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

Prof. A. Armando

(6 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 utilizzi la procedura di cifratura per trasposizione per codificare la sequenza di caratteri ottenuta concatenando il proprio nome e il proprio cognome ed eliminando dalla coda il minimo numero di caratteri in modo tale da ottenere una stringa di lunghezza multipla di 4. Ad esempio, nel mio caso la stringa da considerare è:

A	l	e	s	s	a	n	d	r	o		A	r	m	a	n

   Come chiave si utilizzi la permutazione `2 3 4 1`.

   Soluzione.

   Plaintext: `Alessandro Arman`
   Ciphertext: `lesAandso Armanr`

   Innanzitutto occorre suddividere la sequenza in ingresso in blocchi di 4 caratteri ciascuno. Poi per ciascun blocco il ciphertext corrispondente è dato da:

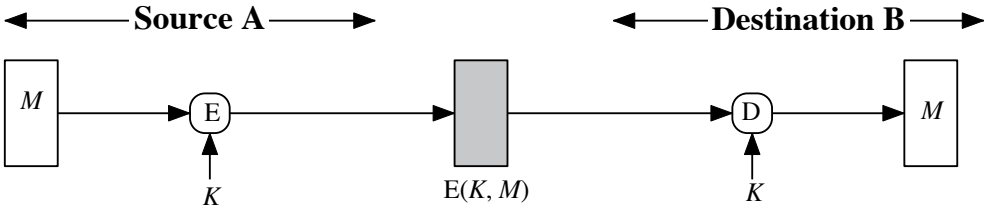
$$E_e(m) = m_{e(1)}m_{e(2)}\cdots m_{e(t)}$$

   dove nel nostro caso $e(1) = 2$, $e(2) = 3$, $e(3) = 4$, $e(4) = 1$.
   Per il primo blocco abbiamo $E_e(\texttt{Ales}) = m_2m_3m_4m_1 = \texttt{lesA}$. Si procede analogamente per il blocchi successivi.

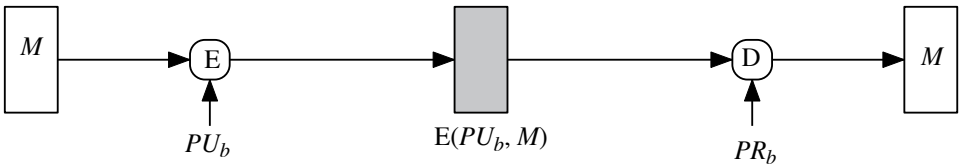
2. ***Message Encryption***

Si indichino le proprietà di sicurezza assicurate dai seguenti schemi crittografici.

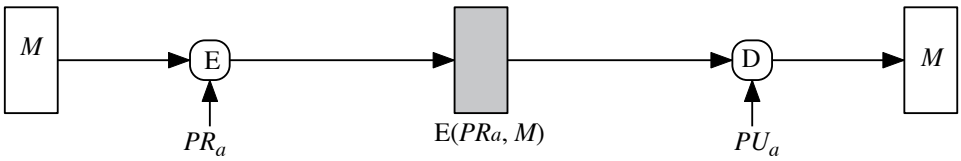
Soluzione.



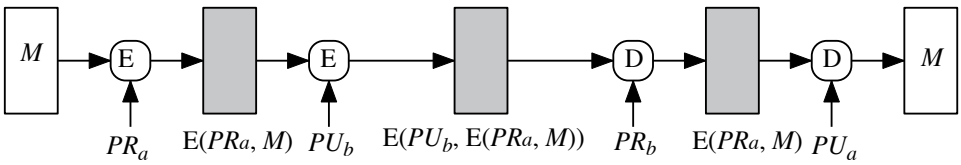
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



(c) Public-key encryption: authentication and non-repudiation



(d) Public-key encryption: confidentiality, authentication, and non-repudiation

3. ***Crittografia a chiave pubblica I***

Si discuta brevemente il ruolo di una smartcard nella produzione di una firma digitale. In particolare, quali informazioni vengono scambiate tra il PC e la smartcard?

Soluzione. La smartcard contiene la chiave privata dell'utente ed è in grado di cifrare stringhe di bit relativamente corte (a causa della limitata capacità computazionale). Ciò è comunque sufficiente ai fini della firma digitale. Infatti il PC calcolerà e invierà alla smartcard solo lo hash del documento da firmare. La smartcard cifrerà tale hash con la chiave privata in essa memorizzata e invierà al PC il risultato. In nessun caso la smartcard trasmette verso l'esterno la chiave privata in essa memorizzata.

4. ***Crittografia a chiave pubblica II***

Si consideri l'algoritmo RSA con $p = 7$, $q = 3$ e $e = 5$.

