CORSO DI SICUREZZA INFORMATICA 1 (A.A. 2005/2006)

Prof. A. Armando

(20 Giugno 2006)

Si risponda alle domande utilizzando lo spazio apposito.
Non è consentito l'utilizzo di libri, appunti, nè dispositivi elettronici di alcun tipo.

Nome e Cognome: _____

Matricola: _____

---

1. ***Crittografia simmetrica***

   Si determini il plaintext corrispondente al ciphertext `dcbdmoet` *ottenuto applicando la procedura di cifratura di Vigegnère ed utilizzando come chiave la sequenza di numeri 4 2 11. Si consideri l'alfabeto italiano.*

   Soluzione.
   Ciphertext: `dcbdmoet`
   Plaintext: `zanzibar`

2. ***Crittografia Simmetrica*** *Dire quali delle seguenti affermazioni sono FALSE, giustificando la risposta data. Si assuma che e_1 ed e_2 sono generiche (ovvero arbitrarie) permutazioni sull'alfabeto considerato.*

   1. *[VERO/FALSO] Se $E^s_e(m)$ è una procedura di sostituzione polialfabetica, allora $E^s_{e_1}(E^s_{e_2}(m))$ è più sicura sia di $E^s_{e_1}(m)$ che di $E^s_{e_2}(m)$.*

      Soluzione. FALSO. Infatti l'applicazione consecutiva di due sostituzioni con permutazioni e_1 ed e_2 equivale ad una singola sostituzione con permutazione $e_1 \circ e_2$.

   2. *[VERO/FALSO] Se $E^t_e(m)$ è una procedura di trasposizione (transposition cipher), allora $E^t_{e_1}(E^t_{e_2}(m))$ è più sicura di $E^t_{e_1}(m)$.*

      Soluzione. FALSO. Infatti l'applicazione consecutiva di due trasposizioni con permutazioni e_1 ed e_2 equivale ad una singola trasposizione con permutazione $e_1 \circ e_2$.

   3. *[VERO/FALSO] Se $E^s_e(m)$ è una procedura di sostituzione polialfabetica e $E^t_e(m)$ è una procedura di trasposizione (transposition cipher), allora $E^s_{e_1}(E^t_{e_2}(m))$ è più sicura sia di $E^s_{e_1}(m)$ che di $E^t_{e_2}(m)$.*

      Soluzione. VERO. Ed infatti la combinazione di sostituzione e trasposizione è alla base della maggior parte degli algoritmi di cifratura simmetrica moderni.

3. ***Sicurezza di Rete***

Indicare la verità o falsità delle seguenti affermazioni.

- *[VERO/FALSO] Con la link encryption è necessaria una coppia di chiavi per ciascuna coppia di utenti.*

  Soluzione. FALSO.

- *[VERO/FALSO] Con la link encryption i messaggi scambiati sono in chiaro nell'host di origine e in quello di destinazione.*

  Soluzione. VERO.

- *[VERO/FALSO] Con la end-to-end encryption i messaggi vengono cifrati nei nodi intermedi*

  Soluzione. FALSO.

- *[VERO/FALSO] Con la link encryption i meccanismi di sicurezza sono "trasparenti" (ovvero invisibili) all'utente.*

  Soluzione. VERO.

- *[VERO/FALSO] Con la link encryption è possibile proteggere o meno i sigoli messaggi inviati ad un determinato host.*

  Soluzione. FALSO.

- *[VERO/FALSO] Con la end-to-end encryption è necessaria una coppia di chiavi per ciascun nodo intermedio.*

  Soluzione. FALSO.

4. ***Funzioni di Hash***

Si scrivano nei riquadri bianchi le proprietà di sicurezza assicurate da ciascuno dei seguenti schemi crittografici.

Soluzione.

