The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CCSDS CRYPTOGRAPHIC ALGORITHMS

INFORMATIONAL REPORT

CCSDS 350.9-G-2

GREEN BOOK

July 2023

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CCSDS CRYPTOGRAPHIC ALGORITHMS

INFORMATIONAL REPORT

CCSDS 350.9-G-2

GREEN BOOK

July 2023

AUTHORITY

Issue:	Informational Report, Issue 2
Date:	July 2023
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 *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

FOREWORD

This document is a companion to the CCSDS Cryptographic Algorithms specification (reference [1]). In this document, the reasoning and rationale for the use of specific algorithms and their respective modes of operation are discussed.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Report is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). 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 sent to the CCSDS Secretariat at the email 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.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (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.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (BELSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Netherlands Space Office (NSO)/The Netherlands.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 350.9-G-1	CCSDS Cryptographic Algorithms, Informational Report, Issue 1	December 2014	Original issue, superseded
CCSDS 350.9-G-2	CCSDS Cryptographic Algorithms, Informational Report, Issue 2	July 2023	Current issue

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

This Informational Report provides background information regarding the standard CCSDS cryptographic algorithms specified in (reference [1]). The CCSDS Cryptographic Algorithms Recommended Standard recommends the use of a single symmetric block-cipher encryption algorithm, the Advanced Encryption Standard (AES), to provide confidentiality. It also recommends several algorithms to provide authentication and integrity.

AES is the sole symmetric encryption algorithm that is recommended for use by all CCSDS missions and ground systems. Regarding the AES mode of operation, while other modes are allowed, the Recommended Standard recommends that the AES Counter Mode be used. Legacy implementations may use 128-bit keys, but for future missions, a key length of 256 bits is recommended. The Recommended Standard recommends the use of one or more alternative authentication/integrity algorithms, which may be chosen for use by individual missions depending on their specific mission environments.

The use of standardized, well-known algorithms and the use of high-quality cryptography helps ensure system security and interoperability. Economies of scale are also achieved when off-the-shelf, standardized, approved algorithms are universally used because they may be purchased rather than having to be implemented for a specific system or mission.

1.2 SCOPE

The algorithms discussed in this Informational Report have been recommended for use on all civilian space missions and ground systems with a requirement for information confidentiality and/or authentication. The algorithms may be employed on any or all mission communications links, such as the forward space link (e.g., telecommand) and the return space link (e.g., telemetry, science data), as well as across the ground data network. They could also be used to ensure confidentiality and authenticity/integrity of stored data (i.e., '*data at rest*').

Key management is not within the scope of this document but is discussed in the CCSDS Key Management documents (reference [10] and reference [11]) from which mission planners may select a key distribution/management technology that is a best fit to a specific mission.

1.3 APPLICABILITY

This Informational Report is applicable to all CCSDS space missions with a requirement for information confidentiality, authentication, or integrity.

While the use of security services is encouraged for all missions, the results of a threat/risk analysis and the realities of schedule and cost drivers may reduce or eliminate the need for them on a mission-by-mission basis.

1.4 RATIONALE

Traditionally, with the exception of commercial telecommunications missions, security mechanisms have not been widely employed on civilian space missions. However, in recognition of increased threat, there has been a steady migration towards the integration of security services and mechanisms.

This CCSDS Cryptographic Algorithm Informational Report discuses the background, rationale, and various other information regarding the specifications found in the *CCSDS Cryptographic Algorithms* Recommended Standard (reference [1]).

1.5 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] *CCSDS Cryptographic Algorithms*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 352.0-B-2. Washington, D.C.: CCSDS, August 2019.

[2] *Information Security Glossary of Terms*. Issue 2. Report Concerning Space Data System Standards (Magenta Book), CCSDS 350.8-M-2. Washington, D.C.: CCSDS, February 2020.

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

[4] *The Keyed-Hash Message Authentication Code (HMAC)*. Federal Information Processing Standards Publication 198-1. Gaithersburg, Maryland: NIST, July 2008.

[5] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*. National Institute of Standards and Technology Special Publication 800-38A. Gaithersburg, Maryland: NIST, December 2001.

[6] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*. National Institute of Standards and Technology Special Publication 800-38B. Gaithersburg, Maryland: NIST, May 2005 (Updated 10/6/2016).

[7] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. National Institute of Standards and Technology Special Publication 800-38D. Gaithersburg, Maryland: NIST, November 2007.

[8] *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-5. Gaithersburg, Maryland: NIST, February 3, 2023.

[9] Quynh Dang. *Recommendation for Applications Using Approved Hash Algorithms*. Revision 1. National Institute of Standards and Technology Special Publication 800-107. Gaithersburg, Maryland: NIST, August 2012.

[10] *Security Guide for Mission Planners*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.7-G-2. Washington, D.C.: CCSDS, April 2019.

[11] *Key Management*. Proposed Recommendation for Space Data System Standards, forthcoming.

[12] *Encryption Algorithm Trade Survey*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 350.2-G-1. Washington, D.C.: CCSDS, March 2008.

[13] *Authentication/Integrity Algorithm Issues Survey*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 350.3-G-1. Washington, D.C.: CCSDS, March 2008.

[14] *Information Technology—Security Techniques—Encryption Algorithms—Part 3: Block Ciphers*. 2nd ed. International Standard, ISO/IEC 18033-3:2010. Geneva: ISO, 2010.

[15] "New Attack on AES." August 18, 2011. Schneier on Security. https://www.schneier.com/blog/archives/2011/08/new_attack_on_a_1.html.

[16] Kenneth G. Paterson and Arnold K. L. Yau. "Cryptography in Theory and Practice: The Case of Encryption in IPsec." In *Advances in Cryptology – EUROCRYPT 2006*. 12–29. LNCS 4004. Berlin, Heidelberg: Springer, 2006.