- *Si calcoli il testo cifrato* C *corrispondente al testo in chiaro* $M = 4$.
- *Si calcoli la chiave di decifrazione* (d, n).

Si giustifichino le risposte date scrivendo tutti i calcoli intermedi.

Soluzione. Siccome $p = 7$, $q = 3$, allora $n = pq = 21$. Quindi $C = M^e \bmod n = 4^5 \bmod 21 = 1024 \bmod 21 = 16$.
La chiave di decifrazione è data da (d, n) dove $d = e^{-1} \bmod \Phi(n) = 5^{-1} \bmod 12 = 5$.
Infatti $\Phi(n) = (p-1)(q-1) = 6 * 2 = 12$.

5. ***Protocolli di Sicurezza***

Si consideri il seguente protocollo:

$$(1)\ B \to A : B$$
$$(2)\ A \to B : \{N_a\}_{K_{ab}}$$
$$(3)\ B \to A : \{f(N_a)\}_{K_{ab}}$$

dove K_{ab} è una chiave segreta condivisa da A e B, N_a è un numero generato con un generatore di numeri pseudo-casuali, e f è una funzione nota sia ad A che a B.

Si assuma che l'agente che esegue il protocollo impersonando il ruolo di A non possa eseguire lo stesso protocollo impersonando il ruolo di B. Ad esempio, B potrebbe essere il ruolo giocato dal telecomando della vostra macchina, mentre A quello giocato dal sistema di apertura istallato sulla vostra macchina.

Si discuta perchè la sicurezza del protocollo dipende dal numero di bit utilizzati per rappresentare il numero N_a.

Soluzione. Se la nonce non è sufficientemente lunga allora la probabilità che N_a assuma lo stesso valore in esecuzioni successive del protocollo non è trascurabile e un attacker non deve fare altro che osservare e memorizzare un certo numero di coppie di valori $\langle \{N_a\}_{K_{ab}}, \{f(N_a)\}_{K_{ab}} \rangle$ prodotte dal telecomando originale e quindi iniziare il protocollo (facendo finta di essere B, ovvero il telecomando) fino a che A (la macchina) invia un valore $\{N_a\}_{K_{ab}}$ già osservato in precedenza. (Come detto sopra, la probabilità che ciò avvenga è tutt'altro che trascurabile.) A questo punto l'attacker non deve fare altro che inviare il valore trasmesso in precedenza dal telecomando originale in risposta a quella specifica sequenza di bit.

6. ***Protocolli di Sicurezza***

Nel seguente protocollo K_A^1 e K_B^1 sono chiavi pubbliche già note, mentre K_A^2 e K_B^2 sono nuove chiavi pubbliche. N_A e N_B sono nonces.

$$\begin{array}{lll} \text{M1.} & A \to B: & \{K_A^2, A\}_{K_B^1} \\ \text{M2.} & B \to A: & \{N_B, K_B^2\}_{K_A^1} \\ \text{M3.} & A \to B: & \{N_B, N_A\}_{K_B^2} \\ \text{M4.} & B \to A: & \{N_A\}_{K_A^2} \end{array}$$

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

Soluzione. Nel passo (M1) A manda confidenzialmente la sua nuova chiave pubblica K_A^2 a B. Nel passo (M2) B manda confidenzialmente ad A la sua nuova chiave pubblica K_B^2 ed una nonce N_B per autenticare A. Nel passo (M3) A invia a B in modo confidenziale la nonce N_B ricevuta per autenticarsi e una nuova nonce N_A per autenticare B. Infine nel passo (M4) B rimanda la nonce N_A ad A in modo confidenziale per autenticarsi. Presumibilmente al termine del protocollo A e B si sono mutuamente autenticati; inoltre A dovrebbe essere certo che K_B^2 è una nuova chiave pubblica di B e B dovrebbe essere certo che K_A^2 è una nuova chiave pubblica di A.

(b) Si discuta se il protocollo garantisce o meno le proprietà di sicurezza descritte alla domanda precedente.

Soluzione. Non le garantisce, come mostrato dalla seguente traccia di esecuzione, al cui termine B è convinto che K_I^2 sia una chiave pubblica di A.

$$\begin{array}{lll} A \rightarrow & I & : \{K_A^2, A\}_{K_I^1} \\ & I \rightarrow B & : \{K_I^2, A\}_{K_B^1} \\ & I \leftarrow B & : \{N_B, K_B^2\}_{K_A^1} \\ A \leftarrow & I & : \{N_B, K_B^2\}_{K_A^1} \\ A \rightarrow & I & : \{N_B, N_A\}_{K_B^2} \\ & I \rightarrow B & : \{N_B, N_A\}_{K_B^2} \\ & I \leftarrow B & : \{N_A\}_{K_I^2} \\ A \leftarrow & I & : \{N_A\}_{K_A^2} \end{array}$$

(c) Nel caso sia stata individuata una vulnerabilità, si identifichi un raffinamento del protocollo che non soffre di tale vulnerabilità.

Soluzione.

$$\begin{array}{lll} \text{M1.} & A \rightarrow B: & \{K_A^2, A\}_{K_B^1} \\ \text{M2.} & B \rightarrow A: & \{N_B, K_B^2\}_{K_A^1} \\ \text{M3.} & A \rightarrow B: & \{B, N_B, N_A\}_{K_B^2} \\ \text{M4.} & B \rightarrow A: & \{N_A\}_{K_A^2} \end{array}$$

In questo modo B ha modo di verificare che il messaggio inviato al passo (M3) è effettivamente creato per lui.

Errata Corrige: La modifica proposta non risolve il problema. Infatti anche questa variante del protocollo soffre dello stesso tipo di attacco.
Il problema risiede nel messaggio $\{K_A^2, A\}_{K_B^1}$ inviato da A a B al passo M1. Questo messaggio asserisce implicitamente che K_A^2 è una nuova chiave pubblica di A, ma B non ha modo di verificare l'autenticità di tale informazione. La soluzione è cambiare tale messaggio in $\{K_A^2, A\}_{inv(K_A^1)}$, dove $inv(K_A^1)$ è la chiave privata corrispondente alla chiave pubblica K_A^1. Questo messaggio è a tutti gli effetti un certificato emesso da A (usando la chiave pubblica K_A^1) in cui viene asserito che K_A^2 è una (nuova) chiave pubblica di A.

7. ***Controllo degli Accessi***

Si consideri il modello MAC di Bell-La Padula e si indichino i permessi concessi ad un utente con security label *(secret, {red, green, blue}) relativamente a documenti classificati nel seguente modo:*

1. *(top secret, {red}):*
2. *(secret, {red}):*
3. *(secret, {red, black}):*
4. *(secret, {white}):*
5. *(confidential, {red, blue, green}):*
6. *(confidential, {white}):*
7. *(top secret, {red, green, blue, black}):*

Soluzione. Ricordiamo che (r_2, c_2) domina (r_1, c_1) (in simboli, $(r_1, c_1) \leq (r_2, c_2)$ se e solo se $r_1 \leq r_2 \wedge c_1 \subseteq c_2$ e che gli accessi nel modello di Bell-LaPadula sono governati dai seguenti due principi:

- **No Read-Up** (detta anche **Simple Security Property**): Un subject con security label x_s può leggere informazione relativa ad una risorsa con security label x_o solo se x_s domina x_o.
- **No Write-Down** (detta anche ***-Property**): Un subject con security label x_s può scrivere informazione su un oggetto con security label x_o solo se x_o domina x_s.

Dunque le risposte sono:

1. (top secret, {red}): Nessun diritto.
2. (secret, {red}): L'utente ha diritto di lettura ma non di scrittura.
3. (secret, {red, black}): Nessun diritto.
4. (secret, {white}): Nessun diritto.
5. (confidential, {red, blue, green}): L'utente ha diritto di lettura ma non di scrittura.
6. (confidential, {white}): Nessun diritto.
7. (top secret, {red, green, blue, black}): L'utente ha diritto di scrittura ma non di lettura.

8. ***Controllo degli accessi***

Si consideri il seguente insieme di diritti {R,W,X,A,L,M,Own}*, dove* A*,* L *e* M *stanno per* Append, List *e* Modify.

(a) Utilizzando la sintassi del modello di Harrison-Ruzzo-Ullman, si scriva il comando per revoke_all_rights(s_1, s_2, o)*, con il quale* s_1 *cancella tutti i diritti di* s_2 *sulla risorsa* o*:*

Soluzione.

command revoke_all_rights(s_1, s_2, o)
  delete R,W,X,A,L,M,Own **from** $M(s2, o)$
end

(b) Si indichi come deve essere modificato il comando affichè la cancellazione avvenga solo se s_1 ha il diritto {Modify} *su o.*

Soluzione.

command revoke_all_rights(s_1, s_2, o)
  if Modify $\in M(s1, o)$
  delete R,W,X,A,L,M,Own **from** $M(s2, o)$
end

(c) Si indichi come deve essere modificato il comando affichè la cancellazione avvenga solo se s_1 ha il diritto {Modify} *su o e s_2 ha il diritto* {Own} *su o.*

Soluzione.

command revoke_all_rights(s_1, s_2, o)
  if Modify $\in M(s1, o)$ **and** Own $\in M(s2, o)$
  delete R,W,X,A,L,M,Own **from** $M(s2, o)$
end