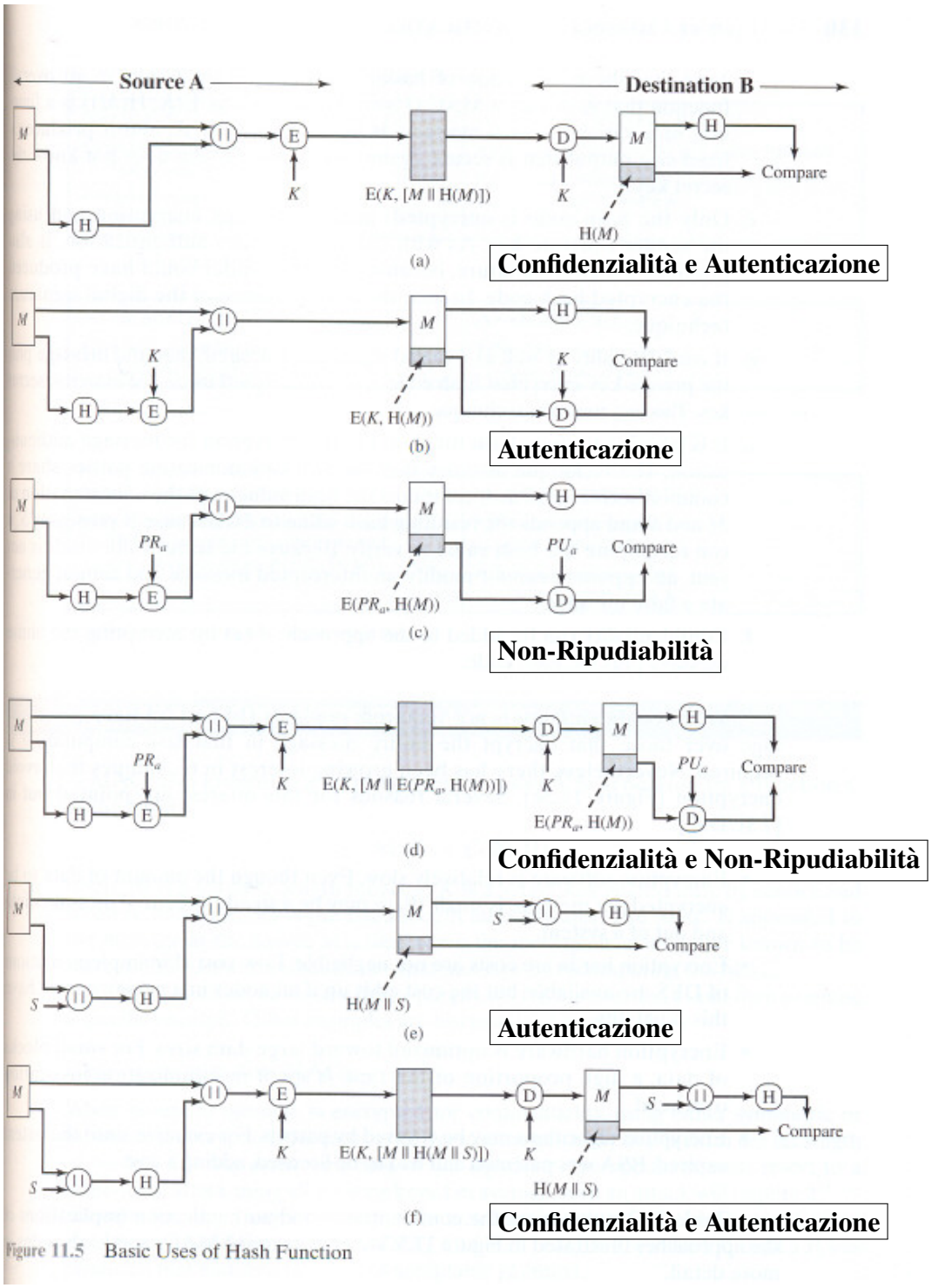

Figure 11.5 Basic Uses of Hash Function

Siccome

- non-ripudio $\Rightarrow$ autenticazione (di messaggio) e
- autenticazione (di messaggio) $\Rightarrow$ integrità

nelle risposte è indicata tra queste solo la condizione più forte.

5. ***Crittografia a chiave pubblica***

Durante l'esecuzione del protocollo per lo scambio di chiave di Diffie-Hellman, viene osservato il seguente scambio di messaggi: Alice manda a Bob il numero 5 e Bob manda ad Alice il numero 8. I valori del modulo e del generatore α sono (pubblicamente) noti essere rispettivamente pari a $q = 11$ e $\alpha = 2$. Si determini la chiave condivisa da Alice e Bob alla fine dell'esecuzione del protocollo.

Nota: *I numeri sono sufficientemente piccoli da rendere possibile lo svolgimento dei calcoli a mano.*

Si giustifichino le risposte date scrivendo tutti i calcoli intermedi.