[17] Gernot R. Bauer, Philipp Potisk, and Stefan Tillich. "Comparing Block Cipher Modes of Operation on MICAz Sensor Nodes." In *Proceedings of 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing, 2009 (18–20 Feburary 2009, Weimar)*. 371–378. Los Alamitos, CA, USA: IEEE Computer Society, 2009.

[18] "NIST Comments on Cryptanalytic Attacks on SHA-1." April 25, 2006. NIST.gov - Computer Security Division - Computer Security Resource Center. http://csrc.nist.gov/groups/ST/hash/statement.html.

[19] "NIST's Policy on Hash Functions." September 28, 2012. NIST.gov—Computer Security Division—Computer Security Resource Center. http://csrc.nist.gov/groups/ST/hash/policy.html.

[20] Andre Adelsbach. *Consultancy on Cryptographic Design*. Luxembourg: Telindus Belgacom ICT, 18 April 2008.

[21] *Space Data Link Security Protocol—Extended Procedures*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.1-B-1. Washington, D.C.: CCSDS, February 2020.

[22] Elaine Barker. *Recommendation for Key Management—Part 1: General*. Revision 5. National Institute of Standards and Technology Special Publication 800-57. Gaithersburg, Maryland: NIST, May 2020.

[23] T. Pornin. *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)*. RFC 6979. Reston, Virginia: ISOC, August 2013.

[24] S. Josefsson and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032. Reston, Virginia: ISOC, January 2017.

2 OVERVIEW

There is increasing awareness of the consequences of attacks against all types of electronic systems. Space flight and ground systems are not immune. While in many cases, sophisticated equipment, large amounts of power, and large antennas are needed to attack spacecraft, the cost and availability of such equipment has been reduced, making such attacks more viable.

As a result, it is in the best interests of all mission planners to ensure that spacecraft, their associated ground systems, and communications systems are adequately protected against attack, and that all transmitted data is protected as required.

The *CCSDS Cryptographic Algorithms* (reference [1]) specifies cryptographic algorithms for use by CCSDS to provide confidentiality, authentication, and integrity for both spacecraft and ground systems. This set of recommended algorithms establishes a common denominator among all missions for implementing information security services.

Undetected data modification or corruption is a major concern. Thus commands, parameters, tables, and software uploaded to a spacecraft must be authenticated to ensure that they are sent from only those individuals or control centers authorized to send commands, parameters, tables, and software. It must be assured that the commands received are exactly the same as sent with no intentional or unintentional errors, which, if not discovered, could result in a mission catastrophe. Similarly, modified or corrupted payload data from the spacecraft could result in erratic or wrong science. In some cases, depending on the mission requirements, the confidentiality of the commands or uploaded data must also be provided.

For the CCSDS community, confidentiality by cryptography is used to provide protection of information between end points that may be located on the ground and in space. In civilian space missions, confidentiality may be employed to ensure non-disclosure of information as it traverses the ground network and as it is transmitted between the ground and the spacecraft, between the spacecraft and the ground, and even onboard a spacecraft.

Engineering and scientific data sent by a spacecraft to the ground should be protected by confidentiality to ensure privacy. For example, access to Earth-observing-satellite data intended to be analyzed by contracted principal investigators should be restricted until the data has been publicly released. Likewise, proprietary engineering data, which might provide information on the internal workings of the system, should be protected.

For human-crewed missions there are concerns regarding the confidentiality of medical information conveyed onboard, across the space link, and over ground communications infrastructures. Similarly, private communications between crew members and their families, such as voice and email, must also be afforded confidentiality.

NOTE – The CCSDS Information Security Glossary (reference [2]) should be consulted for definitions of information security terms used in this document.

3 ENCRYPTION ALGORITHM AND MODE OF OPERATION

3.1 GENERAL

The CCSDS recommended confidentiality algorithm is AES (reference [3]) using the counter mode of operation. This recommendation is the result of an encryption algorithm trade study conducted by CCSDS (reference [12]). In addition, a second, independent study was conducted and resulted in the same recommendation (reference [20]).

3.2 AES OVERVIEW

AES is a symmetric, block-cipher algorithm operating over 128-bit blocks of data. The algorithm operates over a 128-bit plaintext input block and outputs 128-bits of ciphertext (encrypted) data. AES has been adopted by the United States as its official data encryption standard (reference [3]). ISO has also adopted AES as an international data encryption standard (reference [14]). AES has withstood the test of time and has been extremely resilient against attack (reference [15]). An overview flowchart of the AES algorithm is provided in figure 3-1. AES may be keyed using a 128-bit key, a 192-bit key, or a 256-bit key. Existing systems may continue to use 128-bit keys, but new systems should only use 256-bit keys.

3.3 AES MODES OF OPERATION

3.3.1 GENERAL

AES may be used in several modes of operation, such as Cipher-Block Chaining (CBC), Electronic Codebook (ECB), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) (reference [5]). Each of these modes accomplishes the same objective, namely transforming plaintext data into ciphertext data. More information and details on AES modes can be found in reference [5].

Each of these modes operates differently with different security characteristics. The chaining and feedback modes result in linkages from one cryptographic block to another, which means that if a block is lost or damaged, decryption will be significantly affected since the decryption process also relies on the block linkages.

In the ECB mode, each cryptographic block is separately enciphered and separately deciphered. Therefore the encipherment or decipherment of a block is totally independent of other blocks. Likewise, counter mode does not employ any linkage between blocks, and, as a result, cryptographic operations can be implemented in parallel. More information about counter mode can be found in 3.3.2.

When using ECB, identical plaintext blocks are encrypted into identical ciphertext blocks, and thus ECB mode does not hide any patterns in the data. ECB is considered to be extremely weak by the cryptographic community and should not be used.

