INTERNATIONAL STANDARD

ISO/IEC 19772

Second edition
2020-11

Information security — Authenticated encryption

Sécurité de l'information — Chiffrement authentifié

Reference number
ISO/IEC 19772:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

This second edition cancels and replaces the first edition (ISO/IEC 19772:2009) which has been technically revised. It also incorporates the Technical Corrigendum ISO/IEC 19772:2009/Cor 1:2014.

The main changes compared to the previous edition are as follows:

— old Clause 6 has been removed following the deprecation of mechanism 1 (OCB 2.0);

— optional additional authenticated data has been included in mechanism 5.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

When data is sent from one place to another, it is often necessary to protect it in some way while it is in transit, e.g. against eavesdropping or unauthorized modification. Similarly, when data is stored in an environment to which unauthorized parties can have access, it can be necessary to protect it.

If the confidentiality of the data needs to be protected, e.g. against eavesdropping, then one solution is to use encryption, as specified in ISO/IEC 18033 (all parts) and ISO/IEC 10116. Alternatively, if it is necessary to protect the data against modification, i.e. integrity protection, then message authentication codes (MACs) as specified in ISO/IEC 9797 (all parts), or digital signatures as specified in ISO/IEC 9796 (all parts) and ISO/IEC 14888 (all parts), can be used. If both confidentiality and integrity protection are required, then one possibility is to use both encryption and a MAC or signature. While these operations can be combined in many ways, not all combinations of such mechanisms provide the same security guarantees. As a result, it is desirable to define in detail exactly how integrity and confidentiality mechanisms should be combined to provide the optimum level of security. Moreover, in some cases, significant efficiency gains can be obtained by defining a single method of processing the data with the objective of providing both confidentiality and integrity protection.

In this document, authenticated encryption mechanisms are defined. These are methods for processing data to provide both integrity and confidentiality protection. They typically involve either a specified combination of a MAC computation and data encryption, or the use of an encryption algorithm in a special way such that both integrity and confidentiality protection are provided.

The methods specified in this document have been designed to maximize the level of security and provide efficient processing of data. Some of the techniques defined here have mathematical "proofs of security", i.e. rigorous arguments supporting their soundness.

Information security — Authenticated encryption

1 Scope

This document specifies five methods for authenticated encryption, i.e. defined ways of processing a data string with the following security objectives:

— data confidentiality, i.e. protection against unauthorized disclosure of data;

— data integrity, i.e. protection that enables the recipient of data to verify that it has not been modified;

— data origin authentication, i.e. protection that enables the recipient of data to verify the identity of the data originator.

All five methods specified in this document are based on a block cipher algorithm, and require the originator and the recipient of the protected data to share a secret key for this block cipher.

Key management is outside the scope of this document. Key management techniques are defined in ISO/IEC 11770 (all parts).

Four of the mechanisms in this document, namely mechanisms 3, 4, 5 (AAD variant only) and 6, allow data to be authenticated which is not encrypted. That is, these mechanisms allow a data string that is to be protected to be divided into two parts, *D*, the data string that is to be encrypted and integrity-protected, and *A* (the additional authenticated data) that is integrity-protected but not encrypted. In all cases, the string *A* can be empty.

NOTE Examples of types of data that can need to be sent in unencrypted form, but whose integrity is to be protected, include addresses, port numbers, sequence numbers, protocol version numbers and other network protocol fields that indicate how the plaintext is to be handled, forwarded or processed.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9797 (all parts), *Information technology — Security techniques — Message Authentication Codes (MACs)*

ISO/IEC 10116, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

ISO/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
additional authenticated data
AAD
data that is integrity-protected but not encrypted by the *authenticated encryption mechanism* (3.3)

3.2
authenticated encryption
(reversible) transformation of data by a cryptographic algorithm to produce *ciphertext* (3.5) that cannot be altered by an unauthorized entity without detection, i.e. it provides data confidentiality, *data integrity* (3.6), and data origin authentication

3.3
authenticated encryption mechanism
cryptographic technique used to protect the confidentiality and guarantee the origin and integrity of data, and which consists of two component processes: an *encryption* (3.8) algorithm and a *decryption* (3.7) algorithm

3.4
block cipher
symmetric encryption system (3.15) with the property that the *encryption* (3.8) algorithm operates on a block of *plaintext* (3.13), i.e. a string of bits of a defined length, to yield a block of *ciphertext* (3.5)

[SOURCE: ISO/IEC 18033-1:2015, 2.9]

3.5
ciphertext
data which has been transformed to hide its information content

[SOURCE: ISO/IEC 10116:2017, 3.2]

3.6
data integrity
property that data has not been altered or destroyed in an unauthorized manner

[SOURCE: ISO/IEC 9797-1:2011, 3.4]

3.7
decryption
reversal of a corresponding *encryption* (3.8)

[SOURCE: ISO/IEC 18033-1:2015, 2.16]

3.8
encryption
(reversible) transformation of data by a cryptographic algorithm to produce *ciphertext* (3.5), i.e., to hide the information content of the data

[SOURCE: ISO/IEC 18033-1:2015, 2.21]

3.9
encryption system
cryptographic technique used to protect the confidentiality of data, and which consists of three component processes: an *encryption* (3.8) algorithm, a *decryption* (3.7) algorithm, and a method for generating *keys* (3.10)

[SOURCE: ISO/IEC 18033-1:2015, 2.23]

3.10
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encipherment, decipherment)

[SOURCE: ISO/IEC 18033-1:2015, 2.27]

3.11
message authentication code
MAC
string of bits which is the output of a MAC algorithm

[SOURCE: ISO/IEC 9797-1:2011, 3.9]

3.12
partition
process of dividing a string of bits of arbitrary length into a sequence of blocks, where the length of each block is n bits, except for the final block which shall contain r bits, $0 < r \leq n$

3.13
plaintext
unencrypted information

[SOURCE: ISO/IEC 10116:2017, 3.11]

3.14
secret key
key (3.10) used with symmetric cryptographic techniques by a specified set of entities

[SOURCE: ISO/IEC 18033-1:2015, 2.33]

