Sicurezza nei Sistemi Distribuiti

Aspetti di Sicurezza

- La sicurezza nei sistemi distribuiti deve riguardare tutti i componenti del sistema e coinvolge due aspetti principali:
  - Le comunicazioni tra utenti e processi
    - » *soluzione* : **canali sicuri**
  - Autorizzazione di utenti e processi
    - » *soluzione* : **controllo degli accessi**
- Meccanismi : **chiavi crittografiche** e **rimozione di utenti**.

Minacce alla Sicurezza

- Intercettazione

  (accessi non autorizzati)
- Interruzione

  (diniego di servizio - denial of service)
- Modifica

  (modifiche di dati non autorizzate)
- Fabbricazione

  (inserimenti di dati non autorizzati)

Politica di Sicurezza

- Un sistema distribuito sicuro ha bisogno di una

  politica di sicurezza

  che definisce

  le azioni che le entità del sistema possono eseguire e quelle che sono proibite.

- Una politica può essere realizzata tramite **meccanismi di sicurezza**.

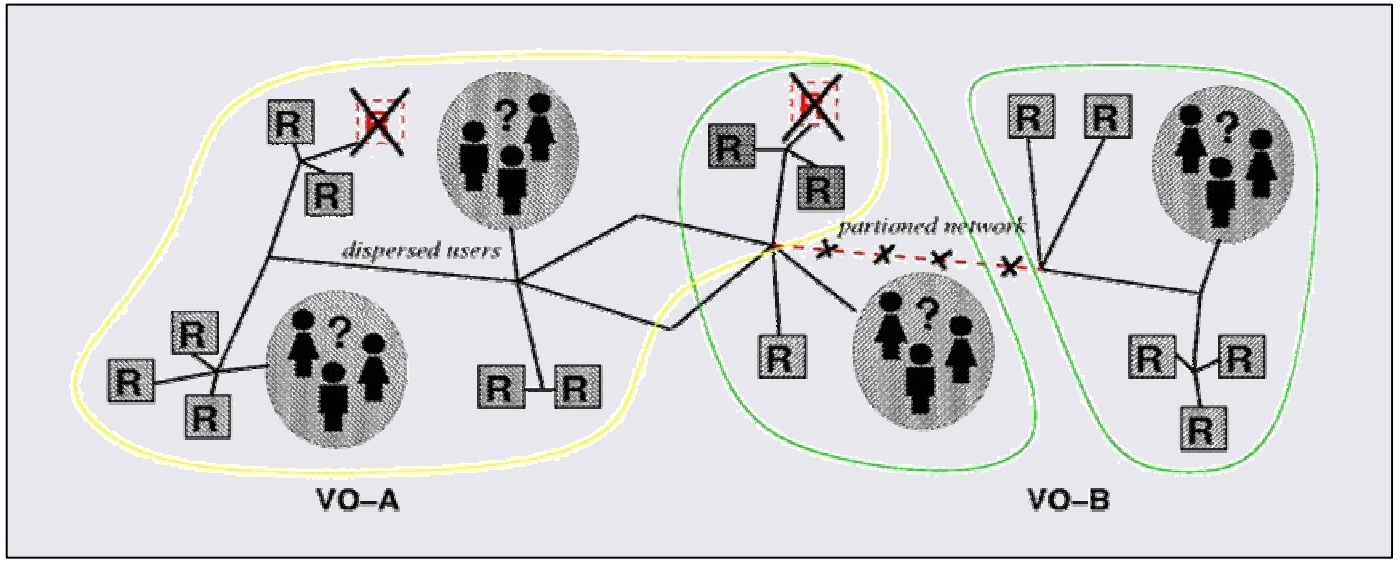
Meccanismi di Sicurezza

- Crittografia
- Autenticazione
- Autorizzazione
- Auditing

Esempio: Il Globus Toolkit

Una Grid è una infrastruttura di distributed computing. Il suo principale obiettivo è : *Condivisione di risorse & problem solving coordinato in organizzazioni virtuali dinamiche e multi-istitutionali*.

Globus Toolkit è un sistema per configurare e usare le Grid.



Domini Multipli in una Grid

Esempio: Politica di Sicurezza di Globus

Le politiche di sicurezza di Globus sono basate sui seguenti principi.

1. **Il sistema consiste di diversi domini amministrativi.**
2. **Le operazioni locali sono soggette solo a politiche di sicurezza locali.**
3. **Le operazioni globali richiedono che il richiedente sia riconosciuto in tutti i domini interessati alle operazioni.**
4. **Le operazioni tra entità in domini differenti richiedono la mutua autenticazione.**
5. **L'autenticazione globale sostituisce la autenticazione locale.**
6. **Il controllo degli accessi alle risorse è soggetto alle politiche locali.**
7. **Gli utenti possono delegare loro diritti ai processi.**
8. **Un gruppo di processi nello stesso dominio può condividere le credenziali.**

Esempio: Globus Security Architecture (1)