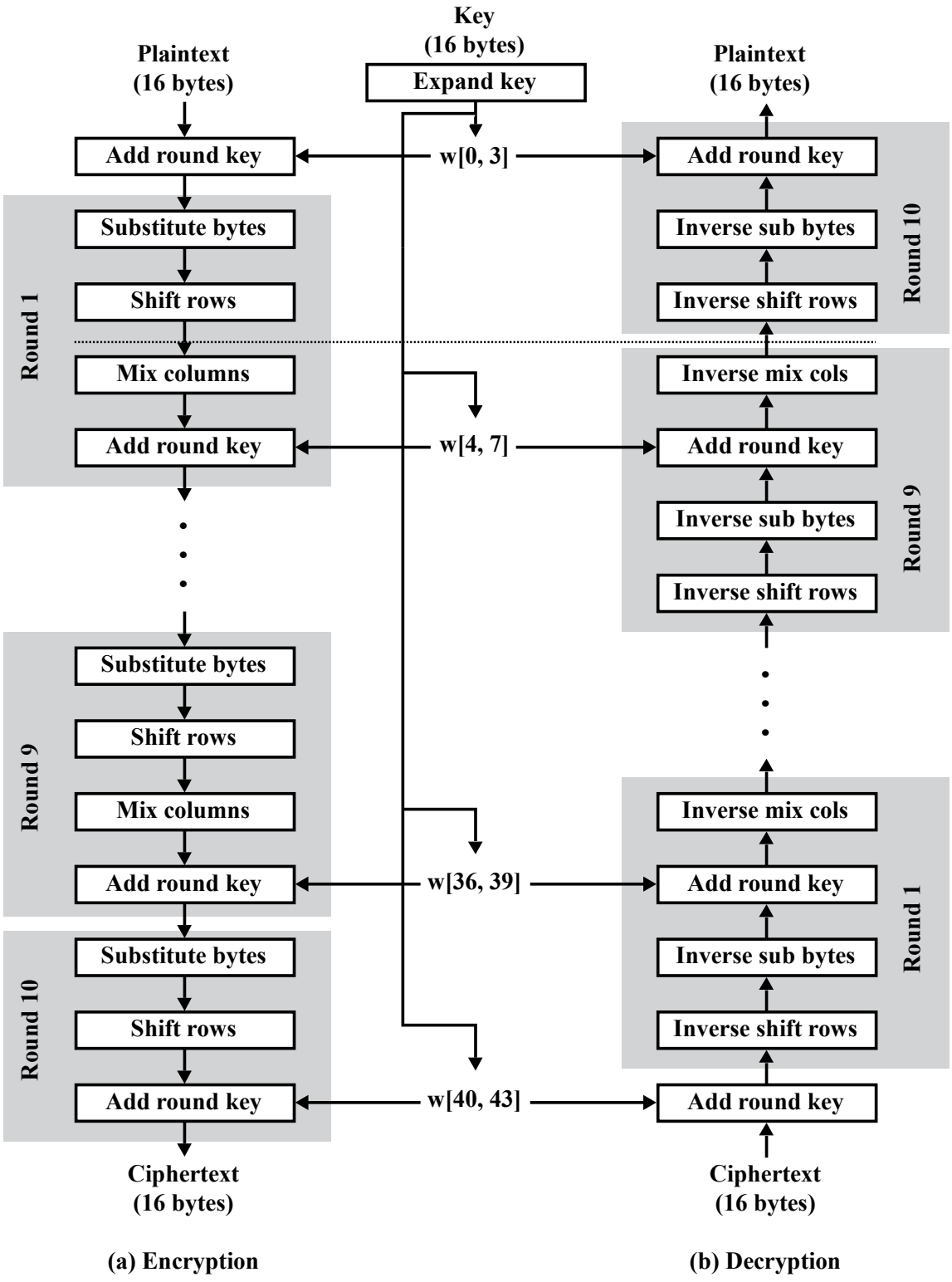

Figure 3-1: Overview of AES[1]

[1] *Source*: Stallings, William, *Network Security Essentials: Applications and Standards*. Used with author's permission.

The feedback and chaining modes, because they use block chaining, are considered more secure. However, CBC encryption requires sequential operation, and therefore cannot be parallelized. It also requires partial blocks to be padded to the full cryptographic block size (i.e., 128 bits). CFB is very similar to CBC but effectively turns the operation into a self-synchronizing stream cipher.

The OFB mode also turns the block cipher operation into a synchronous stream cipher. Like counter mode, it generates keystream blocks that are XORed with the plaintext blocks, resulting in ciphertext.

3.3.2 COUNTER MODE

3.3.2.1 General

The Counter Mode is the recommended mode, and unlike the feedback modes of operation, it creates ciphertext blocks that are entirely independent of one another. Therefore when using counter mode, encryption or decryption of multiple blocks may take place in parallel, since each block is an atomic, standalone entity. Counter mode is illustrated in figure 3-2.

