

IEEE Standard for Wide-Block Encryption for Shared Storage Media

IEEE Computer Society

Developed by the
Cybersecurity and Privacy Committee

IEEE Std 1619.2™-2025
(Revision of IEEE Std 1619.2-2021)

STANDARDS

IEEE Standard for Wide-Block Encryption for Shared Storage Media

Developed by the

Cybersecurity and Privacy Standards Committee
of the
IEEE Computer Society

Approved 10 September 2025

IEEE SA Standards Board

Abstract: EME2-AES wide-block encryption with associated data (EAD) mode of the NIST AES block cipher, providing usage guidelines and test vectors, is described. A wide-block encryption algorithm behaves as a single block cipher with a large plaintext input and ciphertext output, but uses a narrow block cipher (in this case, Advanced Encryption Standard [AES]) internally. This encryption mode is oriented toward random access storage devices that do not provide authentication, but need to reduce the granularity of a potential attack.

Keywords: data-at-rest security, EAD, EME2, encryption, encryption with associated data, encrypt-mix-encrypt-v2 mode of operation, IEEE 1619.2™, security, storage

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2025 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 21 November 2025. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-2702-9 STD28327
Print: ISBN 979-8-8557-2703-6 STDPD28327

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all IEEE standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. “Artificial intelligence” refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](https://development.standards.ieee.org/myproject-web/public/view.html#landing).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/about/contact/>.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

At the time this standard was completed, the Security In Storage Working Group had the following membership:

Paul Suhler, *Chair*
Eric Hibbard, *Vice Chair*

Boon Chong Ang
Walker Arce
Richard Austin
Sridhar Balasubramanian
James Bordonaro
Jerrold Cady
Joseph Chen
John Driscoll
Silverin Duquenoy

John Geldman
Michael Harstrick
Jim Hatfield
Patrick Hery
Glen Jaquette
Chandra Nelogal
Anupam Pandey
Renee Schafer

Dai Shimogaito
Robert Strong
Gary Sutphin
Elochukwu Ukwandu
Lionel Uzzan
Arie van der Hoeven
Peng Wang
Qinggong Yue
Jemmee Yung

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Boon Chong Ang
Richard Austin
Sridhar Balasubramanian
James Bordonaro
Jerrold Cady
Joseph Chen

Thomas Coughlin
John Driscoll
John Geldman
Vinod Goje
Akram Hakiri
Eric Hibbard
Werner Hoelzl

Chandra Nelogal
Paul Suhler
Elochukwu Ukwandu
Hua Wang
Peng Wang
Jemmee Yung

When the IEEE SA Standards Board approved this standard on 10 September 2025, it had the following membership:

Lei Wang, *Chair*
John Walter Rosdahl, *Vice Chair*
David J. Law, *Past Chair*
Alpesh Shah, *Secretary*

Edward Au
Ted Burse
Xiaofeng (Alfred) Chen
Doug Edwards
Nehad El-Sherif
J. Travis Griffith
Deborah R. Hagar

Guido R. Hiertz
Ronald W. Hotchkiss
Tyler L. Jaynes
Thomas Koshy
Howard Li
Xiaohui Liu
Kevin W. Lu
Hiroshi Mano

Daleep C. Mohla
Annette D. Reilly
Robby Robson
Daniel Sabin
F. Keith Waters
Sha Wei
Luyang (Eric) Zhang

Introduction

This introduction is not part of IEEE Std 1619.2-2025, IEEE Standard for Wide-Block Encryption for Shared Storage Media.

The purpose of this standard, like IEEE Std 1619™, is to describe a method of encryption for data stored in logical block-based devices, where the threat model includes possible access to stored data by the adversary. As in IEEE Std 1619, this standard specifies length-preserving encryption algorithms to be applied to the plaintext logical block before storing it on the storage media.

