Esercizi

Numero di trasposizioni e sostituzioni nei cifrari
- Malleabilità di una funzione crittografica

RSA key distribution 🔑

Diffie-Hellman key distribution🔑
- Struttura trasmettitore

Esercizi Crittografia
- Altro foglio di esercizi
  - Proprietà dell’autenticazione dei messaggi
  - RSA algoritmo di encrypt e decrypt
- Needam shroder

Esercizi Controllo degli accessi
- **Bell LaPadula**

Secure Programming

Numero di trasposizioni e sostituzioni nei cifrari

Il numero di trasposizioni possibili, nei transposition cyphers è dato dal fattoriale del numero di caratteri della nostra stringa

EX. stringa di 4bit

4!

Il numero di sostituzioni possibili nei substitution cyphers è dato dal fattoriale numero di caratteri nell'alfabeto

EX. alfabeto inglese di 26 lettere

26!

Il numero di trasformazioni possibili in un block cypher è dato dal fattoriale del numero di caratteri nell'alfabeto elevato alla lunghezza della stringa.

EX. per stringhe di 4 bit in un block cypher

(2^4)!

Malleabilità di una funzione crittografica

Una funzione crittografica si dice malleabile solo se esistono due funzioni F e G tali che :

F(E(K, M)) = E(K, G(M))

RSA key distribution 🔑

Scelta k randomica il messaggio cifrato si ottiene così:

$$c = (k^e \ mod \ n, \ E_k(m))$$

per decifrare il messaggio invece si divide c in c1 e c2 così che:

$$k = {c_1}^d \ mod \ n$$
$$m = D_k(c_2)$$

Diffie-Hellman key distribution🔑

$Y_A = \alpha^{X_A} mod\ q$ $Y_B = \alpha^{X_B} mod\ q$

$K_A = Y_b^{X_A} mod\ q$ $K_B = Y_A^{X_B} mod\ q$

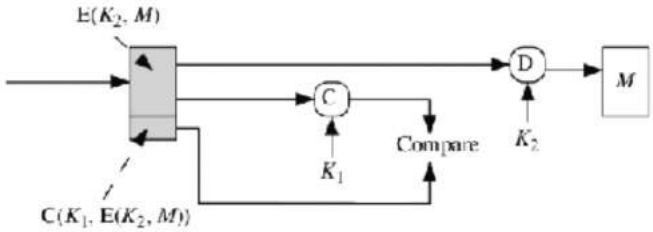
$X_A = \alpha^{Y_A} mod\ q$ $X_B = \alpha^{Y_B} mod\ q$

> dove
> Y: chiave pubblica
> X: chiave privata
> K: chiave segreta, K_A = K_B
> q: numero primo, è informazione pubblica
> α: radice primitiva di q, è informazione pubblica

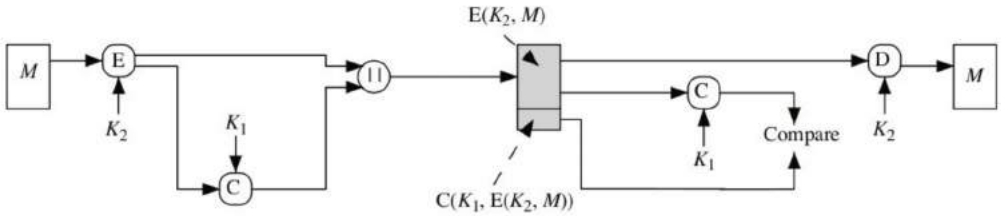
Struttura trasmettitore

4. ***Crittografia II***

Si completi il seguente schema crittografico disegnandone il trasmettitore.



Soluzione.