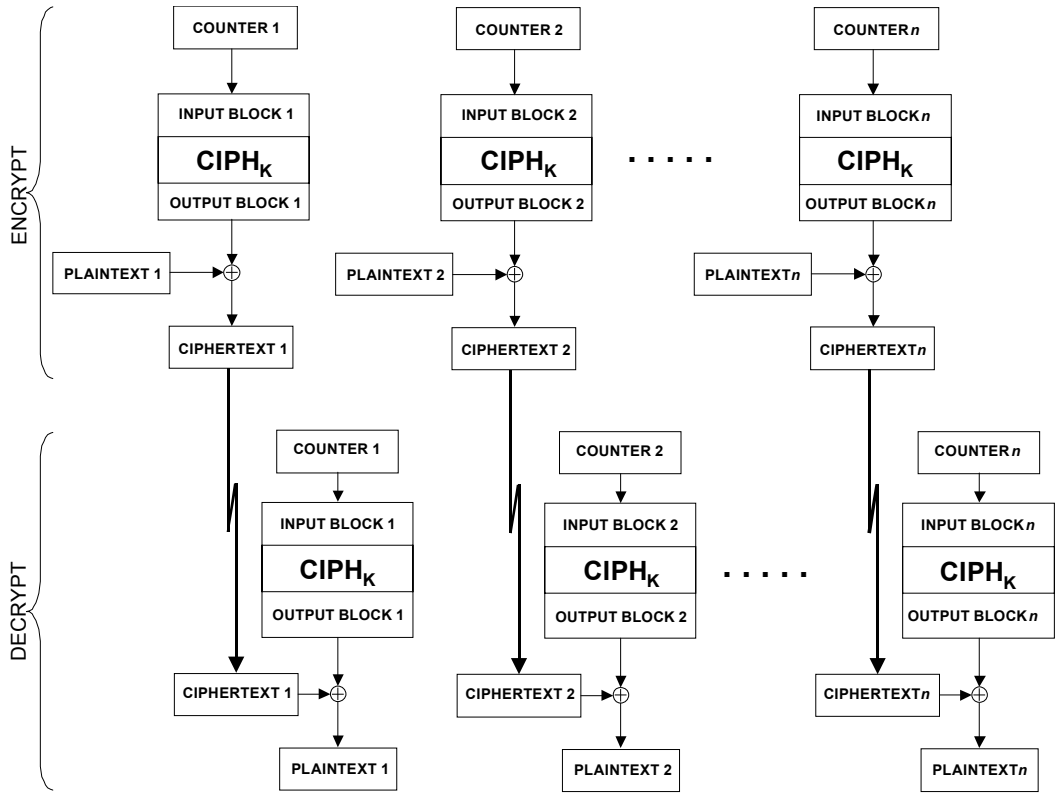

Figure 3-2: Counter Mode[2]

[2] *Source*: NIST Special Publication 800-38A (reference [5]), Figure 5.

Counter mode is a very efficient mode, especially for the CCSDS space environment. Most notably, counter mode operations can be implemented in hardware and pipelined because of the previously discussed block independence.

Counter mode is also efficient because it reduces cryptographic overhead by not requiring that partial blocks be padded as is required by all other AES cryptographic modes for encryption. When counter mode is not used and the input data is less than 128-bits, the input block must be padded with a fill pattern (e.g., all ones, all zeroes, alternating ones and zeroes) to increase the size of the block to 128-bits to create a full cipher block. When counter mode is used, padding is not required and there is no padding overhead.

Counter mode differs from other encryption modes because the plaintext data to be encrypted is not directly input into the AES algorithm. Rather, a counter, which has been combined with a cipher key, is used as the starting input to the algorithm. This produces 128-bit random data blocks. The block bits are XORed with the plaintext data to produce the output cipher blocks (see figure 3-2).

When using counter mode, if the last block of plaintext does not contain 128-bits, only the number of bits remaining are XORed with the previously produced key bits, and all of the other key bits are discarded.

To summarize, it was concluded that counter mode would have the best performance for both TM and TC (reference [20]). The advantages that make counter mode the recommended choice are:

- provable security based on weak assumption (underlying block cipher is a pseudo-random permutation);
- high efficiency;
- no error expansion;
- no block padding required;
- parallelizability;
- random-access property to support partial decryption, e.g., for cross-support services or routing.

3.3.2.2 Counter Mode ‘Counter’ Management

Counter mode requires the creation of a counter which does not have to be kept secret but must never repeat under the same key. If a counter is repeated, the confidentiality of the blocks encrypted under that counter may be compromised. If a counter overflows a new key must be used. Hence, proper counter management concept and implementation are crucial to counter mode security.

In order to ensure the selection of a unique counter, an incrementing function should be used from an initial counter. The initial counter must be chosen to ensure uniqueness across all blocks encrypted under a given key. A random set of bits may be used as the initial counter. Alternatively, a message *nonce* may be chosen and incorporated into every counter block. The specific methods of choosing an initial counter block and generating subsequent counter blocks is described in reference [7] (appendix B, page 18).

3.4 AUTHENTICATED ENCRYPTION

The cryptographic community has recognized that data encryption without data origin authentication often results in degraded security (reference [16]). The preferred approach by the cryptographic community is to combine independent authentication and encryption algorithms with different keys and particular constraints to be respected. However, in the search for efficiency and simplicity, implementers have sought for a cryptographic algorithm that could implement both confidentiality and authentication with the same cryptographic key while providing reasonable security. The result is authenticated encryption which provides both authentication and confidentiality.

Several authenticated encryption counter mode algorithms, providing both encryption and data origin authentication, have been developed (references [17] and [20]). These modes are called *Authenticated Encryption with Associated Data* (AEAD). One such mode, AES/GCM (Galois/counter mode) (reference [7]) can provide very high-speed authenticated encryption in hardware or software. AES/GCM is illustrated in figure 3-3. The CCSDS Cryptographic Algorithm Blue Book (reference [1]) recommends the use of AES/GCM when it is determined that authenticated encryption is required.

AES/GCM can be parallelized and pipelined, methods that can be very advantageous in the space community for low CPU overhead and high speed. In this way, a hardware implementation of AES/GCM can implement multiple encrypt/decrypt threads, and therefore each 128-bit block of data can be processed independently rather than serially. The only limiting factor on parallel operations is the number of independent, parallel paths built into the hardware. Likewise, software could be built to parallelize the encrypt/decrypt processing, increasing the overall speed of the encipherment process.

GCM combines AES counter mode with Galois authentication. Galois authentication employs Galois field multiplication, which can be computed in parallel, resulting in high throughput speeds.

The overall security strength provided by the AES/GCM mode is implementation dependent. For an authenticated encryption function with a given key, if even one Initialization Vector (IV) is ever repeated, the implementation may be vulnerable to forgery attacks. This condition is almost as important as the secrecy of the key. Therefore while there is no requirement that the IV be random or unpredictable, the AES/GCM security requires that the counter be a nonce (i.e., each value used at most once under the same key). Additionally, it is also recommended that CCSDS AES/GCM mode implementations restrict the IV size to 96-bits to promote interoperability, efficiency, and simplicity of design.

