed (MB/sec)[1]
DES	US (IBM)	64 bits/symmetric	56	21.34
Triple DES (3DES)	US	64 bits/symmetric	112 (2 keys) or 168 (3 keys)	9.84
Rijndael	Belgium	128 bits/symmetric	128/192/256	61.01
Serpent	UK, Israel, Norway	128 bits/symmetric	128/192.256	21.09
Twofish	US (Counterpane)	128 bits/symmetric	128/192/256	31.41
Blowfish	US (Counterpane)	64 bits/symmetric	256-448	64.38
RC6	US (RSA)	128 bits (variable)/ symmetric	128/192/256	37.81
TEA	UK (Cambridge)	64 bits/symmetric	128	23.8
IDEA	Switzerland	64 bits/symmetric	128	18.96
MARS	US (IBM)	128 bits/symmetric	128-1248	27.91
Kasumi (A5/3)	Europe (ETSI)	64 bits/symmetric	128	53[2]
SEED	South Korea (Korea Information Security Agency)	128 bit/symmetric	128	16[3] 4[4]
RSA	US (RSA)	N/A/asymmetric	1024-2048	Encrypt: 1024 bit key w/27607 iterations=0.18 ms/operation Decrypt: 1024 bit key w/1050 iterations=4.77 ms/operation

[1] Speeds obtained from Crypto++ 5.2.1 Benchmarks running on Pentium 4 2.1 GHz, Windows XP, www.eskimo.com/~weidai/benchmarks.html.

[2] Speed obtained from Motorola MPC185 Security Co-processor User's Manual (MPC185UM, dated 12/2003 rev 2.3), page 1-12, Table 1-2

[3] Speed obtained from SEED Algorithm Self Examination, Korea Information Security Agency, running on Pentium III 800 MHz MS Visual C++

[4] Speed obtained from SEED Algorithm Self Examination, Korea Information Security Agency, running on Pentium III 866 MHz Java 1.2.2

3.2 AES FINALIST ALGORITHMS

The five NIST AES finalist algorithms were:

- Rijndael;
- Twofish;
- Serpent;
- RC6; and
- MARS.

Ultimately, based on strenuous analysis and testing by the U.S. Government, worldwide private industry, and worldwide academia, Rijndael was selected to become the AES standard.

According to the NIST final selection report (reference [1]), none of the candidates had any cryptographic weaknesses. So the selection of Rijndael over the other algorithms was primarily based on its performance in hardware and software as well as resource frugality, that is, the minimization of CPU and memory requirements.

Table 3-2: Example of Performance of Algorithms (megabits/sec) on Motorola MPC 185 Security Co-Processor

	DES CBC	3DES CBC	AES 128	AES 256	ARC4	MD5	SHA-1	Kasurni	3DES/ HMAC-SHA-1(Rx)
64 byte	204	168	180	153	102	177	162	93	138
128 byte	355	260	281	239	176	311	279	154	237
256 byte	562	358	391	332	279	472	411	230	350
512 byte	815	449	489	415	404	636	540	316	459
1024 byte	1051	513	557	473	521	770	639	391	544
1535 byte	1164	538	585	497	595	828	681	426	579

While there are many public domain analyses of performance of AES algorithms, as illustrated in table 3-1 (typically on general usage hardware), table 3-2 illustrates a few algorithms' performance on a specialized security co-processor (Motorola MPC 185). Not only is Rijndael a performance leader on general purpose hardware, but it is also a performance leader on specialized, security co-processor hardware.

From the NIST AES development report and the independent analyses of algorithm performance and resource utilization, the conclusion is that among the AES finalists, Rijndael should be taken into consideration to be a CCSDS algorithm finalist to the exclusion of the other AES finalist algorithms. While all of them have been deeply analyzed, Rijndael has had even more scrutiny. All of them have been analyzed for resource usage and

performance, but in the past few years there has been a large amount of experience gathered on the resource usage and performance of Rijndael.