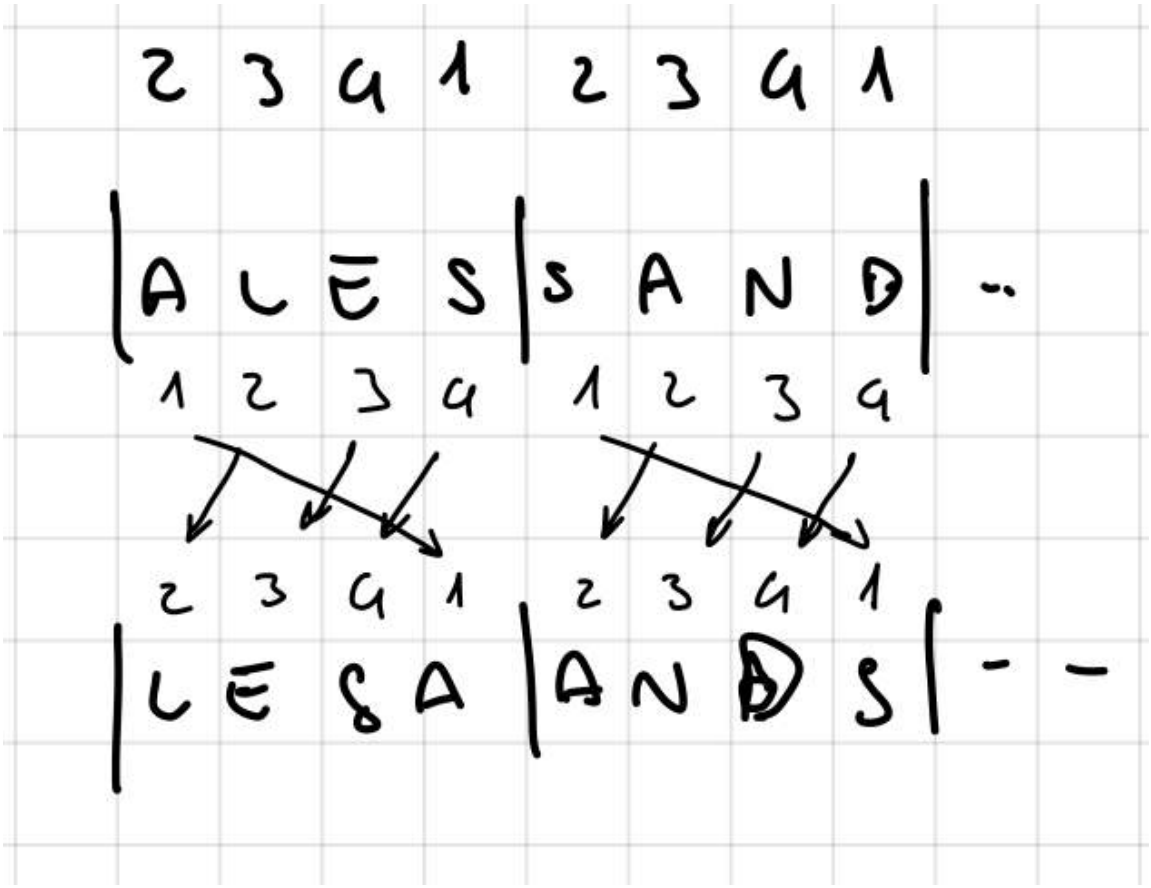
Esercizi Crittografia

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


2 3 4 1 2 3 4 1
| A L E S | S A N D | ..
1 2 3 4 1 2 3 4
2 3 4 1 2 3 4 1
| L E S A | A N D S | - -

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.

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

In questo caso andiamo per sostituzione. Per dimostrare che una chiave di decrypt proposta sia effettivamente quella corretta bisogna provare ad appliacare la chiave di decrypt al testo cifrato e vedere se si ottiene l'elemento di partenza.

Discutere la sicurezza dello schema crittografico:

🍊 **Affine cypher**

Questo è un affine cypher,

Tuttavia gli spazi delle chiavi sono molto grandi di solito e pertanto questi cifrari possono essere "rotti" facilmente con **strumenti di analisi di frequenza** (analizzano quali lettere compaiono con maggiore frequenza nella lingua presa in considerazione e attraverso tentativi si ottiene il plaintext).

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

DES and SP networks

(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 questa soluzione serve a rendere più lenti (e quindi più difficili) password guessing attack

---

Altro foglio di esercizi

Un sistema crittografico a chiave multipla è caratterizzato da un insieme di n chiavi $\mathcal{K} = \{K_1, \ldots, K_n\}$ tali che se se $C_0 = P$ è un generico plaintext e $C_{i+1} = E(C_i, K_i)$ per $i = 0, \ldots, n-1$, allora $C_n = P$. Ovvero cifrando P con tutte le chiavi $K_1, \ldots, K_n$ (in qualunque ordine) si ottiene il plaintext di partenza.

(a) Un sistema crittografico a chiave multipla con $n = 2$ corrisponde ad uno dei sistemi crittografici visti a lezione. Quale? Si giustifichi la risposta data.

▼ Soluzione A

Soluzione. Uno schema crittografico a chiave a chiave pubblica (ad esempio RSA), dove $\mathcal{K}$ è dato dalla chiave pubblica e dalla chiave privata.

Si discutano i possibili utilizzi di un sistema crittografico a chiave multipla con $n > 2$.

▼ Soluzione B

Soluzione. Un primo possibile utilizzo è per la firma digitale congiunta tra due o più agenti. Ad esempio se $\mathcal{K} = \{K_1, \ldots, K_n\}$ e K_i è privata per l'agente A_i (per $i = 1, \ldots, n-1$) e K_n è pubblica, allora A_1 può firmare digitalmente un documento M cifrandone un hash con K_1 e mandando il risultato ad A_2, A_2 cifra quanto ricevuto da A_1 con la propria chiave privata K_2 e invia il risultato ad A_3, e cosìvia fino a A_{n-1} che produce

$$E(K_{n-1}, E(K_{n-2}, \ldots E(K_2, E(K_1, H(M))) \ldots)) \quad (1)$$

La firma può essere verificata cifrando (1) con K_n. Si osservi che la sottoscrizione di un singolo agente è verificabile solo quando è verificabile la sottoscrizione dei co-signatari. Dualmente, sotto le stesse ipotesi sulla distribuzione delle chiavi, cifrando con K_n si ottiene confidenzialità nei confronti $A_1, \ldots A_{n-1}$.

2. ***Crittografia a Chiave Pubblica***

   (a) *Quali delle seguenti attività sono svolte da una smart cart?*

   - A. *memorizzare il certificato digitale del possessore*
   - B. *firmare i documenti utilizzando la chiave pubblica del possessore*
   - ***C. firmare i documenti utilizzando la chiave privata del possessore***
   - D. *verificare la firma digitale dei documenti utilizzando la chiave pubblica del possessore*

   (b) *Quali delle seguenti informazioni devono essere necessariamente presenti in un certificato digitale?*

   - ***A. Identità del possessore del certificato***
   - ***B. Identità dell'Autorità di Certificazione che ha prodotto il certificato***
   - C. *Chiave privata del possessore del certificato*
   - ***D. Firma digitale del certificato stesso prodotta dall'autorità di certificazione***
   - E. *Chiave pubblica dell'autorità di certificazione*
   - F. *Chiave privata dell'autorità di certificazione*
   - ***G. Chiave pubblica del possessore del certificato***

   (c) *Per firmare digitalmente un documento è necessario essere connessi alla rete? Giustificare la risposta data.*

▼ SOLUZIONE

No basta la smartcard

   (d) *Per verificare la firma digitale di un documento è necessario essere connessi alla rete? Giustificare la risposta data.*

▼ SOLUZIONE

Non è strettamente necessario se di dispone del certificato digitale di colui che ha firmato il documento e di una Certificate Revocation List recente

3. ***Digital Signatures***

Which of the following sentences are true?

A. *The private key of a user must be generated by the certification authority and is given to the user together with the digital certificate of the corresponding public key.*

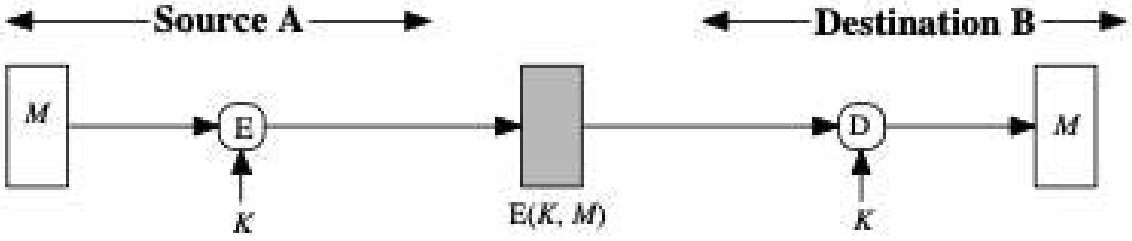
B. *A smartcard used for digital signatures stores the private key of the owner.*

C. *Smartcards play a crucial role in the validation of digital signatures*

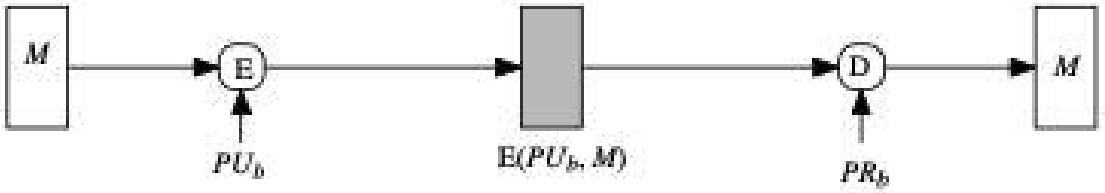
D. *Smartcards play a crucial role in the generation of digital signatures*

