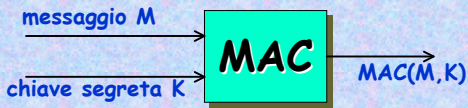




# Message Authentication Code (MAC)

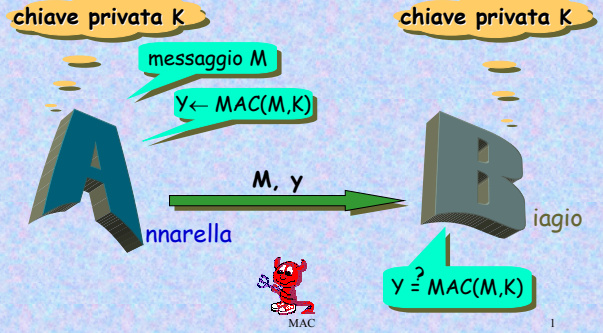


## Applicazioni

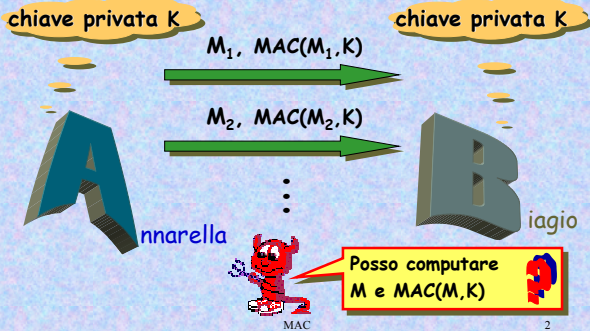
- Verifica autenticità del messaggio  $M$
- Integrità dei dati



# Utilizzo MAC

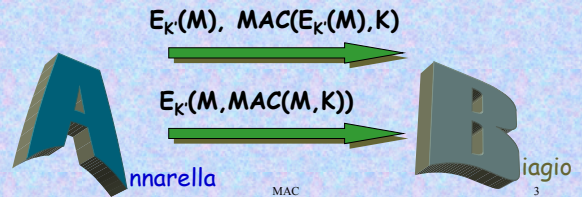


# Sicurezza MAC



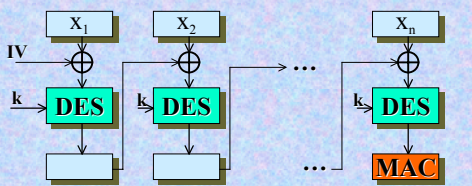
# MAC + confidenzialità

- 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!



## Metodo del segreto prefisso

$H(K, M)$



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

Possibile soluzione:

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

MAC

6



## 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

7



## 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

8



## HMAC

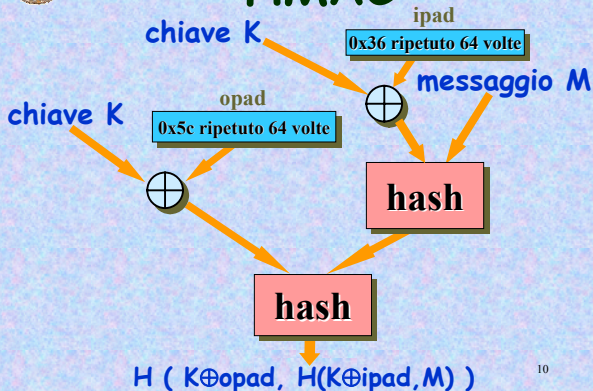
- ❑ RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*, Febbraio 1997
- ❑ Draft FIPS, *Keyed-Hash Message Authentication Code (HMAC)*, pubblicato 5 gen 2001, commenti entro 5 aprile 2001
- ❑ Funzioni Hash usate come black-box
  - Utilizzo delle funzioni hash senza modifiche
  - Facile cambio della funzione hash (più veloci e più sicure)
- ❑ Facile utilizzo e gestione di chiavi

MAC

9



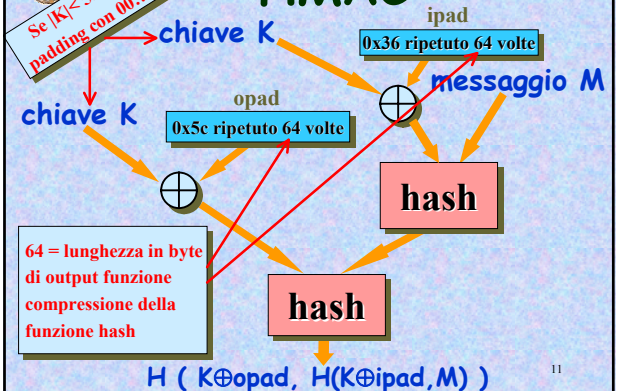
## HMAC



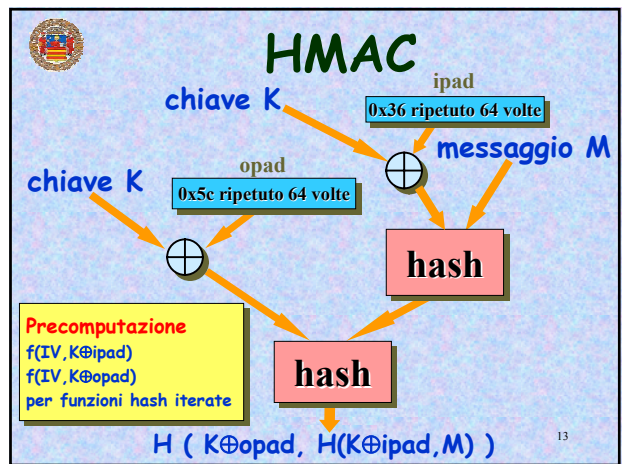
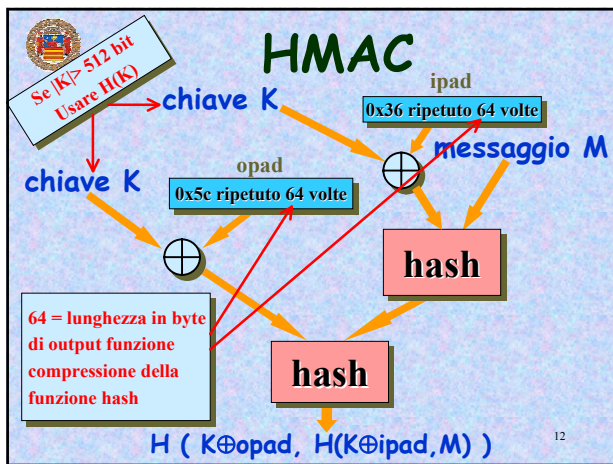
10



## 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),  $t \geq 32$  (Draft FIPS)

MAC 14

### Sicurezza

- ❑ 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 15

### Attacchi

- ❑ Miglior attacco conosciuto [1995,1996] basato sul paradosso del compleanno
  - Occorrono  $2^{\lceil \text{hash}(\cdot) / 2 \rceil}$  coppie  $(M, \text{HMAC}_K(M))$
- ❑ Esempio:
  - Input:  $2^{64}$  coppie  $(M, \text{HMAC-MD5}_K(M))$
  - Output: La chiave  $K$

MAC 16

### Test vectors HMAC-RIPEMD-160

Messaggio	Chiave
"" (stringa vuota)	00112233445566778899aabbccddeeff01234567
"a"	cf387677bfda8483e63b57e06c3b5eccd8b7fc055
"abc"	0d351d71b78e36ddb7391c810a0d2b6240dbafc
"message digest"	f7ef288cb1bbcc6160d76507e0a3bbf712fb67d6
"abcdefghijklmnopqrstuvwxyz"	f83662cc8d339c227e600fcd636c57d2571b1c34
"abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"	843d1c4eb880ac8ac0c9c95696507957d0155ddb
"abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"	60f5ef198a2dd5745545c1f0c47aa3fb5776f881
"A...Za...z0...9"	ehijkiijklklmklmnlmnomnopnopq
8 volte "1234567890"	e49c136a9e5627e0681b808a3b97e6a6e61ae79
1 milione di volte "a"	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...

-  $K_0 \leftarrow \text{MD5}(K', U_0, K')$

-  $K_1 \leftarrow \text{MD5}(K', U_1, K')$

-  $K_2 \leftarrow \text{MD5}(K', U_2, K')$

- ...

Sono le 4 parole di  
inizializzazione

$U_0 U_1 U_2$  costanti  
ottenute come MD5(...)