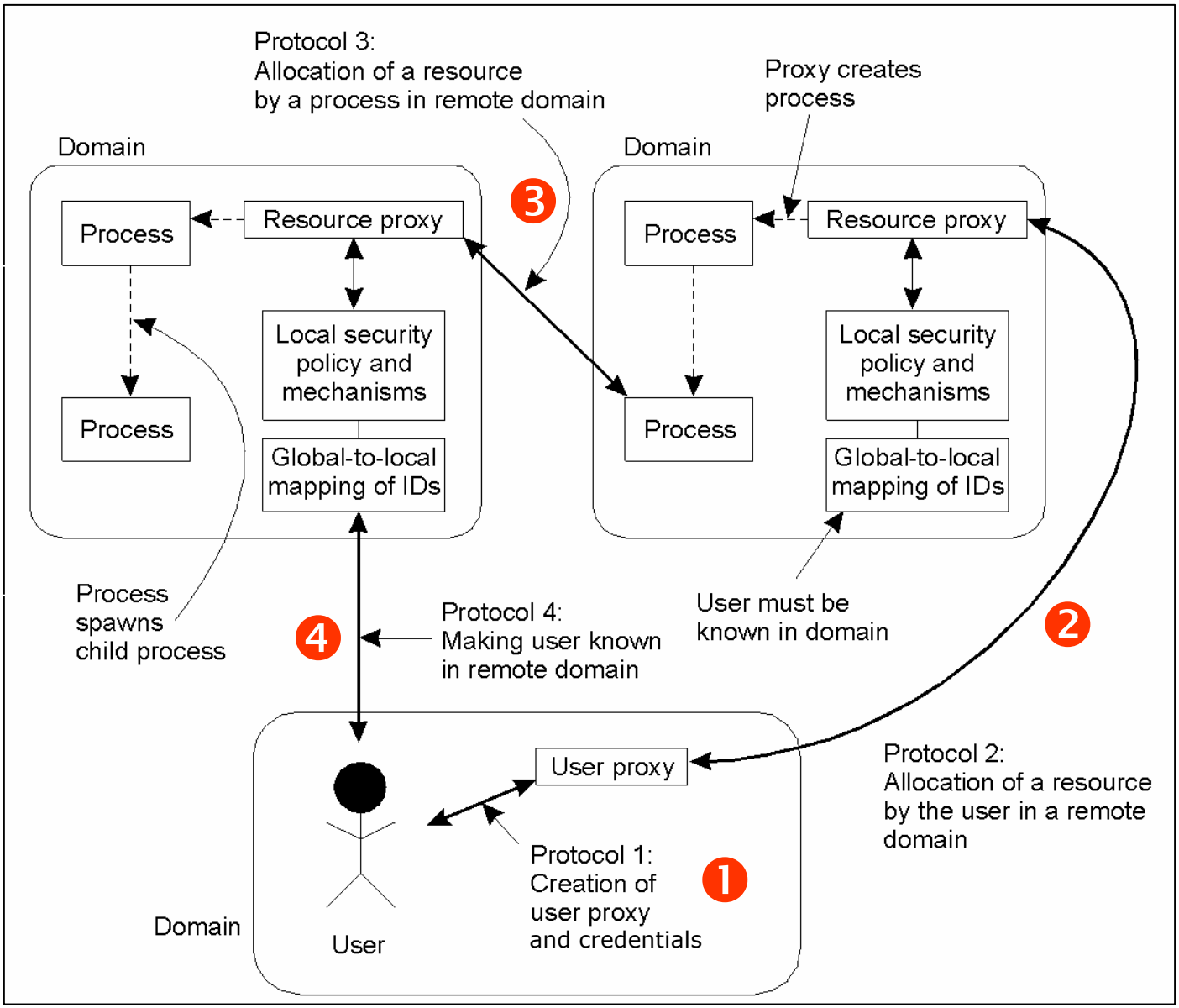
- La *Globus security architecture* consiste di entità come utenti, processi, user proxies, e resource proxies.

- Uno **user proxy** è un processo che ha il permesso di agire per conto di un utente per un limitato periodo di tempo.

- Un **resource proxy** è un processo che in un dominio è usato per tradurre operazioni globali su una risorsa in operazioni locali che rispettano la politica di sicurezza del dominio.

Esempio: Globus Security Architecture (2)

Diagramma della *Globus security architecture* e dei protocolli.



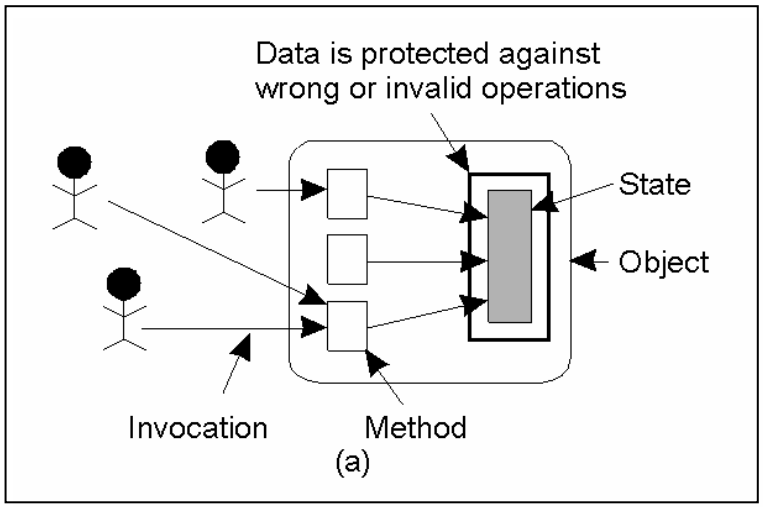
Aspetti di Progettazione della Sicurezza

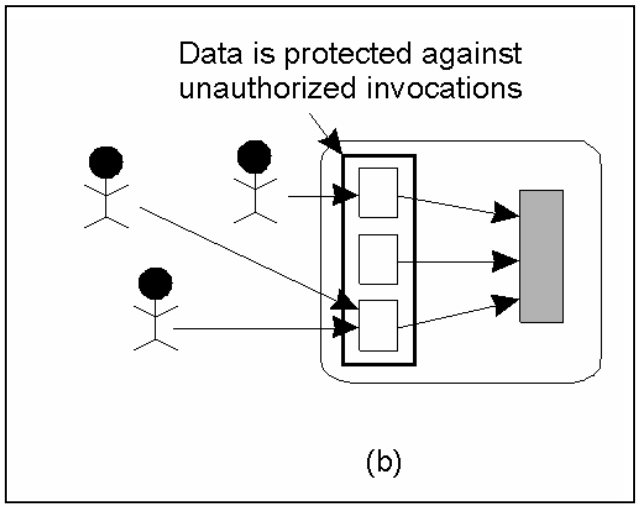
- Politiche di sicurezza possono essere implementate da servizi di sicurezza.