This standard improves on IEEE Std 1619 by defining wide-block encryption algorithms. This means that they act on the whole logical block at once, and each bit on the input plaintext influences every bit of the output ciphertext (and vice versa for decryption). Specifically, this standard specifies the EME2-AES wide-block encryption algorithm.

Wide-block encryption better hides plaintext statistics and provides better protection than the narrow-block encryption, defined in IEEE Std 1619, against attacks that involve traffic analysis and/or manipulations of ciphertext on the raw storage media.

Contents

1. Overview	10
1.1 Scope	10
1.2 Purpose	10
1.3 Word usage	10
2. Normative references.....	11
3. Definitions, acronyms, and abbreviations	11
3.1 Definitions	11
3.2 Acronyms and abbreviations	11
4. Mathematical conventions.....	11
5. Wide-block encryption algorithms	12
5.1 Wide-block encryption overview.....	12
5.2 Encryption with associated data	12
5.3 EME2-AES algorithm	14
6. Compliance.....	20
Annex A (informative) Bibliography	21
Annex B (informative) Implementation guidance	22
B.1 Security considerations	22
B.2 Performance characteristics of EME2-AES.....	22
B.3 Application to logical block-level storage device encryption	23
Annex C (informative) Test vectors	24
C.1 Encoding	24
C.2 EME2-AES test vectors	24

IEEE Standard for Wide-Block Encryption for Shared Storage Media

1. Overview

1.1 Scope

This standard specifies an architecture for encryption of data in random access storage devices, oriented toward applications that benefit from wide encryption-block sizes of 512 bytes and above.

1.2 Purpose

This standard specifies an architecture for media security and enabling components. Wide encryption blocks are well-suited to environments where the attacker has repeated access to cryptographic communication or ciphertext, or is able to perform traffic analysis of data access patterns. The standard is oriented toward fixed-size encryption blocks without data expansion, but anticipates an optional data expansion mode to resist attacks involving data tampering.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{6,7}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

⁶ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁷ The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

NIST Federal Information Processing Standard 197 (FIPS 197), Advanced Encryption Standard (AES).⁸

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁹

associated data: Data that is associated with the plaintext, but that does not need to be encrypted. The associated data input characterizes the plaintext. In other algorithms for encryption of data at rest, the associated data is often referred to as tweak value.

encryption with associated data (EAD): A cryptographic algorithm that consists of an encryption procedure used to encrypt a plaintext and the associated data and a secret key, and a decryption procedure used to decrypt a ciphertext with the same associated data and secret key.

3.2 Acronyms and abbreviations

AES	advanced encryption standard (see Clause 2)
ECB	electronic codebook
EME	ECB-Mix-ECB (see [B1]) ¹⁰
EME2	ECB-Mix-ECB version 2
GF	Galois field
LBA	logical block address

4. Mathematical conventions

This standard uses decimal, binary, and hexadecimal numbers. For clarity, decimal numbers generally represent counts, and binary or hexadecimal numbers describe bit patterns or raw binary data.

Binary numbers are represented by a string of one or more binary digits, followed by the subscript 2. For example, the decimal number 26 is represented as 00011010₂ in binary.

⁸ NIST publications are available from the National Institute of Standards and Technology (<https://www.nist.gov/>).

⁹ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

¹⁰ The numbers in brackets correspond to those of the bibliography in Annex A.

Hexadecimal numbers are represented by a string of one or more hexadecimal digits, prefixed by the string “0x.” For example, the decimal number 135 is represented as 0x87 in hexadecimal.

The number of bits in a bit string X is denoted as $\#X$.