3.15
symmetric encryption system
encryption (3.8) system based on symmetric cryptographic techniques that uses the same *secret key* (3.14) for both the *encryption* (3.8) and *decryption* (3.7) algorithms

[SOURCE: ISO/IEC 18033-1:2015, 2.40]

4 Symbols and abbreviated terms

A additional authenticated data

C authenticated-encrypted data string

D data string to which an authenticated encryption mechanism is to be applied

d block cipher decryption algorithm; $d_K(Y)$ denotes the result of block cipher decrypting the n-bit block Y using the secret key K

e block cipher encryption algorithm; $e_K(X)$ denotes the result of block cipher encrypting the n-bit block X using the secret key K

K secret block cipher key shared by the originator and recipient of the data to which the authenticated encryption mechanism is to be applied

m number of blocks in the partitioned version of D

n block length (in bits) for a block cipher

t tag length (in bits)

0^i block of i zero bits

1^i block of i one bits

$\oplus$ bit-wise exclusive-or of strings of bits (of the same bit-length)

|| concatenation of bit strings, i.e. if A and B are blocks of bits, then $A||B$ is the block of bits obtained by concatenating A and B in the order specified

function converting a number into an a-bit block of bits

If k is an integer ($0 \le k < 2^a$), then $\#_a(k)$ is the a-bit block which, when regarded as the binary representation of a number with the most significant bit on the left, equals k.

$\#^{-1}$ function converting a block of bits to a number

If A is a block of bits, then $\#^{-1}(A)$ is the unique non- negative integer whose binary representation is A. Hence, if A has n bits, then $\#_n(\#^{-1}(A)) = A$.

$X|_s$ left-truncation of the block of bits X

If X has bit-length greater than or equal to s, then $X|_s$ is the s-bit block consisting of the left-most s bits of X.

$X|^s$ right-truncation of the block of bits X

If X has bit-length greater than or equal to s, then $X|^s$ is the s-bit block consisting of the right-most s bits of X.

$X<<1$ left shift of a block of bits X by one position

The rightmost bit of $Y = X<<1$ is always set to zero.

$X>>1$ right shift of a block of bits X by one position

The leftmost bit of $Y = X>>1$ is always set to zero.

len function taking a bit-string X as input, and which gives as output the number of bits in X

mod if a and $b > 0$ are integers, then a mod b denotes the unique integer c such that:

1) $0 \le c < b$; and

2) a-c is an integer multiple of b.

5 Requirements

The authenticated encryption mechanisms specified in this document have the following requirements.

The originator and recipient of the data to which the authenticated encryption mechanism is to be applied, shall:

a) agree on the use of a particular mechanism from those specified in this document;

b) agree on the use of a particular block cipher to be used with the mechanism (one of the block ciphers standardized in ISO/IEC 18033-3 shall be used);

c) share a secret key K: in all mechanisms except for authenticated encryption mechanism 5, this shall be a key for the selected block cipher, and in mechanism 5 it shall be a key used as input to a key derivation procedure.

In addition, each mechanism has specific requirements listed immediately before the mechanism description.

Annex A provides guidance on the use of the mechanisms defined in this document.

Annex B contains numerical examples of the operation of the mechanisms specified in this document.

Annex C provides the object identifiers which shall be used to identify the mechanisms defined in this document.

6 Authenticated encryption mechanism 2 (key wrap)

6.1 General

This clause defines an authenticated encryption mechanism commonly known as key wrap.

NOTE 1 This scheme was originally designed for authenticated encryption of keys and associated information. That is, it is designed for use with short data strings. However, the scheme can be used with arbitrary length data strings (up to a maximum of around 2^{67} bits), although it is not efficient for protecting long messages.

NOTE 2 This mode is known as AES key wrap when the AES block cipher is used, where AES stands for advanced encryption standard, a block cipher algorithm specified in ISO/IEC 18033-3:2010. AES key wrap is also specified in References [7] and [9].

6.2 Specific notation

For the purposes of the specification of this mechanism, the following symbols and notation apply:

$C_0, C_1, \ldots, C_m$	sequence of (m+1) 64-bit blocks obtained as the output of the authenticated encryption process
$D_1, D_2, \ldots, D_m$	sequence of m 64-bit blocks obtained by partitioning D, i.e. $64m = \text{len}(D)$
$R_1, R_2, \ldots, R_m$	sequence of m 64-bit blocks computed during the encryption and decryption processes
Y	64-bit block used during the encryption and decryption processes
Z	128-bit block computed during the encryption and decryption processes

6.3 Specific requirements

The block cipher to be used with this mechanism shall be a 128-bit block cipher, i.e. it shall have n=128.

The data string D to be protected using this mechanism shall contain at least 128 bits and a multiple of 64 bits (i.e. the bit-length of D shall be $64m$ for some integer $m > 1$).

6.4 Encryption procedure

The originator shall perform the following steps to protect a data string D.

a) Partition D into a sequence of m 64-bit blocks $D_1, D_2, \ldots, D_m$, so that D_1 contains the first 64 bits of D, D_2 the next 64 bits, and so on.

b) Let Y be the 64-bit block having hexadecimal representation A6A6A6A6A6A6A6A6, i.e. in binary it equals (10100110 10100110 ... 10100110).

c) For $i = 1, 2, \ldots, m$:

   let $R_i = D_i$.

d) For $i = 1, 2, \ldots, 6m$, perform the following four steps:

   1) Let $Z = e_K(Y \,||\, R_1)$;

2) Let $Y = Z|_{64} \oplus \#_{64}(i)$;

3) For $j = 1, 2, \ldots, m\text{-}1$:

let $R_j = R_{j+1}$;

4) Let $R_m = Z|^{64}$.

e) Let $C_0 = Y$.

f) For $i = 1, 2, \ldots, m$:

let $C_i = R_i$.

The output of the above process, i.e. the authenticated-encrypted version of D, shall be the bit-string:

$C = C_0 \,||\, C_1 \,||\, \ldots \,||\, C_m$

That is, a string of $64(m+1)$ bits, that is C contains precisely 64 bits more than D.

6.5 Decryption procedure

The recipient shall perform the following steps to decrypt and verify an authenticated-encrypted string C.

a) If len(C) is not a multiple of 64 or is less than 192, then halt and output INVALID.

b) Partition C into a sequence of $m+1$ 64-bit blocks $C_0, C_1, \ldots, C_m$ so that C_0 contains the first 64 bits of C, C_1 the next 64 bits, and so on.

c) Let $Y = C_0$.

d) For $i = 1, 2, \ldots, m$:

let $R_i = C_i$.

e) For $i = 6m, 6m\text{-}1$, down to 1, perform the following four steps:

1) Let $Z = d_K([Y \oplus \#_{64}(i)] \,||\, R_m)$;

2) Let $Y = Z|_{64}$;

3) For $j = m, m\text{-}1, \ldots, 2$:

let $R_j = R_{j-1}$;

4) Let $R_1 = Z|^{64}$.

f) If Y = (10100110 10100110 … 10100110), then output $D = R_1 \,||\, R_2 \,||\, \ldots \,||\, R_m$. Otherwise, output INVALID.

7 Authenticated encryption mechanism 3 (CCM)

7.1 General

This clause defines an authenticated encryption mechanism commonly known as CCM (for counter with CBC-MAC).

NOTE CCM is due to Whiting, Housley and Ferguson.[10] The version of CCM defined here is a special case of CCM as defined in References [8] and [10].

7.2 Specific notation

For the purposes of the specification of this mechanism, the following symbols and notation apply:

B	block of bits used in computing the tag value
$B_1, B_2, \ldots, B_v$	sequence of blocks of bits (each of n bits) used in computing the tag value
$C_1, C_2, \ldots, C_m$	sequence of m 128-bit blocks obtained as part of the output of the authenticated encryption process
$D_1, D_2, \ldots, D_m$	sequence of m 128-bit blocks obtained by partitioning a padded version of D
F	flag octet
L	length of D (in octets), excluding padding and the length block D_0
r	the number of octets of D in the block D_m
S	starting variable (of 120-8w bits)
T	plaintext tag value (of t bits)
T'	recomputed tag value, generated during the decryption process
U	encrypted tag value (of t bits)
v	variable used in computing the tag value
w	length of message length field in octets
X	128-bit block computed during the encryption and decryption processes
Y	128-bit block computed during the encryption and decryption processes

7.3 Specific requirements

In advance of any use of the mechanism, the originator and recipient of the data to which the authenticated encryption mechanism is to be applied, shall agree on:

a) t, the bit-length of the tag; t shall be chosen from the set {32, 48, 64, 80, 96, 112, 128}; and

b) w, the octet-length of the message length field; w shall be chosen from the set {2, 3, 4, 5, 6, 7, 8}.

NOTE The choice of w affects the maximum message length which can be protected. The maximum message length is 2^{8w+3} bits, i.e. 2^{8w} octets.

The block cipher to be used with this mechanism shall be a 128-bit block cipher, i.e. it shall have n=128.

The data string D to be protected using this mechanism, and the additional authenticated data string A, shall contain a whole number of octets, i.e. their lengths shall be a multiple of 8 bits [i.e. len(D) and len(A) shall both be an integer multiple of 8].

7.4 Encryption procedure

The originator shall perform the following steps to protect a data string D. Let L = len(D)/8, i.e. L is the number of octets in D.

a) A starting variable S containing 15-w octets (i.e. 120-8w bits) shall be selected. This variable shall be distinct for every message to be protected, and shall be made available to the recipient of the message. However, it is not necessary that this value is unpredictable or secret.

NOTE 1 The value S can, for example, be generated using a counter maintained by the originator, and sent in cleartext along with the protected message.

b) Right pad the data string D with 16-r zero octets (i.e. between 0 and 120 zero bits) so that the padded version of D contains a multiple of 128 bits. Then, partition the padded version of D into a sequence of m 128-bit blocks D_1, D_2, ..., D_m, so that D_1 contains the first 128 bits of D, D_2 the next 128 bits, and so on.

NOTE 2 The value m needs to satisfy $16(m\text{-}1) < L \leq 16m$.

c) If len(A) = 0, then let the flag octet $F = 0^2 \,||\, \#_3((t\text{-}16)/16) \,||\, \#_3(w\text{-}1)$.

d) If len(A) > 0, then let the flag octet $F = 0 \,||\, 1 \,||\, \#_3((t\text{-}16)/16) \,||\, \#_3(w\text{-}1)$.

NOTE 3 The most significant (left-most) bit of F is a "reserved" bit, i.e. it is set to zero for the version of the mechanism specified here but can be used in the future in other (as yet unspecified) versions of the mechanism. The next to the most significant bit of F is set to zero to indicate that all the data being protected by the mechanism is encrypted.

e) Let $X = e_K(F \,||\, S \,||\, \#_{8w}(L))$.

f) If len(A) > 0, then perform the following six steps:

1) if $0 <$ len(A) $< 65\,280$, then let $B = \#_{16}(\text{len}(A)/8) \,||\, A$;

2) if $65\,280 \leq$ len(A) $< 2^{32}$, then let $B = 1^{15} \,||\, 0 \,||\, \#_{32}(\text{len}(A)/8) \,||\, A$;

3) if $2^{32} \leq$ len(A) $< 2^{64}$, then let $B = 1^{16} \,||\, \#_{64}(\text{len}(A)/8) \,||\, A$;

4) partition B into a sequence of blocks: B_1, B_2, ..., B_v, as follows: let B_1 contain the first n bits of B, B_2 the next n bits, and so on, until B_v contains the final k bits, where $0 < k \leq n$. Thus, len(B) = $(v\text{-}1)n+k$;

5) right pad B_v with n-k zeros, i.e. let $B_v = B_v \,||\, 0^{n-k}$;

6) for i = 1, 2, ..., v:

let $X = e_K(\, X \oplus B_i \,)$.

g) For i = 1, 2, ..., m:

let $X = e_K(\, X \oplus D_i \,)$.

h) Let $T = X|_t$.

NOTE 4 The plaintext tag T is equal to a MAC computed on the data string B_1, B_2, ..., B_v, D_1, D_2, ..., D_m using a slight modification of MAC algorithm 1 specified in ISO/IEC 9797-1.

i) Let the flag octet $F = (\, 0^5 \,||\, \#_3(w\text{-}1) \,)$, and let $Y = (\, F \,||\, S \,||\, 0^{8w} \,)$.

NOTE 5 The two most significant (left-most) bits of F are "reserved" bits, i.e. they are set to zero for the version of the mechanism specified here but can be used in the future in other (as yet unspecified) versions of the mechanism. The next three most significant bits of F are set to zero to ensure that this octet is distinct from the flag octet used in step c) above.

j) Let $U = T \oplus [e_K(Y)]|_t$.

k) For i = 1, 2, ..., m-1, perform the following two steps:

1) Let $Y = (\, F \,||\, S \,||\, \#_{8w}(i) \,)$;

2) Let $C_i = D_i \oplus e_K(Y)$.

l) Let $Y = (\, F \,||\, S \,||\, \#_{8w}(m) \,)$, and let $C_m = [D_m \oplus e_K(Y)\,]|_{8r}$.

The output of the above process, i.e. the authenticated-encrypted version of D, shall be the bit-string:

$C = C_1 \,||\, C_2 \,||\, \ldots \,||\, C_{m-1} \,||\, C_m \,||\, U$

That is, a string of $8L+t$ bits, that is C contains precisely t bits more than the original data string D [although it is also necessary to convey the (120-8w)-bit starting variable S and the variable length additional authenticated data A to the recipient].

7.5 Decryption procedure

The recipient shall perform the following steps to decrypt and verify an authenticated-encrypted string C.

a) If C does not contain a whole number of octets, then halt and output INVALID.

b) If the length of C is less than (t+8) bits, then halt and output INVALID.

c) Let m and r be the unique integers such that C contains a total of $128(m\text{-}1) + 8r + t$ bits, where $0 < r \leq 16$. Partition C into a sequence of blocks: $C_1, C_2, \ldots C_m, U$ as follows. Let C_1 contain the first 128 bits of C, C_2 the next 128 bits of C, and so on, until C_m contains the next $8r$ bits of C. Finally, let U be the final t bits of C.

d) Let the flag octet $F = (0^5 \,||\, \#_3(w\text{-}1))$, and let $Y = (F \,||\, S \,||\, 0^{8w})$.

e) Let $T = U \oplus [e_K(Y)]|_t$.

f) For $i = 1, 2, \ldots, m\text{-}1$, perform the following two steps:

   1) Let $Y = (F \,||\, S \,||\, \#_{8w}(i))$;

   2) Let $D_i = C_i \oplus e_K(Y)$.

g) Let $Y = (F \,||\, S \,||\, \#_{8w}(m))$, and let $D_m = C_m \oplus [e_K(Y)]|_{8r}$.

h) Let $D = D_1 \,||\, D_2 \,||\, \ldots \,||\, D_m$, and let $L = 16m - 16 + r$.

i) Right pad D_m with 128-8r zeros, i.e. let $D_m = D_m \,||\, 0^{128-8r}$.

j) If len(A) = 0, then let the flag octet $F = 0^2 \,||\, \#_3((t\text{-}16)/16) \,||\, \#_3(w\text{-}1)$.

k) If len(A) > 0, then let the flag octet $F = 0 \,||\, 1 \,||\, \#_3((t\text{-}16)/16) \,||\, \#_3(w\text{-}1)$.

l) Let $X = e_K(F \,||\, S \,||\, \#_{8w}(L))$.

m) If len(A) > 0, then perform the following six steps:

   1) if $0 <$ len(A) $< 65\,280$ then let $B = \#_{16}(\text{len}(A)/8) \,||\, A$;

   2) if $65\,280 \leq$ len(A) $< 2^{32}$ then let $B = 1^{15} \,||\, 0 \,||\, \#_{32}(\text{len}(A)/8) \,||\, A$;

   3) if $2^{32} \leq$ len(A) $< 2^{64}$ then let $B = 1^{16} \,||\, \#_{64}(\text{len}(A)/8) \,||\, A$;

   4) partition B into a sequence of blocks: $B_1, B_2, \ldots, B_v$, as follows: let B_1 contain the first n bits of B, B_2 the next n bits, and so on, until B_v contains the final k bits, where $0 < k \leq n$. Thus, len(B) = $(v\text{-}1)n+k$;

   5) right pad B_v with n-k zeros, i.e. let $B_v = B_v \,||\, 0^{n-k}$;

   6) for $i = 1, 2, \ldots, v$:

      let $X = e_K(X \oplus B_i)$.

n) For $i = 1, 2, \ldots, m$:

   let $X = e_K(X \oplus D_i)$.

o) Let $T' = X|_t$.

p) If $T = T'$, then output D as computed in step h) and A. Otherwise, output INVALID.

8 Authenticated encryption mechanism 4 (EAX)

8.1 General

This clause defines an authenticated encryption mechanism commonly known as EAX.

NOTE EAX is due to Bellare, Rogaway and Wagner.[2] The letters EAX do not appear to stand for anything in particular.

8.2 Specific notation

For the purposes of the specification of this mechanism, the following symbols and notation apply:

$C_1, C_2, \ldots, C_m$	sequence of blocks of bits (each of n bits, with the possible exception of C_m) obtained as part of the output of the authenticated encryption process
$D_1, D_2, \ldots, D_m$	sequence of blocks of bits (each of n bits, with the possible exception of D_m) obtained by partitioning D
E_0, E_1, E_2	n-bit blocks computed during the encryption and decryption processes
M	function used in the encryption and decryption processes
S	starting variable (n bits)
T	tag (t bits), adjoined to an encrypted message to provide integrity protection
T'	recomputed tag value, generated during the decryption process
W	n-bit block computed during the encryption and decryption processes

8.3 Specific requirements

In advance of any use of the mechanism, the originator and recipient of the data to which the authenticated encryption mechanism is to be applied, shall agree on t, the length of the tag in bits, where $0 < t \leq n$.

8.4 Definition of function *M*

Definition of the encryption and decryption procedures requires the definition of a function M that takes an arbitrary length string of bits and a block cipher key as input and gives an n-bit block as output. The definition of this function is as follows.

If X is a string of bits, and K is a key for the chosen block cipher, then $M_K(X)$ shall equal an (untruncated) message authentication code computed on the string X using key K using MAC algorithm 5 of ISO/IEC 9797-1:2011, where the block cipher used in the MAC algorithm shall be the same as the block cipher algorithm selected for the authenticated encryption process.

NOTE MAC algorithm 5 of ISO/IEC 9797-1:2011 is also known as CMAC.

8.5 Encryption procedure

The originator shall perform the following steps to protect a data string D.

a) A starting variable S containing n bits shall be selected. This variable shall be distinct for every message to be protected, and shall be made available to the recipient of the message. However, it is not necessary that this value is unpredictable or secret.

b) Let $E_0 = M_K(0^n||S)$.

c) Let $E_1 = M_K(0^{n-1}||1||A)$.

d) Let $W = E_0$.

e) Partition D into a sequence of blocks: $D_1, D_2, ..., D_m$, as follows. Let D_1 contain the first n bits of D, D_2 the next n bits, and so on, until D_m contains the final r bits, where $0 < r \leq n$. Thus len(D) = (m-1)n+r.

f) For $i = 1, 2, ..., m$-1, perform the following two steps:

   1) let $C_i = D_i \oplus e_K(W)$;

   2) let $W = \#_n(\#^{-1}(W) +1 \bmod 2^n)$.

g) Let $C_m = D_m \oplus [e_K(W)]|_r$.

h) Let $E_2 = M_K(0^{n-2}||1||0||C_1||C_2||...||C_m)$.

i) Let $T = [E_0 \oplus E_1 \oplus E_2]|_t$.

The output of the above process, i.e. the authenticated-encrypted version of D, shall be the bit-string:

$$C = C_1 \,||\, C_2 \,||\, \ldots \,||\, C_m \,||\, T$$

That is, a string of (m-1)n+r+t bits, that is C contains precisely t bits more than D (although it is also necessary to convey the n-bit starting variable S and the variable length additional authenticated data A to the recipient).

8.6 Decryption procedure

The recipient shall perform the following steps to decrypt and verify an authenticated-encrypted string C.

a) If the length of C is less than t, then halt and output INVALID.

b) Let m and r be the unique integers defined so that C contains a total of (m-1)n + r + t bits, where $0 < r \leq n$. Partition C into a sequence of blocks: $C_1, C_2, ...C_m, T$ as follows. Let C_1 contain the first n bits of C, C_2 the next n bits of C, and so on, until C_m contains the next r bits of C. Finally, let T be the final t bits of C.

c) Let $E_0 = M_K(0^n||S)$.

d) Let $E_1 = M_K(0^{n-1}||1||A)$.

e) Let $E_2 = M_K(0^{n-2}||1||0||C_1||C_2||...||C_m)$.

f) Let $T' = [E_0 \oplus E_1 \oplus E_2]|_t$.

g) If $T \neq T'$, then halt and output INVALID.

h) Let $W = E_0$.

i) For $i = 1, 2, ..., m$-1, perform the following two steps:

   1) let $D_i = C_i \oplus e_K(W)$;

2) let $W = \#_n(\#^{-1}(W) + 1 \bmod 2^n)$.

j) Let $D_m = C_m \oplus [e_K(W)]|_r$.

k) Output D and A.

9 Authenticated encryption mechanism 5 (encrypt-then-MAC)

9.1 General

This clause defines an authenticated encryption mechanism made up of the combination of any encryption mechanism and any MAC scheme. The basic mechanism involves first encrypting the data to be protected, and then computing a MAC on the resulting encrypted data. An AAD variant mechanism allows AAD to be authenticated but not encrypted.

NOTE The encrypt-then-MAC approach (without additional authenticated data) has been analysed by Bellare and Namprempre,[1] who provide a proof of security on the assumption that the method of encryption and the MAC technique possess certain security properties. Although their analysis does not consider extending the technique by including a starting variable or additional authenticated data, it can be shown that security is maintained in this case.

9.2 Specific notation

For the purposes of the specification of this mechanism, the following symbols and notation apply:

C' bit string obtained by encrypting the data string D

δ the decryption function, i.e. a function which takes as input a block cipher key K_1, a starting variable S, and an encrypted data string C' and, using the selected mode of operation, outputs a decrypted data string: the output is written $\delta_{K_1,S}(C')$

ε the encryption function, i.e. a function which takes as input a block cipher key K_1, a starting variable S, and a data string D and, using the selected mode of operation, outputs an encrypted data string: the output is written $\varepsilon_{K_1,S}(D)$

f the MAC function

If X is an input string, and K_2 is a MAC key, then the output MAC is written $f_{K_2}(X)$.

K_1 secret key for the block cipher

K_2 secret key for the MAC function

S starting variable (n bits)

T tag (t bits), adjoined to an encrypted message to provide integrity protection

T' recomputed tag value, generated during the decryption process

9.3 Specific requirements

In advance of any use of the mechanism, the originator and recipient of the data to which the authenticated encryption mechanism is to be applied, shall agree on:

a) a block cipher mode of operation from amongst those specified in ISO/IEC 10116 (the ECB mode shall not be used);

b) a method for MAC computation, which shall be selected from the techniques specified in ISO/IEC 9797 (all parts) (it is supposed that the chosen method generates a tag of length t bits); and

c) a method for obtaining a pair of secret keys (K_1, K_2) from the secret key K, where K_1 is a key for the selected block cipher and K_2 is a key for the selected method of MAC computation.

NOTE 1 K is chosen so that the number of possible values for K is at least as large as the number of possible values for the block cipher key, and also at least as large as the number of possible values for the MAC key.