- Diversi aspetti devono essere considerati nella progettazione di politiche di sicurezza:
  - **focus del controllo,**
  - **meccanismi e livelli,**
  - **semplicità.**

Focus del Controllo

Tre approcci per la protezione da minacce alla sicurezza.

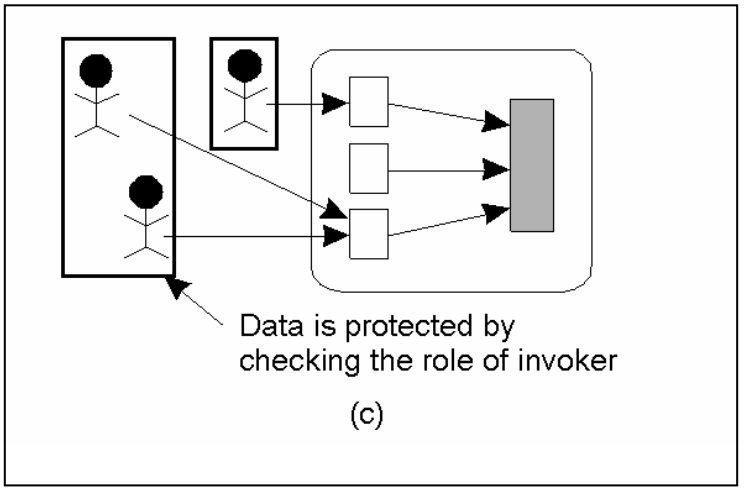




(a) Protezione da operazioni non valide

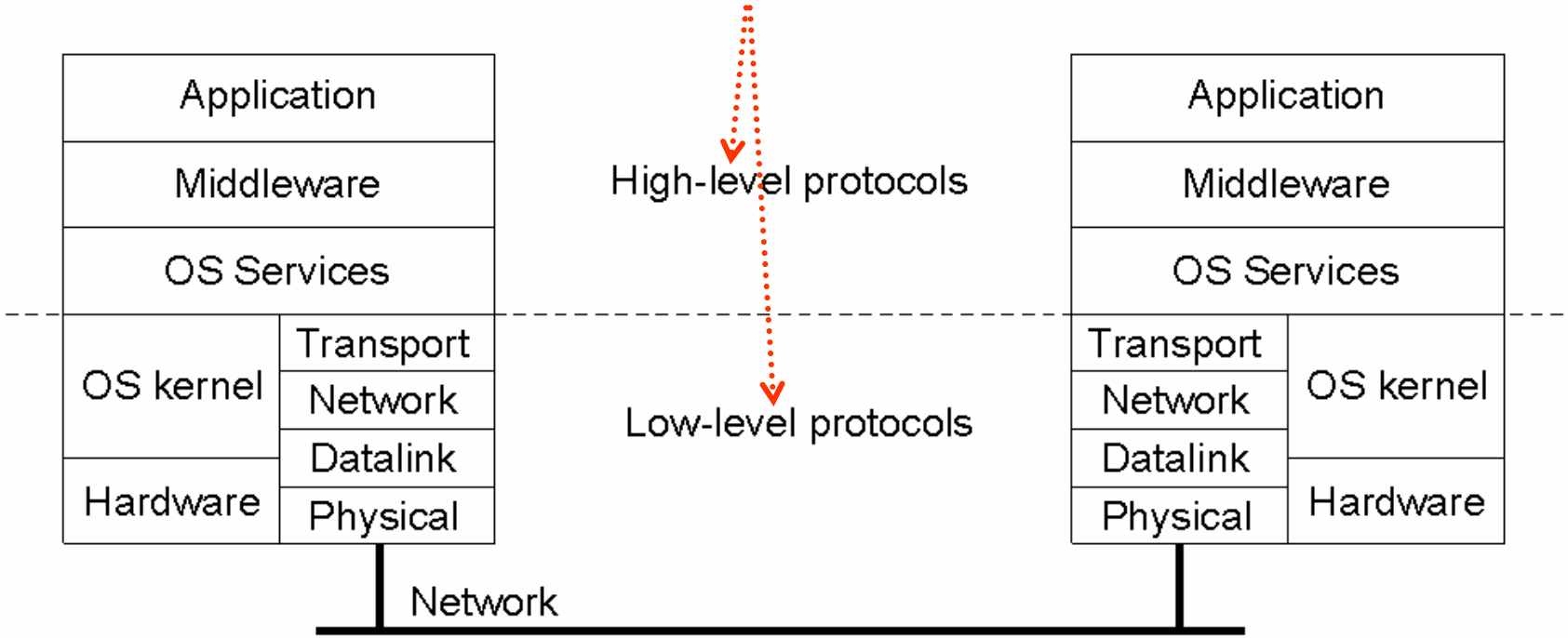
(b) Protezione da invocazioni non autorizzate

(c) Protezione da utenti non autorizzati



Livelli dei Meccanismi di Sicurezza

A quale livello i meccanismi di sicurezza devono essere posti?



NB: I meccanismi di sicurezza nei sistemi distribuiti sono generalmente posti a livello del middleware.

Distribuzione dei Meccanismi di Sicurezza (1)

- Dipendenze tra servizi di sicurezza portano al concetto di

  Trusted Computing Base

  l'insieme di tutti i meccanismi di sicurezza in un sistema distribuito che sono necessari per rispettare la sicurezza del sistema.