E. *Smartcards play a crucial role in the storage of digital signatures*

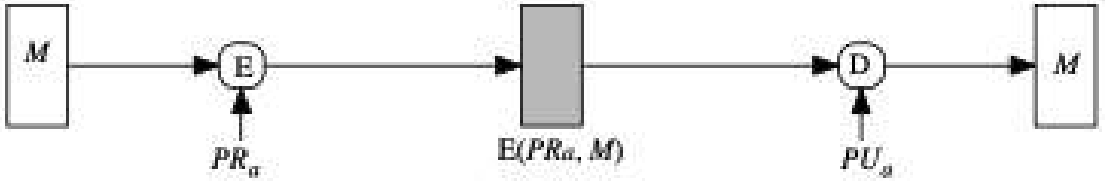
Proprietà dell’autenticazione dei messaggi



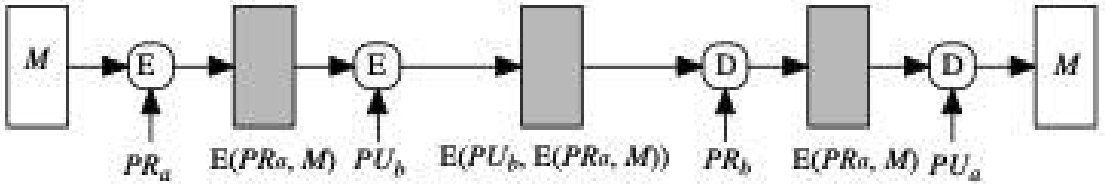
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



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



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

- **Non repudiation**: il messaggio è garantito che venga da una certa persona, (lo9 firma con la sua chiave privata) e lui non può negare di aver effettivamente mandato quel messaggio
- **Authentication**: il messaggio viene mandato tra due persone e l'altra persona sa effettivamente che il messaggio è stato mandato dall'altra persona

- **Confidentiality:** il messaggio rimane segreto e può essere letto solamente dal mittente e dal ricevente

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

Il pc prende il documento, ne crea l'hash, lo fa autenticare dalla smartcard (che può essere l'impronta digitale o qualsiasi altra cosa) e poi lo invia.

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.

RSA algoritmo di encrypt e decrypt

TESTO CIFRATO $\Rightarrow C = M^e \mod (p \cdot q)$

COME SI CALCOLA $d \Rightarrow e \cdot d \mod \phi(n) = 1$

TESTO DECIFRATO $\Rightarrow C^d \mod n = M$

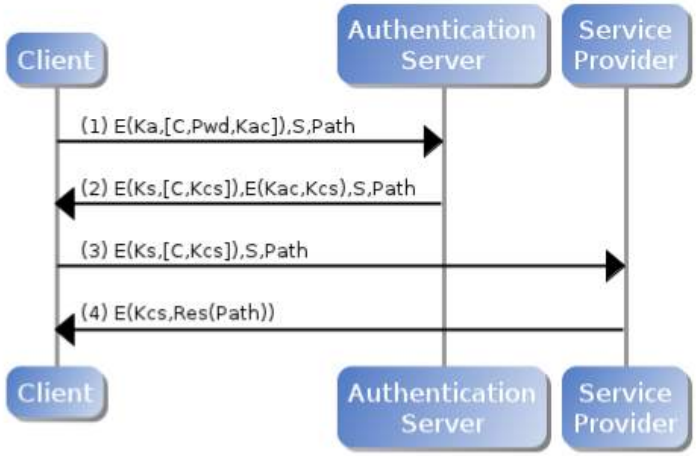
n è p*q

phi(n) = (p-1)(q-1)

Esercizi controlli di sicurezza

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 a

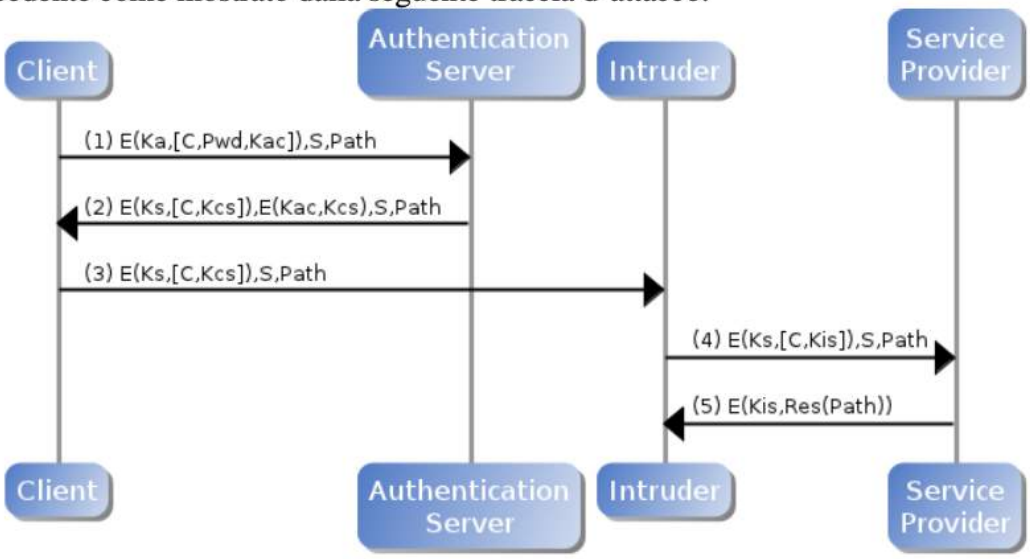
Soluzione.

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 b

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



4. ***Security Protocols***

Suppose Alice wants to send her Bank a message that includes her promise to pay Charlie $50 dollars. Alice and the Bank have a shared secret X. Alice initiates a conversation with the Bank by sending: $A\|B\|n$ (Alice's identity, the Bank's identity, and a nonce).

(a) *Specify a valid reply for the Bank (i.e., a message generated by the Bank to be sent to Alice) that would enable Alice to verify that the reply came from someone who knows the secret X.*

▼ SOLUZIONE

Solution:

$$B \rightarrow A : A\|B\|n\|HMAC(X, A\|B\|n)$$

(b) *In order to setup a secure communication, Alice and the Bank need a secret session key. Suppose that the Bank chooses a session key K by XORing some pseudorandom data with X and includes it with the reply in step* (a) *above. Extend message* (a) *to provide the session key to Alice securely as well.*

▼ SOLUZIONE

Solution:

$$B \rightarrow A : A \| B \| n \| E(X, K) \| HMAC(X, A \| B \| n \| K)$$

(c) Now, Alice can submit her message ("Pay Charlie $50 from my account") to the Bank. Write the message in such a way that Charlie cannot replay it (Hint: you will need to add something to the message that the Bank is capable of checking to prevent replay.).

▼ SOLUZIONE

Solution: *Call M the message and include c a counter for the messages in the session. Then*

$$A \rightarrow B : A \| E(K, M \| c) \| HMAC(K, A \| M \| c)$$

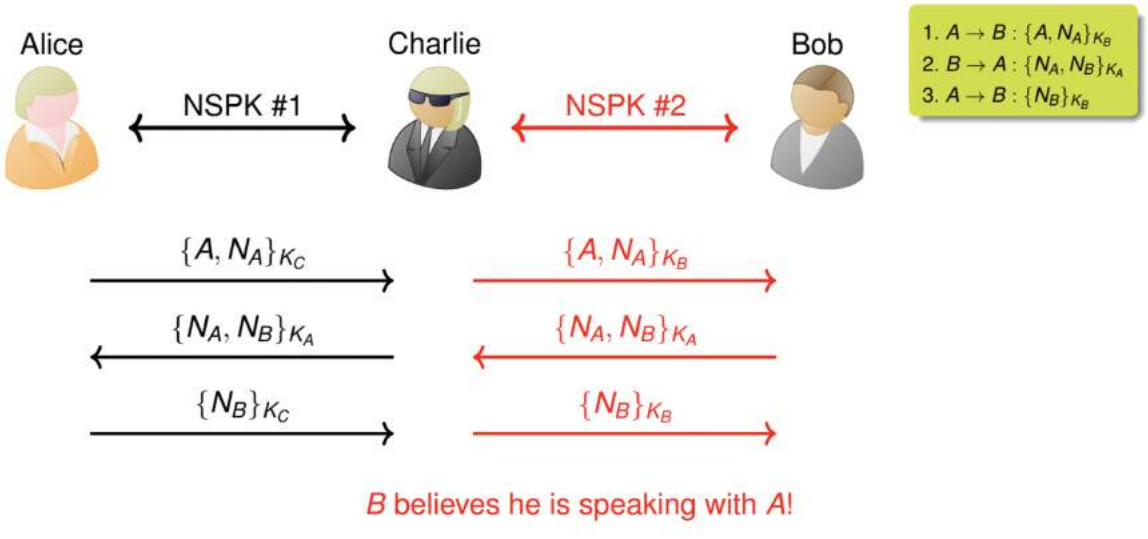
protects M's secrecy, ensures that Charlie cannot replay, and protects the integrity of the message.

