The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CCSDS CRYPTOGRAPHIC ALGORITHMS

INFORMATIONAL REPORT

CCSDS 350.9-G-1

GREEN BOOK
December 2014

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

CCSDS CRYPTOGRAPHIC ALGORITHMS

INFORMATIONAL REPORT

CCSDS 350.9-G-1

GREEN BOOK

December 2014

AUTHORITY

Issue:	Informational Report, Issue 1
Date:	December 2014
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
E-mail: 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 e-mail 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 Federal Science Policy Office (BFSPO)/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.
- Chinese 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.
- 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.
- 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.
- Ministry of Communications (MOC)/Israel.
- 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.
- 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
EC 1	Editorial Correction	May 2015	Corrects erroneous minimum key length for AES on page 1-1.

CONTENTS

Section Page

Figure

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. In addition, a specific mode of operation (counter mode) for the algorithm and a minimum key length (128 bits) are 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-1. Washington, D.C.: CCSDS, November 2012.

[2] *Information Security Glossary of Terms*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 350.8-G-1. Washington, D.C.: CCSDS, November 2012.

[3] *Advanced Encryption Standard (AES)*. Federal Information Processing Standards 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, 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.

[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 186-3. Gaithersburg, Maryland: NIST, June 2009.

[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 1. Report Concerning Space Data System Standards (Green Book), CCSDS 350.7-G-1. Washington, D.C.: CCSDS, October 2011.

[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.

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. The Recommended Standard was developed as a result of the increasing interconnection of ground networks, the movement towards *joy-sticking* of instruments by principal investigators, the decreasing costs for hardware potentially allowing cheap *rogue* ground stations to be established, and national policies requiring enhanced mission security. This set of recommended algorithms establishes a common denominator among all missions for implementing information security services.

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. In some cases, depending on the mission requirements, the confidentiality of the commands or uploaded data must also be provided.

Engineering and scientific data sent by a spacecraft to the ground should be protected by confidentiality to ensure privacy and access controls. 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, medical data must be protected in consonance with national laws and policies for individual privacy.

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, an independent study performed by Telindus 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.

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, turning 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 strengths. 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 affected, since the decryption process also relies on the block linkages.

In the ECB mode, each cryptographic block is separately enciphered and separately deciphered. In other words, the encipherment or decipherment of a block is totally independent of other blocks. Likewise, counter mode also 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. It is considered to be extremely weak by the cryptographic community and should not be used.