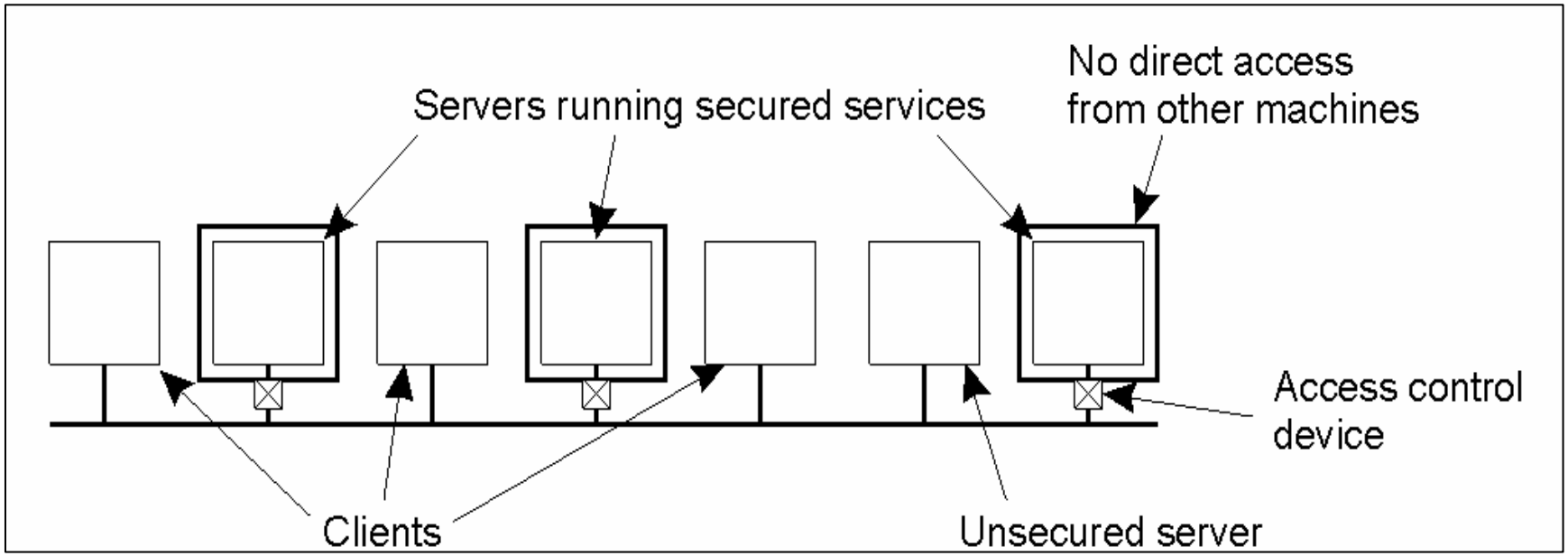
- Una TCB in un sistema distribuito può includere i sistemi operativi locali dei vari nodi del sistema.

- Esempi: file system distribuito, middleware distribuito.

Distribuzione dei Meccanismi di Sicurezza (2)

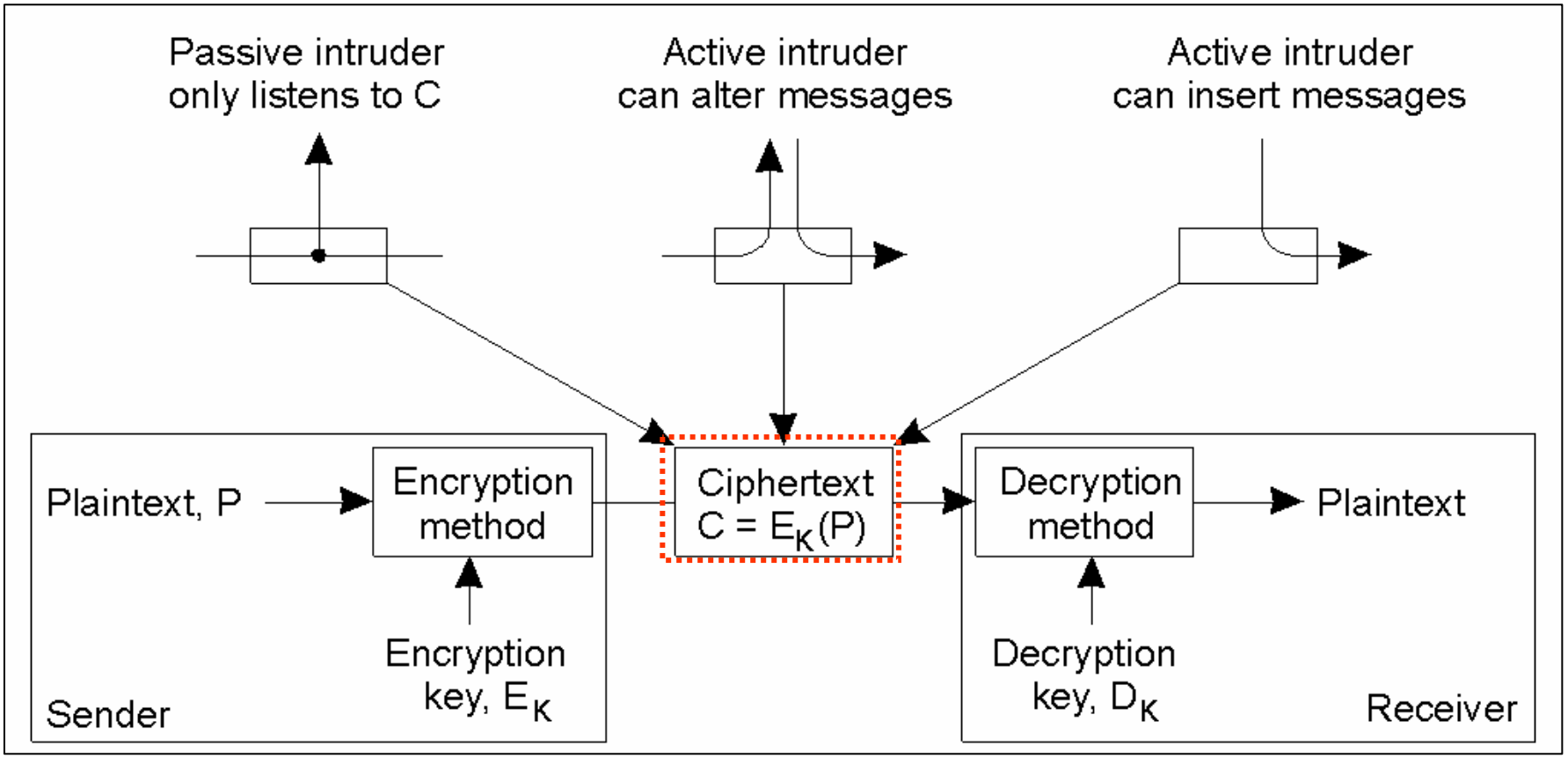
- Servizi di sicurezza possono essere isolati da altri tipi di servizi, reducendo la TCB ad un piccolo sottoinsieme di nodi.