3.3 GSM/UMTS ALGORITHMS

GSM/UMTS 3GPP has specified algorithms for authentication and key generation: functions f1, f1*, f2, f3, f4, f5, and f5* (see reference [2]). The GSM/UMTS 'kernel' example algorithm specified for authentication and key generation is AES (Rijndael).

In addition, GSM UMTS 3GPP has specified a separate confidentiality and integrity algorithm: functions f8 and f9 (see reference [3]). This specification employs an algorithm known as KASUMI. KASUMI is a modification of the MISTY1 algorithm, and because of its use in GSM/UMTS 3G, it has been well analyzed (unlike the original GSM algorithms that were kept secret and were eventually exposed and broken).[5]

As can be seen from table 3-1, KASUMI is block cipher that operates over a 64-bit block (like DES) rather than a 128-bit block as is the case with the AES algorithms. Although the KASUMI algorithm uses a 128-bit key (like the AES algorithms), in fact the 128-bit key is derived from the original GSM/GPRS 64-bit key. The 64-bit key is duplicated and thus the effective key length remains 64 bits (see reference [4]). However, in 3GPP WCDMA systems, KASUMI does use a true 128-bit key but according to one analysis, a KASUMI 64-bit block would be theoretically vulnerable to a 2^{64} lookup-table attack whereas AES would require a significantly larger size 2^{128} table (see reference [4]).

KASUMI, while a strong algorithm, does not appear to be that much better than Rijndael, or for that matter any of the open source AES finalists. From the literature, it appears that KASUMI was chosen by 3GPP because it was felt that it could be implemented in specialized cellular phone hardware and made to run as fast as and with fewer resources than AES while using less power.

3.4 MISCELLANEOUS ALGORITHMS

Besides the AES finalists and the GSM/UMTS algorithm, there are a few others listed in table 3-1 that are discussed:

- Blowfish;
- TEA;
- IDEA; and
- SEED.

[5] Another reason to ensure that the encryption algorithm stand on its own merits of strength and not assume strength by obscurity of the algorithm.

Blowfish is the predecessor of the AES finalist Twofish. Unlike Twofish, Blowfish operates on 64-bit blocks with a variable size key ranging from 256 to 448 bits. The original premise for Blowfish's design was to improve upon the already recognized weaknesses of DES, particularly the small key size. While Blowfish has been well analyzed and appears to be a strong algorithm, there does not appear to be a good reason to adopt Blowfish over its modern relation Twofish, or for that matter any of the AES finalists.

TEA (Tiny Encryption Algorithm) was developed at Cambridge University in the UK with the requirements of being extremely small yet strong. It is a 64-bit block algorithm using a 128-bit key. And it is extremely simple: the encode and decode routines are each only 9 lines of C:

Routine for encoding with key k[0] - k[3]. Data in v[0] and v[1]:

```
void code(long* v, long* k)  {
unsigned long y=v[0],z=v[1], sum=0,   /* set up */
 delta=0x9e3779b9, n=32 ;             /* a key schedule constant */
while (n-->0) {                       /* basic cycle start */
  sum += delta ;
    y += (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1] ;
    z += (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3] ;   /* end cycle */
              }
v[0]=y ; v[1]=z ; }
```

Decode:

```
void decode(long* v,long* k)  {
 unsigned long n=32, sum, y=v[0], z=v[1],
 delta=0x9e3779b9 ;
sum=delta<<5 ;
                      /* start cycle */
while (n-->0) {
    z-= (y<<4)+k[2] ^ y+sum ^ (y>>5)+k[3] ;
    y-= (z<<4)+k[0] ^ z+sum ^ (z>>5)+k[1] ;
   sum-=delta ;  }
                      /* end cycle */
v[0]=y ; v[1]=z ;  }
```

While the code base is extremely small, the algorithm requires many rounds (64) and therefore extreme high speed is not a virtue of this algorithm, although the algorithm has been shown to run as fast as DES and much faster than Triple DES (see table 3-1). While TEA is truly simple, cryptographic weaknesses have been found and the number of rounds required to keep it secure may make it less than suitable for CCSDS environments.