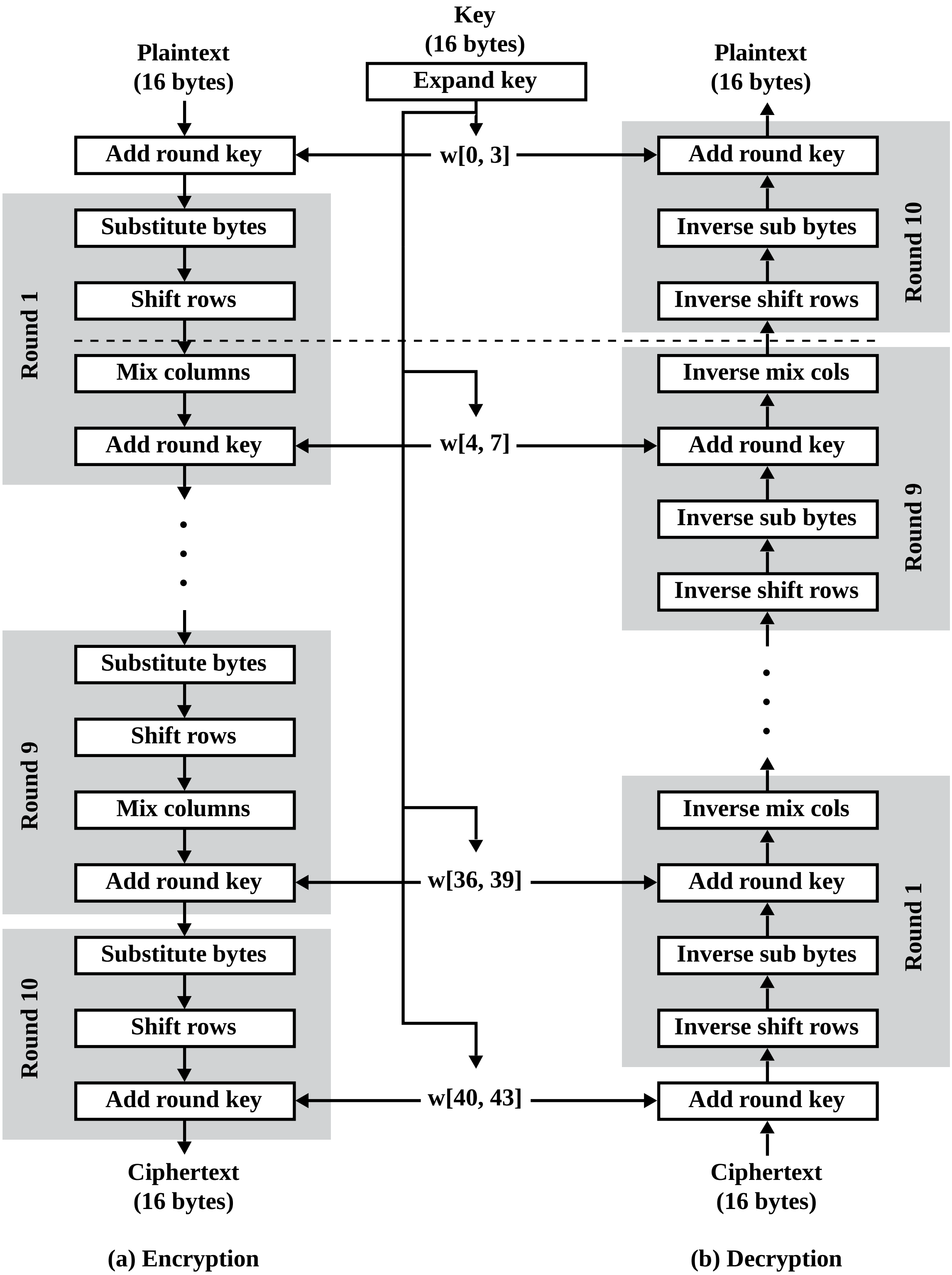

Figure 3-1: Overview of AES[1]

[1] *Source*: STALLINGS, WILLIAM, NETWORK SECURITY ESSENTIALS APPLICATIONS AND STANDARDS, 5th Edition, © 2014, p. 35. Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ.

The feedback and chaining modes, because they use block chaining, are much more secure. However, CBC encryption requires sequential operation, and therefore it 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.

OFB turns the block cipher operation into a synchronous stream cipher. Like counter mode, it generates keystream blocks, which are then XORed with the plaintext blocks to get the ciphertext.