The procedures used in this standard are as follows:

- AES-Enc (K, X), the AES block cipher encryption (see NIST FIPS 197)¹¹, returns the encryption of the input X with the key, K .
- AES-Dec (K, X), the AES block cipher decryption, returns the decryption of the input X with the key, K .
- mult-by-alpha (X), the multiplication-by-alpha of a 16-byte value $X \in$ Galois Field (GF) (2^{128}) by a primitive element α in the field GF(2^{128}), defined in 5.3.1.
- $X \oplus Y$, the addition of two elements $X, Y \in$ GF (2^{128}) that is equivalent to the bitwise exclusive-or operation.
- Bitlength (X) returns an 8-byte string containing the non-negative integer describing the number of bits in the input X , with the most significant bit first and the least significant bit last. The expression 0^n denotes a string of n zero bits, and $A|B$ denotes the concatenation of two bit strings A and B .
- Bytlength (X) returns as an integer the number of bytes in the input X .
- $\text{msb}_t(X)$ returns the initial t bits of the input X . Bit strings are indexed starting from the most significant bit, so that bit zero of X is the most significant bit. $X[a:b]$ denotes the substring of X from the a th bit through the b th.

5. Wide-block encryption algorithms

5.1 Wide-block encryption overview

Wide-block encryption algorithms act on the whole logical block at once, and each bit on the input plaintext influences every bit of the output ciphertext (and vice versa for decryption).

Wide-block encryption better hides plaintext statistics and provides better protection than the narrow-block encryption (see [B2]), against attacks that involve traffic analysis and/or manipulations of ciphertext on the raw storage media.

5.2 Encryption with associated data

5.2.1 Data units and associated data

This standard specifies a length-preserving encryption with associated data (EAD) algorithm that is suitable for the encryption of data at rest. An EAD algorithm consists of an encryption procedure and a decryption procedure. The encryption procedure accepts three inputs as follows:

- Secret key
- Plaintext

¹¹ Information on references can be found in Clause 2.

— Associated data

It returns a single ciphertext value. Each of these inputs is a byte string.

The security of these procedures depends on the secret key consisting of random or pseudo-random data that is unpredictable to the adversary. Each EAD algorithm accepts a key of a fixed length, but some algorithms may have keys of different lengths.

The plaintext input contains the data to be encrypted. Plaintext is divided into data units that, within a particular key scope, are able to have keys of different lengths. An EAD algorithm defines the range of admissible plaintext lengths. A compliant implementation should keep the size of the data units the same within a key scope. If an implementation allows variable-length data units within a key scope, then the implementation's design should be such that it is able to unambiguously determine the length of each data unit without using data that is potentially under the control of an adversary. A compliant implementation shall use distinct associated data for distinct data units.

The associated data input contains data that is associated with the plaintext, but that does not need to be encrypted. The choice of data for this input is described in more detail as follows. Within a particular key scope, associated data units may have different lengths.

The ciphertext returned by the encryption procedure is the same length as the plaintext.

The decryption procedure accepts three inputs as follows:

- Secret key
- Ciphertext
- Associated data value

It returns a single plaintext value. The decryption procedure is the reverse of the encryption procedure; more specifically, if the encryption of the plaintext P with the key K and the associated data A results in the ciphertext C , then the decryption of C with the key K and the associated data A results in the plaintext P .

A conforming implementation shall include in the associated data only information that is available, in plaintext form, at the time of encryption and at the time of decryption.

The associated data input shall uniquely identify the plaintext. This is because whenever the same plaintext is encrypted two different times using the same key but with distinct associated data values, the result is two distinct ciphertext values. Thus, the use of distinct associated data values hides the equality of the plaintexts from an attacker.

5.2.2 Using EAD to protect a string of data blocks

An EAD protects a string of data blocks, such as those in a storage device. In this application, the associated data input to the encryption and decryption procedure should contain the logical index of the block on which the procedure is acting. When this information is included in the associated data, cases in which two distinct data blocks contain identical plaintext values are hidden from an adversary. Figure 1 shows an example of how an EAD performs encryption and decryption.