To prevent counter reuse, two frameworks for constructing IVs are recommended by NIST (reference [7]). These frameworks are defined as: 1) Deterministic Construction and 2) RBG-based Construction. While either framework is applicable to 96-bit IVs, it is required that exactly one of the constructions, but not both, be used across all instances of the authenticated encryption function with a given key.

In order to promote interoperability for the 96-bit IV, it is recommended that the deterministic construction be utilized, where the leading (i.e., leftmost) 32-bits of the IV hold a *fixed field* and the trailing (i.e., rightmost) 64-bits hold the *invocation field*. In principle, the *fixed field* should identify the context for the instance of the authenticated encryption function. The *invocation field* increments upon each invocation of the authenticated encryption function and could be implemented either as an integer counter or a linear feedback shift register driven by a primitive polynomial. To comply with the uniqueness requirement, for any given key, no two distinct devices are allowed to share the same *fixed field* and no two distinct sets of inputs to any single device are allowed to share the same *invocation field*.

A 96-bit IV is also recommended in *Space Data Link Security Protocol—Extended Procedures* (reference [21]). Specifically, the IV field length of the OTAR Command PDU, the Key Verification Reply PDU, and the TM, AOS, and USLP Security Associations is 96-bits.

Additionally, it is recommended that the authentication tags not be truncated to less than 96-bits, to minimize the possibility of successful message-forgery attacks. It is further recommended that the authentication tags be 128-bits in length, similar to the Message Authentication Code of reference [21].

Excessive use of the same key for the AES/GCM algorithm may break its confidentiality and integrity properties. Usage limits for Authenticated Encryption with Associated Data algorithms have been described in multiple venues and usually require complicated calculations. In principle, when the same key is utilized in the same session, the total number of invocations of the authenticated encryption function may not exceed 2^{32}, including all instances of the authenticated encryption function with the given key (reference [7]).

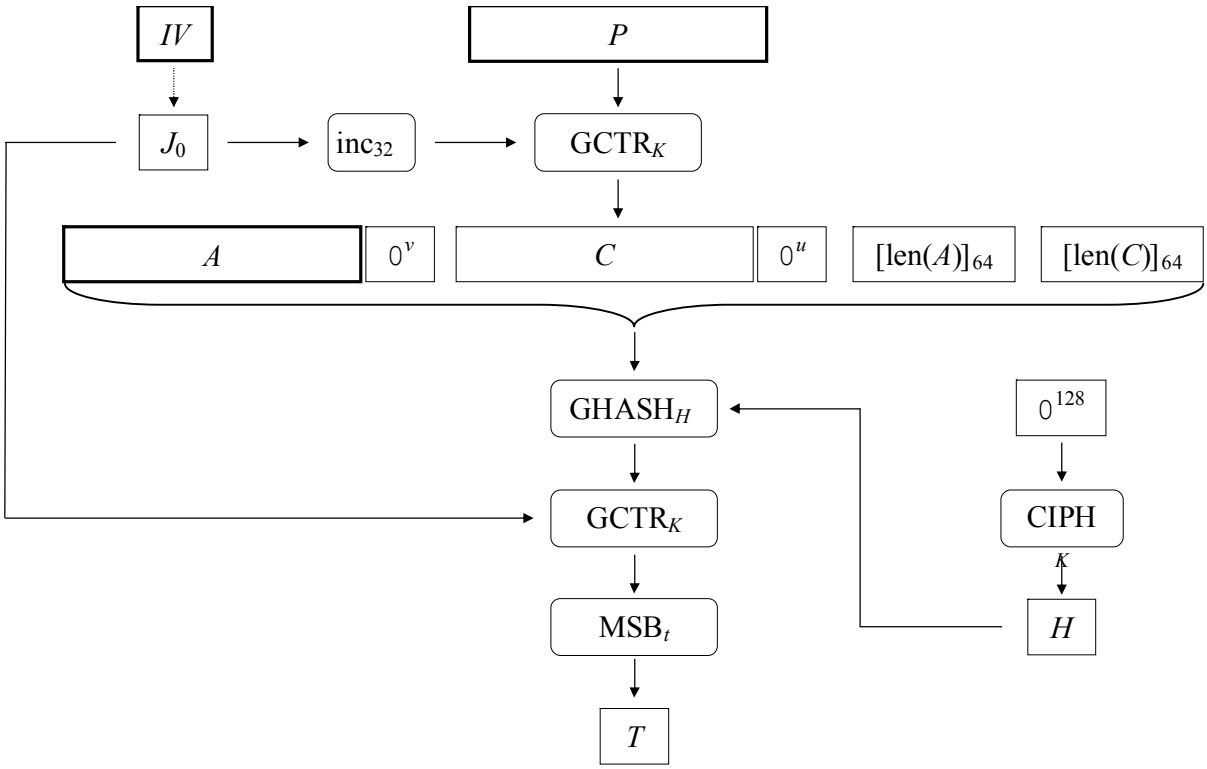

Figure 3-3: GCM Authenticated Encryption Function[3]

[3] *Source*: NIST Special Publication 800-38D (reference [7]), Figure 3.

4 AUTHENTICATION ALGORITHMS AND MODES OF OPERATION

4.1 GENERAL

Authentication and integrity are very important services to protect data communications. As has already been stated, commands must only be accepted from authorized sources and must not contain any errors. In either case, a mission may be threatened if authentication and integrity are not assured. While authenticated encryption provides confidentiality, authentication, and integrity services, in some cases confidentiality is not required (or not permitted) and only authentication/integrity services are needed.

Because a number of different algorithms can provide authentication/integrity services, the *CCSDS Cryptographic Algorithms* Recommended Standard provides several alternatives that may be used depending on mission needs. The recommended algorithms, selected as a result of an authentication algorithms tradeoff study (reference [13]), are:

- HMAC: a keyed hash algorithm; or
- CMAC (based on AES): a cipher-based hash algorithm; or
- RSA: a digital signature algorithm.