Reduced Interface for Secure Systems Components (RISCC)



Il principio della RISSC applicato ad un sistema distribuito sicuro

Crittografia (1)



Intrusioni e ficcanaso nelle comunicazioni.

Crittografia (2)

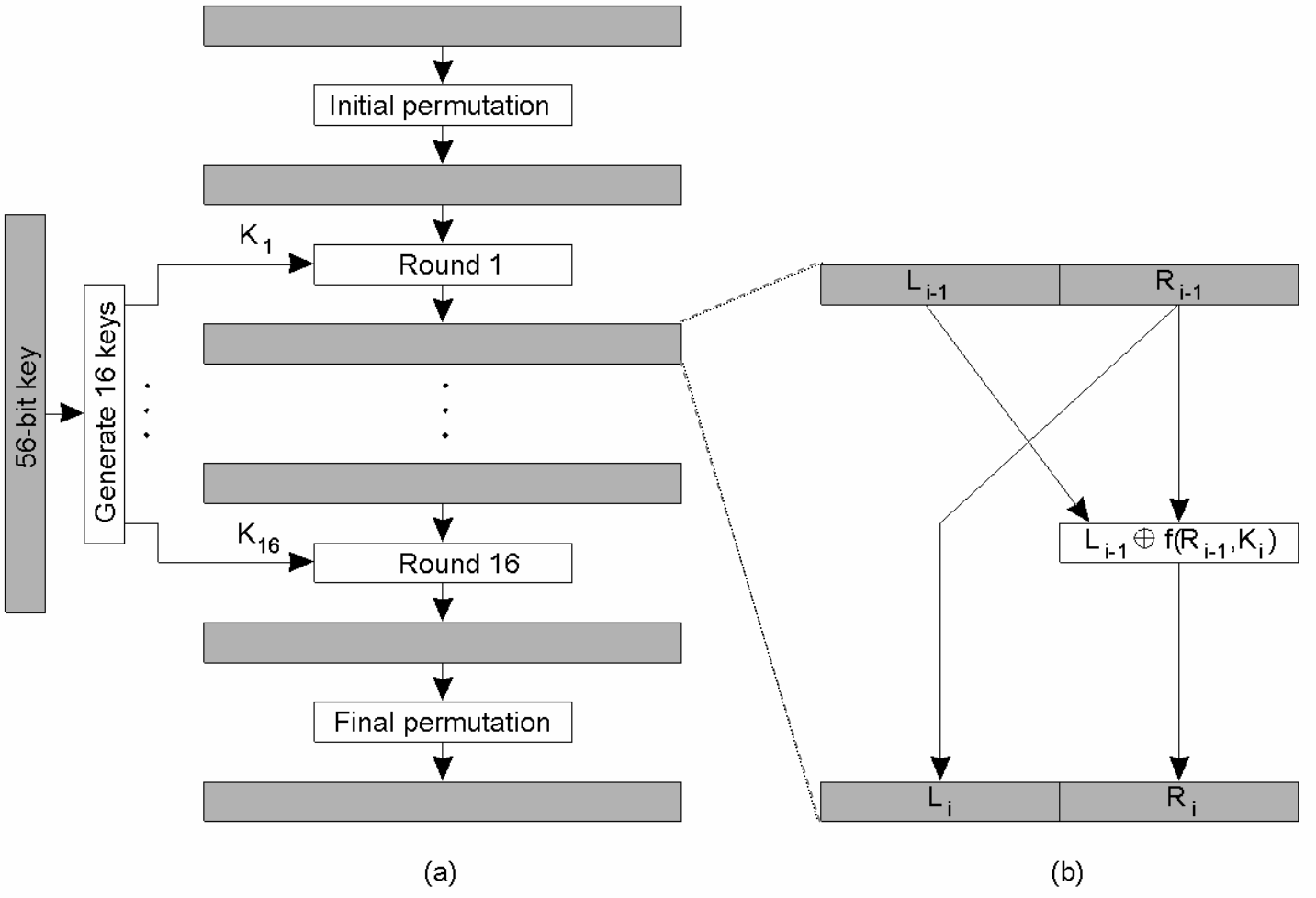
- **Sistema Crittografico Simmetrico** : $P = D_K(E_K(P))$: si usa la stessa chiave.
- **Sistema Crittografico Asimmetrico** : $P = D_{KD}(E_{KE}(P))$: si usano chiavi differenti (una pubblica e una privata): sistema a chiave pubblica

Notazione	Descrizione
$K_{A, B}$	Chiave segreta condivisa tra A e B
K_A^+	Chiave pubblica di A
K_A^-	Chiave privata di A

Crittografia Simmetrica : DES (1)

