AUTENTICITÀ

13/04/2023

Concludiamo questa parte del corso, discutendo di come garantire l'AUTENTICITÀ dei messaggi.

ATTENZIONE: un MESSAGGIO CIFRATO **NON** è necessariamente AUTENTICO

↳ per tale motivo è bene separare i concetti

Abbiamo due modi per farlo:
- AUTENTICAZIONE SIMMETRICA
- FIRMA ELETTRONICA

IDEA: AUTENTICO il messaggio con un **CODICE DI AUTENTICAZIONE**

↳ $MAC_K(M)$ nell' **AUTENTICAZIONE SIMMETRICA** con CIFRARI SIMMETRICI

↳ $Sig_A(M)$ nella **FIRMA ELETTRONICA** con CIFRARI ASIMMETRICI

Tale **CODICE** viene poi inviato in aggiunta al messaggio.

AUTENTICAZIONE
SIMMETRICA

18/04/2023

Il destinatario verifichera' la correttezza dei dati ricevuti

↳ L'Avversario non può farlo poiche' non conosce la chiave

MAC Message Authentication Code

$MAC_K(M)$ — M: messaggio; K: chiave condivisa (simmetrica)

e' possibile ottenerlo con:

- DES-CBC
- funzione HASH: HMAC

- AUTENTICAZIONE: **DES-CBC**

(1) Cifro il messaggio

(2) L'ultimo blocco e' il MAC

N.B. L'ATTACCO DEL COMPLEANNO **NON** funziona ☺

↳ la collisione esiste ma l'avversario non riesce a trovarla poiche' non conosce la chiave K con cui trovare il MAC

⇓

la lunghezza del MAC **NON** e' un problema

AUTENTICAZIONE:

HASH

↓

con aggiunta di chiave

OSSERVAZIONE:

NON è un vero e proprio HASH, poiché come sappiamo, quest'ultimo non vuole chiavi in input

HMAC

è uno standard per scegliere due chiavi K_1, K_2 a partire da quella condivisa K al fine di calcolare:

$$H\left(K_1 | H\left(K_2 | M\right)\right)$$

hash ↑ ↑ concatenazione

Più nello specifico:

$$HMAC_K(M) = H\left(\left(\underbrace{K' \oplus opad}_{K_1}\right) | H\left(\left(\underbrace{K'' \oplus ipad}_{K_2}\right) | M'\right)\right)$$

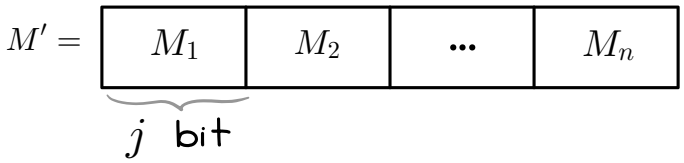
È stato dimostrato che HMAC è sicuro!

N.B. L'ATTACCO DEL COMPLEANNO **NON** funziona ☺

↳ per il medesimo motivo del DES-CBC

HMAC - Algoritmo:

(1) Divido il messaggio M' $(M + padding)$ in n blocchi di j bit



(2) Genero K', K'' → NON sono K_1, K_2

$$K' = \begin{cases} K & \textit{altrimenti} \\ H(K) & \text{se K>j bit} \end{cases}$$

$$K'' = K' + padding \qquad \textit{fino al raggiungimento di j bit}$$

(3) Genero K_1, K_2

(3.1) Prendo due costanti:

$$ipad = 00110110$$
$$opad = 01011010$$

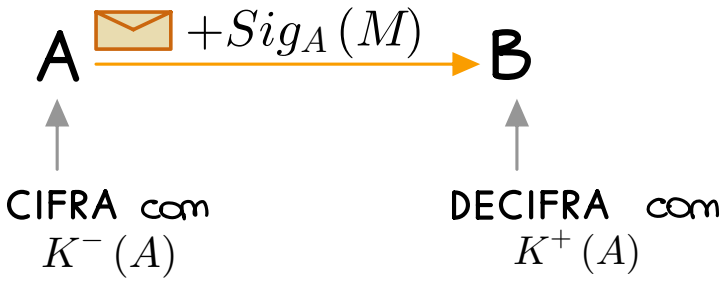
(3.2) Ripeto tali costanti un certo numero di volte...
fino ad eguagliare la lunghezza di K

$$\underbrace{ipad + .. + ipad}_{j/8 \text{ volte}}$$
$$\underbrace{opod + \ldots + opad}_{j/8 \text{ volte}}$$

(3.3) Calcolo MAC

$$HMAC_K(M) = H\left((\underbrace{K' \oplus opad}_{K_1}) \,|H\left((\underbrace{K'' \oplus ipad}_{K_2}) \,|M'\right)\right.$$

FIRMA ELETTRONICA



Contro:

L' ATTACCO DEL COMPLEANNO è possibile ☹

↓

Tale problema è risolvibile usando una lunghezza del codice adeguata

E' possibile ottenerla in due modi:

- RSA con MD5/SHA-1
- DSA con SHA-1 → non lo studieremo

Attacco del compleanno Firma Elettronica:

- Il mittente cifra il messaggio M con la sua chiave chiare privata

... la firma + il messaggio viaggiano ... e l'avversario lo intercetta

- L'avversario:
  - Prende M e ne genera un certo numero di varianti → messaggi equivalenti dal punto di vista semantico
  - Prende un M' falso e ne genera lo stesso numero di varianti
  - Trova collisione e convince il mittente a firmare il messaggio falsificato (non sempre possibile)

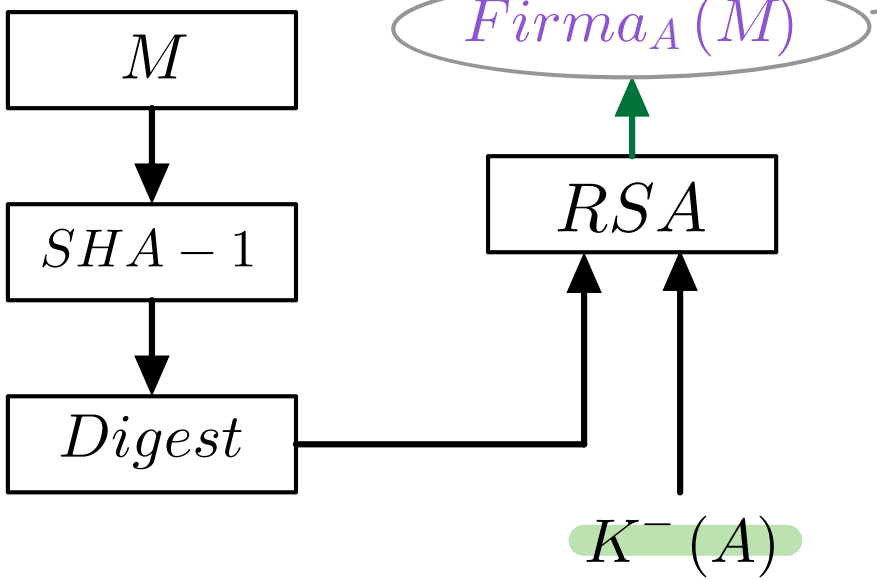
Per il paradosso del compleanno che se M é abbastanza grande la probabilità che ci siano due messaggi, appartenenti ad entrambi gli insiemi, che creano una collisione é >1/2.

É possibile farlo poiché la firma é pubblica; con il MAC questo attacco non funziona poiché non si conosce la chiave condivisa

FIRMA ELETTRONICA: RSA + SHA-1

Lunghezza variabile: $(Digest)^{K^-(A)} \; mod \; n$

MITTENTE:

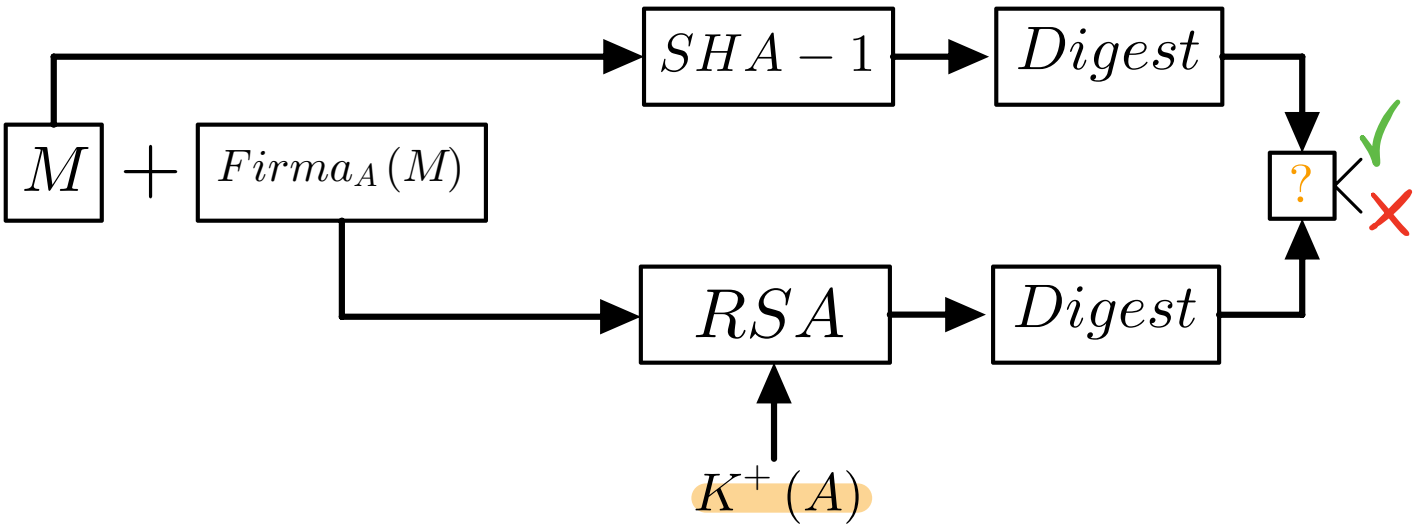


OSSERVAZIONE: poiche' la chiave privata del mittente la conosce **solo** quest'ultimo:

(1) questo rispecchia la realtà
(ciò che avviene nel mondo reale)

(2) Si ha il concetto di **NO-Repudation**:
A non può disconoscere il messaggio

DESTINATARIO:



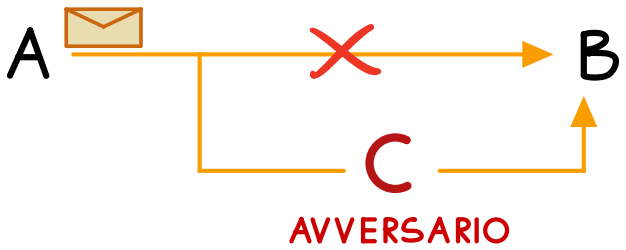
OSSERVAZIONE: poiche' la chiave pubblica del mittente la conoscono tutti, si dice che la firma è **OPPONIBILE A TERZI**:

chiunque può svolgere l'operazione di verifica (e questo rispecchia anche la realtà nel mondo reale)

PROBLEMA:

ATTACCO MAN IN THE MIDDLE

L' avversario può intercettare il messaggio firmato da A e firmarlo con la sua chiave:



ATTACCO: CHIAVE PUBBLICA **FALSA**!

Per cui è buona norma che B sia certo che la chiave pubblica in suo possesso sia veritiera...

27/04/2023

PROBLEMA : DISTRIBUZIONE DELLE CHIAVI

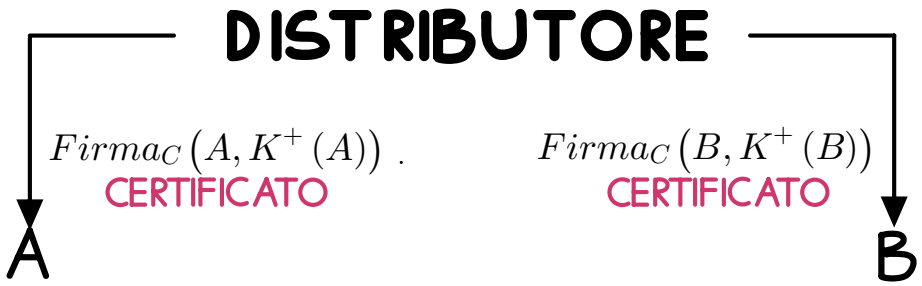
Nella pratica si utilizzano dei **DISTRIBUTORI** (fidati) di chiavi dei quali ottenere **CERTIFICATI**:

Host + Chiave Pubblica

(1) Gli host depositano le proprie chiavi pubbliche al proprio DISTRIBUTORE di fiducia:

ed ottengono de esso stesso, la chiave pubblica del DISTRIBUTORE

(2) Gli host richiedono il proprio CERTIFICATO dal Distributore:



Tale CERTIFICATO sarà poi usato nelle comunicazioni in aggiunta al messaggio e alla firma:

$$M + Sig_A(M) + Certificato$$

EXTRA: Si puo' aggiungere anche una validazione temporale : **Timestamp**