The keyed hash algorithm is considered to be 'light weight' in the sense that it does not use many CPU cycles but provides good authentication. The use of a keyed hash would work very well in the absence of any other algorithms that can be used for authentication on a mission, or in CPU-challenged environments, such as aboard spacecraft.

On the other hand, if an encryption algorithm were to be used on a mission, then the same algorithm could also be used to provide authentication/integrity services without the need for another onboard algorithm. Therefore CMAC may be appropriate when an approved block cipher algorithm is more readily available than an approved hash function.

If a spacecraft were part of a larger network (e.g., an IP-based constellation) and interactive links were available, then the use of digital signature technology, with public/private key negotiation, are desirable.

4.2 HASH MESSAGE BASED AUTHENTICATION

Hash-based authentication employs the properties of hashing algorithms, where an arbitrary input to the hash algorithm produces a fixed-size hash output. The resulting output is also known as a 'check word', a 'message digest', or an 'Integrity Check Value' (ICV). 'Message Authentication Code' (MAC) utilizes a cryptographic hash function in conjunction with a secret key.

In general terms, an n-bit cryptographic hash is defined as a primitive that generates n-bit hash values from arbitrary-length messages, and must be one-way and collision resistant. In

this context, one-way means that given a hash value, it will require about 2^n hash computations to identify a message that hashes to the same value. Collision resistance stipulates that identifying any two different messages that hash to the same value would require about $2^{n/2}$ hash computations. For example, SHA-256 is a 256-bit one-way hash and provides 128 bits of security against collision. SHA-384 is a 384-bit one-way hash and provides 192 bits of security against collision.

Figure 1 of reference [4] illustrates the nine-step HMAC keyed hash-based algorithm. A secret key is generated and shared securely between the source and destination(s) that will be receiving the MAC. The secret key is used in conjunction with the data that is input into the HMAC algorithm. The HMAC-256 algorithm produces a 32-byte MAC that is unique in that no input, other than the original, will generate the same MAC while using the same secret key.

HMAC does not specify a hash algorithm to be used but instead references various approved hash algorithms that may be used. The CCSDS Cryptographic Algorithms Blue Book (reference [1]) specifies the use of SHA-256 as CCSDS's default hash algorithm for use with HMAC. But it also allows the use of other approved hash algorithms as long as they are agreed to by the communicating parties.

As a result of potential attacks, SHA-1 should not be used. Instead SHA-256 is recommended as the minimum hash algorithm. SHA-224, SHA-384, SHA-512, and RIPE-160, among others, may also be used (reference [18], reference [19]).

In FIPS 198a, an earlier version of the specification, HMAC had been specified as a ten-step process. The final step was defined as the truncation of the MAC by selecting only the leftmost t bits from the total of L bits generated by the hash algorithm. For example, using SHA-1, the 160-bit MAC could be truncated to 96 bits. Rather than transmitting the entire 160 bits, only the leftmost 96 bits would be transmitted.

However, the latest HMAC revision described in FIPS 198-1 (reference [4]), removes the final truncation step from the algorithm specification, making it a nine-step process. Truncation issues are now addressed in detail in a separate NIST Special Publication 800-107 (reference [9]).

Truncation results in fewer bits being transmitted over the communications link and thus reduces overhead. By not transmitting the entire message authentication code, some additional security strength may be achieved, since anyone intercepting the message will not obtain the full MAC and the possibility of cryptanalysis will be limited. However, there is potentially more collision strength achievable at the receiver if all of the hash bits must be matched rather than just the truncated t bits. The security community is not unanimous one way or the other on the merits or weakness of truncation.

For CCSDS, truncation is an optional step which should be used only in those situations where bandwidth is critical and a means for reducing overhead is essential. Before the decision is made to employ truncation, the mission managers should first refer to Special Publication 800-107 (reference [9]) in order to fully understand the issues involved with its use.

4.3 CIPHER BASED AUTHENTICATION

CMAC (reference [6]) is a keyed hash function that is based on a symmetric key block cipher such as AES. Cipher-based authentication uses the properties of a block cipher algorithm, rather than a hash function, to create a MAC. Reference [1] specifies the use of CMAC with the AES algorithm for CCSDS. For existing CCSDS implementations, CMAC may use key sizes of 128 bits, 192 bits, or 256 bits. For future CCSDS implementations, only 256-bit keys are recommended.

AES-CMAC provides strong assurance of data integrity. It is stronger than a checksum or an error-detecting code which can only detect unintentional data modification. CMAC is designed to detect intentional, unauthorized data modification as well as unintentional modifications.

AES-CMAC achieves a security goal similar to that of HMAC. Since AES-CMAC is based on the symmetric key block cipher AES, and HMAC is based on a hash function such as SHA-2, AES-CMAC is appropriate for information systems in which AES is more readily available than a hash function.

Recognizing that a spacecraft might require both authentication and encryption services, the use of a cipher-based MAC might ameliorate onboard resources and simplifies flight system certification by requiring only a single algorithm (e.g., AES) for both security services.

For CCSDS, the Galois Message Authentication Code (GMAC) (reference [7]) may be used in place of CMAC when authenticated encryption is downgraded for authentication-only operations. For example, a mission may already employ authenticated encryption and therefore have AES/GCM onboard as part of the flight software. If other elements of the mission require authentication-only services, AES/GCM can be repurposed to provide this service without requiring another onboard algorithm.

CMAC is illustrated in figure 4-1.