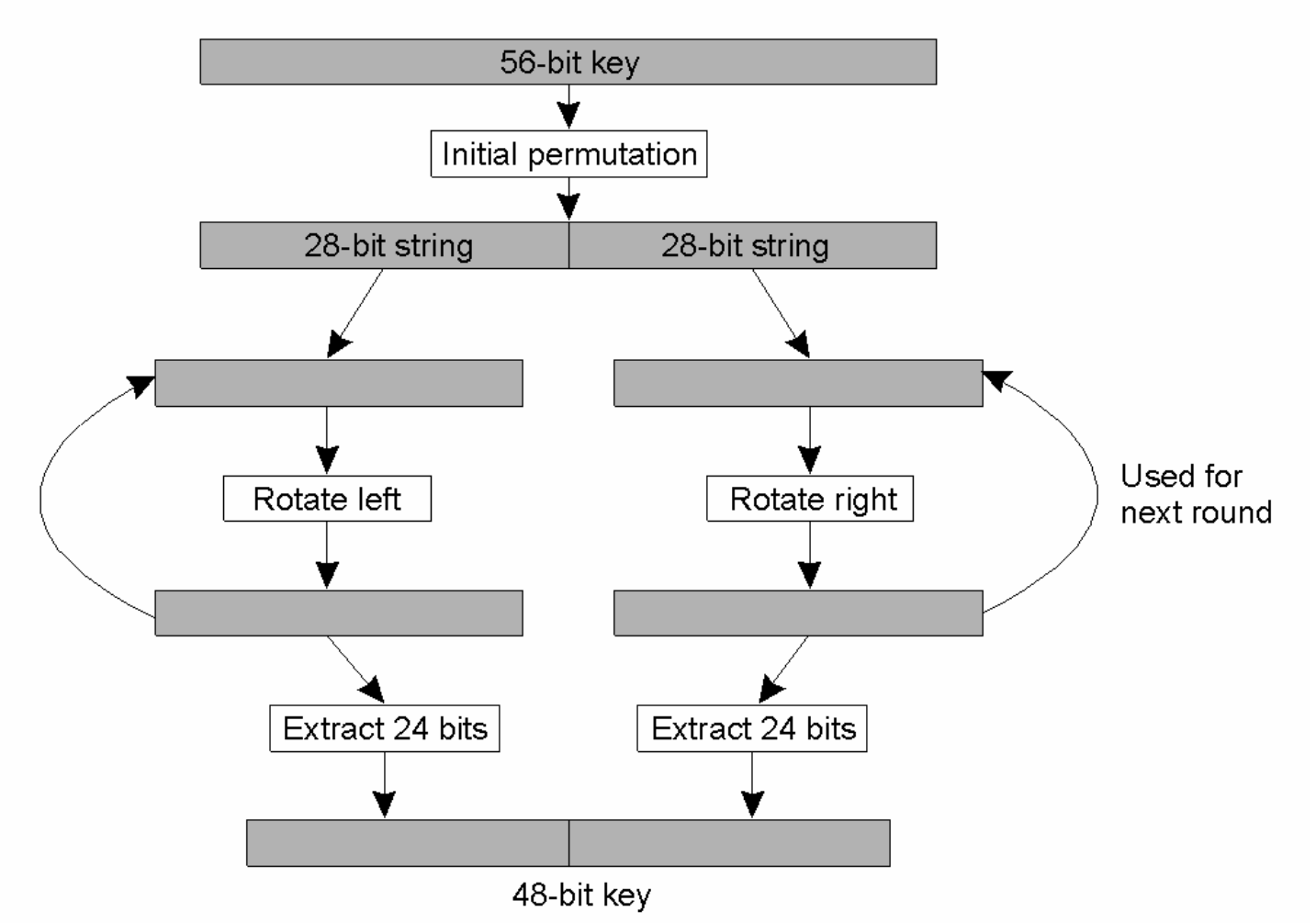
- Sistemi a cifratura simmetrica :*Data Encryption Standard* (**DES**) proposto nel 1970
- I dati sono cifrati operando su blocchi di 64 bit usando 16 chiavi di 48 bit derivati da una chiave master a 56 bit.
- Una *mangler function* ***f*** è usata per cifrare un blocco a 32 bit.
- Estensioni: Triple-DES, DESX e AES (standard)

Crittografia Simmetrica : DES (2)



(a) Il principio del DES **(b) Schema di un ciclo di cifratura**

Crittografia Simmetrica : DES (3)



Dettagli della generazione di chiavi per ciclo nel DES

Sistemi a Chiave Pubblica: RSA (1)

Sistema crittografico asimmetrico definito da Rivest, Shamir e Adleman (RSA) nel 1977, basato sul fatto che è molto complesso trovare i fattori primi di un numero di valore elevato.

- Ogni numero intero può essere scritto come il prodotto di numeri primi.
- Le chiavi pubbliche e private sono costruite a partire da numeri primi molto grandi.
- Decodificare RSA è equivalente a trovare quei numeri primi.

Sistemi a Chiave Pubblica : RSA (2)

La generazione della chiave privata e della chiave pubblica è basata su quattro passi:

1. Selezione di due numeri primi grandi, p e q
2. Calcolo di $n = p \times q$ e $z = (p - 1) \times (q - 1)$
3. Selezione di un numero d che è primo per z
4. Calcolo del numero e tale che $e \times d = 1 \bmod z$

d può essere usato per la decifratura ed e per la cifratura.

Ogni messaggio viene diviso in blocchi m_i e

Per cifrare un blocco: $c_i = m^e_i \pmod n$

Per decifrare un blocco: $m_i = c^d_i \pmod n$

Sicurezza

- La sicurezza nei sistemi distribuiti coinvolge due aspetti principali:
  - Le comunicazioni tra utenti e processi
    - » *soluzione* : **canali sicuri**
  - Autorizzazione di utenti e processi
    - » *soluzione* : **controllo degli accessi**