Soluzione.
Sappiamo che $Y_A = \alpha^{X_A} \bmod q = 2^{X_A} \bmod 11 = 5$. È facile determinare che $X_A = 4$ è soluzione di tale equazione.
Analogamente $Y_B = \alpha^{X_B} \bmod q = 2^{X_B} \bmod 11 = 8$. È facile determinare che $X_B = 3$ è soluzione di tale equazione.
Possiamo quindi calcolare $K_A = {Y_B}^{X_A} \bmod q = 8^4 \bmod 11 = 4096 \bmod 11 = 4$. Verifichiamo infine che la stessa chiave è ottenuta da Bob, infatti $K_B = {Y_A}^{X_B} \bmod q = 5^3 \bmod 11 = 125 \bmod 11 = 4$.

6. ***Protocolli di Sicurezza***

Si consideri il seguente protocollo P_1 utilizzato in ambito di telefonia mobile

$$(1)\ B \to M : B, K_B$$
$$(2)\ M \to B : \{K\}_{K_B}$$
$$(3)\ M \to B : \{M, P\}_K$$

dove M è un terminale mobile con limitate capacità di calcolo, B è la stazione base, K_B è una chiave pubblica di B non nota a M, K è una chiave segreta per crittografia simmetrica generata da M e P una password nota solo a M e a B. Ogni qualvolta viene completato il protocollo la stazione B addebita a M un costo prestabilito.

(a) *Si descrivano i singoli passi del protocollo e le proprietà di sicurezza per il quale è stato presumibilmente progettato.*

Soluzione. Al passo 1 B comunica a M la propria identità e la propria chiave pubblica K_B. Al passo 2 M genera la chiave segreta K e la invia confidenzialmente a B. Siccome B non può essere ancora certo che K provenga da M, nel passo 3 M invia a B in modo confidenziale la propria identità M e la password P. A questo punto B è certo che K sia stata effettivamente generata da M (graze al messaggio ricevuto al passo 3) e che non è nota ad altri (grazie al messaggio ricevuto al passo 2). L'obiettivo del protocollo è dunque quello di stabilire una chiave di sessione tra M e B. Più in generale, il protocollo dovrebbe garantire:

1. la confidenzialità di K e P,
2. l'autenticazione di M nei confronti di B (entity authentication). Quest'ultima proprietà è importante per evitare *replay attacks*, qui particolarmente importante in quanto ad ogni esecuzione del protocollo viene effettuato un addebito.

(b) Spiegare una ragione per cui il protocollo P_1 è preferibile al seguente protocollo P_2 che, pur garantendo le stesse proprietà di sicurezza, è indiscutibilmente più semplice.

$$(1)\ B \to M : B, K_B$$
$$(2)\ M \to B : \{K, M, P\}_{K_B}$$

Soluzione.
Perchè P_2 richiede che M applichi cifratura a chiave pubblica a un plaintext più lungo. Essendo M un dispositivo con limitate capacità computazionali, P_1 è preferibile a P_2.

(c) Si discuta se il protocollo garantisce o meno le proprietà di sicurezza indicate nella risposta alla domanda (a).

Soluzione. La confidenzialità sia di K e di P non sono garantite come mostrato dal seguente attacco:

$$\begin{array}{lll} B \to & I & : B, K_B \\ & I \to M & : B, K_I \\ & I \leftarrow M & : \{K\}_{K_I} \\ B \leftarrow & I & : \{K\}_{K_B} \\ & I \leftarrow M & : \{M, P\}_K \\ B \leftarrow & I & : \{M, P\}_K \end{array}$$

Neppure la entity authentication è garantita, come mostrato dal seguente (ancor più semplice) attacco:

$$\begin{array}{lll} B \to & I \to M & : B, K_B \\ B \leftarrow & I \leftarrow M & : \{K\}_{K_B} \\ B \leftarrow & I \leftarrow M & : \{M, P\}_K \\ & & \vdots \\ B \to & I & : B, K_B \\ B \leftarrow & I & : \{K\}_{K_B} \\ B \leftarrow & I & : \{M, P\}_K \end{array}$$

dove l'intruder dopo aver osservato una normale esecuzione del protocollo "rigioca" i messaggi $\{K\}_{K_B}$ e $\{M, P\}_K$ osservati in precedenza facendo addebitare un ulteriore costo a M senza che questo abbia preso parte alla seconda esecuzione del protocollo eseguita da B. Si noti che in questo semplice caso, la confidenzialità di K e B non è violata.

(d) *Nel caso sia stata individuata una vulnerabilità, si identifichi un raffinamento del protocollo che non soffre di tale vulnerabilità. A tal fine si assuma che*

- B *sia in possesso di un certificato* $\{B, K_B\}_{K_{CA}^{-1}}$ *emesso da un'autorità di certificazione* CA *relativo alla propria chiave pubblica* K_B *e*
- M *sia in possesso di* K_{CA}*, ovvero della chiave pubblica di CA.*

Soluzione. Se B al passo 1 trasmette il certificato $\{B, K_B\}_{K_{CA}^{-1}}$ allora M ha modo di verificare l'autenticità di K_B e il primo dei due attacchi mostrati in precedenza non è più possibile.

$$
\begin{array}{ll}
(1) & B \to M : \{B, K_B\}_{K_{CA}^{-1}} \\
(2) & M \to B : \{K\}_{K_B} \\
(3) & M \to B : \{M, P\}_K
\end{array}
$$

Per far sì che il protocollo non sia soggetto a replay attacks è sufficiente realizzare il solito meccanismo di challenge-response mediante una nonce come indicato qui di seguito:

$$
\begin{array}{ll}
(1) & B \to M : N_B, \{B, K_B\}_{K_{CA}^{-1}} \\
(2) & M \to B : \{K\}_{K_B} \\
(3) & M \to B : \{N_B, M, P\}_K
\end{array}
$$

7. ***Controllo degli Accessi***

Si consideri un sistema informativo universitario, in cui l'informazione relativa agli studenti è memorizzata in due files `piani_di_studio.xls` *e* `voti.xls`*. Siamo interessati a far sì che il sistema realizzi la seguente politica di sicurezza garantendo nel contempo che non vi possa essere alcun flusso informazione dal file* `piani_di_studio.xls` *al file* `voti.xls`*.*

1. *I professori possono leggere e scrivere i files* `piani_di_studio.xls` *e* `voti.xls`*.*
2. *Gli esercitatori possono leggere e scrivere il file* `voti.xls`*, ma non possono accedere in alcun modo al file* `piani_di_studio.xls`*.*
3. *Gli studenti possono solo leggere i files* `piani_di_studio.xls` *e* `voti.xls`*.*

Si descriva un modello reticolare che consente di realizzare tale politica di sicurezza indicando gli utenti che sono autorizzati a fare downgrading all'atto del collegamento e quelli che invece non lo sono.

Si assuma che il sistema escluda le "write-up". Ovvero ogni tentativo di scrittura da parte di un processo p con security label s_p su una risorsa r con security label $s_r > s_p$ viene automaticamente negato dal sistema.

NOTA: *Nel modello reticolare, normalmente un utente si collega al sistema utilizzando il proprio security label s. (Ciò significa che tutti i processi da lui invocati avranno security label s.) Tuttavia, se autorizzato, un utente può fare* downgrading*, ovvero può collegarsi al sistema utilizzando un security label $s' \leq s$. (Ciò significa che tutti i processi da lui invocati avranno security label s'.)*

Soluzione.
Attribuiamo ai files le security labels nel seguente modo:

- `piani_di_studio.xls` $\longrightarrow$ (CONFIDENTIAL, {PIANI})
- `voti.xls` $\longrightarrow$ (CONFIDENTIAL, {VOTI})

e attribuiamo agli utenti le security labels nel seguente modo:

- PROFESSORE $\longrightarrow$ (SECRET, {PIANI, VOTI})
- ESERCITATORE $\longrightarrow$ (SECRET, {VOTI})
- STUDENTE $\longrightarrow$ (CONFIDENTIAL, {PIANI, VOTI})

Infine attribuiamo ai professori e agli esercitatori la possibilità di fare downgrading, mentre escludiamo tale possibilità per gli studenti.
Ora verifichiamo che tale modello realizza la politica di sicurezza richiesta:

1. I professori possono leggere entrambe i files in quando il loro security label domina quello dei files. Inoltre possono anche scrivere sui file purchè facciano downgrading.
2. Gli esercitatori possono leggere e scrivere il file `voti.xls` (la scrittura è possibile previo downgrading), ma non possono accedere in alcun modo al file `piani_di_studio.xls` in quanto il proprio security label non domina quello di `piani_di_studio.xls` (escludendo così la possibilità di lettura).
3. Gli studenti possono leggere i files `piani_di_studio.xls` e `voti.xls` in quanto il loro security label domina quello dei files. Tuttavia non possono scrivere su tali files in quanto non possono fare downgrading.