3.3.2 COUNTER MODE

3.3.2.1 General

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

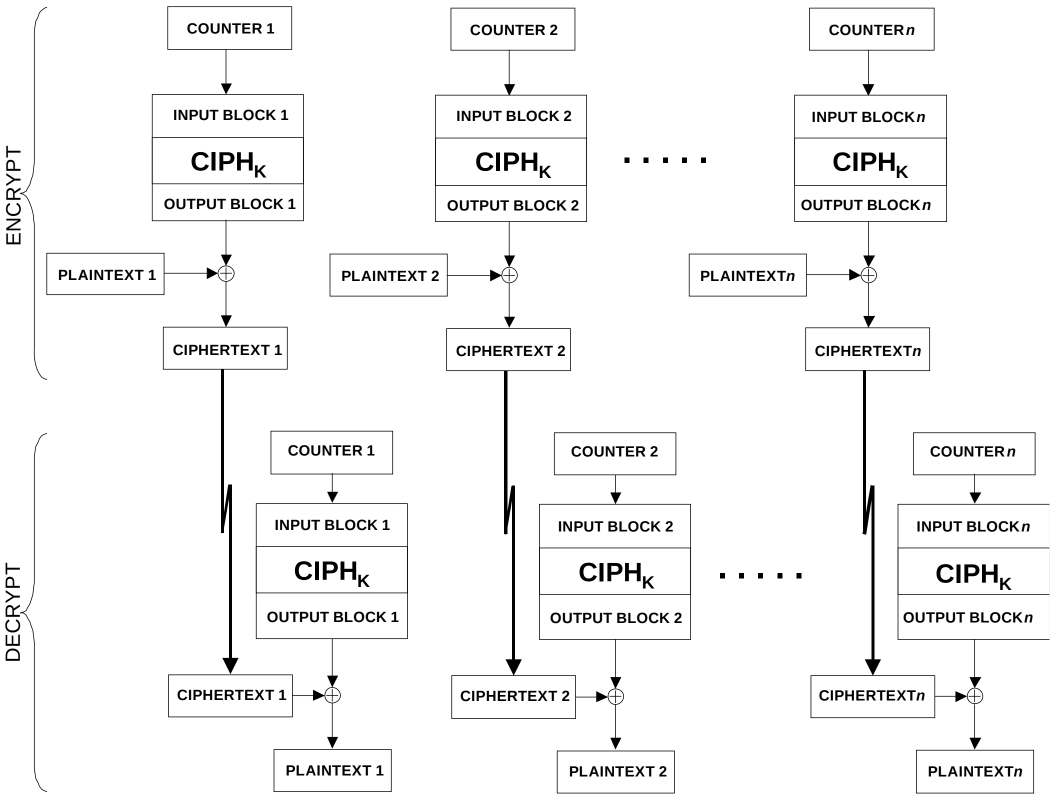

Figure 3-2: Counter Mode[2]

[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 run through 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 while a unique key is being used. If a counter is repeated, then 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 is also recommended in the Telindus study (reference [20]).

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 employees Galois field multiplication, which can easily be computed in parallel, resulting in higher throughput speeds.

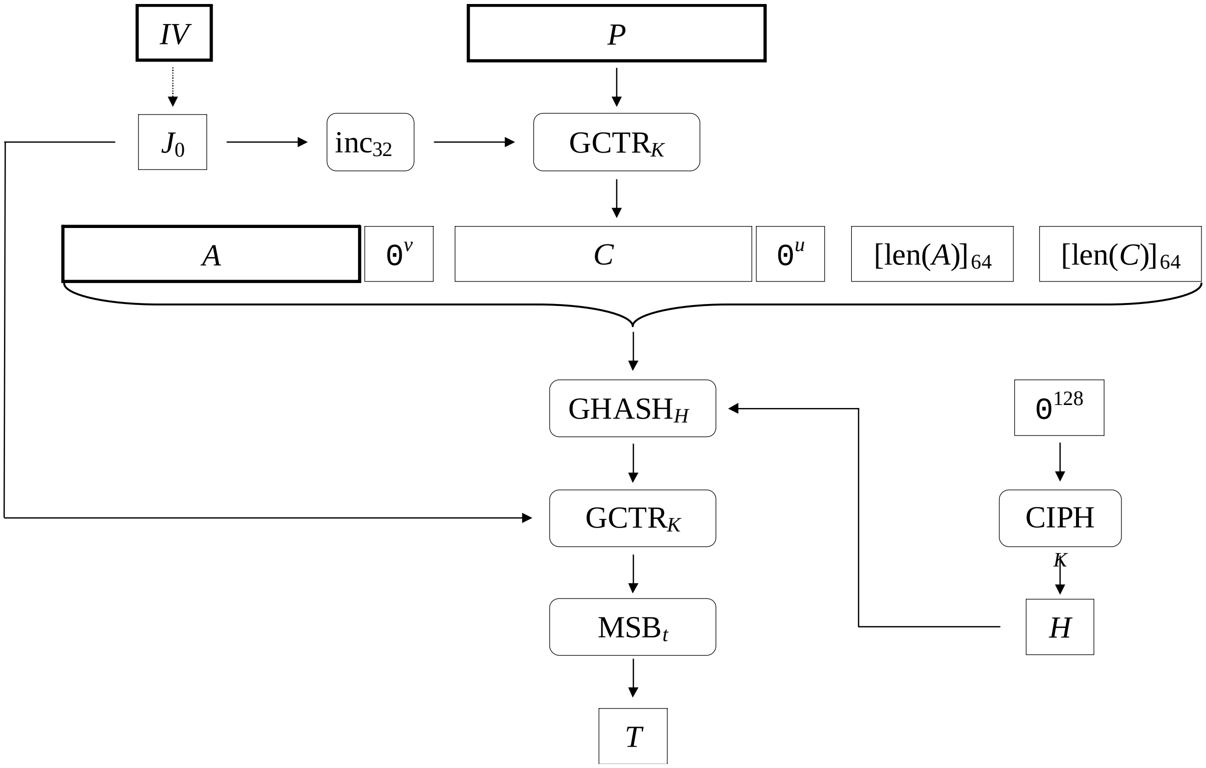

Figure 3-3: GCM Authenticated Encryption Function[3]