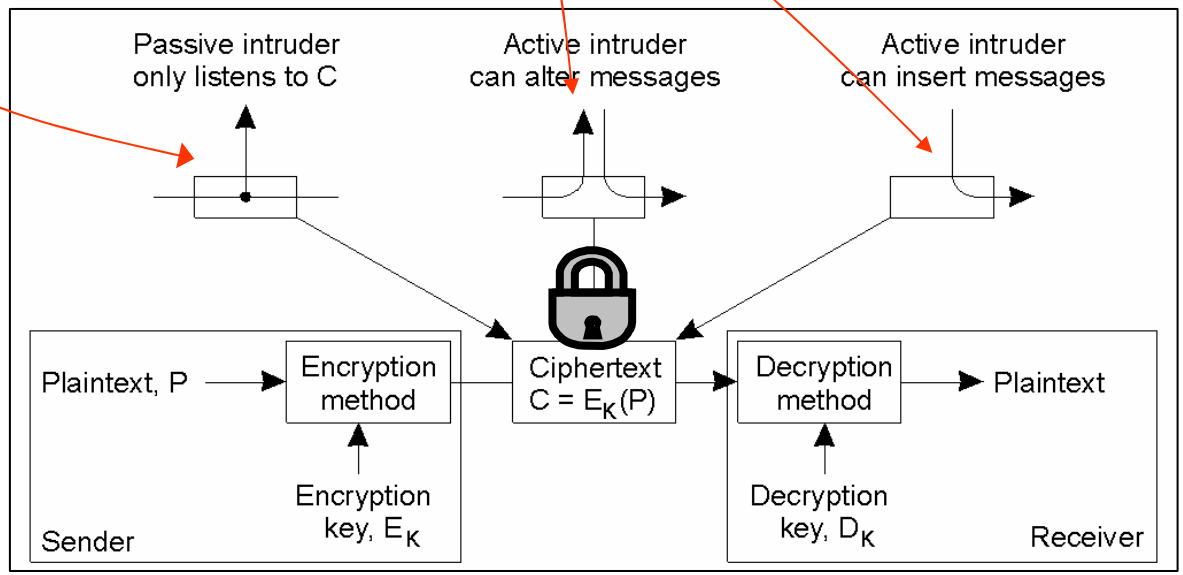
Canali Sicuri (1)

- Un canale sicuro protegge il mittente e il destinatario da
  - **Intercettazione**

    accesso non autorizzato al canale
  - **Modifica**

    modifica non autorizzata di un messaggio
  - **Fabbricazione**

    inserimento non autorizzato di un messaggio

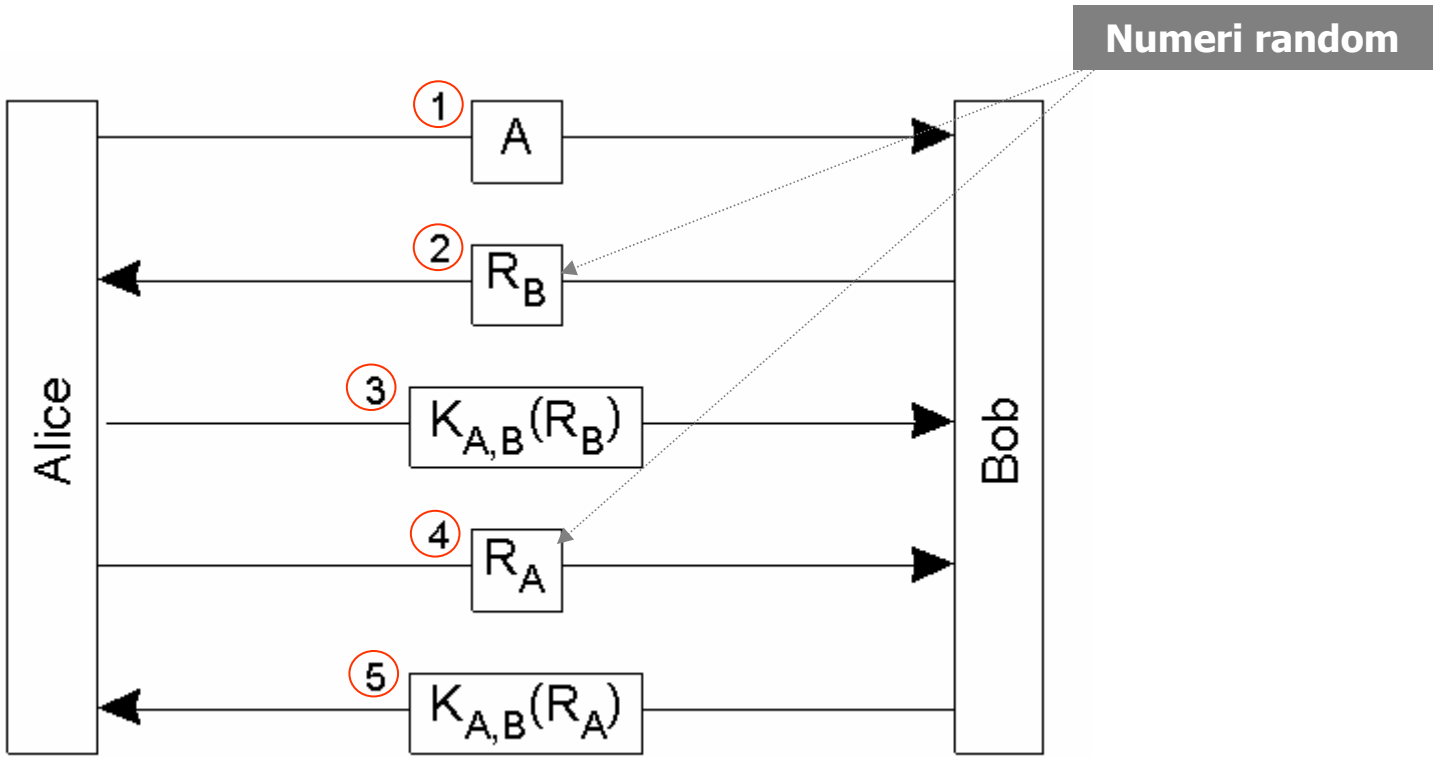


Canali Sicuri (2)

- Autenticazione e integrità dei messaggi devono essere garantiti insieme.
- Viene usata una chiave segreta associata ad un canale: **session key**.
- Quando viene chiuso un canale la *session key* viene eliminata e distrutta.

