



TIÊU CHUẨN QUỐC GIA

TCVN 12854-6:2026
ISO/IEC 29192-6:2019

Xuất bản lần 1

**CÔNG NGHỆ THÔNG TIN – MẬT MÃ HẠNG NHẹ - PHẦN 6:
MÃ XÁC THỰC THÔNG điệp (MAC)**

*Information technology – Lightweight cryptography — Part 6: Message authentication
codes (MACs)*

HÀ NỘI – 2026

Mục lục

Lời nói đầu	4
Lời giới thiệu	5
1 Phạm vi áp dụng	7
2 Tài liệu viện dẫn	7
3 Thuật ngữ và định nghĩa	7
4 Ký hiệu và thuật ngữ viết tắt	9
5 MAC hạng nhẹ dựa trên mã khối	10
5.1 Tổng quát	10
5.2 LightMAC	10
6 MAC hạng nhẹ dựa trên hàm băm	11
6.1 Tổng quát	11
6.2 Tsudik's keymode	11
7 MAC chuyên dụng hạng nhẹ	11
7.1 Tổng quát	11
7.2 Chaskey-12	11
Phụ lục A	15
Phụ lục B	17
Phụ lục C	22
Phụ lục D	24
Thư mục tài liệu tham khảo	25

Lời nói đầu

TCVN 12854-6:2026 hoàn toàn tương đương với ISO/IEC 29192-6:2019.

TCVN 12854-6:2026 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.

Lời giới thiệu

Trong môi trường công nghệ thông tin, thường có yêu cầu về việc kiểm tra xác nhận rằng dữ liệu điện tử không bị sửa đổi trái phép và về việc bảo đảm rằng một thông điệp được tạo ra bởi một thực thể nắm giữ khóa bí mật. Thuật toán MAC (mã xác thực thông điệp) là một cơ chế toàn vẹn dữ liệu được dùng phổ biến, có thể đáp ứng các yêu cầu này.

Có thể theo cách tiếp cận thứ nhất để hiện thực hóa một MAC hạng nhẹ bằng cách sử dụng thuật toán MAC được quy định kết hợp với một mã khối được chọn từ TCVN 12854-2:2020 (ISO/IEC 29192-2) hoặc TCVN 11367-3:2016 (ISO/IEC 18033-3), và kết hợp với một hàm băm được chọn từ TCVN 12854-5:2020 (ISO/IEC 29192-5). Cũng có thể theo cách tiếp cận thứ hai để hiện thực hóa một MAC hạng nhẹ bằng cách sử dụng một hàm chuyên dụng. Tiêu chuẩn này quy định các ví dụ cho cả hai cách tiếp cận.

Công nghệ thông tin — Mật mã hạng nhẹ — Phần 6: Mã xác thực thông điệp (MAC)

Information technology — Lightweight cryptography — Part 6: Message authentication codes (MACs)

1 Phạm vi áp dụng

Tiêu chuẩn này quy định các thuật toán MAC phù hợp với các ứng dụng đòi hỏi cơ chế mật mã hạng nhẹ. Các cơ chế này có thể được dùng làm cơ chế toàn vẹn dữ liệu để kiểm tra xác nhận rằng dữ liệu không bị sửa đổi trái phép. Chúng cũng có thể được dùng làm cơ chế xác thực thông điệp nhằm bảo đảm rằng thông điệp được tạo ra bởi một thực thể nắm giữ khóa bí mật.

Tiêu chuẩn này quy định các thuật toán MAC sau:

- a) LightMAC;
- b) Tsudik's keymode;
- c) Chaskey-12.

2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau đây là cần thiết để áp dụng tiêu chuẩn này. Đối với các tài liệu viện dẫn ghi năm công bố thì áp dụng 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 11367-3:2016 (ISO/IEC 18033-3): Công nghệ thông tin - Các kỹ thuật an toàn - Thuật toán mật mã - Phần 3: Mã khối

TCVN 12854-2:2020 (ISO/IEC 29192-2): Công nghệ thông tin - Các kỹ thuật an toàn - Mật mã hạng nhẹ - Phần 2: Mã khối

TCVN 12854-5:2020 (ISO/IEC 29192-5): Công nghệ thông tin — Các kỹ thuật an toàn mật mã hạng nhẹ - Phần 5: Các hàm băm.

3 Thuật ngữ và định nghĩa

Trong tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa trong tiêu chuẩn TCVN 11367-3:2016 (ISO/IEC 18033-3) và các thuật ngữ, định nghĩa sau đây

3.1

khóa mã khối (block cipher key)

TCVN 12854-6:2026

khóa điều khiển hoạt động của một mã khối

NGUỒN: TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), 3.2

3.2

mã hóa (encryption)

phép toán khả nghịch được thực hiện bởi một thuật toán mật mã nhằm biến đổi dữ liệu thành bản mã để che giấu nội dung thông tin của dữ liệu

NGUỒN: TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), 3.6

3.3

hàm băm (hash-function)

hàm ánh xạ các chuỗi bit có độ dài thay đổi (nhưng thường bị chặn trên) thành các chuỗi bit có độ dài cố định, thỏa mãn hai tính chất sau:

— với một đầu ra cho trước, việc tìm một đầu vào ánh xạ thành đầu ra đó là không khả thi về mặt tính toán;

— với một đầu vào cho trước, việc tìm một đầu vào thứ hai ánh xạ thành cùng một đầu ra với đầu vào đó là không khả thi về mặt tính toán

Chú thích 1: Tính khả thi về mặt tính toán phụ thuộc vào các yêu cầu an toàn và môi trường cụ thể.

NGUỒN: TCVN 11816-1:2017 (ISO/IEC 10118-1:2016), 3.4

3.4

khóa (key)

dãy các ký hiệu điều khiển hoạt động của một phép biến đổi mật mã

Chú thích 1: Ví dụ như phép mã hóa, giải mã, tính toán hàm kiểm tra mật mã, tạo chữ ký hoặc kiểm tra xác nhận chữ ký.

NGUỒN: TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), 3.7

3.5

mã xác thực thông điệp (Message Authentication Code)

MAC

chuỗi các bit là đầu ra của một thuật toán MAC

NGUỒN: TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), 3.9

3.6

thuật toán MAC (MAC algorithm)

thuật toán tính một hàm ánh xạ các chuỗi bit và một khóa bí mật thành các chuỗi bit có độ dài cố định, thỏa mãn hai tính chất sau:

— với mọi khóa và mọi chuỗi đầu vào, hàm có thể được tính một cách hiệu quả;

— với một khóa cố định bất kỳ, khi không biết trước khóa đó, việc tính giá trị hàm trên bất kỳ chuỗi đầu vào mới nào là không khả thi về mặt tính toán, ngay cả khi đã biết một tập các chuỗi đầu vào và các giá trị hàm tương ứng, trong đó chuỗi đầu vào thứ i có thể đã được chọn sau khi quan sát giá trị của $i - 1$ giá trị hàm trước đó (với số nguyên $i > 1$)

NGUỒN: TCVN 11495-1:2016 (ISO/IEC 9797-1:2011), 3.10

3.7

tử (word)

xâu 32 bit được dùng trong thuật toán MAC Chaskey-12

4 Ký hiệu và thuật ngữ viết tắt

$a \leftarrow b$	gán cho biến a giá trị của b
$E(K, P)$	phép mã hóa bản rõ P bằng mã khối E sử dụng khóa K
h	hàm băm
IV	giá trị khởi tạo t bit
$I2BS(x, g)$	hàm nhận đầu vào là một số nguyên không âm x và cho đầu ra là một xâu bit độ dài g tương ứng với biểu diễn nhị phân của số đó
K_i	khóa mã khối được dùng bởi mã khối cơ sở trong LightMAC ($i = 1, 2$)
m	xâu thông điệp đầu vào của thuật toán MAC
m'	xâu thông điệp sau khi đã áp dụng phép đệm
m'_i	khối n bit thứ i của xâu thông điệp đã đệm m'
$m'^{(n-s)}_i$	khối $(n-s)$ bit thứ i của xâu thông điệp đã đệm m' với $i < \ell$, trong đó $m'_1^{(n-s)} m'_2^{(n-s)} \dots m'_\ell = m'$
s	kích thước bộ đếm
t	độ dài của MAC tính bằng bit
v_i	các từ 32 bit dùng để lưu kết quả của các phép tính trung gian
$X _j$	số nguyên không dấu j bit thu được từ số nguyên không dấu u bit X bằng cách lấy j bit có trọng số thấp nhất của X ($1 \leq j \leq u$)
$X \ll 1$	phép dịch trái một bit, nghĩa là nếu X là một từ thì $X \ll 1$ ký hiệu từ thu được bằng cách dịch trái nội dung của X đi một vị trí
$X \lll n$	phép "dịch vòng trái" n vị trí bit, nghĩa là nếu X là một từ và n là một số nguyên không âm thì $X \lll n$ ký hiệu từ thu được bằng cách dịch trái nội dung của X đi n vị trí theo kiểu vòng
0^s	xâu gồm s bit '0'
$\oplus$	phép XOR
$\boxplus$	phép cộng theo modulo 2^{32}
$ X $	độ dài của xâu bit X tính bằng bit
$ $	phép nối các xâu bit
$+_w$	phép cộng theo modulo 2^w , trong đó w là số bit trong một từ; nghĩa là nếu A và B là các từ w bit thì $A +_w B$ là từ thu được bằng cách coi A và B là biểu diễn nhị phân của các số nguyên và tính tổng của chúng theo modulo 2^w , với kết quả bị ràng buộc nằm trong khoảng từ 0 đến $2^w - 1$

5 MAC hạng nhẹ dựa trên mã khối

5.1 Tổng quát

Điều này quy định một thuật toán MAC hạng nhẹ sử dụng một khóa bí mật và một mã khối n bit để tính MAC t bit.

Phụ lục A định nghĩa định danh đối tượng phải được dùng để định danh thuật toán quy định tại Điều 5. Phụ lục B cung cấp các ví dụ bằng số cho các thuật toán MAC theo hệ cơ số mười sáu. Phụ lục C nêu các đặc tính hạng nhẹ của các thuật toán MAC được mô tả trong tiêu chuẩn này.

5.2 LightMAC

5.2.1 Tổng quát

LightMAC là một thuật toán MAC phải được sử dụng cùng với một mã khối bất kỳ thuộc TCVN 12854-2:2020 (ISO/IEC 29192-2) hoặc TCVN 11367-3:2016 (ISO/IEC 18033-3). Người dùng muốn sử dụng LightMAC^[6] phải lựa chọn:

- một mã khối E n bit từ TCVN 12854-2:2020 (ISO/IEC 29192-2) hoặc TCVN 11367-3:2016 (ISO/IEC 18033-3);
- một độ dài t của MAC tính bằng bit;
- một kích thước bộ đếm s , tức là số bit được cấp để biểu diễn giá trị bộ đếm, trong đó $0 \leq s < n$.

Các tham số trên phải được giữ không đổi trong khi sử dụng LightMAC với một khóa cho trước. Không nên dùng các bộ tham số khác nhau với cùng một khóa.

CHÚ THÍCH 1: Nếu bất kỳ tham số nào nêu trên bị thay đổi trong khi đang sử dụng một khóa thì không thể bảo đảm được an toàn.

CHÚ THÍCH 2: Các ví dụ bằng số, bao gồm các trường hợp $s = 8$ hoặc 32 và $t = 64$, được liệt kê trong B.2.

LightMAC nhận đầu vào là hai khóa mã khối K_1 và K_2 được sinh độc lập với nhau, và một thông điệp M có độ dài tối đa $2^s(n-s)$ bit. LightMAC cho đầu ra có độ dài t bit. LightMAC gồm các bước sau: đệm, áp dụng mã khối và cắt ngắn.

5.2.2 Bước 1 (đệm)

Gọi m là thông điệp đầu vào của LightMAC và $d = |m| \bmod (n-s)$. Đệm vào bên phải m một bit '1' duy nhất, tiếp theo là $n-d-1$ bit '0'. Kết quả được ký hiệu là m' .

5.2.3 Bước 2 (áp dụng mã khối)

m' phải được tách thành các chuỗi $m'_1, m'_2, \dots, m'_\ell$, trong đó $m'_1^{(n-s)}, m'_2^{(n-s)}, \dots, m'_{\ell-1}^{(n-s)}$ là các chuỗi $n-s$ bit, m'_ℓ là một chuỗi n bit và $m'_1^{(n-s)} || m'_2^{(n-s)} || \dots || m'_\ell = m'$. Chuỗi S được tính theo thủ tục sau.

$V \leftarrow 0^n$

For $i = 1$ to $\ell-1$:

$V \leftarrow E(K_1, \text{I2BS}(i, s) || m'_i) \oplus V$

$V \leftarrow m'_\ell \oplus V$

$S \leftarrow E(K_2, V)$

Quy định của hàm I2BS được nêu tại Phụ lục D.

5.2.4 Bước 3 (cắt ngắn)

MAC t bit được rút ra bằng cách lấy t bit có trọng số thấp nhất của chuỗi S , nghĩa là:

$\text{MAC} \leftarrow S|_t$

6 MAC hạng nhẹ dựa trên hàm băm

6.1 Tổng quát

Điều này quy định một thuật toán MAC hạng nhẹ sử dụng một hàm băm hạng nhẹ để tính MAC.

Phụ lục A định nghĩa định danh đối tượng phải được dùng để định danh thuật toán quy định tại Điều 6. Phụ lục B cung cấp các ví dụ bằng số cho các thuật toán MAC theo hệ cơ số mười sáu. Phụ lục C nêu các đặc tính hạng nhẹ của các thuật toán MAC được mô tả trong tiêu chuẩn này.

6.2 Tsudik's keymode

6.2.1 Yêu cầu

Tsudik's keymode là một thuật toán MAC sử dụng một hàm băm. Để sử dụng Tsudik's keymode^[1], một hàm băm hạng nhẹ h phải được lựa chọn và thống nhất. Hàm băm này phải được chọn trong số các hàm băm hạng nhẹ quy định trong TCVN 12854-5:2020 (ISO/IEC 29192-5:2016). Thực thể tạo MAC phải được trang bị một khóa bí mật K ; khóa này cũng phải được cung cấp cho mọi bên cần kiểm tra xác nhận MAC.

CHÚ THÍCH 1: Tsudik's keymode được xếp loại hạng nhẹ vì số lần gọi hàm băm cơ sở thường nhỏ hơn so với các MAC dựa trên hàm băm thông dụng như HMAC quy định trong TCVN 11495-2:2016 (ISO/IEC 9797-2).

CHÚ THÍCH 2: Lý do hàm băm cơ sở phải được chọn trong số các hàm băm quy định trong TCVN 12854-5:2020 (ISO/IEC 29192-5:2016) được mô tả tại Phụ lục C.

CHÚ THÍCH 3: Khi lựa chọn hàm băm cơ sở dùng trong Tsudik's keymode, người dùng tự kiểm tra độ an toàn của hàm băm đó trước các tấn công mở rộng độ dài.

6.2.2 Tính MAC

Để tính MAC trên thông điệp m bằng Tsudik's keymode, thực hiện phép toán sau:

$$S \leftarrow h(K||m).$$

MAC t bit được rút ra bằng cách lấy t bit có trọng số thấp nhất của chuỗi S , nghĩa là:

$$\text{MAC} \leftarrow S|_t.$$

7 MAC chuyên dụng hạng nhẹ

7.1 Tổng quát

Điều này quy định một thuật toán MAC chuyên dụng hạng nhẹ.

Phụ lục A định nghĩa định danh đối tượng phải được dùng để định danh thuật toán quy định tại Điều 7. Phụ lục B cung cấp các ví dụ bằng số cho các thuật toán MAC theo hệ cơ số mười sáu. Phụ lục C nêu các đặc tính hạng nhẹ của các thuật toán MAC được mô tả trong tiêu chuẩn này.

7.2 Chaskey-12

7.2.1 Tổng quát

Chaskey-12^[8] là một thuật toán MAC hạng nhẹ xử lý thông điệp m có độ dài tùy ý bằng một khóa K độ dài 128 bit. Thuật toán cho đầu ra là MAC có độ dài 128 bit hoặc ngắn hơn.

CHÚ THÍCH 1: Trong đề xuất ban đầu của thuật toán Chaskey^[7], số vòng được đặt là 8 và thuật toán được gọi là Chaskey-8. Do lo ngại rằng 8 vòng không đủ để bảo đảm mức an toàn cần thiết, lược đồ quy định ở đây có 12 vòng, và do đó được gọi là Chaskey-12^[8].

Chaskey-12 coi một phần tử a của $GF(2^{128})$ theo hai cách tương đương: vừa là chuỗi 128 bit $a[127]a[126]...a[0]$ vừa là đa thức $a(x) = a[127]x^{127} + a[126]x^{126} + ... + a[0]$ với các hệ số nhị phân.

TCVN 12854-6:2026

Gọi $f(x)$ là đa thức bất khả quy $x^{128} + x^7 + x^2 + x + 1$. Để nhân hai phân tử a và b , chúng được biểu diễn thành hai đa thức $a(x)$ và $b(x)$, rồi tính $a(x)b(x) \bmod f(x)$. Ví dụ, cách nhân một phân tử với x được nêu trong Hàm 1. Lưu ý rằng đơn thức x tương ứng với chuỗi bit $0^{126}||10$. Lưu ý rằng chuỗi bit $0^{126}||11$ có thể được coi là số 3 trong hệ thập phân.

Hàm 1: TimesTwo

```
function TimesTwo(a)
```

```
    if a[127] = 0
```

```
        return (a << 1)  $\oplus$   $0^{128}$ 
```

```
    else
```

```
        return (a << 1)  $\oplus$   $0^{120}10000111$ 
```

Khi chuyển đổi giữa các chuỗi 128 bit, các số nguyên không dấu 128 bit và các mảng số nguyên không dấu 32 bit, Chaskey-12 luôn dùng thứ tự byte little endian. Trong mỗi byte, việc đánh số bit bắt đầu từ bit có trọng số thấp nhất.

Chaskey-12 gồm các bước sau: dẫn xuất khóa con, đệm, áp dụng hoán vị và cắt ngắn.

7.2.2 Bước 1 (dẫn xuất khóa con)

Chaskey-12 nhận một khóa K 128 bit; khóa này nên được chọn độc lập và ngẫu nhiên đều trên toàn bộ không gian khóa. Từ K , hai khóa con 128 bit K_1 và K_2 được dẫn xuất, mỗi khóa con bằng một phép dịch 128 bit và một phép XOR theo bit 128 bit. Việc sinh các khóa con được định nghĩa trong Hàm 2.

Hàm 2: Dẫn xuất khóa con

```
function SubKeys(K)
```

```
 $K_1 \leftarrow \text{TimesTwo}(K)$ 
```

```
 $K_2 \leftarrow \text{TimesTwo}(K_1)$ 
```

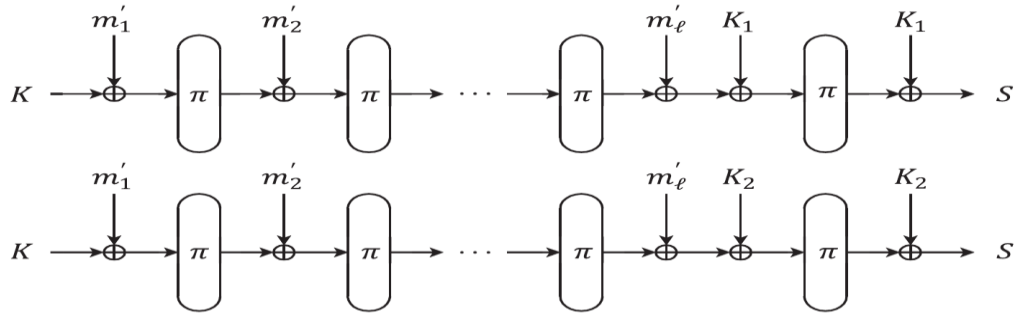
```
return ( $K_1$ ,  $K_2$ )
```

7.2.3 Bước 2 (đệm)

Gọi m là thông điệp đầu vào của Chaskey-12 và gọi d là số dư khi chia độ dài của m cho 128. Trường hợp $d \neq 0$ hoặc $|m| = 0$ thì m phải được đệm bằng cách nối vào nó một bit '1' duy nhất, rồi tiếp theo là $128-d-1$ bit '0'. Bản đã đệm của thông điệp m được ký hiệu là m' , và biến Boolean *padded* được đặt là true. Trường hợp $d = 0$ và $|m| \neq 0$ thì thông điệp không được đệm, khi đó $m' \leftarrow m$ và *padded* được đặt là false.

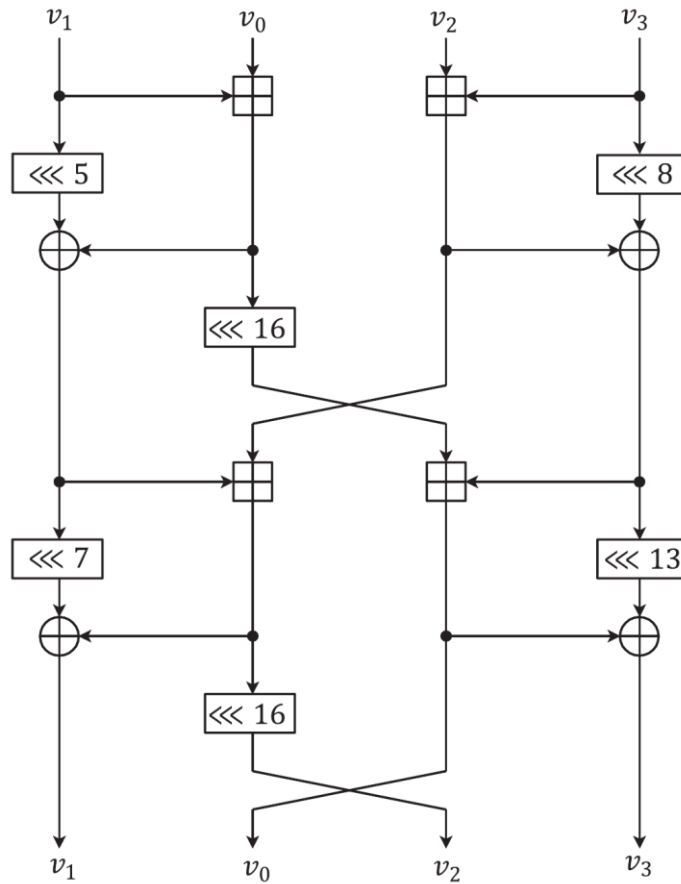
7.2.4 Bước 3 (áp dụng hoán vị)

Thông điệp đã đệm m' phải được tách thành các chuỗi $m'_1, m'_2, \dots, m'_t$, mỗi chuỗi 128 bit, sao cho $m'_1||m'_2||\dots||m'_t = m'$. Để tạo một MAC tối đa 128 bit, Chaskey-12 lập một hoán vị 128 bit π như quy định trong Hàm 3 (xem thêm Hình 1). Hình 1 minh họa việc áp dụng hoán vị (Bước 3) khi *padded* là false (hình trên) và khi *padded* là true (hình dưới). Hàm vòng của hoán vị π được quy định trong Hàm 4, các khóa con K_1 và K_2 được sinh theo Bước 1, và thông điệp m được đệm thành m' theo Bước 2. Đầu ra S có thể được cắt ngắn còn t bit ở Bước 4 (tùy chọn).



Hình 1 — Áp dụng hoán vị

Trong Chaskey-12, hoán vị π gồm 12 lần áp dụng một hàm vòng, như mô tả trong Hàm 4. Hoán vị này thao tác trên một xâu 128 bit, được xem như một mảng gồm bốn số nguyên không dấu 32 bit. Hàm vòng cũng được minh họa trong Hình 2. Hình 2 thể hiện một vòng của hoán vị Chaskey π , được định nghĩa là: $v_0||v_1||v_2||v_3 \leftarrow \pi(v_0||v_1||v_2||v_3)$.



Hình 2 — Một vòng của hoán vị Chaskey π

Hàm 3: Áp dụng hoán vị

function Chaskey(K, K_1, K_2, m')

$V \leftarrow K$

For $j=1$ to $\ell-1$

$V \leftarrow \pi(V \oplus m'_j)$

If *padded* = false

$L \leftarrow K_1$

Else

$L \leftarrow K_2$

EndIf

$S \leftarrow \pi(V \oplus m'_t \oplus L) \oplus L$

return S

Hàm 4: Hoán vị π

function $\pi(v)$

$(v_0, v_1, v_2, v_3) \leftarrow v$

For $i = 1$ to r

$v_0 \leftarrow v_0 +_{32} v_1$

$v_1 \leftarrow v_1 \lll 5$

$v_1 \leftarrow v_1 \oplus v_0$

$v_0 \leftarrow v_0 \lll 16$

$v_2 \leftarrow v_2 +_{32} v_3$

$v_3 \leftarrow v_3 \lll 8$

$v_3 \leftarrow v_3 \oplus v_2$

$v_0 \leftarrow v_0 +_{32} v_3$

$v_3 \leftarrow v_3 \lll 13$

$v_3 \leftarrow v_3 \oplus v_0$

$v_2 \leftarrow v_2 +_{32} v_1$

$v_1 \leftarrow v_1 \lll 7$

$v_1 \leftarrow v_1 \oplus v_2$

$v_2 \leftarrow v_2 \lll 16$

return $v_0 || v_1 || v_2 || v_3$

7.2.5 Bước 4 (cắt ngắn)

MAC t bit được rút ra bằng cách lấy t bit có trọng số thấp nhất của xâu S , nghĩa là:

$MAC \leftarrow S|_t$.

Phụ lục A

(Quy định)

Định danh đối tượng

Phụ lục này liệt kê các định danh đối tượng được gán cho các thuật toán MAC hạng nhẹ quy định trong tiêu chuẩn này.

--

-- Draft object identifiers of ISO/IEC 29192-6

--

LightweightMessageAuthenticationCodes {

iso(1) standard(0) lightweight-cryptography(29192) part6(6)

asn1-module(0) algorithm-object-identifiers(0)}

DEFINITIONS EXPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS

LightweightCryptographyIdentifier

FROM LightweightCryptography-2 {iso(1) standard(0)

lightweight-cryptography (29192) part2(2)

asn1-module(0) algorithm-object-identifiers(0)}

LightweightCryptographyIdentifier

FROM LightweightCryptography-5 {iso(1) standard(0)

lightweight-cryptography (29192) part5(5)

asn1-module(0) algorithm-object-identifiers(0));

OID ::= OBJECT IDENTIFIER -- Alias

-- Synonyms --

is29192-6 OID ::= {iso standard lightweight-cryptography(29192) part6(6)}

-- OID assignments --

light-bc-based-macs OID ::= {is29192-6 bc-based-macs(1)}

lightMAC OID ::= {light-bc-based-macs 1}

light-hf-based-macs OID ::= {is29192-6 hf-based-macs(2)}

tsudik OID ::= {light-hf-based-macs 1}

light-dedicated-macs OID ::= {is29192-6 dedicated-macs(3)}

chaskey-12 OID ::= {light-dedicated-macs 1}

-- note: assign any new OIDs above 3

LightweightMacAlgorithm ::= SEQUENCE {

algorithm ALGORITHM.&id({LightweightMac}),

TCVN 12854-6:2026

```
parameters ALGORITHM.&Type({LightweightMac}{@algorithm}) OPTIONAL
}
LightweightMAC ALGORITHM ::= {
  { OID lightMAC PARMS MacParameters-1 } |
  { OID tsudik PARMS MacParameters-2 } |
  { OID chaskey-12 PARMS NullParms } ,
  ... -- expect additional algorithms --
}
-- MAC parameter types definitions
-- =====
MacParameters-1 ::= SEQUENCE {
  bcAlgo BlockCipher OPTIONAL,
  t INTEGER (1..MAX)
}
MacParameters-2 ::= SEQUENCE {
  hfAlgo HashFunction OPTIONAL,
  t INTEGER (1..MAX)
}
BlockCipher ::=
  LightweightCryptography-2.LightweightCryptographyIdentifier
HashFunction ::=
  LightweightCryptography-5.LightweightCryptographyIdentifier
NullParms ::= NULL
-- Cryptographic algorithm identification --
ALGORITHM ::= CLASS {
  &id OBJECT IDENTIFIER UNIQUE,
  &Type OPTIONAL
}
  WITH SYNTAX { OID &id [PARMS &Type] }
END -- LightweightMessageAuthenticationCodes --
```

Phụ lục B

(Tham khảo)

Ví dụ bằng số

B.1 Tổng quát

Phụ lục này cung cấp các ví dụ bằng số cho từng cơ chế theo hệ cơ số mười sáu.

B.2 Ví dụ bằng số của LightMAC

Mã khối cơ sở: Present-128 như quy định trong TCVN 12854-2:2020 (ISO/IEC 29192-2:2012)

Trường hợp $s = 8$

$K_1 = 00112233\ 44556677\ 8899aabb\ ccddeeff$

$K_2 = 833d3433\ 009f389f\ 2398e64f\ 417acf39$

- $m = (0 \text{ octet})$: MAC = c3c863d3 e954788b
- $m = 00 (1 \text{ octet})$: MAC = 021dff18 0cad82f9
- $m = 0001 (2 \text{ octet})$: MAC = 3f89441a a4492cb9
- $m = 000102 (3 \text{ octet})$: MAC = 858fffa6 d1d238cc
- $m = 00010203 (4 \text{ octet})$: MAC = 7aa56f92 0da21d54
- $m = 000102...04 (5 \text{ octet})$: MAC = cead42e8 f4022b71
- $m = 000102...05 (6 \text{ octet})$: MAC = 939bdb08 9f993715
- $m = 000102...06 (7 \text{ octet})$: MAC = 0870a1d9 8336d9fe
- $m = 000102...07 (8 \text{ octet})$: MAC = cf7fdf18 0e2494a2
- $m = 000102...08 (9 \text{ octet})$: MAC = 957872ff d8474fcc
- $m = 000102...09 (10 \text{ octet})$: MAC = ef35a618 df40a363
- $m = 000102...0a (11 \text{ octet})$: MAC = ea690e1e c15c8817
- $m = 000102...0b (12 \text{ octet})$: MAC = fd0bc212 a16867a4
- $m = 000102...0c (13 \text{ octet})$: MAC = 09bb6867 7241b04d
- $m = 000102...0d (14 \text{ octet})$: MAC = 28eb9983 1d08b850
- $m = 000102...0e (15 \text{ octet})$: MAC = 44cef839 dcd7c1cb
- $m = 000102...0f (16 \text{ octet})$: MAC = 7f30f1f8 8d151ff5
- $m = 000102...10 (17 \text{ octet})$: MAC = 6af5747f ed140aa8
- $m = 000102...11 (18 \text{ octet})$: MAC = 11274bd5 394d388d
- $m = 000102...12 (19 \text{ octet})$: MAC = 99c3a296 a8c8e548
- $m = 000102...13 (20 \text{ octet})$: MAC = dc05c29e e8b9ab4e
- $m = 000102...14 (21 \text{ octet})$: MAC = 95e0f4fa 3774a8cf
- $m = 000102...15 (22 \text{ octet})$: MAC = 102ab033 7011a2f3
- $m = 000102...16 (23 \text{ octet})$: MAC = 88d68a1c 48bbc0b7

TCVN 12854-6:2026

- $m = 000102...17$ (24 octet): MAC = 5f028d39 6326567a
- $m = 000102...18$ (25 octet): MAC = 6f30b07a 025fa7ef
- $m = 000102...19$ (26 octet): MAC = e5ecbaba 994a36bc
- $m = 000102...1a$ (27 octet): MAC = 409b49af 04f3c184
- $m = 000102...1b$ (28 octet): MAC = dc88786e 74c298ad
- $m = 000102...1c$ (29 octet): MAC = 76e6526a f8c125ce
- $m = 000102...1d$ (30 octet): MAC = 942cfea1 68710b4c
- $m = 000102...1e$ (31 octet): MAC = 6f5703a4 048145cb
- $m = 000102...1f$ (32 octet): MAC = 4734b059 b872d41f
- $m = 000102...20$ (33 octet): MAC = 6f8f3297 4cb44284
- $m = 000102...21$ (34 octet): MAC = 12d537fe 00bb6046
- $m = 000102...22$ (35 octet): MAC = cd00fa52 a649e50a
- $m = 000102...23$ (36 octet): MAC = 5a0cdf1f 6e11f546
- $m = 000102...24$ (37 octet): MAC = 36c79cbe 956ed91a
- $m = 000102...25$ (38 octet): MAC = 9c854532 7b31c585
- $m = 000102...26$ (39 octet): MAC = a9a2eeed 33cee786
- $m = 000102...27$ (40 octet): MAC = b3343218 da6aa666
- $m = 000102...28$ (41 octet): MAC = 4f34e65d ba08b06a
- $m = 000102...29$ (42 octet): MAC = eebe3228 f7f21ed4
- $m = 000102...2a$ (43 octet): MAC = 326d89ce bad3f651
- $m = 000102...2b$ (44 octet): MAC = ff99dbfc 72b919d4
- $m = 000102...2c$ (45 octet): MAC = 0b7224ab c148de6c
- $m = 000102...2d$ (46 octet): MAC = 97928b8d f5c5f048
- $m = 000102...2e$ (47 octet): MAC = 9e6e039a a6209f07
- $m = 000102...2f$ (48 octet): MAC = a10deb90 41205b21
- $m = 000102...30$ (49 octet): MAC = 0e8208be 90e1e10b
- $m = 000102...31$ (50 octet): MAC = de0e2416 4f616f79
- $m = 000102...32$ (51 octet): MAC = 1761603e a4fbbd80
- $m = 000102...33$ (52 octet): MAC = eda5d05f b67b528d
- $m = 000102...34$ (53 octet): MAC = 48f4d47e 485b47d2
- $m = 000102...35$ (54 octet): MAC = df07346b bcff6eac
- $m = 000102...36$ (55 octet): MAC = 52d6b140 588dd5ff
- $m = 000102...37$ (56 octet): MAC = cc680cb4 b8b8a9b0
- $m = 000102...38$ (57 octet): MAC = 100160dc fb743c20
- $m = 000102...39$ (58 octet): MAC = d2ee268e c27309d6

- $m = 000102...3a$ (59 octet): MAC = dcdcba02 e6b8a4b3
- $m = 000102...3b$ (60 octet): MAC = 100e1ae6 ced7fede
- $m = 000102...3c$ (61 octet): MAC = d394ff09 dea2010c
- $m = 000102...3d$ (62 octet): MAC = 6e4b1fa6 30d3acc9
- $m = 000102...3e$ (63 octet): MAC = 431f7d96 7c0bc59e

Trường hợp $s = 32$

$K_1 = 01234567\ 89abcdef\ fedcba98\ 76543210$

$K_2 = 9cf35e82\ f26719c4\ f91cf900\ cc2cbcc1$

- $m = (0\ \text{octet})$: MAC = 8634d35b bcdbbbeb9
- $m = 00000001\ 00020003\ \dots\ 007e007f\ 00$ (257 octet): MAC = d60a9a88 90c26393
- $m = 00000001\ 00020003\ \dots\ 00fe00ff\ 0100$ (514 octet): MAC = f93128da b71ecf5c
- $m = 00000001\ 00020003\ \dots\ 017e017f\ 018001$ (771 octet): MAC = e9067e6c 774de5e2
- $m = 00000001\ 00020003\ \dots\ 01fe01ff\ 02000201$ (1 028 octet): MAC = 774d44bb 7ce920a4
- $m = 00000001\ 00020003\ \dots\ 02800281\ 02$ (1 285 octet): MAC = 38a9c371 00b908ff
- $m = 00000001\ 00020003\ \dots\ 03000301\ 0302$ (1 542 octet): MAC = c88d0fb7 32647299
- $m = 00000001\ 00020003\ \dots\ 03800381\ 038203$ (1 799 octet): MAC = d2a1a618 3daacd78
- $m = 00000001\ 00020003\ \dots\ 04000401\ 04020403$ (2 056 octet): MAC = 481dd001 e71a3585

B.3 Ví dụ bằng số của Tsudik's keymode

Hàm băm cơ sở: Lesamnta-LW như quy định trong TCVN 12854-5:2020 (ISO/IEC 29192-5)

Trường hợp $t = 128$

$K = 4c4c4c4c\ 4c4c4c4c\ 4c4c4c4c\ 4c4c4c4c$

Thông điệp = $4c4c4c4c\ 4c4c4c4c\ 4c4c4c4c\ 4c4c4c4c$

MAC = e01f6580 0eea165f 8c85b688 c36afca3

B.4 Ví dụ bằng số của Chaskey-12

Trường hợp $t = 64$

$K = 00112233\ 44556677\ 8899aabb\ ccddeeff$

$K_1 = 87224466\ 88aaccee\ 10335577\ 99bbddff$

$K_2 = 894588cc\ 105599dd\ 2166aaee\ 3277bbff$

$m = (0\ \text{octet})$: MAC = dd3e1849 d6824555

$m = 00$ (1 octet): MAC = ed1da89e c93179ca

$m = 0001$ (2 octet): MAC = 98fe20a3 43cd666f

$m = 000102$ (3 octet): MAC = f6f418ac dd7d9fa1

$m = 00010203$ (4 octet): MAC = 4cf04960 099949f3

$m = 000102...04$ (5 octet): MAC = 75c83252 653d3b57

$m = 000102...05$ (6 octet): MAC = 964b0461 fbe92273

TCVN 12854-6:2026

m = 000102...06 (7 octet): MAC = 141fa08b bf399636
m = 000102...07 (8 octet): MAC = 412d98ed 936d4ab2
m = 000102...08 (9 octet): MAC = fb0d98bc 70e305f9
m = 000102...09 (10 octet): MAC = 36f88e1f da86c8ab
m = 000102...0a (11 octet): MAC = 4d1a1815 868a5aa8
m = 000102...0b (12 octet): MAC = 7a7912c1 999eae81
m = 000102...0c (13 octet): MAC = 9ca11137 b4a34601
m = 000102...0d (14 octet): MAC = 7905142f 3be77e67
m = 000102...0e (15 octet): MAC = 6a3ee3d3 5c043397
m = 000102...0f (16 octet): MAC = d13970d7 be9b2350
m = 000102...10 (17 octet): MAC = 32acd914 bfda3bc8
m = 000102...11 (18 octet): MAC = 8a58d816 cb7a1483
m = 000102...12 (19 octet): MAC = 03f4d666 38efad8d
m = 000102...13 (20 octet): MAC = f9932237 ff05e831
m = 000102...14 (21 octet): MAC = f5fedb13 4862b471
m = 000102...15 (22 octet): MAC = 8bb55486 f38d57ea
m = 000102...16 (23 octet): MAC = 8a3acb94 b5ad591c
m = 000102...17 (24 octet): MAC = 7ce37087 23f7495f
m = 000102...18 (25 octet): MAC = f42f3d2f 405710c2
m = 000102...19 (26 octet): MAC = b3933a16 7e5636ac
m = 000102...1a (27 octet): MAC = 899a7945 423a5e1b
m = 000102...1b (28 octet): MAC = 65e12df5 a695fac8
m = 000102...1c (29 octet): MAC = b82449d8 c8a06ae9
m = 000102...1d (30 octet): MAC = a850dfba defa4229
m = 000102...1e (31 octet): MAC = fd42c39d 08ab71a0
m = 000102...1f (32 octet): MAC = b465c241 2610bf84
m = 000102...20 (33 octet): MAC = 89c4a9dd b53e6991
m = 000102...21 (34 octet): MAC = 5a9af91e b095d331
m = 000102...22 (35 octet): MAC = 8e54914c 151e46b0
m = 000102...23 (36 octet): MAC = fab8ab0b 5beaaec6
m = 000102...24 (37 octet): MAC = 60ad906a cd06c823
m = 000102...25 (38 octet): MAC = 6b1e6bc2 426dad17
m = 000102...26 (39 octet): MAC = 90328fd2 59889a8f
m = 000102...27 (40 octet): MAC = f0f7815e e6f3d516
m = 000102...28 (41 octet): MAC = 97e7e2ce bea826b8

m = 000102...29 (42 octet): MAC = b0fa1845 f72a76d6
m = 000102...2a (43 octet): MAC = a468bdfc df0aa9c7
m = 000102...2b (44 octet): MAC = da84e113 38387da7
m = 000102...2c (45 octet): MAC = b30d5ead 8e39f2bc
m = 000102...2d (46 octet): MAC = 178a43d2 a008503e
m = 000102...2e (47 octet): MAC = 6dfaa705 a8a06c70
m = 000102...2f (48 octet): MAC = aa047f07 c5ae8db4
m = 000102...30 (49 octet): MAC = 305bbb42 0c5d5ecc
m = 000102...31 (50 octet): MAC = 08328031 59750f49
m = 000102...32 (51 octet): MAC = 9080254f b79bab1a
m = 000102...33 (52 octet): MAC = 61c285ca 245774a4
m = 000102...34 (53 octet): MAC = 2aae035c fb61f97a
m = 000102...35 (54 octet): MAC = f5289075 c9ab39e5
m = 000102...36 (55 octet): MAC = e65c4237 32dae795
m = 000102...37 (56 octet): MAC = 4b22cf0d 9da8de3d
m = 000102...38 (57 octet): MAC = 2626ea2f a1f9abcf
m = 000102...39 (58 octet): MAC = d1e17e6e c4a88da6
m = 000102...3a (59 octet): MAC = 16574428 27ff640a
m = 000102...3b (60 octet): MAC = fd155a40 df15f630
m = 000102...3c (61 octet): MAC = ffeb596f 299f58b2
m = 000102...3d (62 octet): MAC = be4ee4ed 3975df87
m = 000102...3e (63 octet): MAC = fc7f9df7 991b87bc

Phụ lục C

(tham khảo)

Thông tin về an toàn và các bảng đặc tính

C.1 Tổng quát

Phụ lục này nêu các đặc tính hạng nhẹ của các MAC được mô tả trong tiêu chuẩn này.

C.2 Thông tin về LightMAC

C.2.1 Các đặc tính hạng nhẹ của LightMAC

Một đặc tính hạng nhẹ của LightMAC là độ dài thông điệp không ảnh hưởng đến giới hạn an toàn, cho phép xử lý lượng dữ liệu nhiều hơn cỡ một bậc độ lớn trên mỗi khóa. Hơn nữa, LightMAC hầu như không có chi phí phụ trội so với mã khối và có thể song song hóa. Nhờ đó, LightMAC không chỉ mang lại khả năng xác thực gọn nhẹ cho các nền tảng hạn chế về tài nguyên mà còn cho phép các cài đặt song song hiệu năng cao. Việc sử dụng các mã khối không thuộc loại hạng nhẹ (từ TCVN 11367-3:2016 (ISO/IEC 18033-3) thay vì từ TCVN 12854-2:2020 (ISO/IEC 29192-2)) không ảnh hưởng đến các đặc tính hạng nhẹ của LightMAC.

Theo TCVN 12854-1:2020 (ISO/IEC 29192-1), một đặc tính hạng nhẹ của LightMAC là nó đòi hỏi ít tài nguyên xét theo số chu kỳ/byte, như nêu trong Bảng C.1.

Bảng C.1 — Hiệu năng cơ sở của các mã khối PRESENT và AES trên Skylake (AVX2, AES-NI)^[6]

Mã khối cơ sở	Mã hóa [chu kỳ/byte]	Lập lịch khóa [chu kỳ]
PRESENT (dựa trên bảng tra)	57,83	353
PRESENT (bitslice 8 khối)	11,23	790
AES (AES-NI, tuần tự)	2,57	116
AES (AES-NI, đường ống)	0,63	116

Các đặc tính hạng nhẹ không phụ thuộc vào mã khối cơ sở được dùng trong LightMAC.

Với LightMAC, được phép dùng $t = 64$ (xem B.2).

C.2.2 Tham số s

Tham số s quyết định độ dài thông điệp tối đa có thể được xử lý bằng LightMAC và ấn định hiệu suất của thuật toán tính theo số lần gọi mã khối trên mỗi khối thông điệp. Cụ thể, LightMAC nhanh hơn khi s nhỏ hơn, nhưng đổi lại độ dài thông điệp tối đa mà nó có thể xử lý sẽ giảm.

C.3 Thông tin về các hàm băm cơ sở được dùng trong chế độ của Tsudik

Tsudik's keymode mô tả tại Điều 6 đòi hỏi một hàm băm kháng va chạm và kháng các tấn công mở rộng độ dài. Để bảo đảm an toàn trước tấn công mở rộng độ dài, hàm băm cơ sở cần có tính chất phù hợp.

So với HMAC — chế độ được dùng rộng rãi nhất để chuyển một hàm băm thành MAC — chế độ của Tsudik đơn giản hơn nhiều. Nhìn chung, mọi chế độ hoạt động đã được chứng minh là không thể phân biệt được với một bộ tiên tri ngẫu nhiên đều phù hợp cho chế độ của Tsudik. Trong đó bao gồm PHOTON và SPONGENT thuộc TCVN 12854 (ISO/IEC 29192). Tài liệu tham khảo [1] đưa ra chứng minh rằng Lesamnta-LW thuộc TCVN 12854 (ISO/IEC 29192) có thể được dùng trong chế độ của Tsudik. Do đó, các hàm băm của TCVN 12854-5:2020 (ISO/IEC 29192-5) có thể được dùng với chế độ này của Tsudik.

Tuy nhiên, các hàm băm của TCVN 11816-3:2017 (ISO/IEC 10118-3) không thể được dùng với chế độ này của Tsudik.

Theo TCVN 12854-1:2020 (ISO/IEC 29192-1), đặc tính hạng nhẹ của chế độ Tsudik là nó đòi hỏi ít tài nguyên xét theo thời gian thực thi. Cụ thể hơn, đối với thông điệp ngắn, số lần gọi hàm băm cơ sở có thể chỉ là một. Đặc tính này không phụ thuộc vào hàm băm cơ sở.

C.4 Thông tin về việc sử dụng Chaskey-12

An toàn của các thuật toán MAC chịu ảnh hưởng bởi số khối thông điệp được xử lý dưới cùng một khóa. Đối với Chaskey-12, cũng như trong hầu hết các thuật toán dựa trên mã khối, số khối được xử lý dưới cùng một khóa nên nhỏ hơn hẳn giới hạn an toàn, tức là 2^{64} khối đối với một mã khối 128 bit. Ví dụ, khóa có thể được làm mới trước khi 2^{48} khối thông điệp 128 bit được xử lý, tương ứng với bốn pebibyte^[2] dữ liệu^[7]. IEC 80000-13^[2] định nghĩa các tiền tố nhị phân chuẩn dùng để ký hiệu các lũy thừa của 1024, chẳng hạn 1024^5 (pebi-).

Theo TCVN 12854-1:2020 (ISO/IEC 29192-1), đặc tính hạng nhẹ của Chaskey-12 là Chaskey-12 chỉ chậm hơn Chaskey-8 từ 15 % đến 30 % tùy nền tảng đích^[8], và nó đòi hỏi ít tài nguyên xét theo dữ liệu, kích thước ROM và số chu kỳ/byte, như nêu trong Bảng C.2.

Bảng C.2 — Kết quả đo hiệu năng của Chaskey-8 trên Cortex-M0/M4^[7]

Vi điều khiển	Dữ liệu [byte]	Kích thước ROM [byte]	Tốc độ [chu kỳ/byte]
Tối ưu theo tốc độ			
Cortex-M0	16	1 308	21,3
Cortex-M4	16	908	10,6
	128	908	7,0
Tối ưu theo kích thước			
Cortex-M0	16	414	21,8
Cortex-M4	16	402	16,1
	128	402	11,2

Phụ lục D

(tham khảo)

Quy định của hàm I2BS**D.1 Tổng quát**

Phụ lục này nêu quy định của hàm I2BS (chuyển đổi số nguyên thành xâu bit).

D.2 I2BS

Để biểu diễn một số nguyên không âm x thành một xâu bit độ dài g (g phải thỏa mãn $2^g > x$), số nguyên đó phải được viết dưới dạng biểu diễn nhị phân duy nhất của nó [xem Công thức (D.1)]:

$$x = 2^{g-1}x_{g-1} + 2^{g-2}x_{g-2} + \dots + 2x_1 + x_0 \quad (\text{D.1})$$

trong đó $0 \leq x_i < 2$ (lưu ý rằng một hoặc nhiều chữ số đứng đầu sẽ bằng 0 nếu $x < 2^{g-1}$). Xâu bit I2BS(x, g) phải như Công thức (D.2):

$$x_{g-1}x_{g-2}\dots x_0 \quad (\text{D.2})$$

Thư mục tài liệu tham khảo

- [1] Hirose S., Ideguchi K., Kuwakado H., Owada T., Preneel B., Yoshida H., "An AES Based 256-bit Hash Function for Lightweight Applications: Lesamnta-LW", IEICE Transactions 95-A (1): 89-99 (2012)
 - [2] IEC 80000-13, *Quantities and units — Part 13: Information science and technology*
 - [3] ISO/IEC 9797-1, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*
 - [4] ISO/IEC 10118-1, *Information technology — Security techniques — Hash-functions — Part 1: General*
 - [5] ISO/IEC 10118-3, *IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions*
 - [6] Luykx A., Preneel B., Tischhauser E., Yasuda K., "A MAC Mode for Lightweight Block Ciphers", In Proceedings of Fast Software Encryption - FSE 2016, Lecture Notes in Computer Science volume 9783, pp. 43-59, Springer-Verlag, 2016
 - [7] Mouha N., Mennink B., Van Herrewege A., Watanabe D., Preneel B., Verbauwhede I., "Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers", In Proceedings of Selected Areas in Cryptography - SAC 2014, Lecture Notes in Computer Science volume 8781, pp. 306-323, Springer-Verlag, 2014
 - [8] Mouha N., "Chaskey: a MAC Algorithm for Microcontrollers – Status Update and Proposal of Chaskey-12 –", Cryptology ePrint Archive, Report 2015/1182, 2015
 - [9] Tsudik G., "Message Authentication with One-Way Hash Functions", ACM, Computer Communications Review, vol. 22, no. 5, pp. 29-38, 1992
-