[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 be accepted only 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 implementation of that same algorithm could also be used to provide an authentication/integrity service without the need for another onboard algorithm. For example, CMAC employs AES to provide a message authentication code for authentication/integrity purposes.

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 its use of public/private key negotiation, might be 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 hash is also known as a “check word,” a “message digest,” a “Message Authentication Code” (MAC), or an “Integrity Check Value” (ICV). For every unique input, a unique cryptographic hash is produced. If only a single bit is changed in the input stream, a different, unique hash is produced. Characteristic of the hash algorithm is that there are no “hash collisions.” That is, no two different input streams will produce the same hash.

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 hash. The key is concatenated to the data that is input into the hash algorithm. The SHA-2(256) hash algorithm produces a 32-byte MAC that is unique, in that no input, other than the original, will generate the same MAC.

For authentication purposes, the data to be authenticated is concatenated with the secret key. The concatenated data is run through the HMAC algorithm. The data and the resulting MAC word are transmitted to the receiver(s). The receivers concatenate the secret key to the raw data to recalculate the MAC. If the resulting (recalculated) check word matches the transmitted MAC, the data has not been altered and is considered to be authentic.

HMAC does not specify a hash algorithm to be used but instead references various hash algorithms that may be used with the algorithm. The CCSDS Cryptographic Algorithms Blue Book (reference [1]) specifies the use of SHA-2(256) as CCSDS's default hash algorithm for use with HMAC. But it also allows the use of other hash algorithms as long as they are agreed to by the communicating parties.

As a result of potential attacks against SHA-1, it was decided that SHA-1 should not be used. Instead SHA-2(256) is recommended as the minimum hash algorithm; SHA-2(324), SHA-2(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 found 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 reduced 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 security strength achievable at the receiver if all of the hash bits must be matched rather than just 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 utilization.

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.

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 a 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 flight system certification by requiring only a single algorithm (e.g., AES) for both security services.

CMAC is illustrated in figure 4-1.

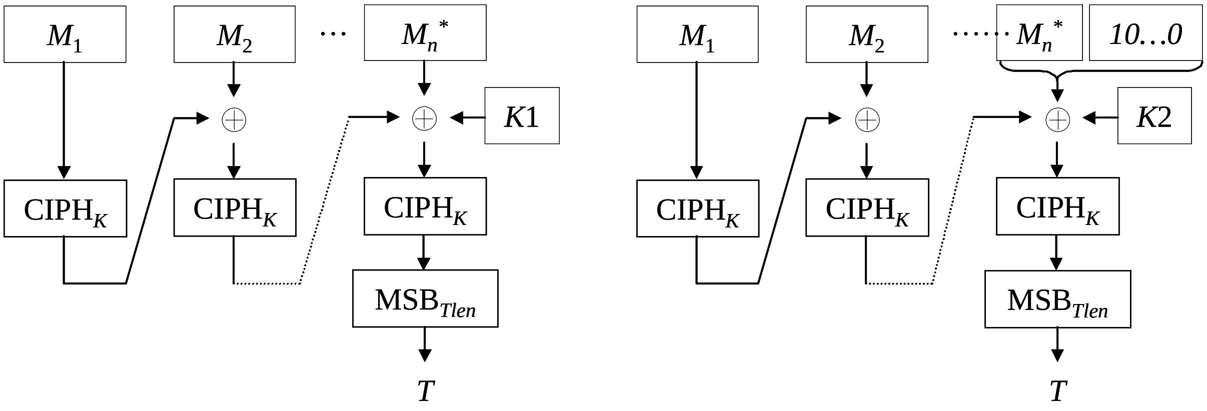

Figure 4-1: CMAC[4]

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

[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 standard, finding its way into many internet applications such as electronic mail. RSA had been a patented algorithm, but since its patents have expired, it is now an open and free algorithm. As a result, it has been chosen as the recommended digital signature algorithm by CCSDS. However, both DSA and ECDSA are viable alternatives for CCSDS and may be used if desired.