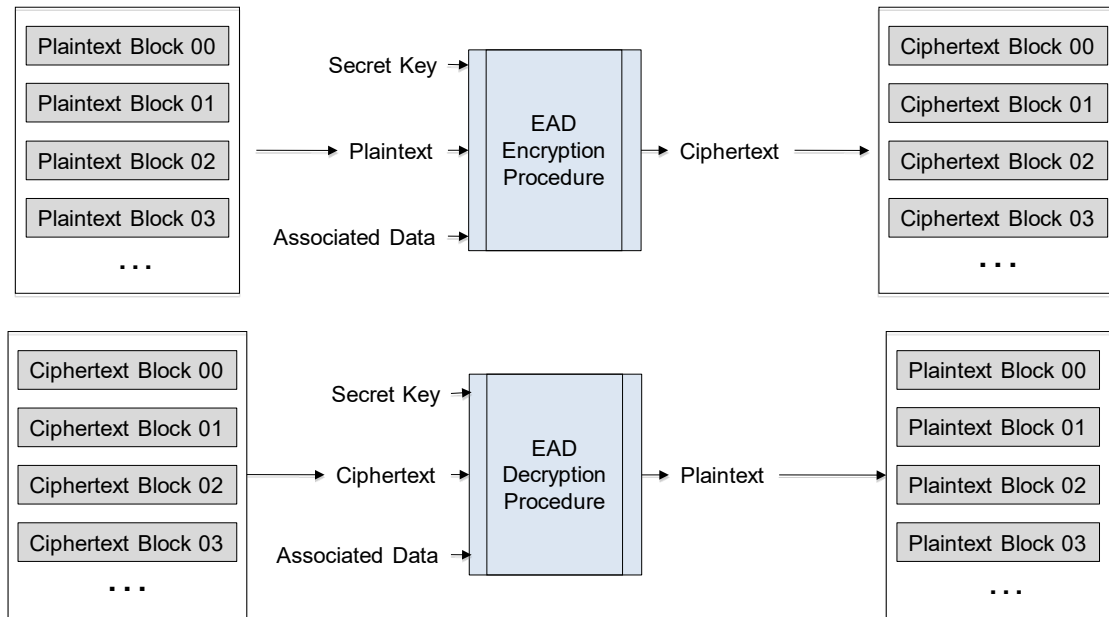


Figure 1—EAD encryption and decryption of a block device

If multiple storage devices are being protected with a single secret key, then the associated data input should contain both the logical index of the block and an additional distinguishing parameter that is unique to each of the storage devices. When this information is included in the associated data, cases in which two distinct data blocks on different storage devices contain identical plaintext values are hidden from an adversary.

5.3 EME2-AES algorithm

5.3.1 The mult-by-alpha procedure

The encryption and decryption procedures described in the following subclauses use a mult-by-alpha(X) procedure (see [B3]) that multiplies a 16-byte value X by a primitive element α in the field $GF(2^{128})$. The input value is first converted into a byte string $X[i]$, $i = 0, 1, \dots, 15$, where $X[0]$ is the first byte of the byte string.

The multiplication by alpha is defined in Table 1 or by a mathematically equivalent procedure.

Table 1—The mult-by-alpha(X) procedure

```
// compute mult-by-alpha(X), where X is a 16-byte value
//
// X[i] denotes the ith byte of X, and indexing starts at 0
//
// i is an integer used in loops
//
// Y[i] denotes the ith byte of the output Y, and indexing starts at 0

for i = 0 to 15 do
    Y[i] = 2*X[i] mod 256
    if (i>0 and X[i-1]>127) then Y[i]=Y[i]+1
end-for
if (X[15] > 127) then Y[0] = Y[0] ⊕ 0x87
return Y
```

Conceptually, the procedure is a left shift of each byte by one bit with carry propagating from one byte to the next. Also, if the 15th (last) byte shift results in a carry, a special value (hexadecimal 0x87) is xor'ed into the first byte. This value is derived from the modulus of the Galois Field (polynomial $x^{128} + x^7 + x^2 + x + 1$).

