Message Authentication Code

Alfredo De Santis

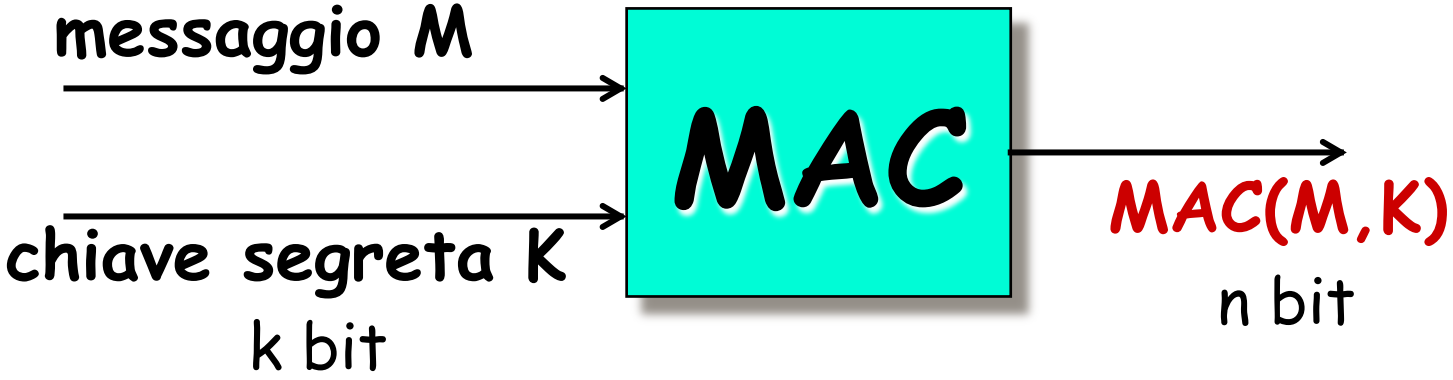
Dipartimento di Informatica
Università di Salerno

ads@unisa.it

http://www.dia.unisa.it/professori/ads

Maggio 2016

Message Authentication Code (MAC)



Applicazioni

- Autenticità del messaggio **M**
- Integrità del messaggio M

Utilizzo MAC

chiave privata K

chiave privata K

messaggio M

Y← MAC(M,K)

Alice

M, y

Bob

Y $\stackrel{?}{=}$ MAC(M,K)

MAC + confidenzialità

- MAC fornisce solo autenticazione ed integrità
- Se si vuole confidenzialità, si può cifrare prima/dopo con una diversa chiave condivisa K'

Alice

$E_{K'}(M)$, $MAC(E_{K'}(M),K)$

$E_{K'}(M, MAC(M,K))$

Bob

MAC + confidenzialità

- MAC fornisce solo autenticazione ed integrità
- Se si vuole confidenzialità, si può cifrare prima/dopo con una diversa chiave condivisa K'

Alice

$E_{K'}(M),\ MAC(E_{K'}(M),K)$

$E_{K'}(M,MAC(M,K))$

Bob

Analizzeremo in seguito la confidenzialità insieme ad autenticazione ed integrità

Sicurezza

- Cosa si intende per sicurezza di uno schema di un MAC?

- Dobbiamo definire
  - Tipo di attacco
  - Scopo dell'attacco

Tipo di attacco

- Known Message Attack
  - Oscar conosce una lista di messaggi ed i relativi MAC
- Chosen Message Attack
  - Oscar sceglie dei messaggi e chiede ad Alice (o Bob) di computarne i MAC

- Adaptive Chosen Message Attack
  - Come nel Chosen Message Attack, ma le scelte dipendono dalle risposte precedenti

Scopo dell'attacco

- Total break
  - Determinare la chiave K
- Selective forgery
  - Dato un messaggio M, determinare y tale che y=MAC(M,K)
- Existential forgery
  - Determinare una coppia (M,y) tale che y=MAC(M,K)

Ricerca esaustiva sulla chiave

- Date le coppie (M_i,y_i), con $y_i=MAC(M_i,K)$, determiniamo K
  - Supponiamo che sia k > n
    - Prova tutte le 2^k chiavi sulla coppia (M_1,y_1): circa 2^{k-n} match
    - Prova le 2^{k-n} chiavi precedenti sulla coppia (M_2,y_2): circa 2^{k-2n} match
    - …
  - In generale, se $k=\beta \cdot n$, necessari circa β round

- Esempio: k=80 bit, n=32 bit
  - Round 1: circa 2^{48} match
  - Round 2: circa 2^{16} match
  - Round 3: 1 match

Ricerca esaustiva sul valore del MAC

- Dato M determiniamo y=MAC(M,K), senza conoscere K
  - Scegli a caso y, prob. successo: $1/2^n$
  - Ma come controlliamo che sia il valore giusto, senza avere K?

Ricerca esaustiva

- Sforzo computazionale richiesto: $\min(2^k, 2^n)$
- Raccomandazioni: $\min(k,n) \geq 128$

MAC: Costruzioni

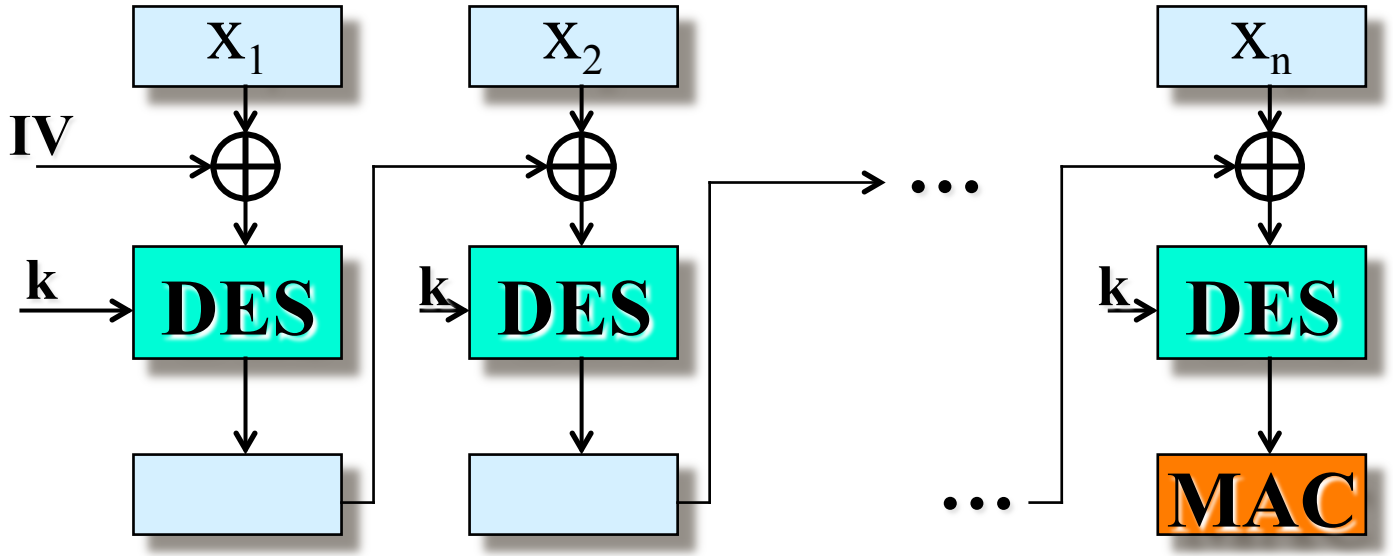
- MAC basati su cifrari a blocchi
  - Data Authentication Algorithm (DAA)
    - CBC-MAC
  - CMAC
- MAC basati su funzioni hash
  - HMAC

MAC: Costruzioni

- MAC basati su cifrari a blocchi
  - Data Authentication Algorithm (DAA)
    - CBC-MAC
  - CMAC
- MAC basati su funzioni hash
  - HMAC

CBC-MAC

- **C**ipher **B**lock **C**haining (con IV=0)
- E' stato molto usato (FIPS PUB 113 e ANSI X9.17)
- Testo $X = X_1 X_2 \ldots X_n$ diviso in blocchi di 64 bit

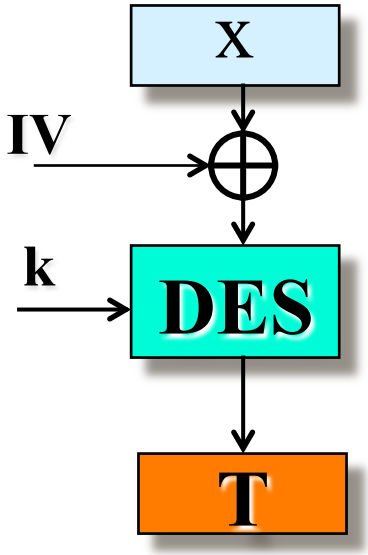


Valore troncato: da 16 a 64 bit più a sinistra

DAA: problemi di sicurezza

Vediamo un semplice esempio:

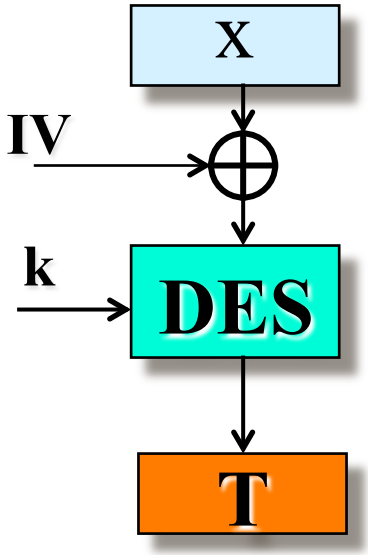
- Supponiamo che T sia il MAC di X

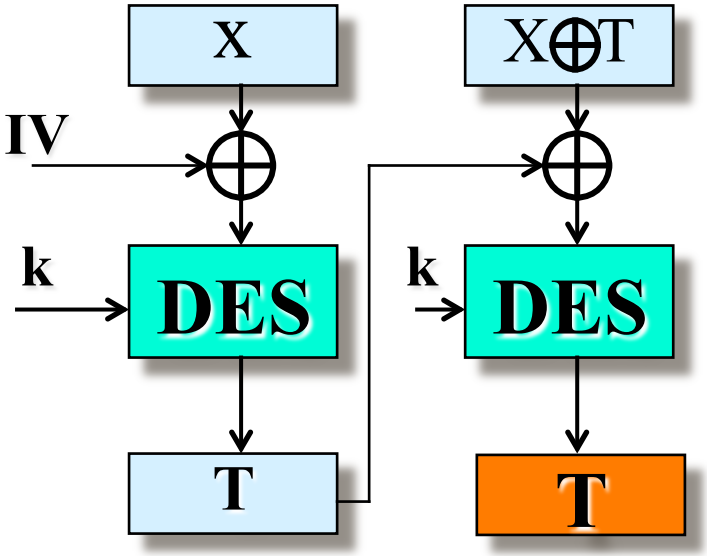


DAA: problemi di sicurezza

Vediamo un semplice esempio:

- Supponiamo che T sia il MAC di X
- Allora T è anche il MAC di X||(X ⊕ T)





Dal DAA al CMAC

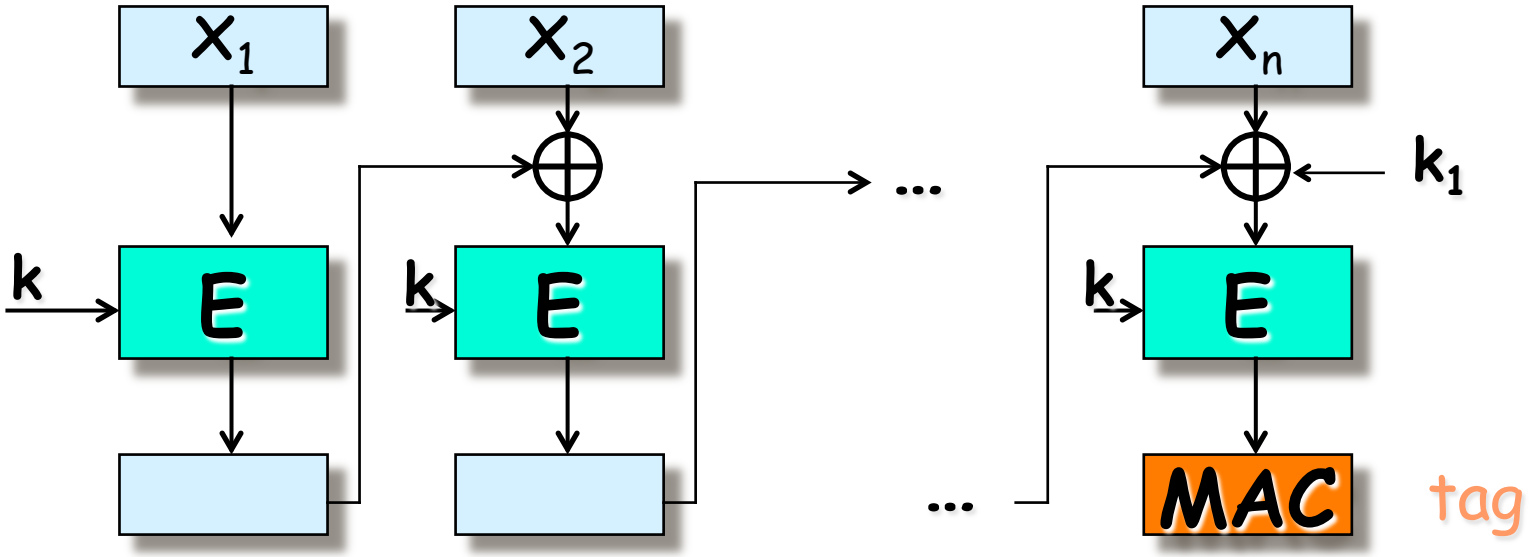
- Modifica del CBC-MAC
  - Proposta al NIST:
    - J. Black, P. Rogaway, A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC [da paper in Crypto 2000]
    - 3 chiavi
- Raffinamento
  - T. Iwata, K. Kurosawa, OMAC: One-key CBC MAC, 2003
  - Una sola chiave
  - Cipher-based MAC (CMAC)

CMAC

- Cipher-based MAC (CMAC)
- NIST SP800-38B, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", maggio 2005
- Utilizzabile con AES e Triplo DES
- Usa una sola chiave k, e 2 chiavi derivate k_0, k_1
- RFC 4493, The AES-CMAC Algorithm (giu 2006)
- RFC 4494, The AES-CMAC-96 Algorithm and its use with Ipsec (giu 2006)

CMAC

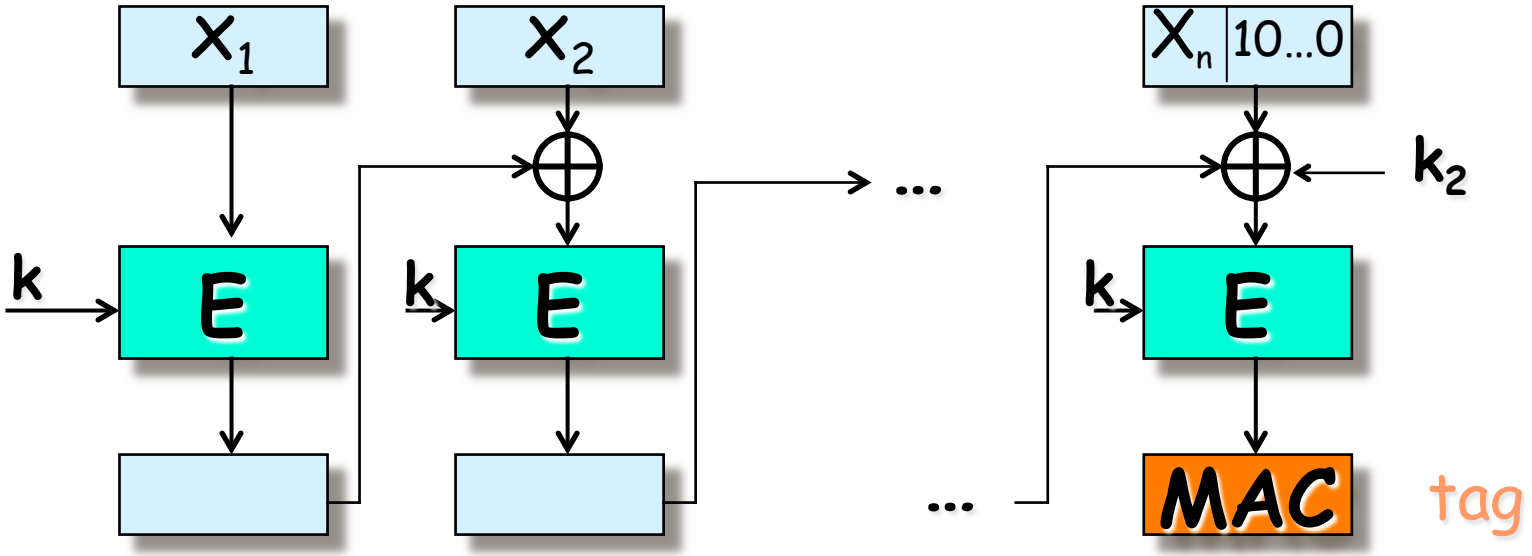
- Testo $X = X_1 X_2 \ldots X_n$ diviso in blocchi uguali (128 AES, 64 TDEA)
- Lunghezza messaggio multipla della lunghezza blocco b
- k_1 dipende da k e da b



Valore troncato: s bit più a sinistra

CMAC

- Testo $X = X_1 X_2 \ldots X_n$ diviso in blocchi uguali (128 AES, 64 TDEA)
- Lunghezza messaggio **non** multipla della lunghezza blocco b
- k_2 dipende da k e da b



Valore troncato: s bit più a sinistra

Chiavi k_1 e k_2

$$k_1 = E_k(0^b) \bullet x$$

$$k_2 = E_k(0^b) \bullet x^2 = (E_k(0^b) \bullet x) \bullet x$$

Moltiplicazione in $GF(2^b)$

Polinomi irriducibili

- $b=64 \rightarrow x^{64}+x^4+x^3+x+1$
- $b=128 \rightarrow x^{128}+x^7+x^2+x+1$

Polinomi in $GF(2^b)$

$0^{b-3}100$ e $0^{b-2}10$

MAC: Costruzioni

- MAC basati su cifrari a blocchi
  - Data Authentication Algorithm (DAA)
    - CBC-MAC
  - CMAC
- MAC basati su funzioni hash
  - HMAC

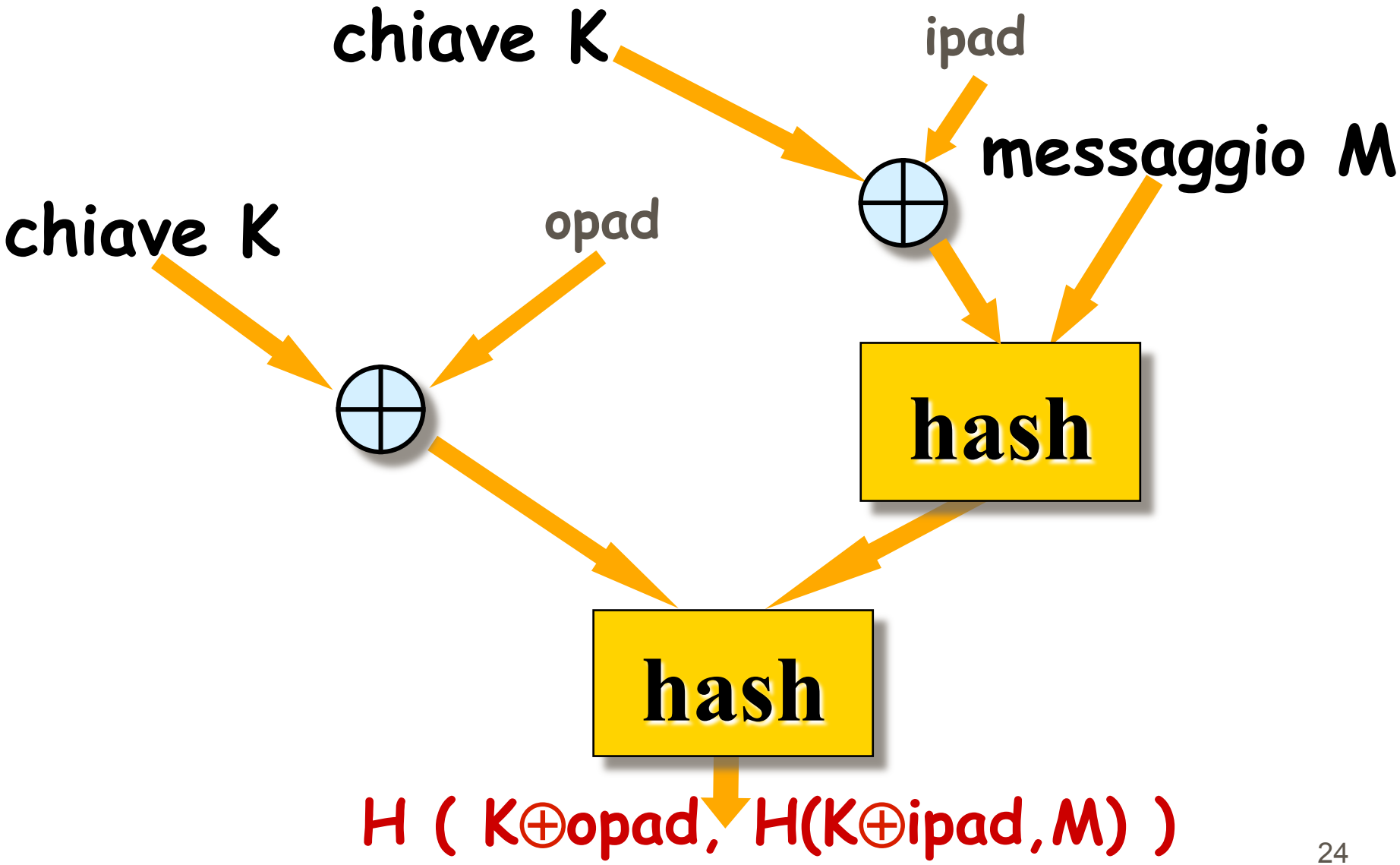
HMAC

- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*, Febbraio 1997
- ANSI X9.71 *Keyed Hash Message Authentication Code, 2000*
- FIPS 198, *The Keyed-Hash Message Authentication Code (HMAC)*, 6 marzo 2002
- FIPS 198-1, *The Keyed-Hash Message Authentication Code (HMAC)*, luglio 2008
- Funzioni Hash usate come black-box
  - Utilizzo delle funzioni hash senza modifiche
  - Facile cambio della funzione hash (più veloci e più sicure)
- Facile utilizzo e gestione di chiavi

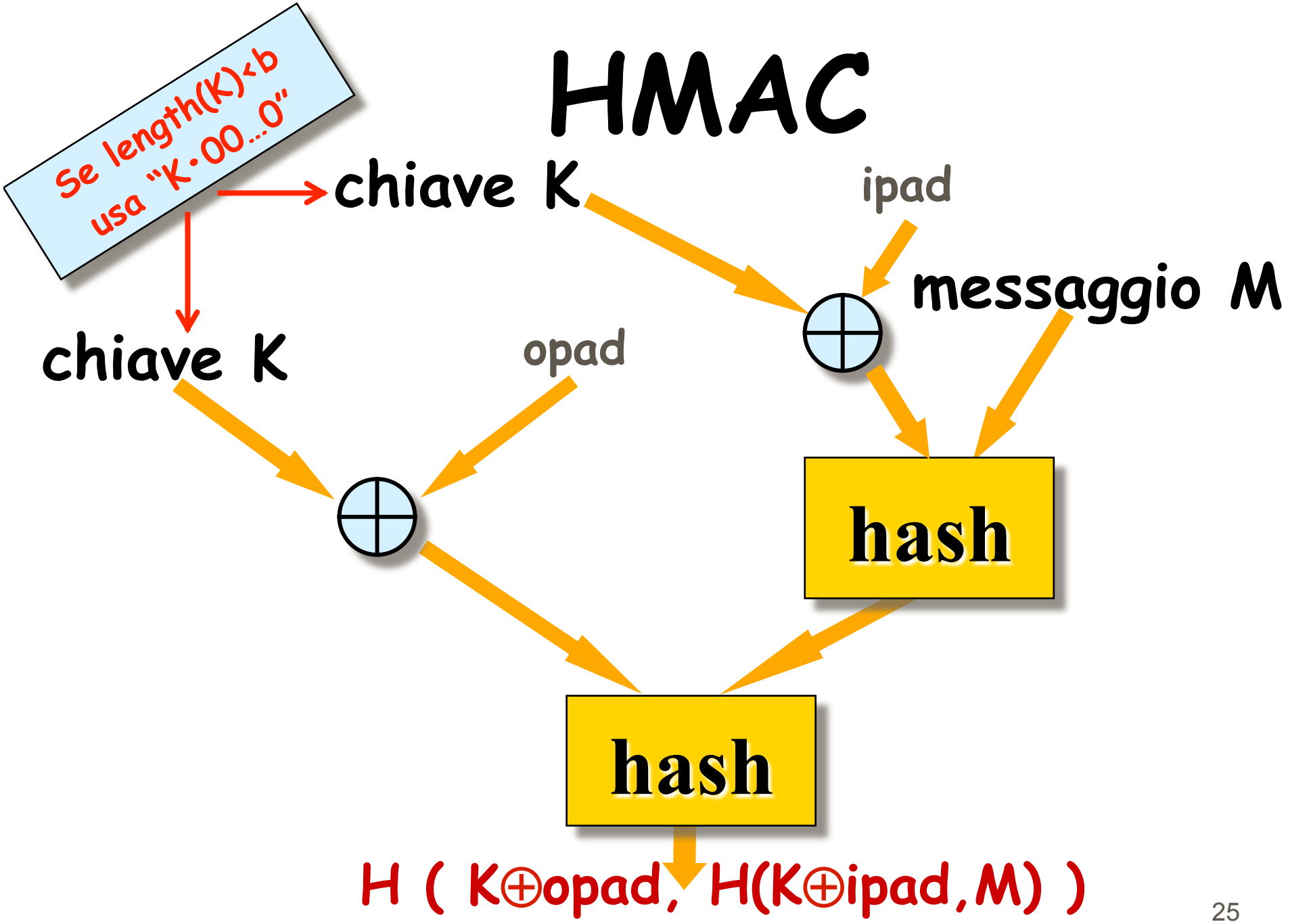
HMAC

- H: funzione hash iterata con inizialization vector IV
  - MD5, SHA-1, RIPEMD-160,…
- n: lunghezza del valore hash
  - n=128, 160,…
- M, suddiviso in L blocchi di b > n bit ciascuno
  - b=512,…
- ipad = byte 00110110 (36 in esadecimale) ripetuto b/8 volte
- opad = byte 01011100 (5C in esadecimale) ripetuto b/8 volte
- K: chiave segreta
  - Se length(K) < b, fai padding con 0…0
  - Se length(K) > b, calcola H(K) di n bit e fai padding con 0…0

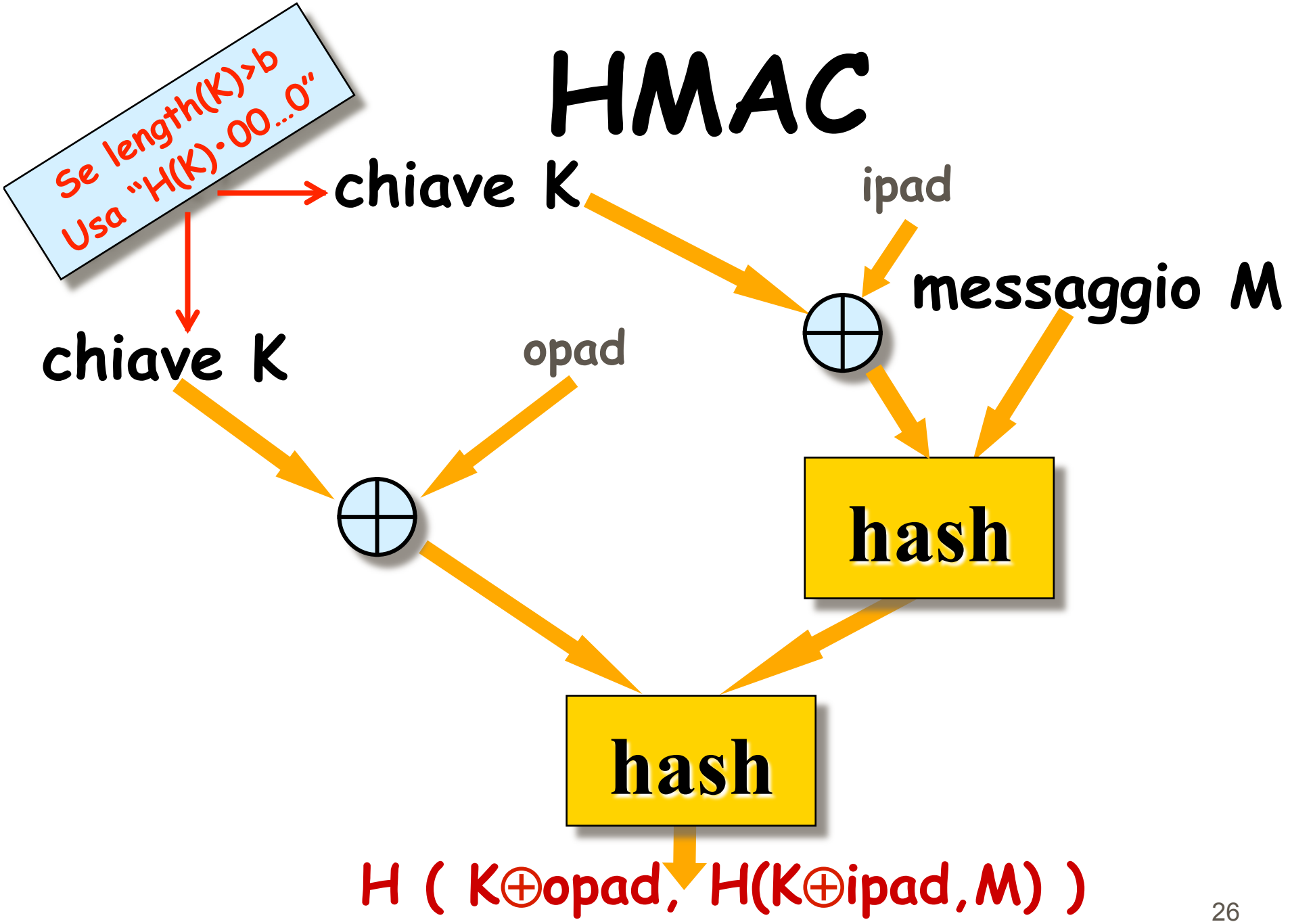
HMAC


chiave K
ipad
messaggio M
chiave K
opad
hash
hash
H (K⊕opad, H(K⊕ipad,M))

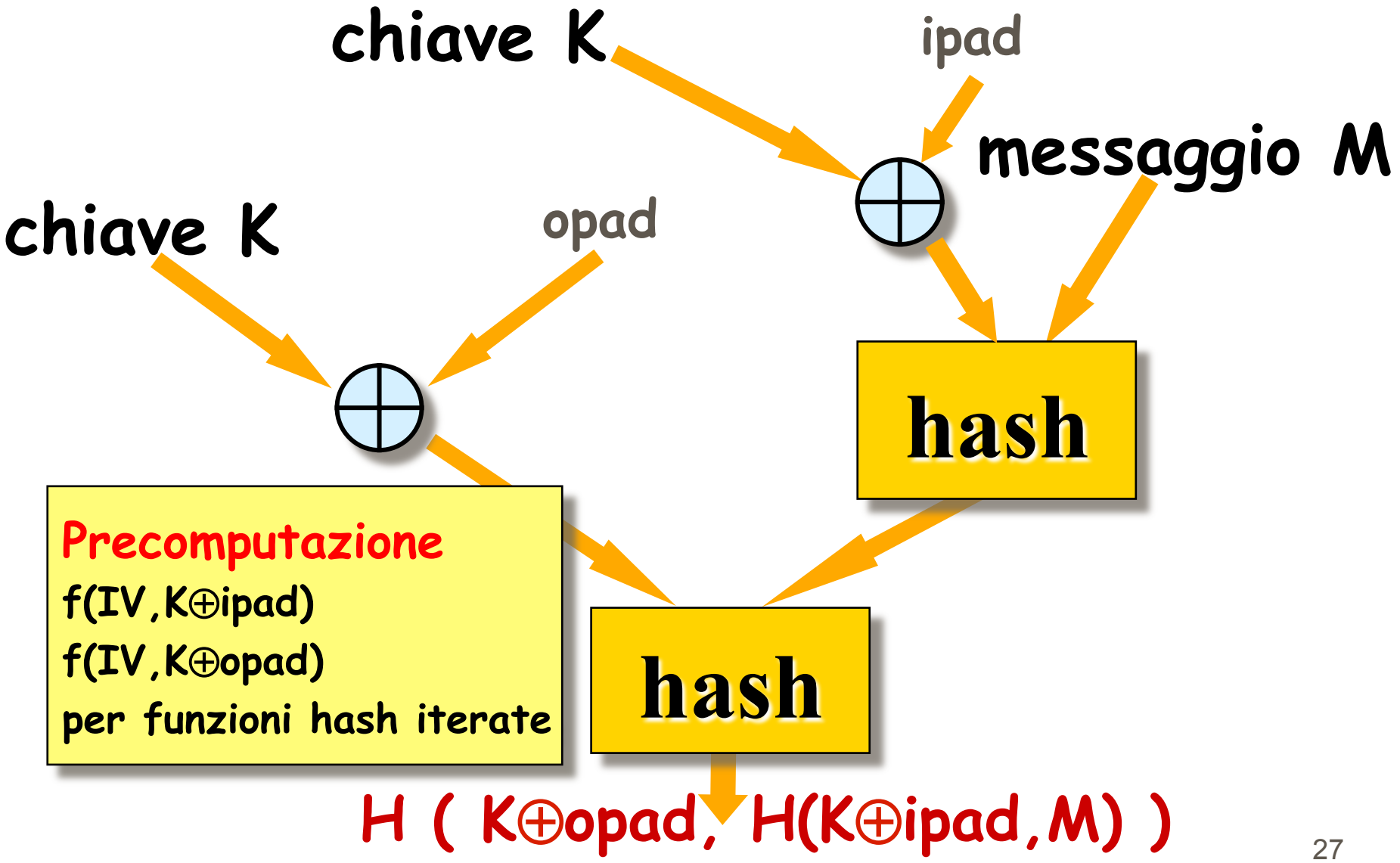
HMAC


Se length(K)<b
usa "K•00…0"
chiave K
ipad
messaggio M
chiave K
opad
hash
hash
H (K⊕opad, H(K⊕ipad,M))

HMAC


Se length(K)>b
Usa "H(K)·00...0"
chiave K
ipad
messaggio M
chiave K
opad
hash
hash
H (K⊕opad, H(K⊕ipad,M))

HMAC


chiave K
ipad
messaggio M
chiave K
opad
hash
Precomputazione
f(IV,K⊕ipad)
f(IV,K⊕opad)
per funzioni hash iterate
hash
H (K⊕opad, H(K⊕ipad,M))

Output troncato

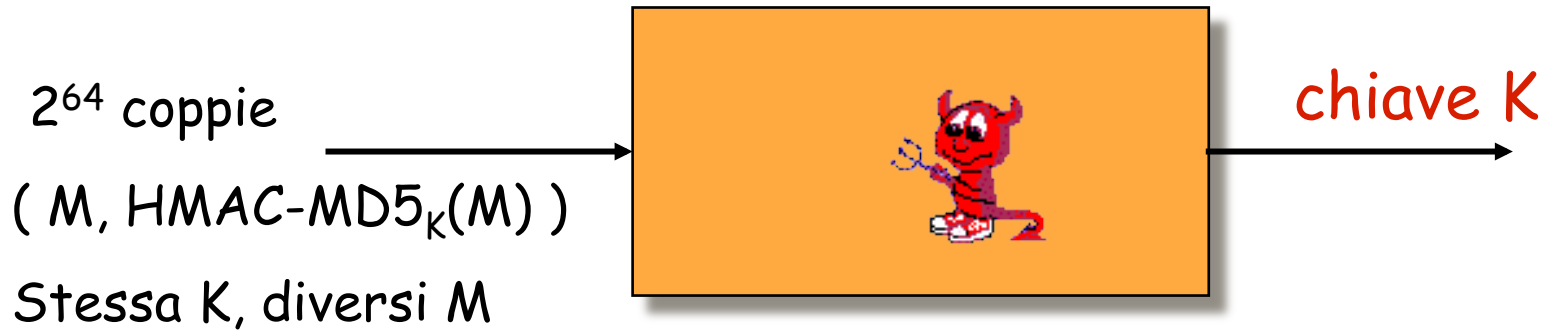
- Diverse volte si usano solo i primi t bit dell'hash
- Esempi:
  - HMAC-SHA1-80 (solo i primi 80 dei 160 bit)
  - HMAC-MD5 (tutti i 128 bit)
- Raccomandazioni:
  - $t \geq n/2$ per una funzione hash di n bit
  - Comunque, $t \geq 80$ (RFC 2104), $t \geq 32$ (FIPS 198)

Sicurezza HMAC

- Sicurezza dipende dalle proprietà della funzione hash usata da HMAC
- Se  ha successo in un attacco ad HMAC allora:
  - Può computare l'output della funzione di compressione anche quando IV è casuale e sconosciuto all'attaccante
  - Può computare collisioni nella funzione hash anche quando IV è casuale e sconosciuto all'attaccante

Attacchi ad HMAC

- Miglior attacco conosciuto [1995,1996] basato sul paradosso del compleanno
  - Occorrono $2^{|hash(\cdot)|/2}$ coppie $(M, HMAC_K(M))$
- Esempio:



Authenticated Encryption (AE)

Vogliamo garantire confidenzialità e autenticità/integrità

- **HtE: Hash-then-encrypt.** $E_k(M||h(M))$
- **MtE: MAC-then-encrypt.** $E_{k2}(M||MAC_{k1}(M))$
  - Usato in SSL/TLS
- **EtM: Encrypt-then-MAC.** $E_{k2}(M),\ MAC_{k1}(E_{k2}(M))$
  - Usato in Ipsec
- **E&M: Encrypt-and-MAC.** $E_{k2}(M),\ MAC_{k1}(M)$
  - Usato in SSH

2 chiavi

Block Cipher Mode del NIST

Ci sono 9 modi approvati in pubblicazioni:

- 6 *confidentiality* mode: ECB, CBC, OFB, CFB, CTR, XTS-AES
- 1 *authentication* mode: CMAC
- 2 *combined* mode: CCM e GCM

CCM

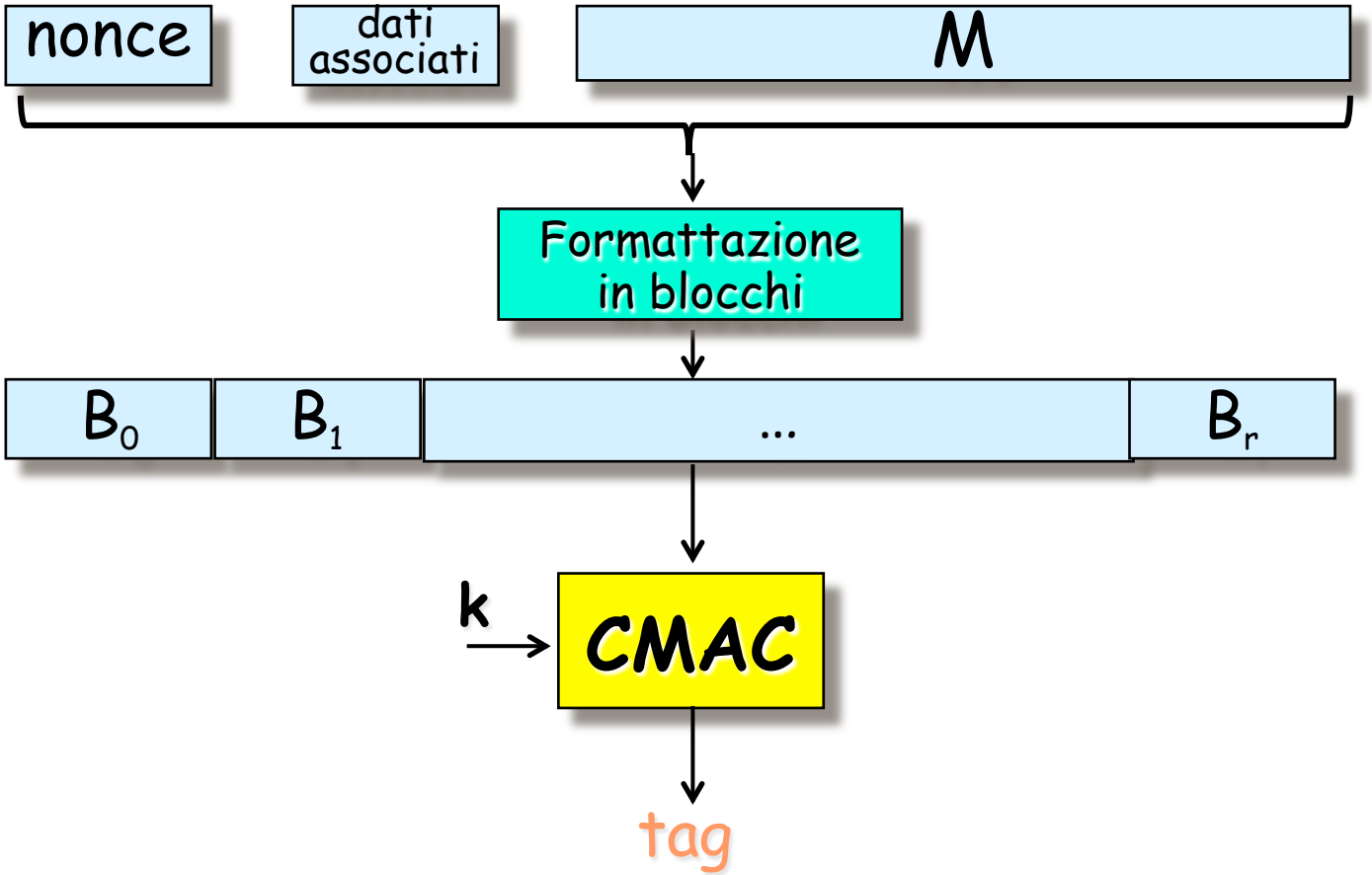
- Counter with Cipher Block Chaining-MAC
- NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, mag 2004 (update lug 2007)
- Sicurezza di IEEE 802.11 WiFi
- Utilizza: AES, CTR modo di operazione, CMAC
- Variazione di Encrypt-and-MAC
  - $E_k(M)$, $MAC_k(M)$ (una sola chiave)

CCM

- Counter with Cipher Block Chaining-MAC
- NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, mag 2004 (update lug 2007)
- Sicurezza di IEEE 802.11 WiFi
- Utilizza: AES, CTR modo di operazione, CMAC
- Variazione di Encrypt-and-MAC
  - $E_k(M)$, $MAC_k(M)$ (una sola chiave)

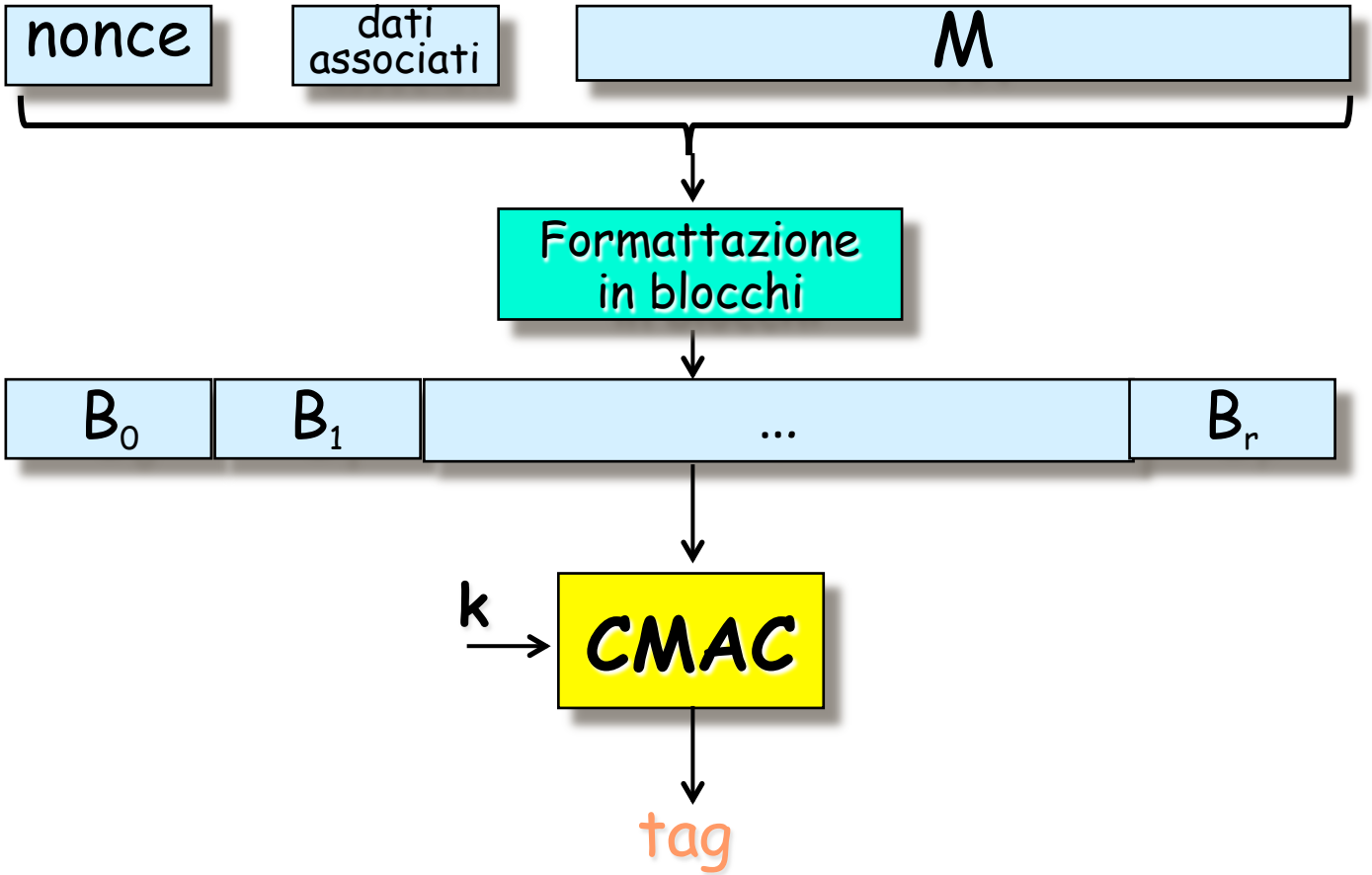
Vediamo prima l'autenticazione e poi la cifratura

CCM: autenticazione

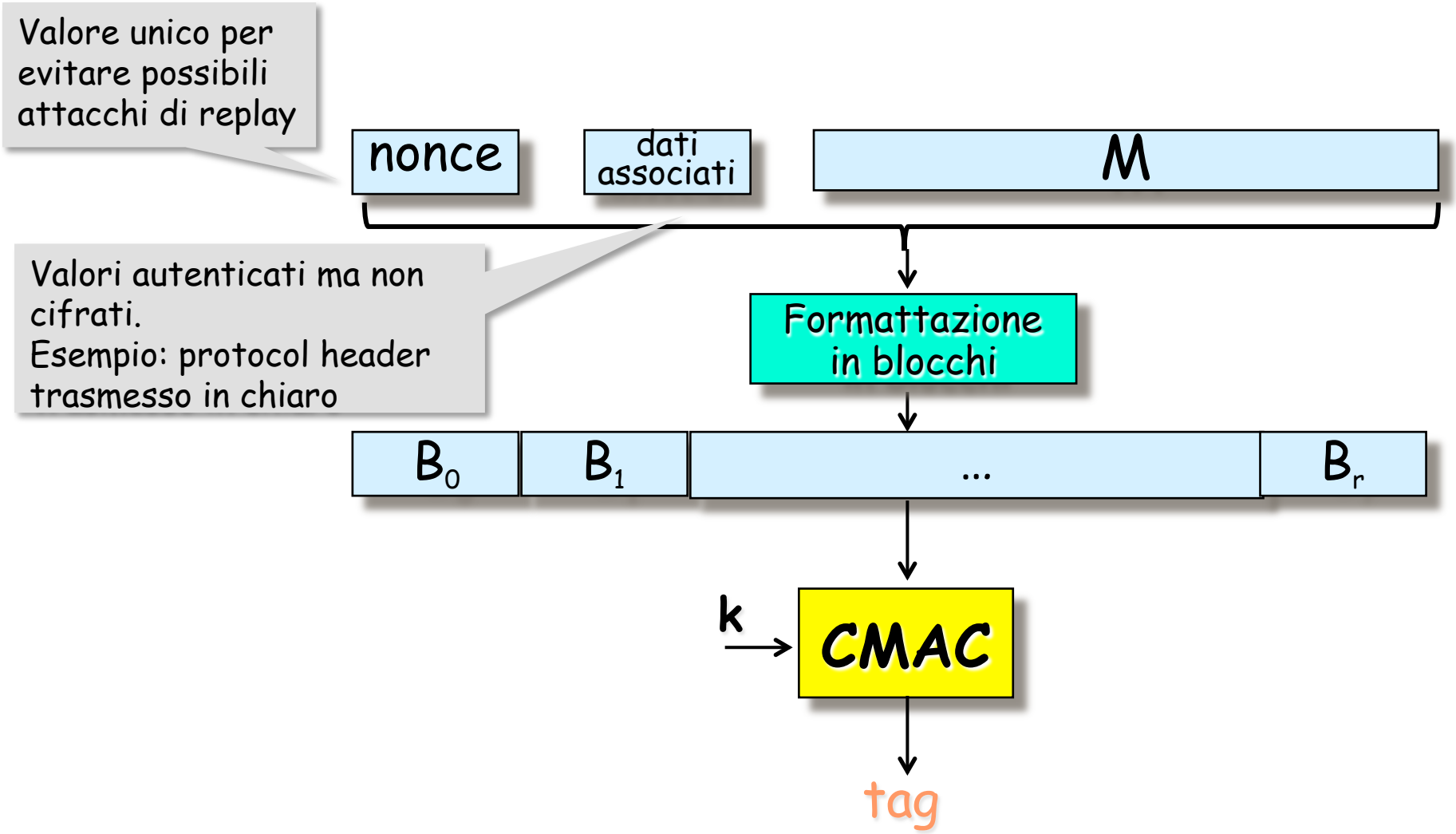


CCM: autenticazione

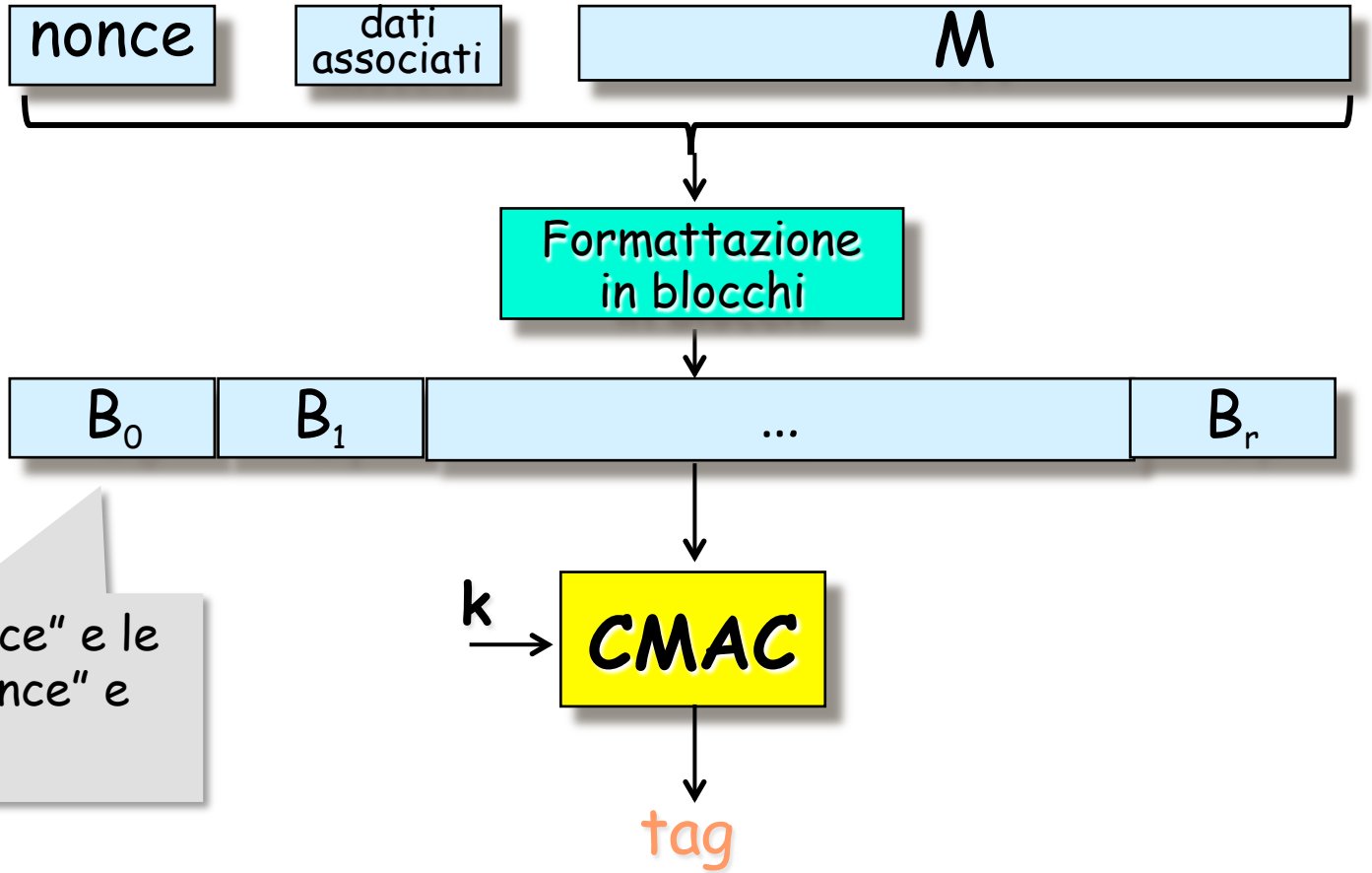
Valore unico per evitare possibili attacchi di replay



CCM: autenticazione

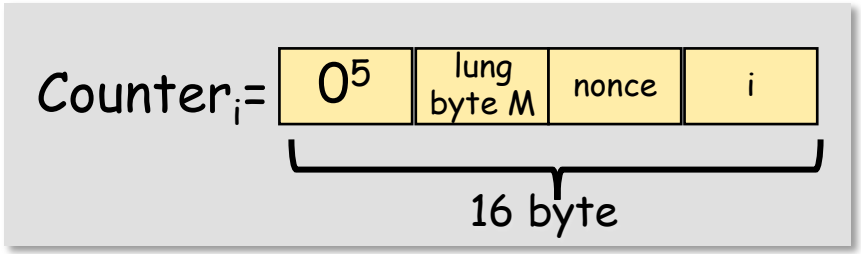


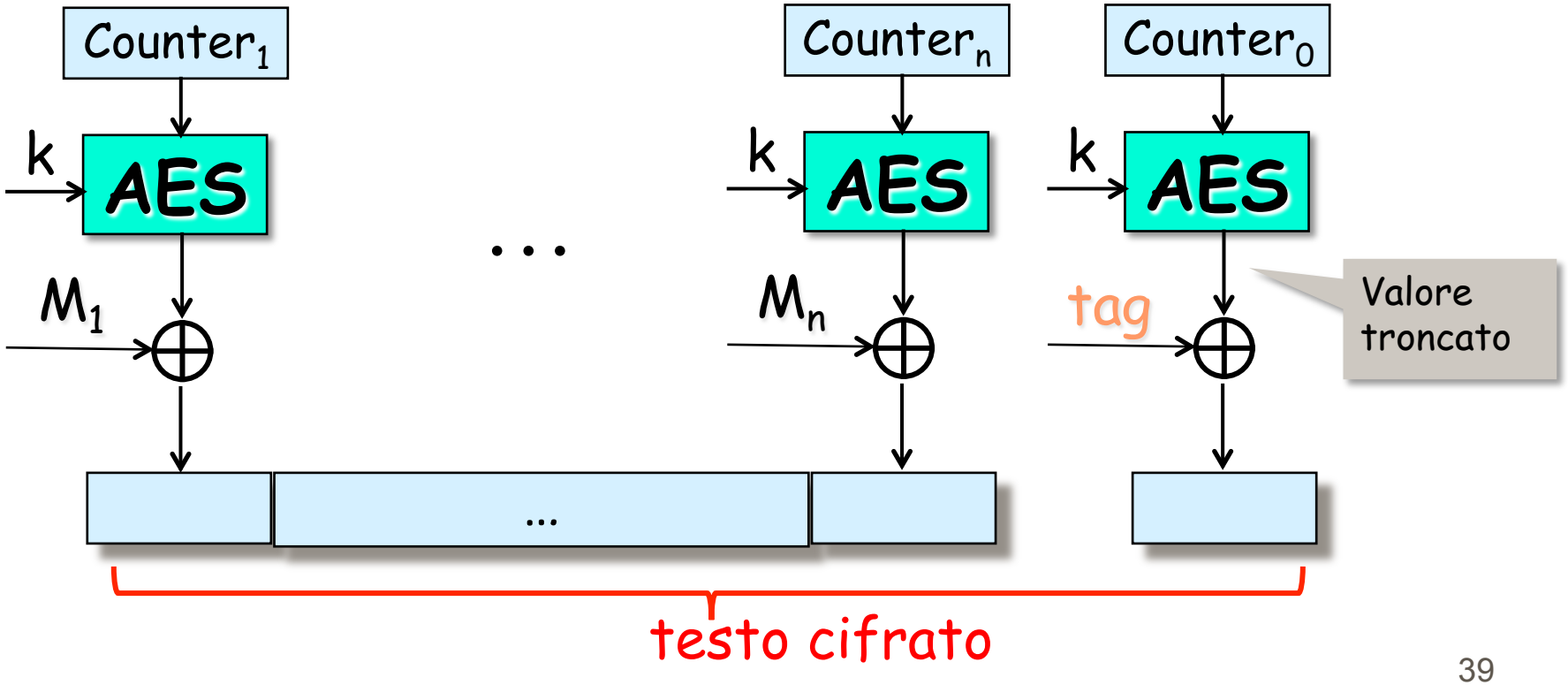
CCM: autenticazione



CCM: cifratura

messaggio in chiaro $M=M_1M_2...M_n$
(diviso in n blocchi di 128 bit)





Galois/Counter Mode (GCM)

- NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/ Counter Mode (GCM) and GMAC, nov 2007
- Parallelizzabile per alto throughput

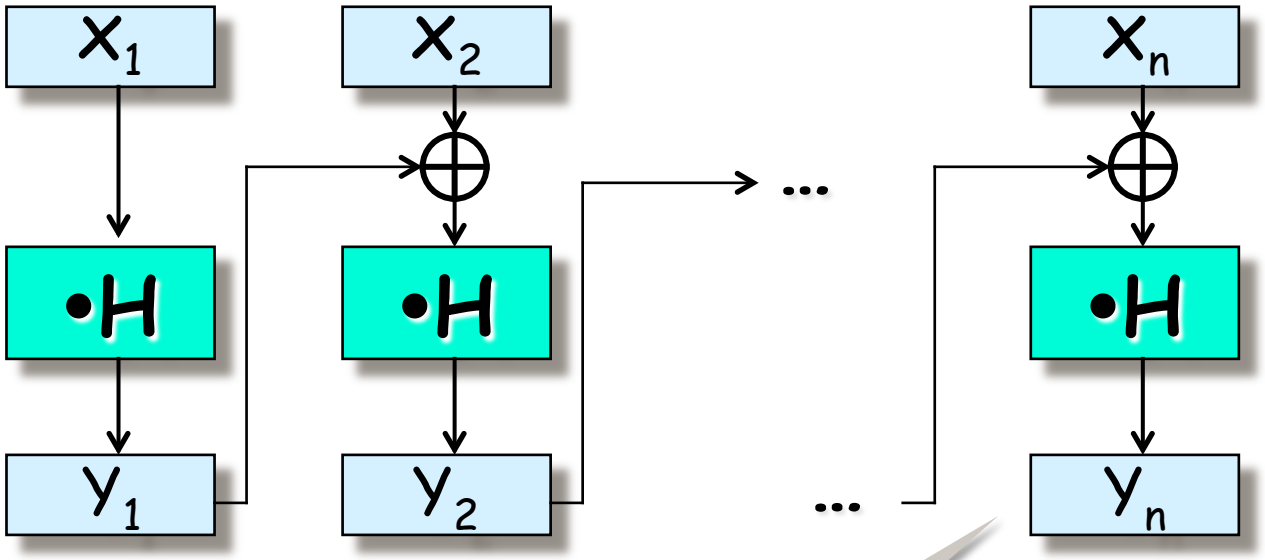
GHASH

Blocchi di 128 bit

$GHASH_H(X_1||X_2||...||X_n)$
$Y=0^{127}1$
for i=1 **to** n do $Y = (Y \bullet X_i) \oplus H$
return Y

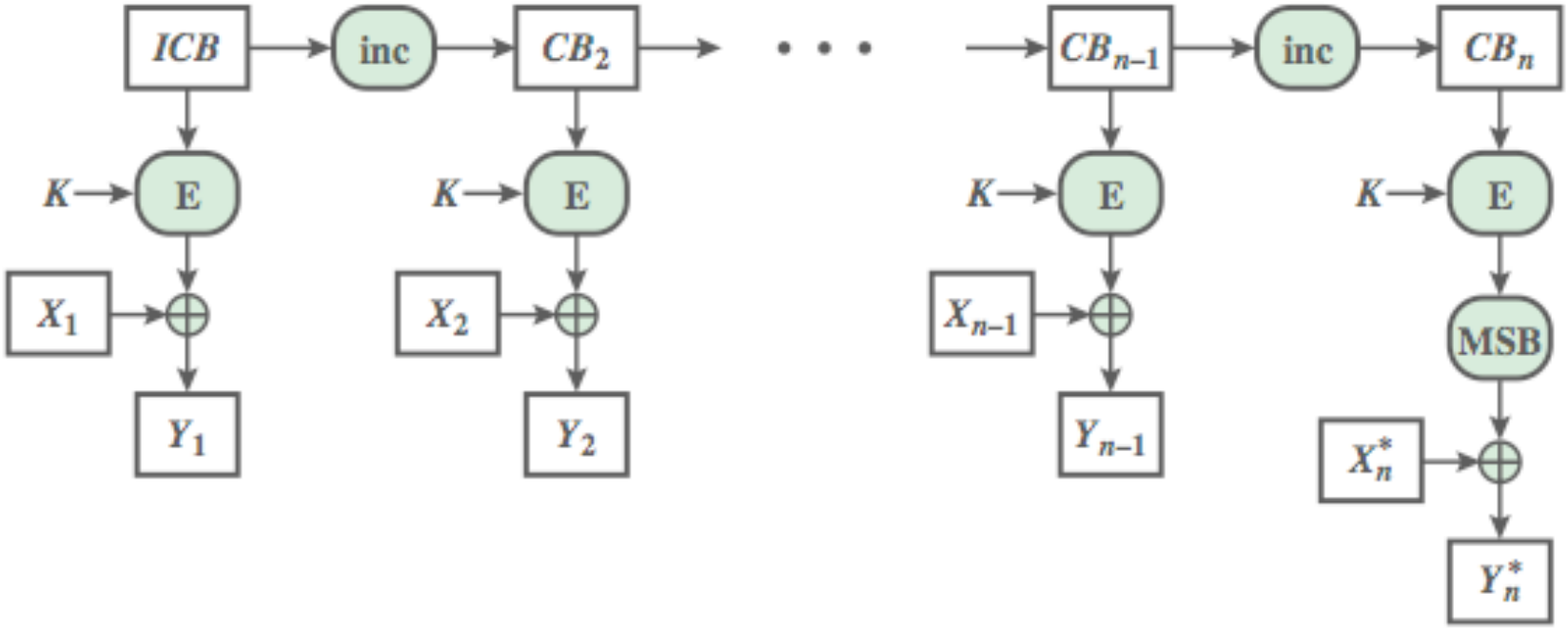
H=hash key

"•" moltiplicazione in $GF(2^{128})$



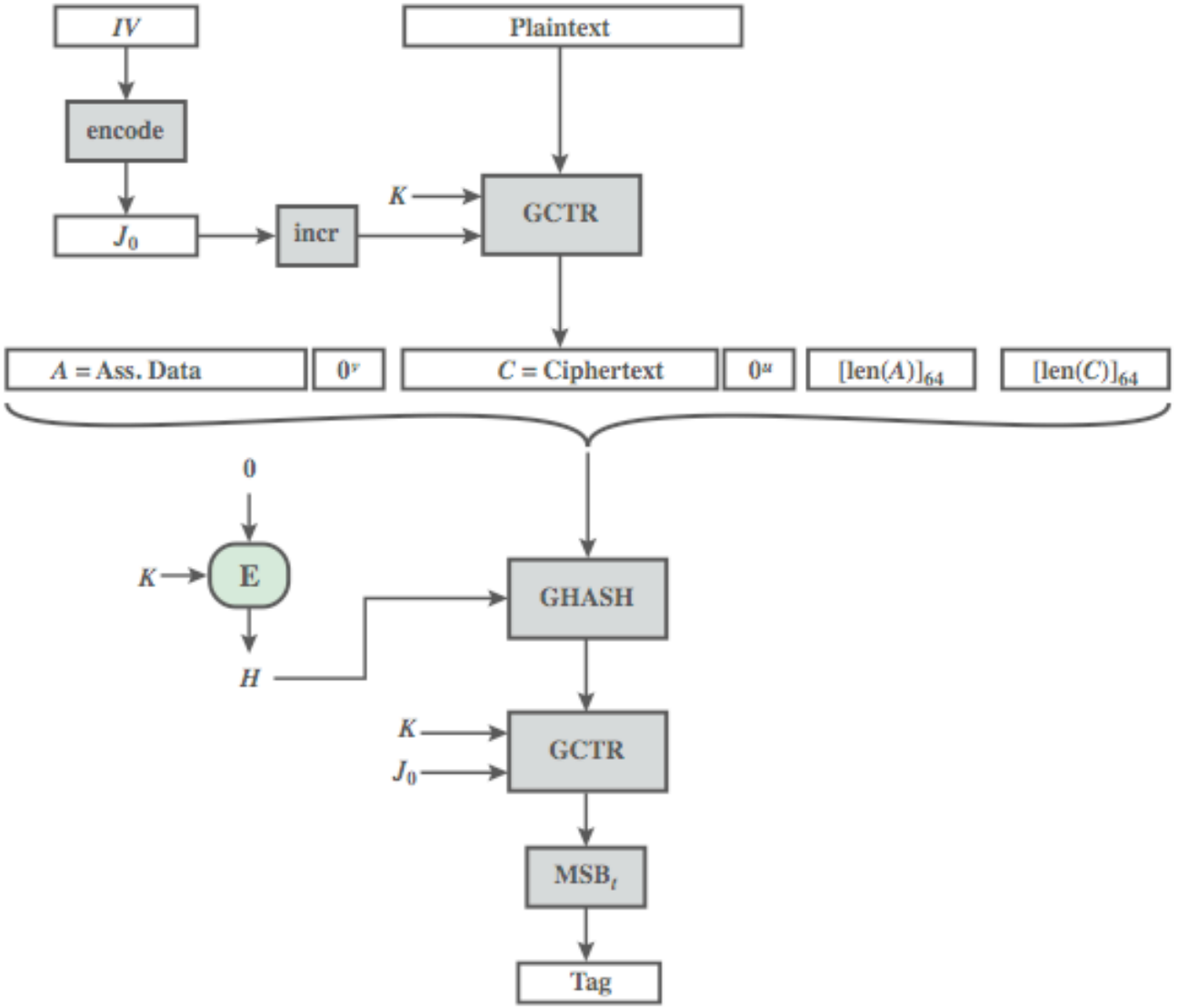
$GHASH_H(X_1||X_2||...||X_n) = Y_n$

GCTR



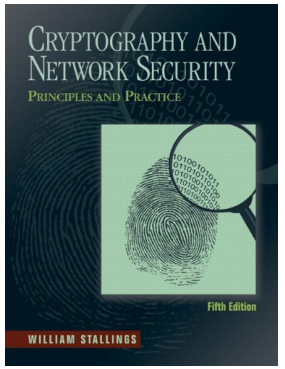
(b) $\text{GCTR}_K(ICB, X_1 \| X_2 \| \ldots \| X_n^*) = Y_n^*$

GCM Mode Overview



Bibliografia

- Cryptography and Network Security
  by W. Stallings, 2010
  - cap. 11 (MAC) e 12 (HMAC)
- Tesina di Sicurezza su reti
  - Message Authentication Code



Domande?