Digital Signatures employ asymmetric cryptography. Asymmetric cryptography does not require that secret keys be generated and distributed to all the communicating endpoints prior to commencement of transmission. Rather, asymmetric cryptography uses public/private key pairs. Each communicating entity possesses a private key that is shared with no other entity. It also possesses a public key that it shares with any other entity as needed.

Using the public/private keys, an originator of data is able to digitally sign it 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.

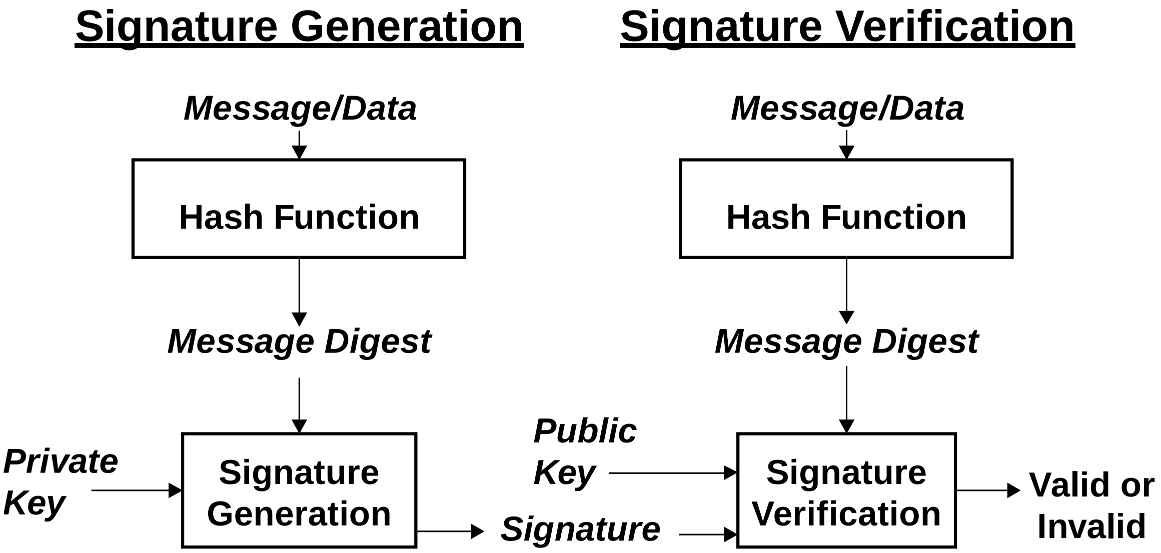

Figure 4-2: Digital Signature Generation and Verification[5]

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. But the 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 provides a repository of public keys available for retrieval on an as-needed basis. For example, if Spacecraft A needs to securely communicate with Ground Station Y and it has never communicated with Ground Station Y previously, it can retrieve Ground Station Y's public key from a key server, assuming one exists and is reachable by the spacecraft.

However, there is a potential trust issue with retrieving public keys from a public server. There needs to be a binding of the public key to the actual identity of the entity that generated and posted the public key. Without such a binding, a retriever of a public key has no idea if it came from the entity identified or from someone masquerading as the entity. Trusted third parties known as Certificate Authorities (CAs), acting as public key notaries, have the ability to provide assurances 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 of the authenticity of the entity's identity, it will countersign the public key. Anyone then 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 (e.g. Thawte, Verisign) 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 public key cache can be preloaded onto the spacecraft prior to launch. This would require that the public keys for all potential communicating entities be preloaded and the result might be that too many or too few keys are loaded.

Alternatively, public keys could be uploaded after launch. An upload could also supply additional keys needed 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 and should not be affected.

NOTE – Cryptographic issues have been noted regarding early versions of the RSA algorithm from RSA Laboratories. As a result, NIST FIPS 186-3 (reference [8]) references only RSA PKCS #1 version 2.1. No earlier versions of RSA should be used.

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
MAC	message authentication code
NIST	National Institute of Standards and Technology
OFB	output feedback
PKCS	Public-Key Cryptography Standard
PKI	public-key infrastructure
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm