

**TCVN**

**TIÊU CHUẨN QUỐC GIA**

**TCVN 12197:2024  
ISO/IEC 19772:2020**

**Xuất bản lần 2**

**AN TOÀN THÔNG TIN – MÃ HÓA CÓ XÁC THỰC**

*Information security – Authenticated encryption*

**HÀ NỘI – 2024**

## Mục lục

<b>Lời nói đầu .....</b>	<b>5</b>
<b>Giới thiệu.....</b>	<b>6</b>
<b>1 Phạm vi áp dụng .....</b>	<b>7</b>
<b>2 Tài liệu viện dẫn .....</b>	<b>7</b>
<b>3 Thuật ngữ và định nghĩa .....</b>	<b>7</b>
<b>4 Ký hiệu và chữ viết tắt.....</b>	<b>9</b>
<b>5 Các yêu cầu.....</b>	<b>10</b>
<b>6 Cơ chế mã hóa có xác thực 2 (Key Wrap).....</b>	<b>11</b>
6.1 Tổng quan .....	11
6.2 Ký hiệu riêng.....	11
6.3 Yêu cầu riêng .....	11
6.4 Quá trình mã hóa.....	11
6.5 Quy trình giải mã .....	12
<b>7 Cơ chế mã hóa có xác thực 3 (CCM).....</b>	<b>12</b>
7.1 Tổng quan .....	12
7.2 Ký hiệu riêng.....	12
7.3 Yêu cầu riêng .....	13
7.4 Quá trình mã hóa.....	13
7.5 Quy trình giải mã .....	14
<b>8 Cơ chế mã hóa có xác thực 4 (EAX).....</b>	<b>15</b>
8.1 Tổng quan .....	15
8.2 Ký hiệu riêng.....	15
8.3 Yêu cầu riêng .....	15
8.4 Định nghĩa hàm $M$ .....	15
8.5 Quá trình mã hóa.....	16
8.6 Quy trình giải mã .....	16
<b>9 Cơ chế mã hóa có xác thực 5 (Encrypt-then-MAC).....</b>	<b>16</b>
9.1 Tổng quan .....	16
9.2 Ký hiệu riêng.....	17
9.3 Yêu cầu riêng .....	17
9.4 Quá trình mã hóa.....	17
9.5 Quy trình giải mã .....	18
<b>10 Cơ chế mã hóa có xác thực (GCM).....</b>	<b>18</b>
10.1 Tổng quan .....	18
10.2 Ký hiệu riêng.....	18

10.3	Yêu cầu riêng.....	19
10.4	Định nghĩa về phép nhân •.....	19
10.5	Định nghĩa hàm $G$ .....	19
10.6	Quá trình mã hóa .....	20
10.7	Quy trình giải mã.....	20
<b>Phụ lục A (tham khảo)</b>	<b>Hướng dẫn sử dụng các cơ chế.....</b>	<b>22</b>
<b>Phụ lục B (tham khảo)</b>	<b>Ví dụ.....</b>	<b>25</b>
<b>Phụ lục C (quy định)</b>	<b>Mô-đun ASN.1 .....</b>	<b>29</b>
<b>Thư mục tài liệu tham khảo.....</b>		<b>30</b>

**Lời nói đầu**

TCVN 12197:2024 (ISO/IEC 19772:2020) thay thế TCVN 12197:2018 (ISO/IEC 19772:2009).

TCVN 12197:2024 (ISO/IEC 19772:2020) do Ban Cơ yếu Chính phủ biên soạn, Bộ Quốc phòng đề nghị, Ủy ban Tiêu chuẩn Đo lường Chất lượng Quốc gia thẩm định, Bộ Khoa học và Công nghệ công bố.

## Giới thiệu

Khi dữ liệu được gửi từ nơi này đến nơi khác, thông thường cần phải bảo vệ dữ liệu đó theo một cách nào đó trong khi nó đang được vận chuyển, ví dụ: chống lại việc nghe trộm hoặc sửa đổi trái phép. Tương tự như vậy, khi dữ liệu được lưu trữ trong một môi trường mà các bên không được phép có thể truy cập, nó có thể cần thiết để bảo vệ nó.

Nếu tính an toàn của dữ liệu cần được bảo vệ, ví dụ: Để chống lại việc nghe trộm, thì một giải pháp là sử dụng mã hóa, như được quy định trong TCVN 11367 (ISO/IEC 18033) và ISO/IEC 10116. Ngoài ra, nếu cần bảo vệ dữ liệu chống lại việc sửa đổi, tức là bảo vệ tính toàn vẹn, thì mã xác thực thông điệp (MAC) như được chỉ định trong TCVN 11495 ((ISO/IEC 9797) (, hoặc chữ ký số như được chỉ định trong TCVN 11495 (ISO/IEC 9797) và TCVN 12214 (ISO/IEC 14888), có thể được sử dụng. Nếu cả tính an toàn và tính toàn vẹn đều được yêu cầu, thì một khả năng là sử dụng cả mã hóa và MAC hoặc chữ ký. Mặc dù các hoạt động này có thể được kết hợp theo nhiều cách, nhưng không phải tất cả sự kết hợp của các cơ chế như vậy đều cung cấp các đảm bảo về độ an toàn giống nhau. Do đó, cần xác định chi tiết chính xác cách kết hợp các cơ chế toàn vẹn và an toàn để cung cấp mức độ an toàn tối ưu. Hơn nữa, trong một số trường hợp, có thể đạt được hiệu quả đáng kể bằng cách xác định một phương pháp xử lý dữ liệu duy nhất với mục tiêu cung cấp cả tính an toàn và tính toàn vẹn.

Trong tiêu chuẩn này, các cơ chế mã hóa có xác thực được xác định. Đây là những phương pháp xử lý dữ liệu để bảo vệ cả tính toàn vẹn và tính an toàn. Chúng thường liên quan đến sự kết hợp cụ thể của tính toán MAC và mã hóa dữ liệu hoặc sử dụng thuật toán mã hóa theo cách đặc biệt để cung cấp cả tính toàn vẹn và bảo vệ bí mật.

Các phương pháp được chỉ định trong tiêu chuẩn này đã được thiết kế để tối đa hóa mức độ an toàn và cung cấp khả năng xử lý dữ liệu hiệu quả. Một số kỹ thuật được định nghĩa ở đây như "chứng minh an toàn" bằng toán học, tức là các lập luận chặt chẽ hỗ trợ tính hợp lý của chúng.

## An toàn thông tin - Mã hóa có xác thực

Information security - Authenticated encryption

### 1 Phạm vi áp dụng

Tiêu chuẩn này quy định cụ thể năm cơ chế mã hóa có xác thực, tức là xác định các cách để xử lý xâu dữ liệu với các mục tiêu an toàn sau:

- Tính bí mật của dữ liệu, tức là bảo vệ chống lại tiết lộ dữ liệu trái phép,
- Tính toàn vẹn của dữ liệu, tức là cho phép người nhận dữ liệu có thể xác minh rằng dữ liệu chưa bị sửa đổi,
- Xác thực nguồn gốc dữ liệu, tức là cho phép người nhận dữ liệu xác minh định danh của người khởi tạo dữ liệu.

Tất cả năm cơ chế được quy định trong tiêu chuẩn này đều dựa trên một thuật toán mã khóa và yêu cầu người khởi tạo và người nhận dữ liệu được bảo vệ phải chia sẻ khóa bí mật cho mã khóa này. Việc quản lý khóa nằm ngoài phạm vi của tiêu chuẩn này; kỹ thuật quản lý khóa được quy định trong tiêu chuẩn TCVN 7817 (ISO/IEC 11770).

Bốn cơ chế trong tiêu chuẩn này, cụ thể là cơ chế 3, 4, 5 (chỉ dành cho biến thể AAD) và 6, cho phép dữ liệu đã xác thực mã không cần phải mã hóa. Nghĩa là, các cơ chế này cho phép một chuỗi dữ liệu đã được bảo vệ được chia thành hai phần:  $D$  là chuỗi dữ liệu được mã hóa và được bảo vệ tính toàn vẹn;  $A$  (dữ liệu xác thực bổ sung) dùng để bảo vệ tính toàn vẹn nhưng không được mã hóa. Trong tất cả các trường hợp, chuỗi  $A$  có thể rỗng.

**CHÚ THÍCH:** Ví dụ về các kiểu dữ liệu có thể cần phải được gửi dưới dạng không mã hóa nhưng tính toàn vẹn phải được bảo vệ, bao gồm: địa chỉ, số cổng, số thứ tự, số phiên bản giao thức và các trường giao thức mạng khác chỉ ra cách thức bản gốc được kiểm soát, được chuyển tiếp hoặc được xử lý.

### 2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau rất cần thiết cho việc áp dụng tiêu chuẩn này. Đối với các tài liệu viện dẫn ghi năm công bố thì áp dụng phiên bản được nêu. Đối với các tài liệu viện dẫn không ghi năm công bố thì áp dụng phiên bản mới nhất, bao gồm cả các sửa đổi, bổ sung (nếu có).

TCVN 11495 (ISO/IEC 6767), Công nghệ thông tin - Các kỹ thuật an toàn - Mã xác thực thông điệp (MAC).

TCVN 12213:2018 (ISO/IEC 10116:2017), Công nghệ thông tin - Các kỹ thuật an toàn - Chế độ hoạt động của mã khối n-bit.

TCVN 11367-3:2016 (ISO/IEC 18033-3:2010), Công nghệ thông tin - Các kỹ thuật an toàn - Thuật toán mật mã - Phần 3: Mã khóa.

### 3 Thuật ngữ và định nghĩa

Tiêu chuẩn này sử dụng các thuật ngữ và định nghĩa sau:

#### 3.1

**Dữ liệu xác thực bổ sung** (additional authenticated data)

#### 3.2

**Mã hóa có xác thực** (authenticated encryption)

(Có tính đảo ngược) Biến đổi dữ liệu bằng thuật toán mã hóa để tạo ra *bản mã* (3.5) không thể bị truy xuất bởi thực thể trái phép mà không bị phát hiện, tức là nó cung cấp bảo vệ tính bí mật dữ liệu, *tính toàn vẹn dữ liệu* (3.6) và xác thực nguồn gốc dữ liệu.

### 3.3

#### Cơ chế mã hóa có xác thực (authenticated encryption mechanism)

Kỹ thuật mã hóa được sử dụng để bảo vệ tính bí mật, đảm bảo nguồn gốc, tính toàn vẹn của dữ liệu, trong đó bao gồm hai quá trình thành phần: thuật toán *mã hóa* (3.8) và thuật toán *giải mã* (3.7).

### 3.4

#### Mã khối (block cipher)

Hệ mật đổi xứng (3.15) với tính chất là thuật toán *mã hóa* (3.8) thao tác trên một khối của *bản rõ* (3.13), nghĩa là trên một xâu bit có độ dài xác định, kết quả cho ra một khối của *bản mã* (3.5).

[Nguồn: 2.8, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.5

#### Bản mã (ciphertext)

Dữ liệu đã được biến đổi để giấu thông tin trong đó.

[Nguồn: 2.10, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.6

#### Tính toàn vẹn dữ liệu (data integrity)

Tính chất của dữ liệu đã không bị thay đổi hay bị làm hư hại theo một cách trái phép.

[Nguồn: 3.4, TCVN 11495-1:2016 (ISO/IEC 9797-1:2011)].

### 3.7

#### Giải mã (decryption)

Phép toán ngược với phép mã hóa tương ứng.

[Nguồn: 2.13, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.8

#### Mã hóa (encryption)

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là giấu nội dung thông tin của dữ liệu.

[Nguồn: 2.15, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.9

#### Hệ mật (encryption system)

Kỹ thuật mật mã sử dụng để bảo vệ bí mật dữ liệu bao gồm ba quá trình thành phần: thuật toán mã hóa, thuật toán giải mã và phương pháp tạo khóa.

[Nguồn: 2.17, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)]

### 3.10

#### Khóa (key)

Dãy các ký tự điều khiển sự vận hành của các thuật toán mật mã (ví dụ: phép mã hóa, giải mã).

[Nguồn: 2.21, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.11

#### Mã xác thực thông điệp (message authentication code)

#### MAC

Chuỗi các bit được tạo ra của một thuật toán MAC.

[Nguồn: 3.9, TCVN 11495-1:2016 (ISO/IEC 9797-1:2011)].

### 3.12

#### **Phân chia** (partition)

Quá trình chia mỗi chuỗi các bit có độ dài tùy ý thành một chuỗi các khối, trong đó độ dài của mỗi khối là  $n$  bit, ngoại trừ khối sau cùng chứa  $r$  bit với  $0 < r \leq n$ .

### 3.13

#### **Bản rõ** (plaintext)

Thông tin chưa được mã hóa.

[Nguồn: 2.24, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.14

#### **Khóa bí mật** (secret key)

Khóa sử dụng cho kỹ thuật mật mã đối xứng và được dùng bởi một tập thực thể xác định.

[Nguồn: 2.27, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 3.15

#### **Hệ mật đối xứng** (symmetric encryption system)

Hệ mật dựa trên kỹ thuật mật mã đối xứng.

[Nguồn: 2.33, TCVN 11367-1:2016 (ISO/IEC 18033-1:2015)].

### 4 Ký hiệu và chữ viết tắt

- A Dữ liệu được xác thực bổ sung.
- C Chuỗi dữ liệu được mã hóa - xác thực.
- D Chuỗi dữ liệu sẽ áp dụng phương pháp mã hóa có xác thực.
- d Thuật toán giải mã khôi bản mã;  $d_K(Y)$  biểu thị kết quả giải mã khôi bản mã khôi  $Y$   $n$ -bit bằng cách sử dụng khóa bí mật  $K$ .
- e Thuật toán mã hóa khôi bản mã;  $e_K(X)$  biểu thị kết quả giải mã khôi bản mã khôi  $X$   $n$ -bit bằng cách sử dụng khóa bí mật  $K$ .
- K Khối bản mã với khóa bí mật được chia sẻ bên gửi và bên nhận dữ liệu áp dụng cơ chế mã hóa có xác thực.
- m Số lượng khôi biến thể của  $D$  được phân chia.
- n Kích thước khôi (tính theo bit) cho một khôi bản mã.
- t Kích thước tag (tính theo bit).
- $0^i$  Khối có  $i$  bit "0".
- $1^i$  Khối có  $i$  bit "1".
- $\oplus$  Phép toán thao tác bit XOR. Thực hiện tính toán (trên từng bit) với hai chuỗi bit có cùng độ dài để tạo ra một chuỗi bit mới có cùng độ dài với hai chuỗi bit ban đầu.

	Nối các chuỗi bit, tức là nếu $A$ và $B$ là các khối bit, thì $A \parallel B$ là khối các bit có được qua việc nối $A$ và $B$ lại theo các thứ tự cụ thể.
#	Hàm chuyển đổi số $a$ thành một khối có kích thước $a$ -bit. Nếu $k$ là một số nguyên ( $0 \leq k < 2^a$ ) thì $\#_a(k)$ là khối $a$ -bit, khi khối này được xem như là biểu diễn dưới dạng nhị phân với bit trọng số cao nhất ở bên trái, bằng $k$ .
$\#^{-1}$	Hàm chuyển đổi một khối bit thành số. Nếu $A$ là một khối bit thì $\#^{-1}(A)$ là một số nguyên không âm mà biểu diễn nhị phân là $A$ . Do đó, nếu $A$ có $n$ bit thì $\#_n(\#^{-1}(A)) = A$ .
$X _s$	Cắt ngắn bên trái khối bit $X$ . Nếu $X$ có độ dài bit lớn hơn hoặc bằng $s$ thì $X _s$ là khối $s$ -bit ngoài cùng bên trái của $X$ .
$X ^s$	Cắt ngắn bên phải khối bit $X$ . Nếu $X$ có độ dài bit lớn hơn hoặc bằng $s$ thì $X ^s$ là khối $s$ -bit ngoài cùng bên phải của $X$ .
$X \ll 1$	Dịch trái khối bit $X$ đi 1 vị trí. Bit tận cùng phía phải của $Y = X \ll 1$ sẽ luôn được đặt về "0".
$X \gg 1$	Dịch phải khối bit $X$ đi 1 vị trí. Bit tận cùng phái trái của $Y = X \gg 1$ sẽ luôn được đặt về "0".
<i>len</i>	Hàm lấy một chuỗi bit $X$ làm đầu vào và cho đầu ra là số lượng các bit có trong $X$ .
<i>mod</i>	Nếu $a$ và $b > 0$ là các số nguyên thì $a \bmod b$ biểu thị số nguyên $c$ duy nhất sao cho: 1) $0 \leq c < b$ ; 2) $a - c$ là bội số của $b$ .

## 5 Các yêu cầu

Các cơ chế mã hóa có xác thực được quy định trong tiêu chuẩn này có các yêu cầu sau đây:

Bên khởi tạo và bên nhận dữ liệu sử dụng cơ chế mã hóa có xác thực phải:

- Thỏa thuận về việc sử dụng một cơ chế cụ thể từ những cơ chế được quy định trong tiêu chuẩn này.
- Thỏa thuận về việc sử dụng một mã khối cụ thể được sử dụng với cơ chế (phải sử dụng một mã khối được tiêu chuẩn hóa trong TCVN 11367-3:2016 (ISO/IEC 18033-3:2010)).
- Chia sẻ một khóa bí mật  $K$ : trong tất cả các cơ chế ngoại trừ cơ chế mã hóa có xác thực 5, đây sẽ là một khóa cho mã khối được chọn và trong cơ chế 5 nó sẽ là một khóa được sử dụng làm đầu vào cho quá trình dẫn xuất khóa.

Ngoài ra, mỗi cơ chế phải có các yêu cầu cụ thể được liệt kê ở ngay trước phần mô tả cơ chế.

Phụ lục A: Cung cấp hướng dẫn về việc sử dụng các cơ chế được xác định trong tiêu chuẩn này.

Phụ lục B: Các ví dụ bằng số về hoạt động của các cơ cấu được chỉ định trong tiêu chuẩn này.

Phụ lục C: Cung cấp các mã nhận dạng đối tượng sẽ được sử dụng để xác định các cơ chế được định nghĩa trong tiêu chuẩn này.

## 6 Cơ chế mã hóa có xác thực 2 (Key Wrap)

### 6.1 Tổng quan

Phần này trình bày về cơ chế mã hóa có xác thực thường được biết đến với tên gọi là bọc khóa (Key Wrap - KW).

**CHÚ THÍCH 1:** Lược đồ này được thiết kế cơ bản cho mã hóa có sử dụng xác thực khóa và các thông tin đi kèm. Nó được thiết kế để sử dụng với các chuỗi dữ liệu ngắn. Tuy nhiên, lược đồ có thể được sử dụng với các chuỗi dữ liệu có độ dài tùy ý (tối đa lên tới  $2^{67}$ -bit), dù không đạt hiệu quả cho việc bảo vệ với các thông điệp dài.

**CHÚ THÍCH 2:** Chế độ này được biết đến với tên gọi là bọc khóa (KW) khi mã khối AES được sử dụng, trong đó AES là chữ viết tắt của Advanced Encryption Standard, một thuật toán mã khối được trình bày chi tiết trong TCVN 11367-3:2016 (ISO/IEC 18033-3:2010). AES-KW cũng được trình bày chi tiết trong tài liệu tham khảo [7] và [9].

### 6.2 Ký hiệu riêng

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

$C_0, C_1, \dots, C_M$	Chuỗi ( $m + 1$ ) khối 64-bit thu được là đầu ra của quá trình mã hóa có xác thực.
$D_1, D_2, \dots, D_m$	Chuỗi $m$ khối 64-bit thu được bằng cách phân chia $D$ , tức là $64m = \text{len}(D)$ .
$R_1, R_2, \dots, R_m$	Chuỗi $m$ khối 64-bit được tính toán trong quá trình mã hóa và giải mã.
$Y$	Khối 64-bit được sử dụng trong quá trình mã hóa và giải mã.
$Z$	Khối 128-bit được tính toán trong quá trình mã hóa và giải mã.

### 6.3 Yêu cầu riêng

Mã khối được sử dụng trong cơ chế này phải là mã khối 128-bit, tức là phải có  $n = 128$ .

Chuỗi dữ liệu  $D$  được bảo vệ bằng cách áp dụng cơ chế này phải chứa ít nhất 128-bit và phải là bội của 64-bit (tức độ dài của  $D$  phải là  $64m$  với một số nguyên  $m > 1$ ).

### 6.4 Quá trình mã hóa

Bên gửi thực hiện theo các bước sau để bảo vệ chuỗi dữ liệu  $D$ :

- d) Phân chia  $D$  thành một chuỗi  $m$  khối 64-bit  $D_1, D_2, \dots, D_m$ , do đó  $D_1$  chứa 64-bit đầu tiên của  $D$ ,  $D_2$  chứa 64-bit tiếp theo của  $D$  và tiếp tục cho đến hết  $D$ .
- e) Đặt  $Y$  là khối 64-bit biểu diễn theo hệ hexa A6A6A6A6A6A6A6A6 hoặc theo hệ nhị phân (10100110 10100110 ... 10100110).
- f) Cho  $i = 1, 2, \dots, m$ :

$$\text{Đặt } R_i = D_i.$$

- g) Cho  $i = 1, 2, \dots, 6m$  thực hiện 04 bước sau:
- h) Đặt  $Z = e_K(Y \parallel R_1)$ ;
- i) Đặt  $Y = Z|_{64} \oplus \#_{64}(i)$ ;
- j) Với  $j = 1, 2, \dots, m-1$ :

$$\text{Đặt } R_j = R_{j+1};$$

- k) Đặt  $R_m = Z|^{64}$ .
- l) Đặt  $C_0 = Y$ .
- m) Với  $i = 1, 2, \dots, m$ :

$$\text{Đặt } C_i = R_i.$$

Đầu ra của quá trình trên hay bản mã hóa có xác thực của  $D$  là chuỗi bit:

$$C = C_0 \parallel C_1 \parallel \dots \parallel C_m$$

Hay  $C$  là một chuỗi  $(64(m + 1))$ -bit, nghĩa là  $C$  chứa nhiều hơn  $D$  64-bit.

## 6.5 Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi  $C$  đã được mã hóa có xác thực.

- n) Nếu  $\text{len}(C)$  không là một bội số của 64 hoặc nhỏ hơn 192, dừng và thông báo "Không hợp lệ".
- o) Chia  $C$  thành chuỗi  $(m+1)$  khối 64 bit  $C_0, C_1, \dots, C_m$ , do đó  $C_0$  chứa 64 bit đầu của  $C$ ,  $C_1$  chứa 64 bit tiếp theo và cứ thế tiếp tục.
- p) Đặt  $Y = C_0$ .
- q) Cho  $i = 1, 2, \dots, m$ :

$$\text{Đặt } R_i = C_i.$$

- r) Cho  $i = 6m, 6m-1, \dots, 1$ , giảm đi 1, thực hiện 04 bước sau:
- s) Đặt  $Z = d_K([Y \oplus \#_{64}(i)] || R_m)$ ;
- t) Đặt  $Y = Z|_{64}$ ;
- u) Cho  $j = m, m-1, \dots, 2$ :

$$\text{Đặt } R_j = R_{j+1};$$

- v) Đặt  $R_1 = Z|^{64}$ .
- w) Nếu  $Y = (10100110 10100110 \dots 10100110)$ , thì đầu ra là  $D = R_1 || R_2 || \dots || R_m$ . Ngược lại thông báo "Không hợp lệ".

## 7 Cơ chế mã hóa có xác thực 3 (CCM)

### 7.1 Tóm quan

Phần này quy định một cơ chế mã hóa có xác thực thường được gọi là CCM (Counter with CBC-MAC).

CHÚ THÍCH: CCM là do Whiting, Housley và Ferguson đề xuất [10]. Phiên bản CCM được quy định ở đây là trường hợp đặc biệt của CCM được định nghĩa trong [8] và [10].

### 7.2 Ký hiệu riêng

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

$B$	Khối bit được sử dụng trong tính toán giá trị thẻ (tag).
$B_1, B_2, \dots, B_v$	Chuỗi khối bit (mỗi khối $n$ -bit) được sử dụng trong tính toán giá trị tag.
$C_1, C_2, \dots, C_m$	Chuỗi gồm $m$ khối có kích thước 128-bit là một phần đầu ra thu được của cơ chế mã hóa có xác thực.
$D_1, D_2, \dots, D_m$	Chuỗi gồm $m$ khối có kích thước khối 128-bit thu được bằng việc phân chia phiên bản có đệm của $D$ .
$F$	Bộ tám (octet) làm cờ.
$L$	Độ dài của $D$ (tính theo octet), ngoại trừ phần đệm và độ dài khối $D_0$ .
$r$	Số lượng octet của $D$ trong khối $D_m$ .
$S$	Biến khởi tạo (trong 120-8w bit).
$T$	Giá trị tag của bản rõ (trong $t$ -bit).
$T'$	Giá trị tag được tính toán lại, sinh ra trong quá trình giải mã.
$U$	Giá trị tag đã được mã hóa (trong $t$ -bit).
$v$	Biến được sử dụng trong tính toán giá trị tag.

w	Độ dài trường thông điệp tính theo octet.
X	Khối 128-bit được tính toán trong quá trình mã hóa và giải mã.
Y	Khối 128-bit được tính toán trong quá trình mã hóa và giải mã.

### 7.3 Yêu cầu riêng

Trước khi bắt đầu sử dụng cơ chế này, bên gửi và bên nhận dữ liệu có áp dụng cơ chế mã hóa có xác thực phải thỏa thuận:

- a)  $t$  là độ dài thẻ (tag) tính theo bit;  $t$  phải được chọn từ tập hợp {32, 48, 64, 80, 96, 112, 128};
- b)  $w$  là độ dài tính bằng bộ tám (octet) của trường độ dài thông điệp;  $w$  phải được chọn từ tập hợp {2, 3, 4, 5, 6, 7, 8}.

CHÚ THÍCH: Việc chọn  $w$  ảnh hưởng đến độ dài tối đa của thông điệp cần được bảo vệ. Độ dài tối đa của thông điệp là  $2^{8w+3}$  bit hay  $2^{8w}$  octet.

Mã khối được sử dụng trong cơ chế này phải là mã khối 128-bit, tức mã khối phải có  $n = 128$ .

Chuỗi dữ liệu  $D$  đã được bảo vệ qua áp dụng cơ chế này và chuỗi dữ liệu  $A$  được xác thực bổ sung, phải chứa toàn bộ số lượng octet, tức độ dài của chuỗi phải là một bội của 8-bit (tức là  $\text{len}(D)$  và  $\text{len}(A)$  phải là bội số nguyên của 8).

### 7.4 Quá trình mã hóa

Bên khởi tạo sẽ thực hiện các bước sau để bảo vệ chuỗi dữ liệu  $D$ . Đặt  $L = \frac{\text{len}(D)}{8}$ , tức là  $L$  là số octets trong  $D$ .

- a) Phải chọn biến bắt đầu  $S$  chứa 15-w octet (tức là 120-8w bit) sẽ được chọn. Biến này sẽ khác biệt đối với mọi thông điệp được bảo vệ và sẽ được cung cấp cho bên nhận thông điệp. Tuy nhiên, không yêu cầu giá trị này là không thể đoán trước được hoặc phải giữ bí mật.

CHÚ THÍCH 1: Ví dụ, giá trị  $S$  có thể được tạo bằng bộ đếm của bên khởi tạo và được gửi dưới dạng rõ ràng với thông điệp được bảo vệ.

- b) Đếm vào bên phải chuỗi dữ liệu  $D$  với 16-r octet "0" (tức là trong khoảng 0 đến 120-bit "0") sao cho phần đếm là biến thể của  $D$  là bội số của 128-bit. Sau đó, phân chia phần đếm là biến thể của  $D$  thành một chuỗi, gồm  $m$  khối 128-bit  $D_1, D_2, \dots, D_m$ , sao cho  $D_1$  chứa 128-bit đầu tiên của  $D$ ,  $D_2$  chứa 128-bit tiếp theo, ...

CHÚ THÍCH 2: Giá trị  $m$  cần thỏa mãn  $16(m - 1) < L \leq 16m$ .

- c) Nếu  $\text{len}(A) = 0$ , thì đặt cờ octet  $F = 0^2 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w-1)$ .
- d) Nếu  $\text{len}(A) > 0$ , thì đặt cờ octet  $F = 0 \parallel 1 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w-1)$ .

CHÚ THÍCH 3: Bit có trọng số cao nhất (ngoài cùng bên trái) của  $F$  là bit "đự trữ", tức là bit đó bằng 0 cho phiên bản của cơ chế được quy định ở đây, nhưng có thể được sử dụng cho các phiên bản khác của cơ chế trong tương lai (chưa được quy định). Bit tiếp theo của bit có trọng số cao nhất trong  $F$  được đặt bằng 0 để chỉ ra rằng tất cả dữ liệu được bảo vệ bởi cơ chế này đã được mã hóa.

- e) Đặt  $X = e_X(F \parallel S \parallel \#_{8w}(L))$ .

- f) Nếu  $\text{len}(A) > 0$ , thì thực hiện 06 bước sau:

- 1) Nếu  $0 < \text{len}(A) < 65280$ , thì đặt  $B = \#_{16}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;
- 2) Nếu  $65280 \leq \text{len}(A) < 232$ , thì đặt  $B = 1^{15} \parallel 0 \parallel \#_{32}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;
- 3) Nếu  $232 \leq \text{len}(A) < 264$ , thì đặt  $B = 1^{16} \parallel \#_{64}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;
- 4) Phân chia  $B$  thành một dãy các khối:  $B_1, B_2, \dots, B_v$ , như sau: đặt  $B_1$  chứa  $n$ -bit đầu tiên của  $B$ ,  $B_2$  chứa  $n$ -bit tiếp theo, ..., cho đến khi  $B_v$  chứa  $k$ -bit cuối cùng, trong đó  $0 < k \leq n$ . Do đó,  $\text{len}(B) = (v-1)n + k$ ;
- 5) Đếm bên phải  $B_v$  với  $n - k$  số không, tức là để  $B_v = B_v \parallel 0^{n-k}$ ;
- 6) Cho  $i = 1, 2, \dots, v$ :

Đặt  $X = e_K(X \oplus B_i)$ .

g) Cho  $i = 1, 2, \dots, m$ :

Đặt  $= e_K(X \oplus D_i)$ .

h) Đặt  $T = X|_t$ .

CHÚ THÍCH 4: Tag bắn rỗ  $T$  bằng với giá trị MAC được tính toán trên chuỗi dữ liệu  $B_1, B_2, \dots, B_p, D_1, D_2, \dots, D_m$ , sử dụng một sửa đổi nhỏ của Thuật toán MAC 1 được chỉ định trong TCVN 11495-1:2016 (ISO/IEC 9797-1:2011).

i) Đặt cờ octets  $F = (0^5 \parallel \#_3(w - 1))$ , và đặt  $Y = (F \parallel S \parallel 0^{8w})$ .

CHÚ THÍCH 5: Hai bit có trọng số cao nhất (ngoài cùng bên trái) của  $F$  là bit "dự trữ", tức là bit đó bằng 0 cho phiên bản của cơ chế được quy định ở đây, nhưng có thể được sử dụng cho các phiên bản khác của cơ chế trong tương lai (chưa được quy định). Ba bit có trọng số cao nhất tiếp theo của  $F$  được đặt bằng 0 để đảm bảo rằng bộ tám này khác với bộ tám làm cờ được sử dụng tại bước c ở trên.

j) Đặt  $U = T \oplus [e_K(Y)]|_t$ .

k) Với  $i = 1, 2, \dots, m - 1$ , thực hiện hai bước sau:

1) Đặt  $Y = (F \parallel S \parallel \#_{8w}(i))$ ;

2) Đặt  $C_i = D_i \oplus e_K(Y)$ .

l) Đặt  $Y = (F \parallel S \parallel \#_{8w}(m))$ , và đặt  $C_m = [D_m \oplus e_K(Y)]|_{8r}$ .

Đầu ra của quá trình trên, tức là biến thể được mã hóa có xác thực của  $D$ , phải là chuỗi bit:

$$C = C_1 \parallel C_2 \parallel \dots \parallel C_{m-1} \parallel C_m \parallel U$$

Nghĩa là, một chuỗi gồm  $(8L+t)$ -bit, nghĩa là  $C$  chứa chính xác  $t$ -bit nhiều hơn chuỗi dữ liệu gốc  $D$  [mặc dù cũng cần truyền tải  $(120-8w)$ -bit biến khởi tạo  $S$  và độ dài biến bổ sung được xác thực dữ liệu  $A$  đến bên nhận].

## 7.5 Quy trình giải mã

Bên nhận sẽ thực hiện các bước sau để giải mã và xác minh mã hóa - xác thực chuỗi  $C$ .

a) Nếu  $C$  không chứa toàn bộ số octet thì tạm dừng và thông báo "Không hợp lệ".

b) Nếu độ dài của  $C$  nhỏ hơn  $(t + 8)$ -bit, thi tạm dừng và thông báo "Không hợp lệ".

c) Gọi  $m$  và  $r$  là các số nguyên duy nhất sao cho  $C$  chứa tổng cộng  $(128(m-1) + 8r + t)$ -bit, trong đó  $0 < r \leq 16$ . Chia  $C$  thành một dãy các khối:  $C_1, C_2, \dots, C_m, U$  như sau. Đặt  $C_1$  chứa 128-bit đầu tiên của  $C$ ,  $C_2$  chứa 128-bit tiếp theo của  $C$ , ..., cho đến khi  $C_m$  chứa  $8r$ -bit tiếp theo của  $C$ . Cuối cùng, đặt  $U$  là  $t$ -bit cuối cùng của  $C$ .

d) Đặt cờ octet  $F = (0^5 \parallel \#_3(w - 1))$  và đặt  $Y = (F \parallel S \parallel 0^{8w})$ .

e) Đặt  $T = U \oplus [e_K(Y)]|_t$ .

f) Với  $i = 1, 2, \dots, m - 1$ , thực hiện hai bước sau:

1) Đặt  $Y = (F \parallel S \parallel \#_{8w}(i))$ ;

2) Đặt  $D_i = C_i \oplus e_K(Y)$ .

g) Đặt  $Y = (F \parallel S \parallel \#_{8w}(m))$ , và đặt  $D_m = C_m \oplus [e_K(Y)]|_{8r}$ .

h) Đặt  $D = D_1 \parallel D_2 \parallel \dots \parallel D_m$ , và đặt  $L = 16m - 16 + r$ .

i) Đếm bên phải  $D_m$  với 128-8r số "0", tức là đặt  $D_m = D_m \parallel 0^{128-8r}$ .

j) Nếu  $\text{len}(A) = 0$ , thì đặt cờ octets  $F = 0^2 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w - 1)$ .

k) Nếu  $\text{len}(A) > 0$ , thì đặt cờ octets  $F = 0 \parallel 1 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w - 1)$ .

l) Đặt  $X = e_K(F \parallel S \parallel \#_{8w}(L))$ .

m) Nếu  $\text{len}(A) > 0$ , sau đó thực hiện 06 bước sau:

1) Nếu  $0 < \text{len}(A) < 65280$  thì đặt  $B = \#_{16}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;

2) Nếu  $65280 \leq \text{len}(A) < 2^{32}$  thì đặt  $B = 1^{15} \parallel 0 \parallel \#_{32}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;

3) Nếu  $2^{32} \leq \text{len}(A) < 2^{64}$  thì đặt  $B = 1^{16} \parallel \#_{64}\left(\frac{\text{len}(A)}{8}\right) \parallel A$ ;

- 4) Phân chia  $B$  thành một dãy các khối:  $B_1, B_2, \dots, B_v$ , như sau: đặt  $B_1$  chứa  $n$ -bit đầu tiên của  $B$ ,  $B_2$  chứa  $n$ -bit tiếp theo, ..., cho đến khi  $B_v$  chứa  $k$ -bit cuối cùng, trong đó  $0 < k \leq n$ . Do đó,  $\text{len}(B) = (v - 1)n + k$ ;
- 5) Đệm bên phải  $B_v$  với  $n - k$  số "0", tức là để  $B_v = B_v \parallel 0^{n-k}$ ;
- 6) Cho  $i = 1, 2, \dots, v$ :

Đặt  $X = e_K(X \oplus B_i)$ .

- n) Cho  $i = 1, 2, \dots, m$ :

Đặt  $X = e_K(X \oplus D_i)$ .

- o) Đặt  $T' = X|_t$ .
- p) Nếu  $T = T'$ , thì xuất  $D$  như đã tính ở bước h) và A. Ngược lại, thông báo "Không hợp lệ".

## 8 Cơ chế mã hóa có xác thực 4 (EAX)

### 8.1 Tổng quan

Phần này trình bày về cơ chế mã hóa có sử dụng xác thực thường được biết đến với tên gọi EAX.

**CHÚ THÍCH:** EAX do Bellare, Rogaway và Wagner [2], EAX không phải là một từ viết tắt cụ thể nào.

### 8.2 Ký hiệu riêng

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

$C_1, C_2, \dots, C_m$	Chuỗi khối bit (mỗi khối $n$ -bit, ngoại trừ khối $C_m$ ) là thành phần đầu ra thu được của quá trình mã hóa có xác thực.
$D_1, D_2, \dots, D_m$	Chuỗi các khối bit (mỗi khối $n$ -bit, ngoại trừ khối $D_m$ ) thu được bằng cách phân chia $D$ .
$E_0, E_1, E_2$	Các khối $n$ -bit được tính toán trong quá trình mã hóa và giải mã.
$M$	Hàm được sử dụng trong quá trình mã hóa và giải mã.
$S$	Biến khởi tạo $S$ ( $n$ -bit).
$T$	Thẻ (tag) ( $t$ -bit), đính kèm với một thông điệp đã được mã hóa để bảo vệ tính toàn vẹn.
$T'$	Giá trị tag được tính toán lại, sinh ra trong quá trình giải mã.
$W$	Khối $n$ -bit được tính toán trong quá trình mã hóa và giải mã.

### 8.3 Yêu cầu riêng

Trong bất kỳ trường hợp nào sử dụng cơ chế này, bên gửi và bên nhận dữ liệu có áp dụng cơ chế mã hóa có sử dụng xác thực, phải thỏa thuận về  $t$ , độ dài tag được tính bằng bit, trong đó:  $0 < t \leq n$ .

### 8.4 Định nghĩa hàm $M$

Định nghĩa các thủ tục mã hóa và giải mã yêu cầu định nghĩa hàm  $M$  nhận một chuỗi bit có độ dài tùy ý và một khóa mã khối ở đầu vào và đưa một khối  $n$ -bit ở đầu ra. Định nghĩa hàm  $M$  như sau:

Nếu  $X$  là một chuỗi bit và  $K$  là một khóa cho mã khối được chọn, thì  $M_K(X)$  bằng một mã xác thực thông điệp (chưa bị cắt) được tính toán trên chuỗi  $X$  bằng cách sử dụng khóa  $K$  qua sử dụng thuật toán MAC 5 trong tiêu chuẩn TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), trong đó mã khối được sử dụng trong thuật toán MAC sẽ giống như thuật toán mã khối được chọn trong quá trình mã hóa có sử dụng xác thực.

**CHÚ THÍCH:** Thuật toán MAC 5 trong tiêu chuẩn TCVN 11495-1:2016 (ISO/IEC 9797-1:2011) còn được biết đến với tên gọi CMAC.

### 8.5 Quá trình mã hóa

Bên gửi thực hiện các bước sau để bảo vệ chuỗi dữ liệu  $D$ :

- x) Lựa chọn biến khởi tạo  $S$  chứa  $n$ -bit. Biến  $S$  là riêng biệt đối với mỗi bản tin được bảo vệ, phải sẵn sàng tại bên nhận bản tin. Tuy nhiên, giá trị này không nhất thiết phải bí mật hoặc không thể đoán trước.
- y) Đặt  $E_0 = M_K(0^n \parallel S)$ .
- z) Đặt  $E_1 = M_K(0^{n-1} \parallel 1 \parallel A)$ .
- aa) Đặt  $W = E_0$ .
- bb) Chia  $D$  thành một chuỗi các khối:  $D_1, D_2, \dots, D_m$ , như sau: đặt  $D_1$  chứa  $n$ -bit đầu tiên của  $D$ ,  $D_2$  chứa  $n$ -bit kế tiếp và cứ thế tiếp tục đến khi  $D_m$  chứa  $r$ -bit sau cùng, trong đó  $0 < r \leq n$ ; Do đó  $\text{len}(D) = (m-1)n + r$ .
- cc) Với  $i = 1, 2, \dots, m-1$ , thực hiện 02 bước sau:
- dd) Đặt  $C_i = D_i \oplus e_K(W)$ .
- ee) Đặt  $W = \#_n(\#^{-1}(W) + 1 \bmod 2^n)$ .
- ff) Đặt  $C_m = D_m \oplus [e_K(W)]_r$ .
- gg) Đặt  $E_2 = M_K(0^{n-2} \parallel 1 \parallel 0 \parallel C_1 \parallel C_2 \parallel \dots \parallel C_m)$ .
- hh) Đặt  $T = [E_0 \oplus E_1 \oplus E_2]_t$ .

Đầu ra của quá trình trên, tức là bản mã hóa có sử dụng xác thực của  $D$  là chuỗi bit:

$$C = C_1 \parallel C_2 \parallel \dots \parallel C_{m-1} \parallel C_m \parallel T$$

Hay  $C$  là một chuỗi  $((m-1)n + r + t)$ -bit, nghĩa là chứa nhiều hơn  $t$ -bit so với chuỗi dữ liệu gốc  $D$  (mặc dù nó phải cần mang  $n$ -bit biến khởi tạo  $S$  và dữ liệu  $A$  đã được xác thực bổ sung có độ dài thay đổi đến bên nhận).

### 8.6 Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi  $C$  đã được mã hóa có xác thực.

- ii) Nếu độ dài của  $C$  nhỏ hơn  $t$ , dừng và thông báo "Không hợp lệ".
- jj) Đặt  $m$  và  $r$  là các số nguyên duy nhất xác định sao cho  $C$  chứa tổng  $((m-1)n + r + t)$ -bit, trong đó  $0 < r \leq n$ . Chia  $C$  thành một chuỗi các khối:  $C_1, C_2, \dots, C_m, T$ . Đặt  $C_1$  chứa  $n$ -bit đầu tiên của  $C$ ,  $C_2$  chứa  $n$ -bit kế tiếp và cứ thế cho đến khi  $C_m$  chứa  $r$ -bit kế tiếp của  $C$ . Sau cùng, đặt  $T$  là  $t$ -bit cuối cùng của  $C$ .
- kk) Đặt  $E_0 = M_K(0^n \parallel S)$ .
- ll) Đặt  $E_1 = M_K(0^{n-1} \parallel 1 \parallel A)$ .
- mm) Đặt  $E_2 = M_K(0^{n-2} \parallel 1 \parallel 0 \parallel C_1 \parallel C_2 \parallel \dots \parallel C_m)$ .
- nn) Đặt  $T' = [E_0 \oplus E_1 \oplus E_2]_t$ .
- oo) Nếu  $T \neq T'$ , thì dừng và thông báo "Không hợp lệ".
- pp) Đặt  $W = E_0$ .
- qq) Với  $i = 1, 2, \dots, m-1$ , thực hiện 02 bước sau:
- rr) Đặt  $D_i = C_i \oplus e_K(W)$ ;
- ss) Đặt  $W = \#_n(\#^{-1}(W) + 1 \bmod 2^n)$ .
- tt) Đặt  $D_m = C_m \oplus [e_K(W)]_r$ .
- uu) Đầu ra là  $D$  và  $A$ .

## 9 Cơ chế mã hóa có xác thực 5 (Encrypt-then-MAC)

### 9.1 Tổng quan

Phần này trình bày về cơ chế mã hóa có xác thực được tạo nên từ sự kết hợp giữa một cơ chế mã hóa và một lược đồ MAC được xác định. Lược đồ này yêu cầu mã hóa dữ liệu được bảo vệ trước, sau đó tính toán MAC trên kết quả dữ liệu đã được mã hóa.

**CHÚ THÍCH:** Cơ chế Encrypt-then-MAC đã được phân tích bởi Bellare và Namprempre [1], là những người chứng minh sự an toàn của cơ chế này dựa trên giả thiết phương thức mã hóa và kỹ thuật MAC có các đặc tính an toàn nhất định.

## 9.2 Ký hiệu riêng

Trong tiêu chuẩn này áp dụng các ký hiệu và chữ viết tắt dưới đây:

- $C'$  Chuỗi bit thu được qua việc mã hóa chuỗi dữ liệu  $D$
- $\delta$  Hàm giải mã, tức là một hàm với đầu vào là một khóa mã khôi  $K_1$ , một biến khởi tạo  $S$  và một chuỗi dữ liệu đã được mã hóa  $C'$  và thông qua sử dụng chế độ vận hành đã chọn, đầu ra là chuỗi dữ liệu đã được giải mã; đầu ra được viết là  $\delta_{K_1, S}(C')$ .
- $\epsilon$  Hàm mã hóa, tức là hàm với đầu vào là một khóa mã khôi  $K_1$ , một biến khởi tạo  $S$ , một chuỗi dữ liệu  $D$  và thông qua sử dụng chế độ vận hành đã chọn, đầu ra là chuỗi dữ liệu đã được mã hóa; đầu ra được viết là  $\epsilon_{K_1, S}(D)$
- $f$  Hàm MAC; nếu  $X$  là chuỗi đầu vào và  $K_2$  là khóa MAC, thì đầu ra của hàm MAC được viết là  $f_{K_2}(X)$
- $K_1$  Khóa bí mật của mã khôi
- $K_2$  Khóa bí mật của hàm MAC
- $S$  Biến khởi tạo ( $n$ -bit)
- $T$  Tag ( $t$ -bit), đính kèm với thông điệp đã được mã hóa để bảo vệ tính toàn vẹn
- $T'$  Giá trị tag được tính toán lại, sinh ra trong quá trình giải mã.

## 9.3 Yêu cầu riêng

Trong bất kỳ trường hợp nào sử dụng cơ chế này, bên nhận và bên gửi dữ liệu áp dụng cơ chế mã hóa có sử dụng xác thực, phải thỏa thuận:

- a) Chế độ hoạt động của mã khôi được mô tả chi tiết trong tiêu chuẩn TCVN 12213:2018 (ISO/IEC 10116:2017) (Không được sử dụng chế độ ECB).
- b) Cơ chế tính toán MAC phải được chọn từ các kỹ thuật được trình bày chi tiết trong tiêu chuẩn TCVN 11495 (ISO/IEC 9797) (giả thiết rằng phương thức được chọn tạo ra tag có độ dài  $t$ -bit).
- c) Phương pháp để trích xuất một cặp khóa bí mật  $(K_1, K_2)$  từ khóa bí mật  $K$ , trong đó:  $K_1$  là khóa cho mã khôi được chọn và  $K_2$  là khóa cho cơ chế tính toán MAC được chọn.

CHÚ THÍCH 1:  $K$  phải được chọn sao cho số lượng các giá trị có thể có cho  $K$  ít nhất lớn hơn hoặc bằng số lượng các giá trị có thể có của khóa mã khôi và ít nhất cũng phải lớn hơn hoặc bằng số lượng giá trị khóa MAC có thể có.

CHÚ THÍCH 2: Các cơ chế có thể có thể trích xuất  $(K_1, K_2)$  từ khóa bí mật  $K$  bằng cách lấy các chuỗi bit (riêng biệt) từ  $K$  hoặc từ  $h(K)$ , trong đó  $h$  là hàm băm được chọn ra từ các hàm băm được trình bày chi tiết trong tiêu chuẩn TCVN 11816 (ISO/IEC 10118) (Tất cả các phần). Tổng quát hơn, có thể nhận được  $(K_1, K_2)$  từ khóa bí mật  $K$  bằng cách sử dụng hàm dẫn xuất được quy định trong ISO/IEC 11770-6:2016.

- d) Sử dụng cơ chế cơ bản (không hỗ trợ AAD) hay sử dụng cơ chế biến thể AAD. Nếu sử dụng cơ chế là biến thể AAD, thì chuỗi dữ liệu được xác thực bổ sung  $A$  sẽ chứa toàn bộ số octet (có thể bằng "0"), tức là  $len(A)$  sẽ là bội số nguyên của 8, nhưng phải chứa ít hơn  $2^{64}$  octet (hoặc thậm chí ít hơn tùy thuộc vào yêu cầu của lược đồ MAC được sử dụng).

Một khóa  $K$  duy nhất sẽ chỉ được sử dụng với một biến thể, tức là chỉ với biến thể cơ bản hoặc chỉ với biến thể AAD.

## 9.4 Quá trình mã hóa

Bên gửi phải thực hiện các bước sau để bảo vệ chuỗi dữ liệu  $D$  và nếu sử dụng cơ chế biến thể AAD, để đảm bảo tính toàn vẹn của chuỗi dữ liệu  $A$  đã được xác thực bổ sung.

- a) Chọn biến khởi tạo  $S$  thích hợp để sử dụng với chế độ hoạt động của mã khôi của các hoạt động sẽ chọn. Biến  $S$  là riêng biệt đối với mỗi thông điệp được bảo vệ theo khóa đã cho và phải sẵn sàng tại

bên nhận thông điệp. Các yêu cầu có thể có đối với  $S$  được mô tả chi tiết trong các phần tương ứng của tiêu chuẩn TCVN 12213:2018 (ISO/IEC 10116:2017) và hướng dẫn được đưa ra trong Phụ lục A, A.6

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

Nếu không sử dụng biến thể AAD:

- c) Đặt  $T = f_{K_2}(S \parallel C')$ .  
c) Nếu  $\text{len}(A)$  không phải là bội của 8 hoặc  $\geq 2^{67}$ , sau đó tạm dừng và thông báo "Không hợp lệ".

$$\text{Đặt } T = f_{K_2}\left(\frac{\text{len}(A)}{8}\right) \parallel A \parallel S \parallel C'.$$

Đầu ra của quá trình trên, tức là biến thể của  $D$  được mã hóa - xác thực, phải là chuỗi bit:

$$C = C' \parallel T, \text{ cùng với biến khởi tạo } SS.$$

### 9.5 Quy trình giải mã

Bên nhận phải thực hiện các bước sau để giải mã và xác minh một chuỗi  $C$  được mã hóa - xác thực, với biến khởi tạo  $S$  kèm theo và nếu biến AAD sử dụng cơ chế để xác minh toàn vẹn của dữ liệu được xác thực bổ sung  $A$ .

- a) Nếu độ dài của  $C$  nhỏ hơn  $t$  thì dừng, thông báo "Không hợp lệ".  
b) Đặt  $T$  là  $t$ -bit ngoài cùng bên phải của  $C$  và đặt  $C'$  bằng  $C$  với  $t$ -bit ngoài cùng bên phải bị loại bỏ, tức là  $C = C' \parallel t..$

Nếu không sử dụng biến AAD:

- c) Đặt  $T' = f_{K_2}(S \parallel C')$ .

Nếu sử dụng biến AAD:

- d) Nếu  $\text{len}(A)$  không phải là bội của 8 hoặc  $\geq 2^{67}$  thì tạm dừng và thông báo "Không hợp lệ".

$$\text{Đặt } T' = f_{K_2}\left(\#_{64}\left(\frac{\text{len}(A)}{8}\right)\right) \parallel A \parallel S \parallel C'.$$

- e) Nếu  $T \neq T'$  thì tạm dừng và thông báo "Không hợp lệ".

- f) Đặt  $D = \delta_{K_1, S}(C')$ .

- g) Đầu ra là  $D$ .

## 10 Cơ chế mã hóa có xác thực (GCM)

### 10.1 Tổng quan

Phần này trình bày về cơ chế mã hóa có xác thực thường được biết đến với tên gọi GCM (Galois/Counter Mode).

CHÚ THÍCH: GCM là do McGrew và Viega đề xuất [6].

### 10.2 Ký hiệu riêng

$C_1, C_2, \dots, C_m$  Chuỗi gồm  $m$  khối 128-bit (ngoại trừ khối  $C_m$  có thể chứa từ 1 đến 128-bit) là đầu ra của quá trình mã hóa có xác thực.

$D_1, D_2, \dots, D_m$  Chuỗi gồm  $m$  khối 128-bit (ngoại trừ khối  $D_m$ ) thu được bằng cách phân chia  $D$ .

$G$  Hàm được sử dụng trong quá trình mã hóa và giải mã (10.5)

$H$  Khối 128-bit được sử dụng trong quá trình mã hóa và giải mã.

$inc$  Hàm có đầu vào là một khối 128-bit và xuất ra một khối 128-bit, trong đó có  $X$  là một khối 128-bit:

$$inc(X) = (X|_{96}) \parallel \#_{32}(\#^{-1}(X|^{32}) + 1 \bmod 2^{32})$$

$r$	Số lượng bit trong khối cuối cùng của thông điệp được mã hóa, sau khi nó đã được phân chia thành các khối $n$ -bit, tức là thông điệp chứa $((m - 1)n + r)$ -bit.
$R$	Khối 128-bit được sử dụng trong tính toán phép nhân $GF(2^{128})$ .
$S$	Biến khởi tạo $S$ (độ dài tùy ý).
$T$	Tag ( $t$ -bit), được đính kèm với thông điệp đã được mã hóa để bảo vệ tính toàn vẹn.
$T'$	Giá trị tag được tính toán lại, sinh ra trong quá trình giải mã.
$U, V, W, Z$	Các khối 128-bit được sử dụng trong xác định tính toán phép nhân $GF(2^{128})$ .
$X_0, X_1, \dots, X_{k+l+1}$	Các khối 128-bit được sử dụng trong tính toán hàm $G$ .
$Y_0, Y_1, \dots, Y_m$	Chuỗi gồm các khối 128-bit được sử dụng trong quá trình mã hóa và giải mã.
{ }	Chuỗi bit có độ dài bằng 0.
•	Phép nhân trên trường $GF(2^{128})$ . Đa thức được dùng để xác định biểu diễn của $GF(2^{128})$ là $1 + \alpha + \alpha^2 + \alpha^7 + \alpha^{128}$ .

### 10.3 Yêu cầu riêng

Trong bất kỳ trường hợp nào sử dụng cơ chế này, bên gửi và bên nhận dữ liệu có áp dụng cơ chế mã hóa có sử dụng xác thực phải thỏa thuận: Độ dài tag  $t$ -bit, trong đó:  $t$  phải là một bội của 8 thỏa mãn  $96 \leq t \leq 128$  ( $t = 32$  và  $t = 64$  cũng được cho phép đối với các ứng dụng đặc biệt).

Mã khối được sử dụng với cơ chế này phải là mã khối 128-bit, tức là mã khối phải có  $n = 128$ .

### 10.4 Định nghĩa về phép nhân •

Giả thiết  $U$  và  $V$  là các khối 128-bit; thì  $W = U \bullet V$  được xác định như sau, trong đó:  $W$  cũng là một khối 128-bit. Lưu ý rằng, trong mô tả dưới đây,  $V$  biểu thị là bit thứ  $i$  của  $V$ , tức là  $V = v_0 \parallel v_1 \parallel \dots \parallel v_{127}$ . Ngoài ra,  $z_{127}$  biểu thị là bit ngoài cùng bên phải của  $Z$ .

- Đặt  $R = 11100001 \parallel 0^{120}$ .
- Đặt  $W = 0^{128}$ .
- Đặt  $Z = U$ .
- Cho  $i = 0, 1, \dots, 127$  thực hiện 02 bước sau:
  - Nếu  $v_i = 1$ , thì đặt  $W = W \oplus Z$ ;
  - Nếu  $z_{127} = 0$ , thì đặt  $Z = Z \gg 1$ . Cách khác, đặt  $Z = (Z \gg 1) \oplus R$ .

### 10.5 Định nghĩa hàm $G$

Các quá trình mã hóa và giải mã sử dụng hàm  $G$ , nhận đầu vào là khối 128-bit và hai chuỗi bit có độ dài tùy ý, đầu ra là một khối 128-bit. Đặt  $H$  là một khối 128-bit,  $W$  và  $Z$  là hai chuỗi bit có độ dài tùy ý (có thể rỗng). Giả thiết rằng  $k$  và  $u$  là các số nguyên duy nhất sao cho  $\text{len}(W) = 128(k - 1) + u$  và  $0 < u \leq 128$ ;

Tương tự cũng giả thiết rằng  $l$  và  $v$  là các số nguyên duy nhất sao cho  $\text{len}(Z) = 128(l - 1) + v$  và  $0 < v \leq 128$ . Đặt  $W_1, W_2, \dots, W_k$  là chuỗi có kích thước khối 128-bit (có thể không bao gồm  $W_k$  chứa  $u$ -bit sau cùng của  $W$ ) thu được qua việc phân chia  $W$ ; tương tự, đặt  $Z_1, Z_2, \dots, Z_l$ , là chuỗi có kích thước khối 128-bit (có thể không bao gồm  $Z_l$  chứa  $v$ -bit sau cùng của  $Z$ ) thu được qua việc phân chia  $Z$ .

Thì  $G(H, W, Z)$  là giá trị  $X_{k+l+1}$  128-bit, trong đó:  $X_i$  là xác định đệ quy với  $i = 0, 1, \dots, k + l + 1$ , như sau:

- Đặt  $X_0 = 0^{128}$

- b) Đặt  $X_i = (X_{i-1} \oplus W_i) \bullet H$   $1 \leq i \leq k-1$  (bỏ qua bước này nếu  $k \leq 1$ ).
- c) Đặt  $X_k = (X_{k-1} \oplus (W_k || 0^{128-u})) \bullet H$  (bỏ qua bước này nếu  $k = 0$ ).
- d) Đặt  $X_i = (X_{i-1} \oplus Z_{i-k}) \bullet H$   $k+1 < i \leq k+l-1$  (bỏ qua bước này nếu  $l \leq 1$ )
- e) Đặt  $X_{k+l} = (X_{k+l-1} \oplus (Z_l || 0^{128-v})) \bullet H$  (bỏ qua bước này nếu  $l = 0$ ).
- f) Đặt  $X_{k+l+1} = (X_{k+l} \oplus [\#_{64}(len(W)) || \#_{64}(len(Z))]) \bullet H$

## 10.6 Quá trình mã hóa

Bên gửi thực hiện các bước sau để bảo vệ chuỗi dữ liệu  $D$  và đảm bảo tính toàn vẹn của chuỗi dữ liệu đã được xác thực bổ sung  $A$ .

- a) Chọn biến khởi tạo  $S$  có độ dài tùy ý. Giá trị biến  $S$  phải là riêng biệt đối với mỗi thông điệp được bảo vệ và phải sẵn sàng tại bên nhận thông điệp. Tuy nhiên, giá trị này không nhất thiết phải bí mật hoặc không thể đoán trước.

CHÚ THÍCH: Giá trị  $S$  có thể được tạo ra bằng cách sử dụng một bộ đếm được duy trì bởi bên gửi và gửi đi trong một bản gốc định kèm với thông điệp đã được bảo vệ.

- b) Phân chia  $D$  thành một chuỗi các khối có kích thước 128-bit:  $D_1, D_2, \dots, D_m$ . Đặt  $D_1$  chứa 128-bit đầu tiên của  $D$ ,  $D_2$  chứa 128-bit kế tiếp và cứ thế tiếp tục cho đến khi  $D_m$  chứa  $r$ -bit sau cùng của  $D$ , trong đó  $0 < r \leq 128$ . Do đó,  $D$  chứa tổng số  $((m-1)n + r)$ -bit.
- c) Đặt  $H = e_K(0^{128})$ .
- d) Nếu  $len(S) = 96$  thì đặt  $Y_0 = S || 0^{31} || 1$ . Cách khác, đặt  $Y_0 = G(H, \{\}, S)$ .
- e) Cho  $i = 1, 2, \dots, m-1$  thực hiện 02 bước sau:
- 1) Đặt  $Y_i = inc(Y_{i-1})$ ;
  - 2) Đặt  $C_i = D_i \oplus e_K(Y_i)$ .
- f) Đặt  $Y_m = inc(Y_{m-1})$ .
- g) Đặt  $C_m = D_m \oplus (e_K(Y_m))|_r$ .

Đầu ra của quá trình trên, biến thể mã hóa - xác thực của  $D$  phải là chuỗi bit:

$$C = C_1 || C_2 || \dots || C_m || T$$

Hay  $C$  là một chuỗi  $(m-1)n + r + t$ , nghĩa là chuỗi  $C$  chứa nhiều hơn so với chuỗi dữ liệu gốc  $D$   $t$ -bit (mặc dù nó phải cần mang thêm  $n$ -bit biến khởi tạo  $S$  và dữ liệu  $A$  đã được xác thực bổ sung có độ dài thay đổi đến bên nhận).

## 10.7 Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi  $C$  đã được mã hóa có sử dụng xác thực và xác nhận dữ liệu  $A$  đã được xác thực bổ sung.

- a) Nếu độ dài của  $C$  nhỏ hơn  $t$  thì dừng và thông báo "Không hợp lệ".
- b) Đặt  $m$  và  $r$  là các số nguyên duy nhất xác định sao cho  $C$  chứa một tổng  $((m-1)n + r + t)$ -bit, trong đó  $0 < r \leq n$ . Chia  $C$  thành một chuỗi các khối:  $C_1, C_2, \dots, C_m, T$ . Đặt  $C_1$  chứa  $n$ -bit đầu tiên của  $C$ ,  $C_2$  chứa  $n$ -bit kế tiếp và cứ thế cho đến khi  $C_m$  chứa  $r$ -bit tiếp theo của  $C$ . Cuối cùng, đặt  $T$  là  $t$ -bit cuối cùng của  $C$ .
- c) Đặt  $H = e_K(0^{128})$ .
- d) Nếu  $len(S) = 96$  thì đặt  $Y_0 = S || 0^{31} || 1$ . Cách khác, đặt  $Y_0 = G(H, \{\}, S)$ .
- e) Nếu  $T' = (G(H, A, C_1 || C_2 || \dots || C_m) \oplus e_K(Y_0))|_t$ .
- f) Nếu  $T \neq T'$  thi dừng và thông báo "Không hợp lệ".
- g) Cho  $i = 1, 2, \dots, m-1$ , hãy thực hiện 02 bước sau:
- 1) Đặt  $Y_i = inc(Y_{i-1})$ ;
  - 2) Đặt  $D_i = C_i \oplus e_K(Y_i)$ .

- h) Đặt  $Y_m = inc(Y_{m-1})$ .
- i) Đặt  $D_m = C_m \oplus (e_K(Y_m))|_r$ .
- j) Đầu ra  $D$  và dữ liệu xác thực bổ sung  $A$ .

**Phụ lục A**  
 (tham khảo)  
**Hướng dẫn sử dụng các cơ chế**

#### A.1. Giới thiệu

Mục đích của phụ lục này là nhằm cung cấp hướng dẫn sử dụng các cơ chế đã được định nghĩa trong tiêu chuẩn này. Việc sử dụng từng cơ chế đòi hỏi cần chọn các tham số cụ thể cho mỗi cơ chế, chọn các tham số đánh giá khuyến nghị được đưa ra trong các phần từ A.2 đến A.7. Phần còn lại là các khuyến nghị liên quan đến các yêu cầu áp dụng tất cả các cơ chế trong tiêu chuẩn này (Điều khoản 5).

Tất cả các cơ chế yêu cầu chọn một mã khối từ tập hợp các mã khối đã được quy định trong TCVN 11367-3:2016 (ISO/IEC 18033-3:2010). Độ dài khối  $n$  của mã khối tối thiểu phải bằng 64, khuyến nghị có thể sử dụng mã khối với  $n = 128$ . Việc sử dụng mã khối với  $n = 128$  là yêu cầu bắt buộc đối với các cơ chế 2, 3 và 6.

Tất cả các cơ chế cũng yêu cầu bên gửi và bên nhận dữ liệu đã được bảo vệ phải chia sẻ một khóa bí mật  $K$ . Khóa bí mật này chỉ được hai bên biết và có thể có thêm bên thứ ba có sự tin tưởng của cả bên gửi và bên nhận. Có nhiều phương thức để thiết lập khóa bí mật này; tuy nhiên, khuyến nghị nên sử dụng cơ chế thiết lập khóa được trình bày chi tiết trong tiêu chuẩn TCVN 7817-2:2010 (ISO/IEC 11770-2:2008) hoặc tiêu chuẩn TCVN 7817-3:2007 (ISO/IEC 11770-3:1999).

Cả sáu cơ chế yêu cầu chọn một độ dài thẻ (tag). Lựa chọn tham số này ảnh hưởng đến mức độ đảm bảo được cung cấp cho bên nhận liên quan đến tính toàn vẹn và nguồn gốc của một thông điệp được bảo vệ. Để biết thêm thông tin chi tiết, xem phụ lục C của TCVN 11495-1:2016 (ISO/IEC 9797-1:2011).

#### A.2 Lựa chọn cơ chế

Tất cả các cơ chế được trình bày trong tiêu chuẩn này được cho là cung cấp mức độ an toàn cao. Tuy nhiên, một số cơ chế phù hợp hơn các cơ chế khác cho ứng dụng cụ thể. Khi chọn một cơ chế để sử dụng, cần xem xét các tính chất được nêu trong Bảng A.2 và danh sách liệt kê dưới đây cần được chú ý xem xét.

**Bảng A.1 – Thuộc tính các cơ chế**

Cơ chế số	2	3	4	5	6
Số lượng xấp xỉ của các phép tính mã khối cần thiết để mã hóa một thông điệp $q$ -bit	$12[q/n]$	$2q/n$	$2q/n$	Phụ thuộc vào cơ chế mã hóa và MAC sử dụng	$q/n$
Yêu cầu bản quyền	Không	Không	Không	Phụ thuộc vào cơ chế mã hóa và MAC sử dụng	Không
Thiết kế đặc biệt để sử dụng với các thông điệp ngắn	Có	Không	Không	Không	Không
Độ dài thông điệp phải biết trước khi bắt đầu mã hóa	Không	Có	Không	Không	Không
Yêu cầu giá trị khởi tạo	Không	Có	Có	Có	Có
Đã được chuẩn hóa trước đó	Có	Có	Không	Không	Có

a) Cơ chế 3 và 4 là các phương pháp kết hợp mã hóa sử dụng chế độ CTR trong mã khối (TCVN 12213:2018 (ISO/IEC 10116:2017)) với một mã xác thực thông điệp.

b) Cơ chế 5 cung cấp một phương thức để kết hợp phương pháp đã tiêu chuẩn hóa để mã hóa và tính toán MAC. Nếu việc triển khai các chức năng như vậy đã có sẵn, thì cơ chế 5 có thể có một số lợi thế khi triển khai.

c) Cơ chế 6 phù hợp với việc triển khai phần cứng thông lượng cao, vì có thể được thực thi mà không cần khoảng thời gian rõ ràng trong kỹ thuật pipeline.

#### A.3 Cơ chế 2 (Key Wrap)

Cơ chế này yêu cầu phải sử dụng mã khôi có  $n = 128$ . Bắt buộc phải sử dụng một trong các mã khôi được trình bày chi tiết trong tiêu chuẩn TCVN 11367-3:2016 (ISO/IEC 18033-3:2010), mục 5.

#### A.4 Cơ chế 3 (CCM)

Cơ chế này yêu cầu phải sử dụng mã khôi có  $n = 128$ . Bắt buộc phải sử dụng một trong các mã khôi được trình bày chi tiết trong tiêu chuẩn TCVN 11367-3:2016 (ISO/IEC 18033-3:2010), mục 5.

Cơ chế này yêu cầu sự chọn tham số độ dài tag  $t$  (từ tập hợp {32, 48, 64, 80, 96, 112, 128}). Việc chọn độ dài tag  $t$  phụ thuộc vào môi trường sử dụng cơ chế này, tuy nhiên, trừ khi vì một số nguyên nhân bắt buộc nào đó mà phải chọn độ dài tag khác, còn lại khuyến nghị nên chọn độ dài tag  $t$  thỏa mãn  $t \geq 64$ .

Cơ chế này yêu cầu sự chọn độ dài trường thông điệp  $w$  (tính theo octet) (từ tập hợp {2, 3, 4, 5, 6, 7, 8}). Việc chọn độ dài octet của trường độ dài thông điệp  $w$  cũng phụ thuộc vào môi trường sử dụng cơ chế này. Việc chọn  $w$  không ảnh hưởng đến mức an toàn mà cơ chế cung cấp. Giá trị  $w$  càng lớn càng cho phép độ dài thông điệp dài hơn, mặc dù chúng cũng làm giảm đi độ dài phần còn lại của biến khởi tạo. Tuy nhiên, ngay cả khi  $w$  được chọn bằng giá trị lớn nhất có thể, tức là  $w = 8$ , 56-bit của biến khởi tạo  $S$  có thể được chọn để đảm bảo rằng một biến khởi tạo  $S$  khác được sử dụng cho mỗi thông điệp, phải đủ cho phần lớn, nếu không đủ cho tất cả các ứng dụng thiết thực. Với phần lớn các ứng dụng có giá trị  $w = 4$ , tức là cho một độ dài thông điệp lớn nhất  $2^{32} \approx 4 \times 10^9$  octet, như vậy là đủ.

#### A.5 Cơ chế 4 (EAX)

Cơ chế này yêu cầu sự chọn tham số độ dài thẻ (tag)  $t$  ( $t \leq n$ ). Việc chọn độ dài tag  $t$  phụ thuộc vào môi trường sử dụng cơ chế này, tuy nhiên, trừ khi vì một số nguyên nhân bắt buộc nào đó mà phải chọn độ dài tag khác, còn lại khuyến nghị nên chọn độ dài tag  $t$  thỏa mãn  $t \geq 64$ .

#### A.6 Cơ chế 5 (Encrypt-then-MAC)

Cơ chế này yêu cầu chọn phương thức vận hành và cơ chế tính toán MAC. Mức an toàn được cung cấp qua kết quả lược đồ mã hóa có sử dụng xác thực phụ thuộc vào mức độ an toàn của hai yêu cầu trên.

Đối với chế độ hoạt động mã hóa, cần tuân theo cơ chế an toàn trong TCVN 12213:2018 (ISO/IEC 10116:2017). Đặc biệt, nếu chế độ CBC được sử dụng để mã hóa nhiều bản rõ với cùng một khóa, thì:

a) Một biến khởi tạo ngẫu nhiên nên được sử dụng để mã hóa mỗi bản rõ;

b) Khối bản rõ đầu tiên phải được đặt một cách đáng tin cậy thành một giá trị duy nhất cho bản rõ (ví dụ: bộ đếm).

Việc chọn cơ chế tính toán MAC trong trường hợp sử dụng kỹ thuật mã hóa được xác thực và cần tuân thủ cẩn thận lời khuyên được cung cấp trong TCVN 11495 (ISO/IEC 9797) (Tất cả các phần). Đặc biệt, nếu MAC dựa trên mã khôi từ TCVN 11495-1:2016 (ISO/IEC 9797-1:2011) được chọn, thì:

a) Thuật toán MAC 1 chỉ nên được sử dụng nếu độ dài thông điệp là cố định;

b) Chỉ nên sử dụng phương pháp đệm 1 nếu độ dài thông báo được cố định.

#### A.7 Cơ chế 6 (GCM)

Cơ chế này yêu cầu phải sử dụng mã khôi có  $n = 128$ . Bắt buộc phải sử dụng một trong các mã khôi có đặc tính này được trình bày chi tiết trong tiêu chuẩn TCVN 11367-3:2016 (ISO/IEC 18033-3:2010), mục 5.

Độ dài biến khởi tạo  $S$  có thể thay đổi, phải được chọn sao cho  $1 \leq \text{len}(S) \leq 2^{64}$ . Yêu cầu biến khởi tạo không bao giờ được tái sử dụng lại trong toàn bộ vòng đời của khóa là rất quan trọng đối với an toàn của cơ chế này.

Độ dài tag  $t$  phải được chọn ra sao cho  $t$  là một bội của 8 thỏa mãn  $96 \leq t \leq 128$  ( $t = 32$ ) và  $t = 64$  cũng được cho phép với các ứng dụng đặc biệt, mặc dù các tùy chọn này chỉ nên được sử dụng cực kỳ thận trọng - hướng dẫn chi tiết về sử dụng các độ dài tag đó được đưa ra trong phần Phụ lục C của [6]).

Chuỗi dữ liệu  $D$  phải thỏa mãn điều kiện sau thì mới áp dụng cơ chế mã hóa có xác thực:

$$\text{len}(D) \leq 2^{39} - 256$$

và chuỗi dữ liệu đã được xác thực bổ sung phải thỏa mãn  $\text{len}(A) \leq 2^{64}$ . Tổng số các khối dữ liệu và các khối dữ liệu đã được xác thực bổ sung khi áp dụng GCM với một khóa  $K$  được ấn định phải ở mức tối đa  $2^{64}$ . Ngoài ra, tổng số viện dẫn thủ tục mã hóa đối với bất kỳ khóa nào được đưa ra phải mức tối đa  $2^{32}$ , trừ khi  $\text{len}(S) = 96$  đối với mọi trường hợp sử dụng khóa đó.

**Phụ lục B**  
 (tham khảo)  
**Ví dụ**

**B.1 Giới thiệu**

Phụ lục bao hàm các ví dụ về vận hành các cơ chế được trình bày chi tiết trong tiêu chuẩn này.

**B.2 Cơ chế 2 (Key Wrap)**

Ví dụ về vận hành cơ chế này với mã khối AES được đưa ra trong tiêu chuẩn IETF RFC 3394 [9].

**B.3. Cơ chế 3 (CCM)**

Sáu ví dụ về bộ ba thông điệp ( $D_i$ ), bản mã ( $C_i$ ) và tag ( $T_i$ ) được tạo ra bằng cách sử dụng mã khối AES, trong môi trường hợp sử dụng  $t = 128$  và  $w = 2$  (và vì vậy  $S$  phải chứa 104-bit). Tất cả các ví dụ đều được biểu diễn theo hệ Hexa. Khóa  $K$  và biến khởi tạo  $S$  được sử dụng trong 06 ví dụ:

$K$ : 000102030405060708090A0B0C0D0E0F

$S$ : 000102030405060708090A0B0C

$D_1$ : Chuỗi rỗng (tức là  $L = 0$ )

$C_1$ : Chuỗi rỗng

$T_1$ : 54C92FE45510D6B3B0D46EAC2FEE8E63

$D_2$ : 0001020304050607

$C_2$ : 1635B68B570CFC85

$T_2$ : 2734A0447531C02916CF8B9A494C3AD1

$D_3$ : 000102030405060708090A0B0C0D0E0F

$C_3$ : 1635B68B570CFC85529E39AC913910D7

$T_3$ : C7C5C394B685B08B3F00DCD81256F0D0

$D_4$ : 000102030405060708090A0B0C0D0E0F

1011121314151617

$C_4$ : 1635B68B570CFC85529E39AC913910D7

F3111631623867F1

$T_4$ : BB85D5BEEA595F573A9B4733D3E04887

$D_5$ : 000102030405060708090A0B0C0D0E0F  
101112131415161718191A1B1C1D1E1F

$C_5$ : 1635B68B570CFC85529E39AC913910D7  
F3111631623867F134E6E441904FD504

$T_5$ : C80A98AAFDFF79C23FB4D775A71C29D0

$D_6$ : 000102030405060708090A0B0C0D0E0F  
101112131415161718191A1B1C1D1E1F  
2021222324252627

$C_6$ : 1635B68B570CFC85529E39AC913910D7  
F3111631623867F134E6E441904FD504  
F5746D6BF189815F

$T_6$ : 1A6F75C612B703E25E47260BABCCB06E

CHÚ THÍCH: Các ví dụ khác về vận hành cơ chế này với mã khôi AES được đưa ra trong IETF RFC 3610 [10].

#### B.4 Cơ chế 4 (EAX)

Sáu ví dụ về bộ ba: thông điệp ( $D_i$ ), bản mã ( $C_i$ ) và tag ( $T_i$ ) được tạo ra bằng cách sử dụng mã khôi AES, trong môi trường hợp sử dụng  $t = 128$ . Tất cả các ví dụ dưới đây đều được biểu diễn theo hệ hexa. Khóa  $K$  và biến khôi tạo  $S$  được sử dụng trong 06 ví dụ:

$K$ : 000102030405060708090A0B0C0D0E0F

$S$ : 000102030405060708090A0B0C0D0E0F

$D_1$ : Chuỗi rỗng (tức là  $L = 0$ )

$C_1$ : Chuỗi rỗng

$T_1$ : 1CE10D3EFFD4CADBE2E44B58D60AB9EC

$D_2$ : 0001020304050607

$C_2$ : 29D878D1A3BE857B

$T_2$ : 9E1F336E2D9058EE57BF181EDF49395B

$D_3$ : 000102030405060708090A0B0C0D0E0F

$C_3$ : 29D878D1A3BE857B6FB8C8EA5950A778

$T_3$ : BD55E38C169E77135C2AE42309004C04

$D_4$ : 000102030405060708090A0B0C0D0E0F  
1011121314151617

$C_4$ : 29D878D1A3BE857B6FB8C8EA5950A778  
331FBF2CCF33986F

$T_4$ : 7E72C073D72CB70D1129C56FA0794573

$D_5$ : 000102030405060708090A0B0C0D0E0F  
101112131415161718191A1B1C1D1E1F

$C_5$ : 29D878D1A3BE857B6FB8C8EA5950A778  
331FBF2CCF33986F35E8CF121DCB30BC

$T_5$ : EF07F23F26E1DC3BEEFF83B18A9E2687

$D_6$ : 000102030405060708090A0B0C0D0E0F  
101112131415161718191A1B1C1D1E1F  
5C87F59B057A40E9

$C_6$ : 29D878D1A3BE857B6FB8C8EA5950A778  
331FBF2CCF33986F35E8CF121DCB30BC  
5C87F59B057A40E9

$T_6$ : A0FA15E39A14811AE5AC0E7353C2BAB6

CHÚ THÍCH: Các ví dụ khác về vận hành cơ chế này với mã khôi AES được đưa ra trong [2]

#### B.5 Cơ chế 5 (Encrypt-then-MAC)

Tham khảo trong tiêu chuẩn TCVN 11495 (ISO/IEC 9797) và TCVN 12213:2018 (ISO/IEC 10116:2017).

#### B.6 Cơ chế 6 (GCM)

Hai ví dụ tiếp theo về bộ ba: thông điệp ( $D_t$ ), bản mã ( $C_t$ ) và tag ( $DT_t$ ) được tạo ra bằng cách sử dụng mã khôi AES, trong mỗi trường hợp sử dụng  $t = 128$  và dữ liệu xác thực bổ sung bao hàm chuỗi rỗng. Tất cả các ví dụ dưới đây đều được biểu diễn theo Hệ hexa. Khóa  $K$  và biến khôi tạo  $S$  được sử dụng trong 02 ví dụ:

*K:* 00000000000000000000000000000000

*S:* 00000000000000000000000000

$D_1$ : Chuỗi rỗng (tức là  $L = 0$ )

### *C<sub>1</sub>: Chuỗi rỗng*

**T<sub>1</sub>:** 58E2FCCEFA7E3061367F1D57A4E7455A

**C<sub>2</sub>:** 0388DACE60B6A392F328C2B971B2FE78

**T<sub>2</sub>:** AB6E47D42CEC13BDF53A67B21257BDDF

**CHÚ THÍCH:** Các ví dụ khác về vận hành cơ chế này với mã khởi AES được đưa ra trong phần tham khảo [6].

**Phụ lục C**  
 (quy định)  
**Mô-đun ASN.1**

### C.1 Định nghĩa định dạng

```

AuthenticatedEncryption {
    iso(1) standard(0) authenticated-encryption(19772) asn1-module(0)
        authenticated-encryption-mechanisms(0) }
DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- IMPORTS None; --

OID ::= OBJECT IDENTIFIER

AuthenticatedEncryptionMechanism ALGORITHM ::= {
    ae-mechanism2
    ae-mechanism3
    ae-mechanism4
    ae-mechanism5
    ae-mechanism5-AAD
    ae-mechanism6
}
-- Synonyms --
is19772 OID ::= { iso(1) standard(0) authenticated-encryption(19772) }
mechanism OID ::= { is19772 mechanisms(1) }
    ae-mechanism2 OID ::= { mechanism 2 }
    ae-mechanism3 OID ::= { mechanism 3 }
    ae-mechanism4 OID ::= { mechanism 4 }
    ae-mechanism5 OID ::= { mechanism 5 }
    ae-mechanism5-AAD OID ::= { mechanism 7 }
    ae-mechanism6 OID ::= { mechanism 6 }
END -- AuthenticatedEncryption --

```

### C.2 Sử dụng nhận diện đối tượng tiếp sau

Mỗi một cơ chế xác thực được trình bày trong tiêu chuẩn này sử dụng thuật toán mã khôi, trong trường hợp sử dụng cơ chế mã hóa có xác thực (cơ chế 5), sử dụng chế độ hoạt động tương tự và sử dụng thuật toán MAC. Do đó, mã nhận dạng đối tượng cơ chế mã hóa có xác thực có thể được tuân theo bởi một trong các mã nhận dạng thuật toán cơ chế mã hóa đối với mã khôi được chỉ định trong TCVN 11367-3:2016 (ISO/IEC 18033-3:2010) và bất kỳ tham số liên quan nào. Đối với trường hợp cơ chế 5, nhận dạng các chế độ hoạt động của mã khôi (TCVN 12213:2018 (ISO/IEC 10116:2017)) và thuật toán MAC [TCVN 11495 (ISO/IEC 9797) (Tất cả các phần)] có thể được cung cấp.

#### Thư mục tài liệu tham khảo

- [1] TCVN 11816 (ISO/IEC 10118), Công nghệ thông tin - Các kỹ thuật an toàn - Hàm băm.
- [2] TCVN 7817-4:2010 (ISO/IEC 11770-4:2006), Công nghệ thông tin - Kỹ thuật an ninh quản lý khoá.
- [3] TCVN 11367-1:2016 (ISO/IEC 18033-1:2015), Công nghệ thông tin - Các kỹ thuật an toàn - thuật toán mật mã - Phần 1: Tổng quan.
- [4] BeLLARE M., NAMpREMpRE C.'Authenticated encryption: Relations among notions and analysis of the generic composition paradigm'. In: T. Okamoto (ed.), Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Lecture Notes in Computer Science 1976, Springer-Verlag (2000) pp. 531-545
- [5] BeLLARE M., RogAwAy P., WagneR D.'The EAX mode of operation, In: B. K. Roy, W.Meier (eds.): Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers. Lecture Notes in Computer Science 3017, Springer-Verlag (2004) pp.389-407
- [6] National Institute of Standards and Technology, NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. November 2007.
- [7] National Institute of Standards and Technology, AES Key Wrap Specification. NIST, November 2001.
- [8] National Institute of Standards and Technology, NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality. May 2004.
- [9] SchAAd J., HousLEy R.RFC 3394: Advvanced Encryption Standard (AES); Key Wrap Algorithm, IETF, September 2002.
- [10] WhitiNg D., HousLey R., FeRguson N. RFC 3610: Counter with CBC-MAC (CCM). IETF, September 2003.