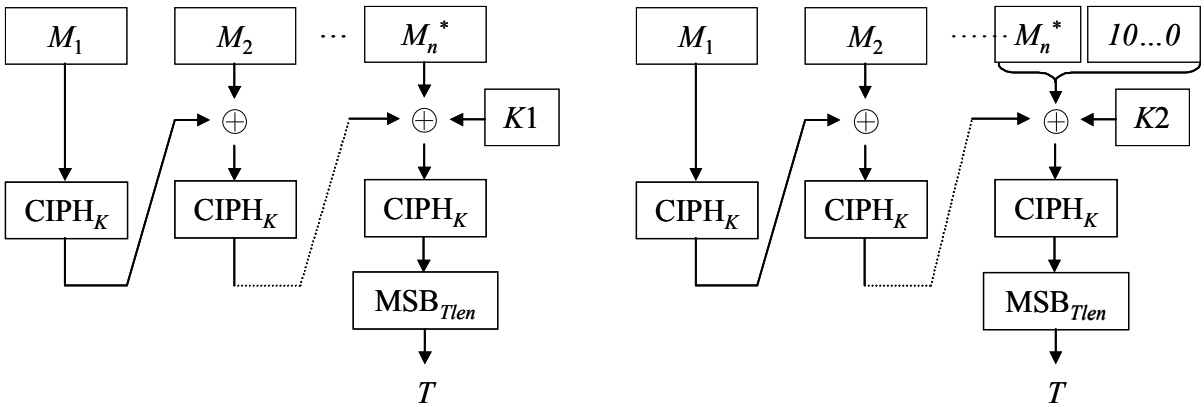

Figure 4-1: CMAC[4]

[4] *Source*: NIST Special Publication 800-38B (reference [6]), Figure 1.

4.4 DIGITAL SIGNATURE BASED AUTHENTICATION

The *Digital Signature Standard* (reference [8]) specifies several algorithms to construct and verify digital signatures: the Digital Signature Algorithm (DSA), the Rivest-Shamir-Adleman (RSA) Digital Signature Algorithm, and the Elliptic Curve Digital Signature Algorithm (ECDSA).

The RSA algorithm has become the de-facto commercial digital signature standard finding its way into internet applications such as electronic mail. RSA had been patented, but the patents have expired, and it is now a free and open algorithm. Therefore it has been chosen as the recommended digital signature algorithm for use in CCSDS missions. Both DSA and ECDSA are viable alternatives for CCSDS and may be used.

Digital Signatures employ asymmetric cryptography with public/private key pairs. Each communicating entity possesses a private key that is not shared with any other entity. It also possesses a public key that it shares with any other entity as needed.

Using public/private keys, a data originator is able to digitally sign using its private key. Recipients of the signed data use the originator's public key to authenticate the data. The high-level digital signature operation is illustrated in figure 4-2.

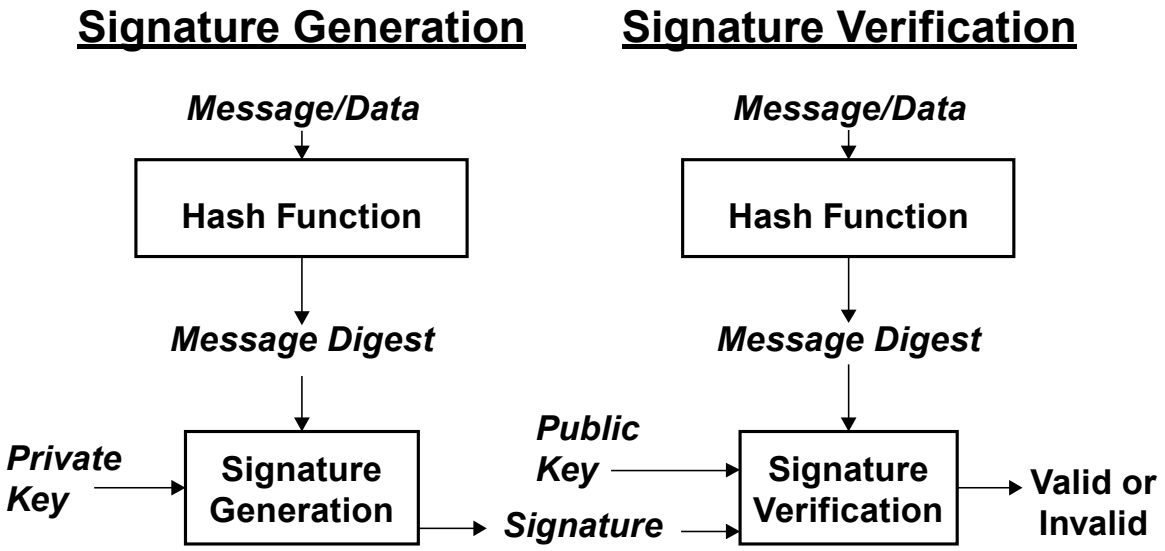

Figure 4-2: Digital Signature Generation and Verification[5]

The use of digital signature authentication is predicated on the ability to generate and share key pairs. Private keys are held by the key owner and are never shared. Public keys are shared either directly in a peer-to-peer manner or by storage and retrieval from a public key server.

[5] *Source*: http://itlaw.wikia.com/wiki/File:Snapshot_2009-07-29_20-53-38.gif.

A public key server may provide a repository of public keys available for retrieval on an as-needed basis. However, there is a potential trust issue when retrieving public keys from a public server. The public key needs to be bound to the actual identity of the entity that generated and posted the public key. Without such a binding, the retriever of a public key has no assurance that it came from the entity identified or from someone masquerading as the entity. Trusted third parties, known as Certificate Authorities (CAs), act as public key notaries and have the ability to provide assurance of public key authenticity by countersigning public keys after the generating entity has provided adequate proof-of-identity to the trusted third party.

For example, Person A generates a public key and provides proof-of-identity (e.g., government issued identity document) to a trusted third party. When the trusted third party is satisfied with the authenticity of the entity's identity, it will countersign the public key. Then anyone retrieving Person A's public key from the public key server is assured that the correct, authentic key has been received.

In a CCSDS environment, public keys stored on a key server would be generated by a national space agency, which would countersign the keys using its own, a government owned, or a publicly recognized CA. The national space agency's public key would be shared among the other national space agencies so that any countersigned public keys could be authenticated.

For spacecraft without the ability to contact a key server to obtain public keys, a local cache of public keys can be preloaded onto the spacecraft prior to launch. This requires that the public keys for all potential communicating entities be preloaded but can result in too many or too few keys loaded.

