COMPUTER SECURITY

Corso di Laurea Magistrale in Ingegneria Informatica
Prof. Alessandro Armando

8 luglio 2008

Attenzione: Si risponda alle domande utilizzando lo spazio apposito.
Tempo per lo svolgimento: 2 ore.

Nome e Cognome: _____

Matricola: _____

---

1. ***Crittografia I***

   Si consideri lo schema crittografico che trasforma ciascun carattere x del plaintext nel carattere $E(x)$ dato dalla seguente formula:

   $$E(x) = ((a * x + b) \bmod m)$$

   dove

   - *l'alfabeto ha m lettere e la prima lettera dell'alfabeto è rappresentata dal numero 0, il secondo dal numero 1, ... e l'ultimo dal numero $m - 1$;*
   - *a e b sono numeri interi che fungono da chiave di cifratura e a è relativamente primo con m;*
   - *$(y \bmod z)$ indica il resto della divisione intera tra y e z.*

   (a) Si dimostri che l'algoritmo di decifratura è dato da

   $$D(x) = a^{-1}(x - b) \bmod m$$

   dove a^{-1} è un inverso moltiplicativo di a, ovvero $aa^{-1} = 1 \bmod m$.

   Soluzione.

   $$\begin{aligned} \mathrm{D}(\mathrm{E}(x)) &= a^{-1}(\mathrm{E}(x) - b) \mod m \\ &= a^{-1}(((ax + b) \mod m) - b) \mod m \\ &= a^{-1}(ax + b - b) \mod m \\ &= a^{-1}ax \mod m \\ &= x \mod m. \end{aligned}$$

(b) Si discuta la sicurezza dello schema crittografico.

Soluzione. È facile vedere che lo schema crittografico proposto (noto come *affine encryption*) è vulnerabile a crittoanalisi basata su analisi di frequenza.

2. ***Crittografia II***

La funzione one-way utilizzata in UNIX per calcolare e memorizzare l'hash delle password degli utenti è derivata dall'algoritmo di cifratura DES modificato in modo tale che non esiste una chiave di cifratura che consenta di calcolare PW a partire dall'hash code $h(PW)$.

(a) *Inoltre l'algoritmo è stato deliberatamente modificato in modo tale da essere molto più lento di DES. Perchè?*

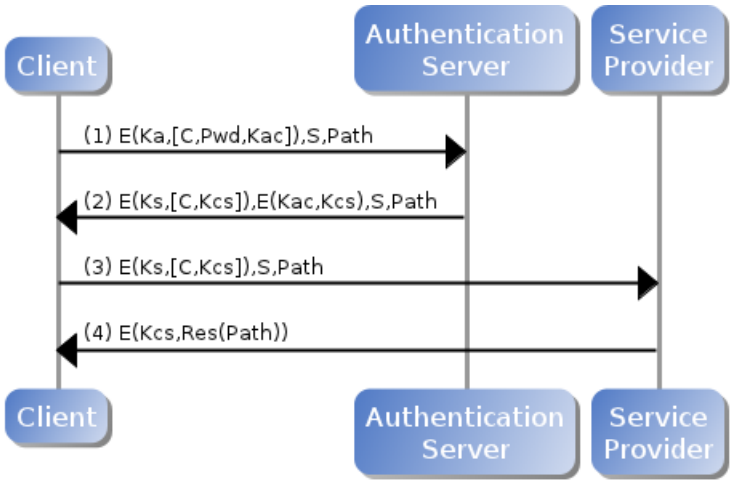
Soluzione. Per rendere più lenti (e quindi più difficili) password guessing attacks.

(b) *Spesso, il file delle password invece di memorizzare coppie della forma $\langle U, h(PW_U)\rangle$ dove PW_U è la password dell'utente U, memorizza triple della forma $\langle U, R_U, h(PW_U \| R_U)\rangle$, dove R_U è un numero generato in modo (pseudo)random e $\|$ denota concatenazione. Perché?*

Soluzione. Anche in questa soluzione serve a rendere più lenti (è quindi più difficili) password guessing attacks.

3. ***Protocolli di Sicurezza***

Si consideri il seguente protocollo dove un client C vuole accedere ad una risorsa protetta che risiede su un server S la cui posizione all'interno di S è identificata da Path. Prima di accedere alla risorsa, il client C deve autenticarsi presso un Authentication Server (A) per farsi rilasciare un'asserzione di autenticazione (ovvero il messagggio E(Ks,[C,Kcs])) che poi presenterà al Service Provider assieme alla richiesta della risorsa.



Al passo (1) C invia ad A le proprie credenziali (C, Pwd) ed una nuova chiave di sessione Kac tra C ed A, il tutto cifrato con Ka, la chiave pubblica di A; a tale messaggio vengono inoltre aggiunte S e Path.

Se le credenziali sono corrette, allora al passo (2) A invia a C l'asserzione di autenticazione E(Ks,[C,Kcs])), dove Kcs è una nuova chiave di sessione tra C e S e Ks è la chiave pubblica di S; tale asserzione è accompagnata Kcs cifrata con Kac oltre che da S e Path.

Al passo (3), C non fa altro che inviare a S l'asserzione di autenticazione ricevuta da A aggiungendo in coda S e Path.

Infine, al passo (4), S invia Res(Path) (ovvero la risorsa identificata da Path) a C cifrandola con la chiave di sessione Kcs.

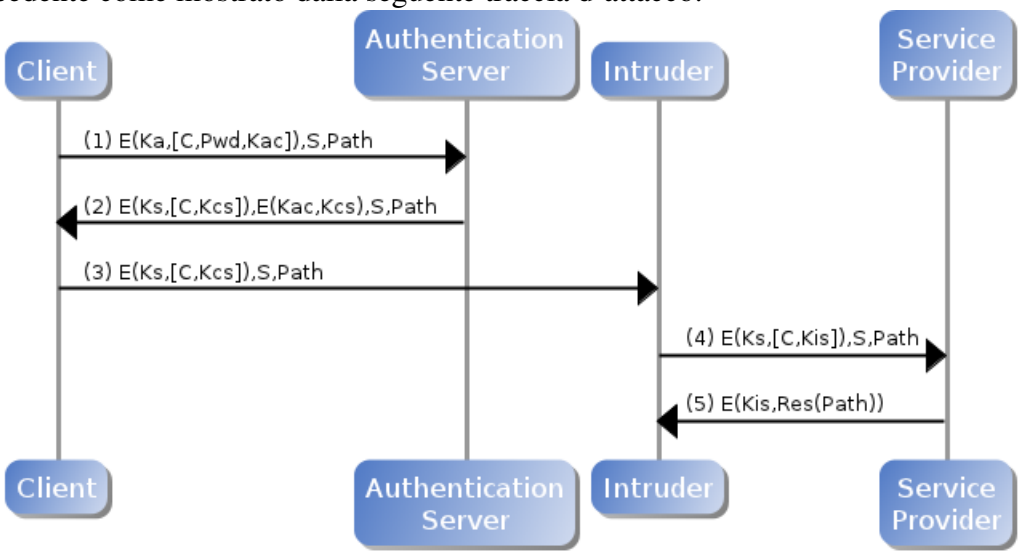
(a) *Quali sono le proprietà di sicurezza che dovrebbe garantire un protocollo di questo tipo?*

Soluzione.

1. Sia A che S devono autenticare C, ovvero se A ed S completano la loro parte del protocollo C deve aver iniziato la propria con gli stessi valori di S e Path.
2. C deve autenticare S, ovvero se C completa la sua parte del protocollo S deve aver iniziato la sua con lo stesso valore di Path.
3. La risorsa $Res(Path)$ deve rimanere segreta.
4. Si potrebbe anche richiedere al segretezza di $Path$ per garantire ad esempio la privacy del client, ma questa proprietà è chiaramente violata in quanto Path è inviato in chiaro.

(b) Si discuta la sicurezza del protocollo.

Soluzione. Il protocollo non garantisce nessuna delle proprietà indicate al punto precedente come mostrato dalla seguente traccia d'attacco:



4. ***Controllo degli Accessi***

Si consideri la politica di controllo degli accessi RBAC presentata a lezione:

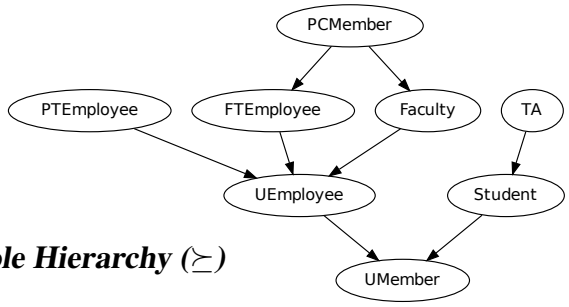
User Assignment (UA)

User	***Role***
Alice	*PCMember*
Bob	*Faculty*
Charlie	*Faculty*
David	*TA*
David	*Student*
Eve	*UEmployee*
Fred	*Student*
Greg	*UMember*

Permission Assignment (PA)

Role	***Permission***
PCMember	*GrantTenure*
Faculty	*AssignGrades*
TA	*AssignHWScores*
UEmployee	*ReceiveHBenefits*
Student	*Register4Courses*
UMember	*UseGym*

Role Hierarchy ($\succeq$)



e la politica ARBAC (sempre presentata a lezione):

- *can_assign1:* $UEmployee : \{Student, \neg TA\} \Longrightarrow PTEmployee$
- *can_assign2:* $UEmployee : \{UEmployee, \neg Faculty\} \Longrightarrow Student$
- *can_revoke:* $UEmployee : \{Faculty\} \Longrightarrow \neg Faculty$

(a) È possibile trasformare la politica in modo tale che David acquisisca il permesso AssignGrades? Giustificare la risposta data.

Soluzione. No, non è possibile assegnare il ruolo Faculty a David.

(b) È possibile trasformare la politica in modo tale che Eve acquisisca il permesso Register4Courses? Giustificare la risposta data.

Soluzione. Si. È possibile assegnare il ruolo Student ad Eve applicando can_assign2.

(c) È possibile trasformare la politica in modo tale che Charlie acquisisca il permesso Register4Courses? Giustificare la risposta data.

Soluzione. ~~No. Essendo Faculty, non puo' acquisire il ruolo Student.~~
Errata Corrige [2/1/2019]: Si, applicando can_revoke e poi can_assign2.