NOTE 2 (K_1, K_2) can be obtained from the secret key K by taking (disjoint) bit strings from K or from $h(K)$, where h is a hash-function in ISO/IEC 10118 (all parts). More generally (K_1, K_2) can be obtained from the secret key K using a derivation function specified in ISO/IEC 11770-6.

d) whether to use the basic mechanism (that does not support AAD) or to use the AAD variant mechanism. If using the AAD variant mechanism, then the additional authenticated data string A shall contain a whole number of octets (possibly zero), i.e. len(A) shall be an integer multiple of 8, but shall contain fewer than 2^{64} octets (or even less depending on the requirements of the MAC scheme used).

A single key K shall only be used with one variant, i.e. only with the basic variant or only with the AAD variant.

9.4 Encryption procedure

The originator shall perform the following steps to protect a data string D and, if using the AAD variant mechanism, to ensure the integrity of an additional authenticated data string A.

a) A starting variable S appropriate for use with the selected block cipher mode of operation shall be selected. Security requirements for S are as described in the appropriate clauses of ISO/IEC 10116, and further guidance is given in A.6.

b) Let $C' = \varepsilon_{K_1,S}(D)$.

If not using the AAD variant:

c) Let $T = f_{K_2}(S \| C')$.

If using the AAD variant:

c) If len(A) is not a multiple of 8 or is $\geq 2^{67}$, then halt and output INVALID.

Let $T = f_{K_2}(\#_{64}(\text{len}(A)/8) \| A \| S \| C')$.

The output of the above process, i.e. the authenticated-encrypted version of D, shall be the bit-string:

$C = C' \| T$, together with the starting variable S.

9.5 Decryption procedure

The recipient shall perform the following steps to decrypt and verify an authenticated-encrypted string C, with accompanying starting variable S and, if using the AAD variant mechanism, to verify the integrity of additional authenticated data A.

a) If the length of C is less than t, then halt and output INVALID.

b) Let T be the rightmost t bits of C, and let C' be equal to C with the rightmost t bits removed, i.e. $C = C' \| T$.

If not using the AAD variant:

c) Let $T' = f_{K_2}(S \| C')$.

If using the AAD variant:

c) If len(A) is not a multiple of 8 or is ≥ 2^{67}, then halt and output INVALID.

Let $T' = f_{K_2}(\#_{64}(\text{len}(A)/8) \,||\, A \,||\, S \,||\, C')$.

d) If $T \neq T'$, then halt and output INVALID.

e) Let $D = \delta_{K_1,S}(C')$.

f) Output D.

10 Authenticated encryption mechanism 6 (GCM)

10.1 General

This clause defines an authenticated encryption mechanism commonly known as GCM (for Galois/Counter Mode).

NOTE GCM is due to McGrew and Viega.[6]

10.2 Specific notation

For the purposes of the specification of this mechanism, the following symbols and notation apply:

$C_1, C_2, ..., C_m$	sequence of m 128-bit blocks (with the possible exception of C_m, which may contain between 1 and 128 bits) obtained as part of the output of the authenticated encryption process
$D_1, D_2, ..., D_m$	sequence of m 128-bit blocks of bits (with the possible exception of D_m) obtained by partitioning D
G	function used in the encryption and decryption processes (defined in 10.5)
H	128-bit block used in the encryption and decryption processes
inc	function taking a 128-bit block as input and giving a 128-bit block as output, where, if X is a 128-bit block: $\text{inc}(X) = (X\|_{96}) \,||\, \#_{32}(\#^{-1}(X\|^{32})+1 \bmod 2^{32})$
r	the number of bits in the final block of the message to be encrypted, after it has been divided into n-bit blocks, i.e. the message contains $(m\text{-}1)n+r$ bits
R	128-bit block used in the computation of a GF(2^{128}) multiplication
S	starting variable (variable length)
T	tag (t bits), adjoined to an encrypted message to provide integrity protection
T'	recomputed tag value, generated during the decryption process
U, V, W, Z	128-bit blocks used in defining the computation of a GF(2^{128}) multiplication
$X_0, X_1, ..., X_{k+l+1}$	128-bit blocks used in computing the function G

$Y_0, Y_1, ..., Y_m$ sequence of 128-bit blocks used in the encryption and decryption processes

{ } a bit-string with zero length

• multiplication in the field GF(2^{128})
The polynomial to be used to determine the representation of GF(2^{128}) is $1 + \alpha + \alpha^2 + \alpha^7 + \alpha^{128}$.

10.3 Specific requirements

In advance of any use of the mechanism, the originator and recipient of the data to which the authenticated encryption mechanism is to be applied, shall agree on the tag length t in bits, where t shall be a multiple of 8 satisfying $96 \leq t \leq 128$ ($t = 32$ and $t = 64$ are also permitted for specialized applications).

The block cipher to be used with this mechanism shall be a 128-bit block cipher, i.e. it shall have n=128.

10.4 Definition of multiplication operation •

Suppose U and V are 128-bit blocks; then $W = U \bullet V$ is defined as follows, where W is also a 128-bit block. Note that, in the description below, v_i denotes the ith bit of V, i.e. $V = v_0||v_1||...||v_{127}$. Analogously, z_{127} denotes the rightmost bit of Z.

a) Let $R = 11100001 || 0^{120}$.

b) Let $W = 0^{128}$.

c) Let $Z = U$.

d) For $i = 0, 1, ..., 127$, perform the following two steps:

1) if $v_i = 1$, then let $W = W \oplus Z$;

2) if $z_{127} = 0$, then let $Z = Z>>1$. Otherwise let $Z = (Z>>1) \oplus R$.

10.5 Definition of function *G*

The encryption and decryption procedures make use of a function G, that takes as input a 128-bit block and two arbitrary length strings of bits, and gives a 128-bit block as output. Let H be a 128-bit block, and W and Z be two arbitrary length (possible empty) strings of bits. Suppose that k and u are the unique integers such that $\text{len}(W) = 128(k\text{-}1)+u$ and $0 < u \leq 128$.

Similarly, suppose that l and v are the unique integers such that $\text{len}(Z) = 128(l\text{-}1)+v$ and $0 < v \leq 128$. Let $W_1, W_2, ..., W_k$ be the sequence of 128-bit blocks (with the possible exception of W_k which contains the final u bits of W) obtained by partitioning W; similarly, let $Z_1, Z_2, ..., Z_l$ be the sequence of 128-bit blocks (with the possible exception of Z_l, which contains the final v bits of Z) obtained by partitioning Z.

Then $G(H,W,Z)$ is the 128-bit value X_{k+l+1}, where X_i is recursively defined for $i = 0, 1, ..., k+l\text{-}1$, as follows:

a) $X_0 = 0^{128}$

b) $X_i = (X_{i\text{-}1} \oplus W_i) \bullet H$ $1 \leq i \leq k\text{-}1$ (this step is omitted if $k \leq 1$).

c) $X_k = (X_{k\text{-}1} \oplus (W_k||0^{128\text{-}u})) \bullet H$ (this step is omitted if k=0).

d) $X_i = (X_{i\text{-}1} \oplus Z_{i\text{-}k}) \bullet H$ $k+1 \leq i \leq k+l\text{-}1$ (this step is omitted if $l \leq 1$).

e) $X_{k+l} = (X_{k+l\text{-}1} \oplus (Z_l||0^{128\text{-}v})) \bullet H$ (this step is omitted if l=0).

f) $X_{k+l+1} = (X_{k+l} \oplus [\#_{64}(\text{len}(W)) || \#_{64}(\text{len}(Z))]) \bullet H$

10.6 Encryption procedure

The originator shall perform the following steps to protect a data string D and ensure the integrity of an additional authenticated data string A.

a) A variable length starting variable S shall be selected. This value shall be distinct for every message to be protected, and shall be made available to the recipient of the message. However, it is not necessary that this value be unpredictable or secret.

NOTE The value S can, for example, be generated using a counter maintained by the originator, and sent in clear text along with the protected message.

b) Partition D into a sequence of 128-bit blocks: $D_1, D_2, ..., D_m$, as follows. Let D_1 contain the first 128 bits of D, D_2 the next 128 bits, and so on, until D_m contains the final r bits, where $0 < r \leq 128$. Thus, D contains a total of $(m\text{-}1)n+r$ bits.

c) Let $H = e_K(0^{128})$.

d) If $\text{len}(S) = 96$ then let $Y_0 = S \,||\, 0^{31} \,||\, 1$. Otherwise let $Y_0 = G(H, \{\}, S)$.

e) For $i = 1, 2, ..., m\text{-}1$, perform the following two steps:

1) let $Y_i = \text{inc}(Y_{i-1})$;

2) let $C_i = D_i \oplus e_K(Y_i)$.

f) Let $Y_m = \text{inc}\,(Y_{m-1})$.

g) Let $C_m = D_m \oplus (e_K(Y_m))|_r$.

h) Let $T = (G(H, A, C_1 \,||\, C_2 \,||\, ... \,||\, C_m) \oplus e_K(Y_0))|_t$.

The output of the above process, i.e. the authenticated-encrypted version of D, shall be the bit-string:

$C = C_1 \,||\, C_2 \,||\, ... \,||\, C_m \,||\, T$

That is, a string of $(m\text{-}1)n+r+t$ bits, that is C contains precisely t bits more than D (although it is also necessary to convey the variable length starting variable S and the variable length additional authenticated data A to the recipient).

10.7 Decryption procedure

The recipient shall perform the following steps to decrypt and verify an authenticated-encrypted string C and to verify the additional authenticated data A.

a) If the length of C is less than t, then halt and output INVALID.

b) Let m and r be the unique integers defined so that $\text{len}(C) = (m\text{-}1)n+r+t$, where $0 < r \leq n$. Partition C into a sequence of blocks: $C_1, C_2, ..., C_m, T$ as follows. Let C_1 contain the first n bits of C, C_2 the next n bits of C, and so on, until C_m contains the next r bits of C. Finally, let T be the final t bits of C.

c) Let $H = e_K(0^{128})$.

d) If $\text{len}(S) = 96$ then let $Y_0 = S \,||\, 0^{31} \,||\, 1$. Otherwise, let $Y_0 = G(H, \{\}, S)$.

e) Let $T' = (G(H, A, C_1 \,||\, C_2 \,||\, ... \,||\, C_m) \oplus e_K(Y_0))|_t$.

f) If $T \neq T'$, then halt and output INVALID.

g) For $i = 1, 2, ..., m\text{-}1$, perform the following two steps:

1) let $Y_i = \text{inc}(Y_{i-1})$;

2) let $D_i = C_i \oplus e_K(Y_i)$.

h) Let $Y_m = \text{inc}\ (Y_{m\text{-}1})$.

i) Let $D_m = C_m \oplus (e_K(Y_m))|_r$.

j) Output D and the additional authenticated data A.

Annex A
(informative)

Guidance on the use of the mechanisms

A.1 General

The purpose of this annex is to provide guidance on the use of the mechanisms defined in this document. Use of each mechanism requires the choice of mechanism-specific parameters, and recommendations regarding choices of parameters are provided in A.2 to A.7. This clause provides recommendations made with respect to the requirements applying to all mechanisms in this document (see Clause 5).

All mechanisms require the selection of a block cipher from amongst those standardized in ISO/IEC 18033-3. The block length n of the block cipher is to be at least 64, and wherever possible use of a block cipher with $n = 128$ is recommended. The use of a block cipher with $n = 128$ is mandatory for mechanisms 2, 3 and 6.

All mechanisms also require that the originator and recipient of protected data share a secret key K. This key should be known only to these two parties and, possibly, by third parties trusted for this purpose by both originator and receiver. There are many ways in which this key can be established. However, the use of a key establishment mechanism specified in ISO/IEC 11770-2 or ISO/IEC 11770-3 is recommended.

All five mechanisms require the choice of a tag length. The choice of this parameter affects the degree of assurance provided to the recipient regarding the integrity and origin of a protected message. For further details, see ISO/IEC 9797-1:2011, Annex C.