5.3.2 EME2-AES encryption

Equation (1) describes the EME2-AES encryption procedure:

$$C = \text{EME2-AES-Enc}(Key, T, P) \quad (1)$$

where

- Key* is the 48- or 64-byte EME2-AES key
- T* is the associated data, of arbitrary byte length (zero or more bytes)
- P* is the plaintext, of length 16 bytes or more
- C* is the ciphertext resulting from the operation, of the same byte length as *P*

The input to the EME2-AES encryption procedure is parsed as follows:

- The key is partitioned into three fields, $Key = K_{AD} \mid K_{ECB} \mid K_{AES}$, with K_{AD} (the associated data key) consisting of the first 16 bytes, K_{ECB} (the ECB pass key) consisting of the following 16 bytes, and K_{AES} (the AES encryption/decryption key) consisting of the remaining 16 or 32 bytes.
- If not empty, the associated data *T* is partitioned into a sequence of blocks $T = T_1 \mid T_2 \mid \dots \mid T_r$, where each of the blocks $T_1, T_2, \dots, T_{r-1}$ is of length exactly 16 bytes, and T_r is of length between 1 and 16 bytes.
- The plaintext *P* is partitioned into a sequence of blocks $P = P_1 \mid P_2 \mid \dots \mid P_m$, where each of the blocks $P_1, P_2, \dots, P_{m-1}$ is of length exactly 16 bytes, and P_m is of length between 1 and 16 bytes.

As part of the ciphertext computation, the associated data T are processed by the sequence of steps in Table 2 or by a mathematically equivalent procedure.

Table 2—The function H for processing the associated data T

```
// Function H( $K_{AES}$ ,  $K_{AD}$ ,  $T = (T_1 \mid \dots \mid T_{r-1} \mid T_r)$ ) :  
if bytelength( $T$ ) == 0 then  
   $T_{star} = \text{AES-Enc}(K_{AES}, K_{AD})$   
else  
   $K_{AD} = \text{mult-by-alpha}(K_{AD})$   
  for  $j = 1$  to  $r-1$   
     $TT_j = \text{AES-Enc}(K_{AES}, K_{AD} \oplus T_j) \oplus K_{AD}$   
     $K_{AD} = \text{mult-by-alpha}(K_{AD})$   
  next  
  if bytelength( $T_r$ ) < 16 then  
     $T_r = T_r \mid 0x80 \mid 0x00 \mid \dots$  // pad  $T_r$  to 16 bytes, 0x80 followed by 0's  
     $K_{AD} = \text{mult-by-alpha}(K_{AD})$   
  endif  
   $TT_r = \text{AES-Enc}(K_{AES}, K_{AD} \oplus T_r) \oplus K_{AD}$   
   $T_{star} = TT_1 \oplus TT_2 \oplus \dots \oplus TT_r$   
endif  
return  $T_{star}$  // return the 16 byte value  $T_{star}$ 
```

A conforming implementation shall compute the ciphertext by executing the sequence of steps in Table 3 or a mathematically equivalent procedure.

Table 3—The EME2-AES encryption procedure

```
// EME2-AES-Enc( $K_{AES}$ ,  $K_{ECB}$ ,  $K_{AD}$ ,  $T$ ,  $P = (P_1 \mid \dots \mid P_{m-1} \mid P_m)$ ):
 $T\_star = H(K_{AES}, K_{AD}, T)$  // Process the associated data
if bytelength( $P_m$ ) == 16 then
    lastFull = m
else
    lastFull = m-1
     $PPP_m = P_m \mid 0x80 \mid 0x00 \mid \dots$  // Pad  $P_m$  to 16 bytes, 0x80 followed by 0's
endif

// First ECB pass
 $L = K_{ECB}$ 
for j = 1 to lastFull
     $PPP_j = AES-Enc(K_{AES}, L \oplus P_j)$ 
     $L = \text{mult-by-alpha}(L)$ 
next

// Intermediate mixing
 $MP = PPP_1 \oplus PPP_2 \oplus \dots \oplus PPP_m \oplus T\_star$ 
if bytelength( $P_m$ ) < 16 then
     $MM = AES-Enc(K_{AES}, MP)$ 
     $MC = MC_1 = AES-Enc(K_{AES}, MM)$ 
else
     $MC = MC_1 = AES-Enc(K_{AES}, MP)$ 
endif
 $M = M_1 = MP \oplus MC$ 
for j = 2 to lastFull
    if (j-1 mod 128 > 0) then // use the current mask M
         $M = \text{mult-by-alpha}(M)$ 
         $CCC_j = PPP_j \oplus M$ 
    else // calculate a new mask M
         $MP = PPP_j \oplus M_1$ 
         $MC = AES-Enc(K_{AES}, MP)$ 
         $M = MP \oplus MC$ 
         $CCC_j = MC \oplus M_1$ 
    endif
next
if lastFull < m then
     $C_m = P_m \oplus (MM \text{ truncated to bytelength}(P_m) \text{ bytes})$ 
     $CCC_m = C_m \mid 0x80 \mid 0x00 \mid \dots$  // Pad  $C_m$  to 16 bytes, 0x80 followed by 0's
endif
 $CCC_1 = MC_1 \oplus CCC_2 \oplus \dots \oplus CCC_m \oplus T\_star$ 

// Second ECB Pass
 $L = K_{ECB}$ 
for j = 1 to lastFull
     $C_j = AES-Enc(K_{AES}, CCC_j) \oplus L$ 
     $L = \text{mult-by-alpha}(L)$ 
next

return  $C = (C_1 \mid \dots \mid C_{m-1} \mid C_m)$ 
```

An illustration of the EME2-AES encryption procedure (for plaintext of 130 full blocks and one partial block) is provided in Figure 2.

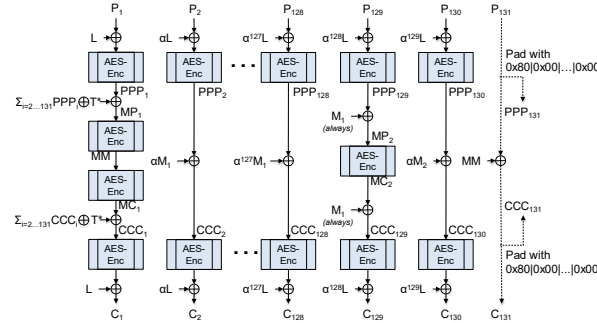


Figure 2—An illustration of EME2-AES encryption

In Figure 2, each AES-Enc block has a key input that is not shown. T^* is computed from the associated data using the function H from Table 2, and the M_i 's are computed as $M_i = MP_i \text{ xor } MC_i$.

5.3.3 EME2-AES decryption

Equation (2) describes the EME2-AES decryption procedure:

$$P = \text{EME2-AES-Dec}(Key, T, C) \quad (2)$$

where

- Key is the 48- or 64-byte EME2-AES key
- T is the associated data, of arbitrary byte length (zero or more bytes)
- C is the ciphertext, of length 16 bytes or more
- P is the plaintext resulting from the operation, of the same byte length as C

The input to the EME2-AES decryption procedure is parsed as follows:

- The key is partitioned into three fields, $Key = K_{AD} \mid K_{ECB} \mid K_{AES}$, with K_{AD} (the associated data key) consisting of the first 16 bytes, K_{ECB} (the ECB pass key) consisting of the following 16 bytes, and K_{AES} (the AES encryption/decryption key) consisting of the remaining 16 or 32 bytes.
- If not empty, the associated data is partitioned into a sequence of blocks $T = T_1 \mid T_2 \mid \dots \mid T_r$, where each of the blocks $T_1, T_2, \dots, T_{r-1}$ is of length exactly 16 bytes, and T_r is of length between 1 and 16 bytes.
- The ciphertext C is partitioned into a sequence of blocks $C = C_1 \mid C_2 \mid \dots \mid C_m$, where each of the blocks $C_1, C_2, \dots, C_{m-1}$ is of length exactly 16 bytes, and C_m is of length between 1 and 16 bytes.

A conforming implementation shall compute the plaintext by executing the sequence of steps in Table 4 or a mathematically equivalent procedure.

NOTE—The only difference between the encryption and decryption procedures is that all the AES-Enc calls in Table 3 are replaced in Table 4 by calls to AES-Dec. The function H is defined in Table 2.¹²

¹² Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Table 4—The EME2-AES decryption procedure

```
// EME2-AES-Dec( $K_{AES}$ ,  $K_{ECB}$ ,  $K_{AD}$ ,  $T$ ,  $C = (C_1 \mid \dots \mid C_{m-1} \mid C_m)$  ):
T_star = H( $K_{AES}$ ,  $K_{AD}$ ,  $T$ ) // Process the associated data
if bytelength( $C_m$ ) == 16 then
    lastFull = m
else
    lastFull = m-1
    CCC_m =  $C_m \mid 0x80 \mid 0x00 \mid \dots$  // Pad  $C_m$  to 16 bytes, 0x80 followed by 0's
endif

// First ECB pass
L =  $K_{ECB}$ 
for j = 1 to lastFull
    CCC_j = AES-Enc( $K_{AES}$ ,  $L \oplus C_j$ )
    L = mult-by-alpha(L)
next

// Intermediate mixing
MC = CCC_1  $\oplus$  CCC_2  $\oplus \dots \oplus$  CCC_m  $\oplus$  T_star
if bytelength( $C_m$ ) < 16 then
    MM = AES-Dec( $K_{AES}$ , MC)
    MP = MP_1 = AES-Dec( $K_{AES}$ , MM)
else
    MP = MP_1 = AES-Dec( $K_{AES}$ , MC)
endif
M = M_1 = MP  $\oplus$  MC
for j = 2 to lastFull
    if (j-1 mod 128 > 0) then // use the current mask M
        M = mult-by-alpha(M)
        PPP_j = CCC_j  $\oplus$  M
    else // calculate a new mask M
        MC = CCC_j  $\oplus$  M_1
        MP = AES-Dec( $K_{AES}$ , MC)
        M = MP  $\oplus$  MC
        PPP_j = MP  $\oplus$  M_1
    endif
next
if lastFull < m then
    P_m =  $C_m \oplus$  (MM truncated to bytelength( $C_m$ ) bytes)
    PPP_m = P_m  $\mid 0x80 \mid 0x00 \mid \dots$  // Pad P_m to 16 bytes, 0x80 followed by 0's
endif
PPP_1 = MP_1  $\oplus$  PPP_2  $\oplus \dots \oplus$  PPP_m  $\oplus$  T_star

// Second ECB Pass
L =  $K_{ECB}$ 
for j = 1 to lastFull
    P_j = AES-Dec( $K_{AES}$ , PPP_j)  $\oplus$  L
    L = mult-by-alpha(L)
next

return P = (P_1  $\mid \dots \mid$  P_{m-1}  $\mid$  P_m)
```

6. Compliance

The EME2-AES encryption and decryption procedures described in 5.3.2 and 5.3.3 use AES as the basic building block with a key of either 48 or 64 bytes. EME2-AES-128 uses a 48-byte key, and EME2-AES-256 uses a 64-byte key.

An implementation may claim compliance with this standard if it fulfills all the requirements for either the encryption or decryption procedures or both of at least one of the following algorithms:

- EME2-AES-128
- EME2-AES-256

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Halevi, S., “EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data,” *INDOCRYPT 2004, Lecture Notes in Computer Science*, vol. 3348, pp. 315–327, Springer-Verlag, 2004.

[B2] IEEE Std 1619™, IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices.^{13,14}

[B3] Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A., *Handbook of Applied Cryptography*, CRC Press, October 1996.

¹³ IEEE publications are available from The Institute of Electrical and Electronics Engineers, Inc., (<https://standards.ieee.org/>).

¹⁴ The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.