Alternatively, public keys could be uploaded after launch. An upload could supply additional keys or update keys that have expired or been compromised.

Ground systems are assumed to have robust network communications and access to a Public Key Infrastructure (PKI) or CAs.

NOTE – Cryptographic issues have been noted regarding early versions of the RSA algorithm from RSA Laboratories. As a result, NIST FIPS 186-5 (reference [8]) references only RSA PKCS #1 version 2.1. No earlier versions of RSA should be used.

For existing CCSDS implementations, the RSA Digital Algorithm key length may be 2048-bits. For future implementations it is recommended that the key length be 4096-bits.

In addition to the RSA Digital Algorithm, the CCSDS Cryptographic Algorithm Blue Book (reference [1]) states that DDS-specified algorithms such as the Elliptic Curve Digital Signature Algorithm (ECDSA (reference [8]) may also be used.

ECC is an attractive public-key cryptosystem, in particular for stressed environments. Compared to currently prevalent cryptosystems (e.g., RSA), ECC provides equivalent or

higher security using much smaller key sizes. In this context, security strength is a number associated with the amount of operations required to break a cryptographic algorithm or system. The security strength is specified in bits and is a specific value from the set {80, 112, 128, 192, 256}. Table 4-1 lists estimated comparable maximum-security strengths for integrity and confidentiality algorithms and key lengths (see reference [22]).

Table 4-1: ECC Security Strength vs. IFC

Security Strength	Symmetric Key Algorithms	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80		k = 1024	f = 160-223
112		k = 2048	f = 224-255
128	AES-128	k = 3072	f = 256-383
192	AES-192	k = 7680	f = 384-511
256	AES-256	k = 15360	f = 512+

In table 4-1, the first column shows the estimated maximum-security strength (in bits) provided by the algorithms and key sizes in a particular row. The second column identifies the symmetric-key confidentiality algorithms that can provide the security strength indicated in the first column. The third column indicates the value for k (the size of the modulus n and commonly considered to be the key size) for algorithms based on Integer-Factorization Cryptography (IFC), such as RSA. Column 4 shows the range of f (the size of n, where n is the order of the base point G, and commonly considered to be the key size) for algorithms based on ECC that are specified for digital signatures and key establishment.

The 192-bit and 256-bit key strengths identified for the IFC algorithms (e.g., RSA) are not included in the NIST standards for interoperability and efficiency reasons. Shown with a yellow background, the 7,680-bit and 15,360-bit RSA keys at these security strengths are too inefficient for space or terrestrial use. Additionally, algorithm/key-size combinations that have been estimated at a maximum-security strength of less than 112 bits (red background) are no longer recommended.

Therefore utilizing ECC with a curve such as the P-384 that uses a private key of 384-bits will provide a security strength of 192-bits, which is greater than RSA with a key length of 4096-bits. Smaller key sizes result in savings for power, memory, bandwidth, and computational cost, and this makes ECC especially attractive for constrained environments.

FIPS PUB 186-5 (Digital Signature Standard) (reference [8]) defines additional methods for digital signature generation and for the verification and validation of those digital signatures. These methods may be considered for future missions once their validation is covered in the NIST FIPS 140-3.

One of these methods is the deterministic ECDSA, which is a variant of ECDSA, where a *per-message secret number* is a function of the message that is signed, thereby resulting in a deterministic mapping of messages to signatures. This variant does not impact the signature verification process. IETF RFC 6979, *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)* (reference [23]), describes this deterministic digital signature generation procedure. The use of deterministic ECDSA may be desirable for devices that do not have an adequate source of quality random numbers.

FIPS PUB 186-5 also approves the use of the Edwards Curve Digital Signature Algorithm (EdDSA) and specifies additional requirements. EdDSA is a digital signature scheme that uses a variant of a Schnorr signature based on twisted Edwards curves. Prehash EdDSA (HashEdDSA) is a version of EdDSA in which the EdDSA signature is generated on the hash of the message rather than the message itself.

RFC 8032, *Edwards-Curve Digital Signature Algorithm* (reference [24]), describes EdDSA and specifies parameters for the *edwards25519* (Ed25519) and *edwards448* (Ed448) curves. EdDSA signatures are deterministic. A unique value computed with the hash of the private key and the message is used in the signature generation process. This process may protect against attacks arising from generating signatures with insufficient randomness for the per-message secret number. Ed25519 is intended to provide approximately 128-bits of security, and Ed448 is intended to provide approximately 224-bits of security.

HashEdDSA is a version of the EdDSA digital signature scheme. The main difference is that HashEdDSA generates a signature on the hash of the message *M*, unlike EdDSA, which signs the message *M* directly. The domain parameters and key generation for HashEdDS are exactly the same as for EdDSA, with the two options denoted Ed25519ph and Ed448ph.

5 SUMMARY

This Informational Report discusses the technology behind the selection of the algorithms specified in the *CCSDS Cryptographic Algorithms* Recommended Standard (reference [1]). It provides information concerning encryption, authentication, authenticated encryption, and modes of operation such as counter mode.

The CCSDS Recommended Standard provides specifics regarding the selected algorithms; this report provides additional insight and background material for use by CCSDS mission planners, or by others who require cryptographic security services.

ANNEX A

ABBREVIATIONS AND ACRONYMS

Term	Meaning
AEAD	Authenticated Encryption With Associated Data
AES	Advanced Encryption Standard
CA	Certificate Authority
CBC	Cipher-Block Chaining
CFB	Cipher Feedback
CMAC	Cipher-Based Message Authentication Code
CPU	Central Processing Unit
CTR	Counter
DSA	Digital Signature Algorithm
ECB	Electronic Codebook
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Keyed-Hash Message Authentication Code
ICV	Integrity Check Value
IFC	Integer Factorization Cryptography
IV	Initialization Vector
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OFB	Output Feedback
PKCS	Public-Key Cryptography Standard
PKI	Public-Key Infrastructure
RBG	Random Bit Generator
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm