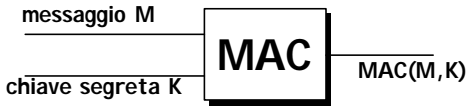




# Message Authentication Code (MAC)

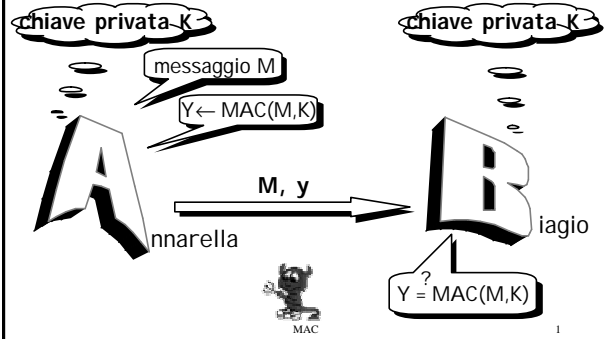


## Applicazioni

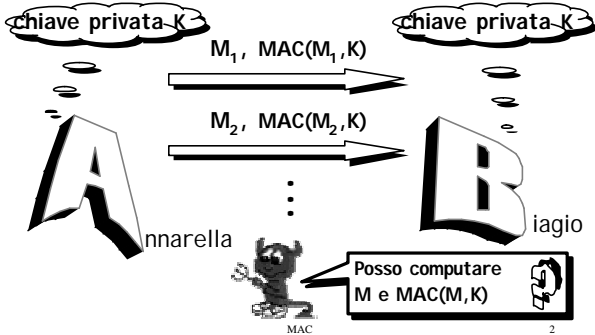
- Autenticità del messaggio M
- Integrità del messaggio M



# Utilizzo MAC

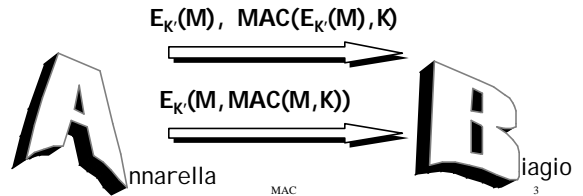


# Sicurezza MAC



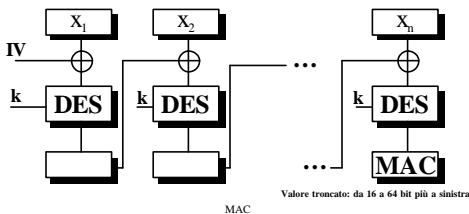
# MAC + confidenzialità

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



# CBC-MAC

- Cipher Block Chaining (con IV=0)
- Uno dei più usati (FIPS PUB 113 e ANSI X9.17)
- Testo  $X = X_1 X_2 \dots X_n$  diviso in blocchi di 64 bit



# MAC basati su Funzioni Hash

- Vantaggi uso funzioni hash
  - Sono più veloci dei cifrari a blocchi, in genere
  - Sono incluse nelle funzioni di libreria, in genere
  - Non ci sono restrizioni sull'esportazione dagli USA
- Attenzione alla costruzione!



## Alcune proposte naive

- ❑  $H(M) \circ K$
- ❑  $H(K \circ M) \circ K$
- ❑  $H(K) \circ M$



Attenzione alla costruzione!

MAC

6



## Metodo del segreto prefisso

### $H(K, M)$



Per funzioni hash iterate:  
aggiunta blocco  $y$  ad  $M$   
ottenendo  $f(H(K, M), y) = H(K, M'y)$

Possibile soluzione:

$H(K, L, M)$  con  $L$ =lunghezza di  $M$

MAC

7



## Metodo del segreto suffisso

### $H(M, K)$



Attacco compleanno  $2^{|\text{hash}(\cdot)|}$  per  
calcolo collisione  $H(M) = H(M')$   
(o meglio, funzione di iterazione)

Quindi,  $H(M, K) = H(M', K)$

MAC

8



## MAC basati su Funzioni Hash

- ❑  $H(K, M, K)$  o meglio  $H(K_1, M, K_2)$
- ❑  $H(K, H(K, M))$
- ❑  $E_K(H(M))$  dove  $E_K(\cdot)$  è un cifrario a blocchi



MAC

9



## HMAC

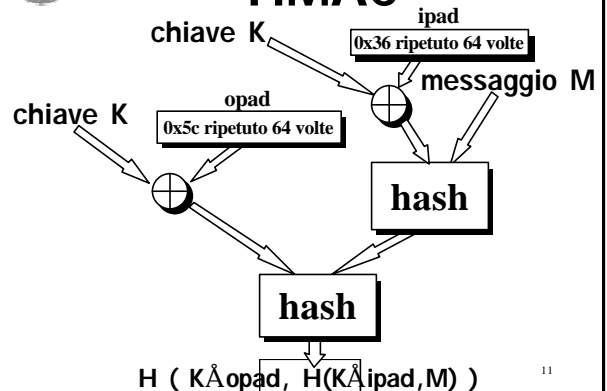
- ❑ RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*, Febbraio 1997
- ❑ ANSI X9.71 *Keyed Hash Message Authentication Code*, 2000
- ❑ FIPS 198, *The Keyed-Hash Message Authentication Code (HMAC)*, pubblicato 6 marzo 2002
  - Standard effettivo dal 6 agosto 2002
  - Draft pubblicato 5 gennaio 2001, review e commenti pubblici
- ❑ 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

MAC

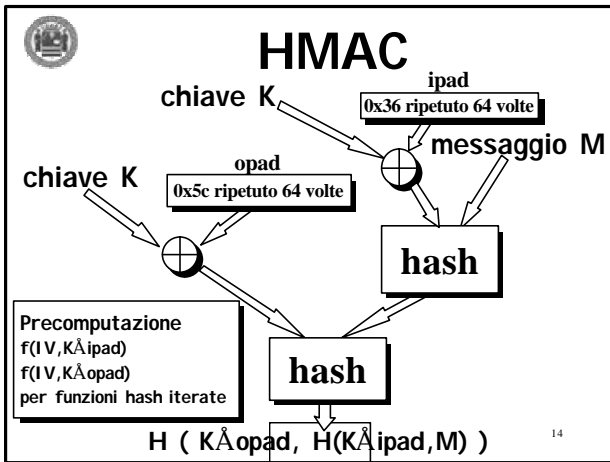
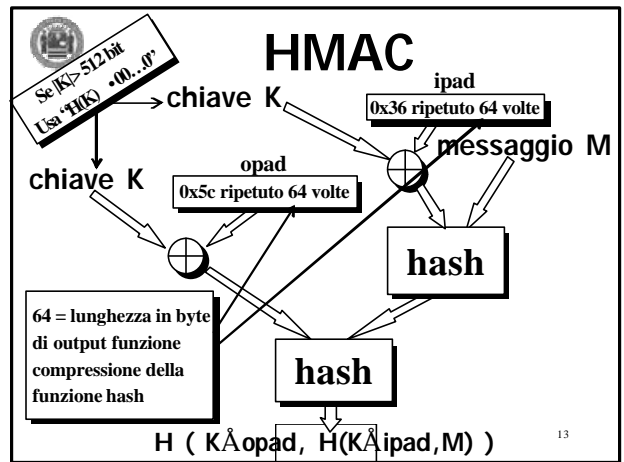
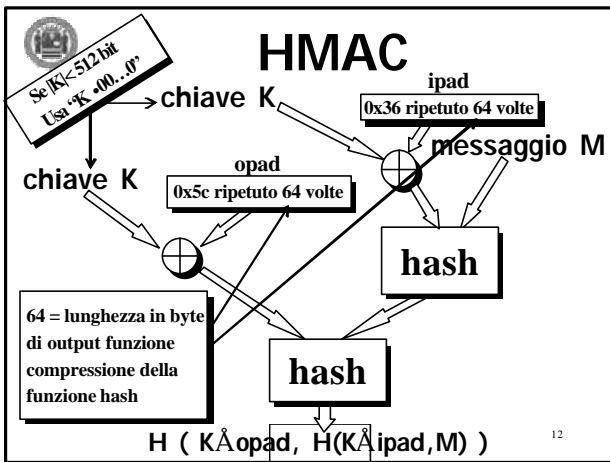
10



## HMAC



11



## Output troncato

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

MAC

15

## Sicurezza HMAC

- Sicurezza dipende dalle proprietà della funzione hash
- 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

MAC

16

## Attacchi ad HMAC

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

$2^{64}$  coppie  
 $(M, \text{HMAC-MD5}_k(M))$

chiave K

Stessa K, diversi M

MAC

17



## Test vectors HMAC-RIPEND-160

Messaggio	Chiave
** (stringa vuota)	00112233445566778899aabbccddeeff01234567
"a"	cf387677bfda8483e63b57e06c3b5ecd8b7fc055
"abc"	0d351d71b78e36ddb7391c810a0d2b6240ddbafc
"message digest"	f7ef288cb1bbcc6160d76507e0a3bbf712fb67d6
"abcdefghijklmnopqrstuvwxyz"	f83662cc8d339c227e600fcd636c57d2571b1c34
"abcdefghijklmnopqrstuvwxyz"	843d1c4eb880ac8ac0c9c95696507957d0155ddb
"abcdefghijklmnopqrstuvwxyz jklmnopq"	60f5ef198a2dd5745545c1f0c47aa3fb5776f881
"A...Za...z0...9"	e49c136a9e5627e0681b808a3b97e6a6e661ae79
8 volte "1234567890"	31be3cc98cee37b79b0619e3e1c2be4f1aa56e6c
1 milione di volte "a"	c2aa88c6405658dc225e485488371fb2433fa735



## Modifica di Funzioni Hash

❑ Non usare la funzione hash come black-box

❑ Esempio: MD5-MAC

- $K' \leftarrow$  primi 128 bit di  $KKK\dots$
- $K_0 \leftarrow MD5(K', U_0, K')$
- $K_1 \leftarrow MD5(K', U_1, K')$
- $K_2 \leftarrow MD5(K', U_2, K')$
- ...

Sono le 4 parole di  
inizializzazione

$U_0 U_1 U_2$  costanti  
ottenute come  $MD5(\dots)$