IDEA (International Data Encryption Algorithm) was developed in Switzerland in the early '90s as a joint program of the Swiss Federal Institute of Technology (ETHZ) and Ascom AG. IDEA was developed as a DES replacement with the intent of being strong, small, fast, and resistant to known cryptographic attacks. Phil Zimmerman's PGP (Pretty Good Privacy) used the IDEA algorithm because, at the time, it was unencumbered by U.S. patents and was not developed in the U.S., hence (apparently) thwarting U.S. encryption legal issues. The

algorithm operates over a 64-bit block using a 128-bit key. Unlike DES, however, IDEA does not use substitution boxes (S-Boxes) as part of its underlying cryptology.

While IDEA is a strong algorithm, its speed as shown in table 3-1 is not terribly good. However, it has been shown to be faster in other implementations including hardware. But a major issue is that IDEA is both patented (covering most of Europe, Japan, and the U.S.) and licensed. Therefore for CCSDS to adopt IDEA, a license would have to be purchased from MediaCrypt. Given the number of other good, unencumbered algorithms available, it does not seem to make a lot of sense to adopt IDEA for CCSDS.

SEED is a 128-bit block, 128-bit key algorithm developed by the Korea Information Security Agency (KISA). It has been offered to the Internet Engineering Task Force (IETF) in the form of an RFC (reference [5]) for use with IETF security protocols. SEED has been analyzed both by KISA and independent analysts but has not been met with wide community acceptance to date. It has been adopted as a South Korean standard encryption algorithm. However, as seen in table 3-1, its performance in software was not outstanding (on an 800 MHz P-III). While it has been offered to the IETF, SEED has not found itself in wide-spread usage (as far as can be confirmed) other than by the Koreans. While it may have had a good deal of scrutiny, it certainly is not as widespread as other equivalently strong algorithms and therefore all things being equal, its not clear why CCSDS would standardize on the use of SEED.

4 CONCLUSIONS AND RECOMMENDATIONS

Regarding the AES finalist algorithms, it would appear to make the most sense to eliminate the ones not selected through the NIST selection process. NIST spent several years, in concert with public forums and academia to make their selection of Rijndael as the Advanced Encryption Standard. There does not appear to be a compelling reason to run counter to this selection. Therefore, from the list of AES candidates, it is proposed that all except for Rijndael be eliminated from contention for the CCSDS standard.

With respect to the miscellaneous list of algorithms, given that they each have their own pros and cons (e.g., Blowfish has been essentially superceded by Twofish, IDEA is patented and licensed, SEED has not seen widespread acceptance, and TEA is tiny but….) it is proposed that all of these be eliminated from contention as well.

With respect to the GSM/UMTS 3GPP KASUMI algorithm, it has seen and will see a great deal of use, exposure, and acceptance given the environment in which it is being used. However, the genesis of the algorithm was one that would run fast in custom hardware while using minimal power with adequate strength over airlinks. This sounds vaguely reminiscent of the CCSDS space environments, although the customer base for GSM custom hardware is significantly larger than the custom (and radiation hardened) hardware customer base for the space community. Also, the GSM 3GPP community is already using AES as the recommended core of their MILENAGE algorithm suite, so it is not clear what the compelling reason is to go with another algorithm that has not been as widely analyzed or as widely deployed.

As a result of performing the research and examining the data, it would appear the two best potential candidates are Rijndael and KASUMI. However, given the large, but relatively narrow focus of usage for KASUMI and the very large and very wide focus of usage for Rijndael, it is **proposed that CCSDS adopt Rijndael**, as specified in FIPS PUB 197 (reference [6]) as the CCSDS standard encryption algorithm. It is also recommended that standard modes of operation of the algorithm should be determined and specified in a CCSDS profile independent of this trade survey.