A.2 Selection of mechanism

All the mechanisms specified in this document are believed to provide a high level of security. However, some mechanisms are more suitable than others for particular applications. When selecting a mechanism for use, the facts given in Table A.1 and those listed below should be taken into consideration.

Table A.1 — Properties of mechanisms

Mechanism number	2	3	4	5	6
Approximate number of block cipher operations required to encrypt a q-bit message	$12\lceil q/n \rceil$	$2q/n$	$2q/n$	Depends on encryption and MAC methods used	q/n
Licence possibly required	No	No	No	Depends on encryption and MAC methods used	No
Specifically designed for use with short messages	Yes	No	No	No	No
Message length is to be known prior to starting encryption	No	Yes	No	No	No
Starting value required	No	Yes	Yes	Yes	Yes
Previously standardized	Yes	Yes	No	No	Yes

a) Mechanisms 3 and 4 are methods for combining block cipher encryption in CTR mode (see ISO/IEC 10116) with a message authentication code.

b) Mechanism 5 provides a method for combining standardized methods for encryption and MAC computation. If implementations of such functions are already available, then mechanism 5 can have some implementation advantages.

c) Mechanism 6 is suitable for high-throughput hardware implementations, since it can be implemented without pipeline stalls.

A.3 Mechanism 2 (key wrap)

This mechanism requires the block cipher used to have n = 128. One of the block ciphers with this property specified in ISO/IEC 18033-3 shall be used (see Clause 5).

A.4 Mechanism 3 (CCM)

This mechanism requires the block cipher used to have n = 128. One of the block ciphers with this property specified in ISO/IEC 18033-3 shall be used (see Clause 5).

This mechanism requires the selection of a tag length parameter t (from the set {32, 48, 64, 80, 96, 112, 128}). The choice of the tag length t depends on the environment within which the mechanism is to be used. However, unless there are strong reasons to make a different choice, use of $t \geq 64$ is recommended.

This mechanism requires the selection of the length w (in octets) of the message field (from the set {2, 3, 4, 5, 6, 7, 8}). The choice of the octet-length of the message length field w also depends on the environment within which the mechanism is to be used. The choice of w does not affect the level of security provided by the mechanism. Larger values of w allow longer message lengths, although they also reduce the length of the remainder of the starting variable. Nevertheless, even if w is chosen to be the maximum possible value, i.e. w=8, 56 bits of the starting variable can be selected to ensure that a different starting variable is used for every message, which should be sufficient for most, if not all, practical applications. For the majority of applications, a value of w = 4, i.e. giving a maximum message length of $2^{32} \approx 4 \times 10^9$ octets, is likely to be adequate.

A.5 Mechanism 4 (EAX)

This mechanism requires the selection of a tag length parameter t ($t \leq n$). The choice of the tag length t depends on the environment within which the mechanism is to be used. However, unless there are strong reasons to make a different choice, use of $t \geq 64$ is recommended.

A.6 Mechanism 5 (encrypt-then-MAC)

This mechanism requires the choice of a mode of operation and a method for MAC computation. The security offered by the resulting authenticated encryption scheme depends on the security of the two underlying primitives.

For the encryption mode of operation, the security advice in ISO/IEC 10116 should be followed. In particular, if CBC mode is used to encrypt multiple plaintexts under the same key, then either:

a) a random starting variable should be used for the encryption of each plaintext; or

b) the first plaintext block should be reliably set to a value unique to the plaintext (e.g. a counter).

The choice of MAC technique should take into account the context of use of the authenticated encryption technique, and the advice provided in ISO/IEC 9797 (all parts) should be carefully followed. In particular, if a block cipher-based MAC from ISO/IEC 9797-1 is chosen, then:

a) MAC algorithm 1 should only be used if the message length is fixed; and

b) padding method 1 should only be used if the message length is fixed.

A.7 Mechanism 6 (GCM)

This mechanism requires the block cipher used to have n = 128. One of the block ciphers with this property specified in ISO/IEC 18033-3:2010 shall be used (see Clause 5).

The variable length starting variable, S, should be selected such that $1 \leq \text{len}(S) \leq 2^{64}$. The requirement that starting variables are never re-used during the lifetime of a key is critical to the security of this mechanism.

The tag length t should be selected such that t is a multiple of 8 satisfying $96 \leq t \leq 128$ (t=32 and t=64 are also permitted for specialized applications, although these options should only be used with great care – detailed guidance on use of these tag lengths is provided in Appendix C of Reference [6]).

The data string, D, to which the authenticated encryption mechanism is to be applied shall satisfy $\text{len}(D) \leq 2^{39}-256$. The additional authenticated data string A should satisfy $\text{len}(A) \leq 2^{64}$.

The total number of data blocks and additional authenticated data blocks to which GCM should be applied for a fixed key K should be at most 2^{64}. In addition, the total number of invocations of the encryption procedure for any given key should be at most 2^{32}, unless $\text{len}(S) = 96$ for every use of that key.

Annex B
(informative)

Numerical examples

B.1 General

This annex contains worked examples of the operation of the mechanisms specified in this document.

B.2 Mechanism 2 (key wrap)

Examples of the operation of this mechanism with the AES block cipher are provided in IETF RFC 3394.[9]

B.3 Mechanism 3 (CCM)

The following six examples of message (D_i), ciphertext (C_i) and tag (T_i) triples were all generated using the AES block cipher, in each case using t=128 and w=2 (and hence S should contain 104 bits). The examples are all presented using hexadecimal notation. The same key K and starting variable S are used for each of these six examples, namely:

K: 000102030405060708090A0B0C0D0E0F

S: 000102030405060708090A0B0C

D_1: The empty string (i.e. L=0)

C_1: The empty string

T_1: 54C92FE45510D6B3B0D46EAC2FEE8E63

D_2: 0001020304050607

C_2: 1635B68B570CFC85

T_2: 2734A0447531C02916CF8B9A494C3AD1

D_3: 000102030405060708090A0B0C0D0E0F

C_3: 1635B68B570CFC85529E39AC913910D7

T_3: C7C5C394B685B08B3F00DCD81256F0D0

D_4: 000102030405060708090A0B0C0D0E0F
1011121314151617