Needam shroder

A manda messaggio a B ma soffre di man in the middle.

A pensa che C sia B percioò cripta i sui messaggi con la chiave pubblica di C.

Nel momento in cui B manda il suo identificativo cade l'asino perchè A confronta l'id di C con Quello di B e si accorge di parlare in verità con due persone



Si consideri il seguente protocollo:

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

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

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.

Esercizi Controllo degli accessi

4. ***Controllo degli Accessi***

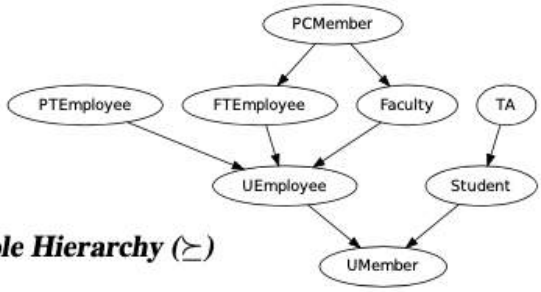
Si consideri la politica di controllo degli accessi RBAC presentata a lezione:

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.

▼ SPOILER Soluzione

NO non è possibile assegnare il ruolo Faculty a David. poor david :(

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

▼ SPOILER Soluzione

SI è possibile assegnare il ruolo di 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.

▼ SPOILER Soluzione

SI applicando `can_revoke` e poi `can_assign2`

---

6. ***Controllo degli Accessi*** *Si consideri un sistema con tre utenti: Alice, Bob e Charlie. Alice possiede il file* `alice.bat`*, Bob può solo leggerlo e scriverlo, mentre Charlie può solo eseguirlo. Charlie può solo leggere il file* `bob.bat`*, che è posseduto da Bob, mentre Alice lo può solo leggere e scrivere. Charlie possiede il file* `charlie.bat`*; Alice lo può solo scrivere e Bob può solo eseguirlo. Ogni file può essere letto, scritto ed eseguito dagli utenti che lo posseggono.*

   (a) Si scriva la matrice di controllo degli accessi corrispondente a tale situazione.

▼ SOLUZIONE

Soluzione.

	`alice.bat`	`bob.bat`	`charlie.bat`
Alice	*rwx*	*rw*	*w*
Bob	*rw*	*rwx*	*x*
Charlie	*x*	*r*	*rwx*

   (b) Si scriva la matrice di controllo degli accessi che si ottiene se Charlie dà ad Alice il permesso di leggere `charlie.bat` *e Alice revoca a Bob il permesso di scrivere* `alice.bat`*.*

▼ SOLUZIONE

Soluzione.

	`alice.bat`	`bob.bat`	`charlie.bat`
Alice	*rwx*	*rw*	*wr*
Bob	*r*	*rwx*	*x*
Charlie	*x*	*r*	*rwx*

R/W/X/OWN	alice.bat	bob.bat	charlie.bat
Alice	OWN	R/W	W
Bob	R/W	OWN	X
Charlie	X	R	OWN

R/W/X/OWN	alice.bat	bob.bat	charlie.bat
Alice	OWN	R/W	RW
Bob	**R**	OWN	**X**
Charlie	X	R	OWN

Controllo degli accessi

Trasferisci permesso di scrittura da s1 a s2 se lo ha (s1 lo perde dopo averlo passato)

```
command transfer.write(s1, s2, o)
  if W in M(s1, o)
  then enter W into M(s2, o)
       delete W from M(s1,o)
end
```

Bell LaPadula

lettura : **xo** ≤ **xs** e se **top secret** ≤ **secret** && **design** è contenuto in **personnel design assistance**

scrittura : **xs** ≤ **xo** e se **top secret** ≥ **secret** && **personnel design assistance** è contenuto in **design**

ALTO	BASSO
no write down	no read up
can read up	can write up

> s è soggetto (user)
> o è l'oggetto (file)

Secure Programming

5. ***Secure Programming***

Consider the program below.

```
void example(char *s) {
 char array[1024];
 strcpy(array,s);
}

int main(int argc, char **argv) {
 example(argv[1]);
}
```

What threat exists in the above program? How would you resolve it?