Autenticazione Basata su Chiave Segreta (1)

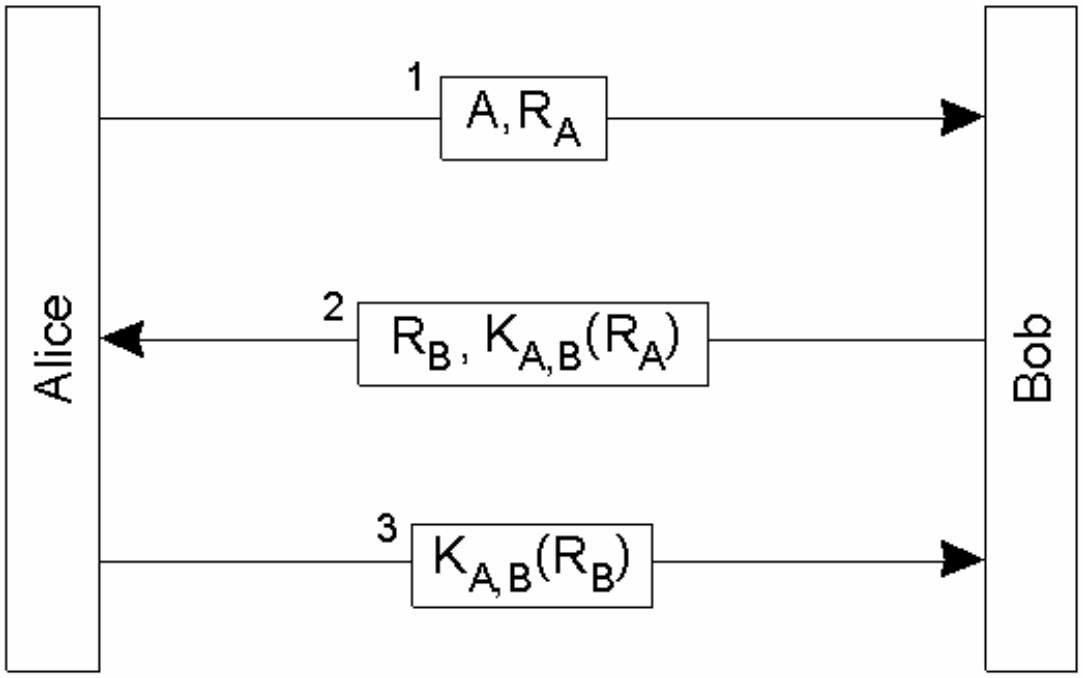
Protocolli Challenge-Response per la mutua autenticazione di due "parti" che condividono una chiave segreta



Autenticazione basata su una ***shared secret key*** ($K_{A,B}$)

Autenticazione Basata su Chiave Segreta (2)

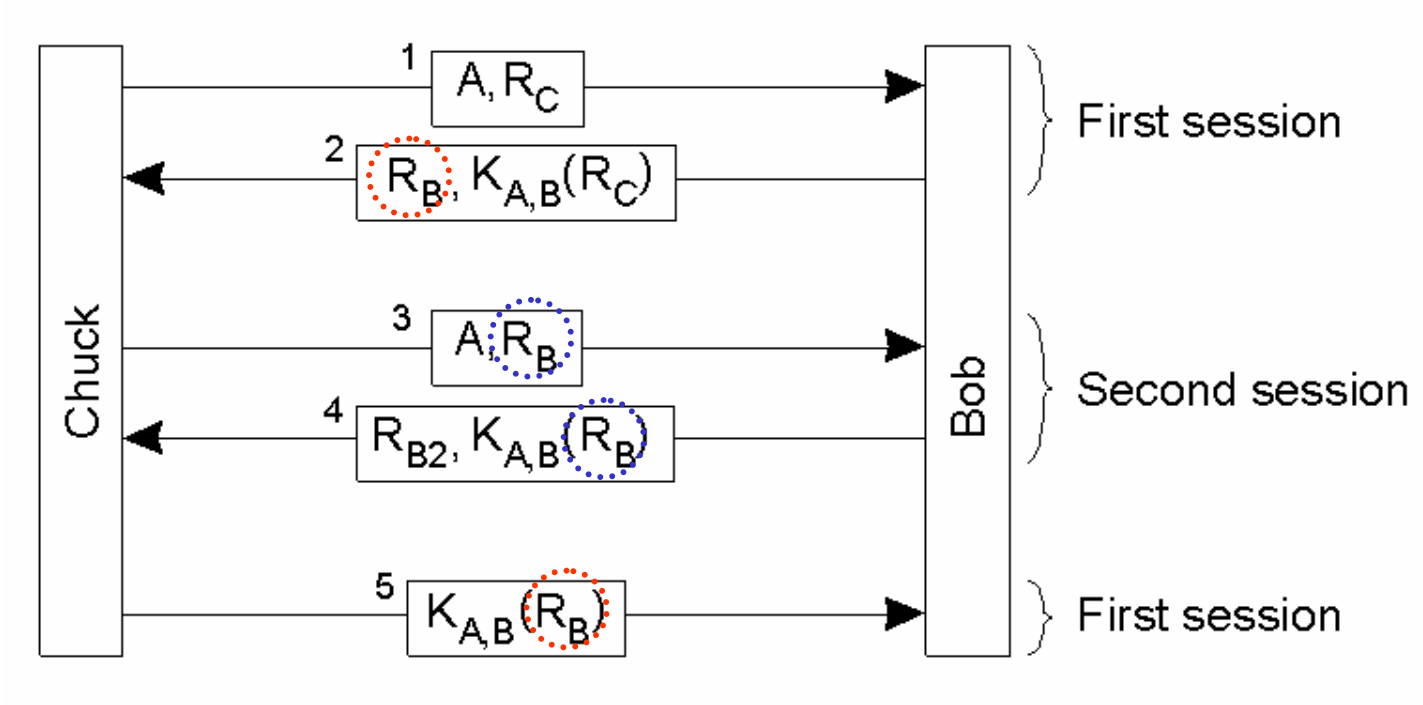
E’ possibile realizzare una ottimizzazione del Protocollo Challenge-Response ??



Possibile protocollo di autenticazione basato su chiave segreta che usa solo 3 invece di 5 messaggi.

Autenticazione Basata su Chiave Segreta (3)

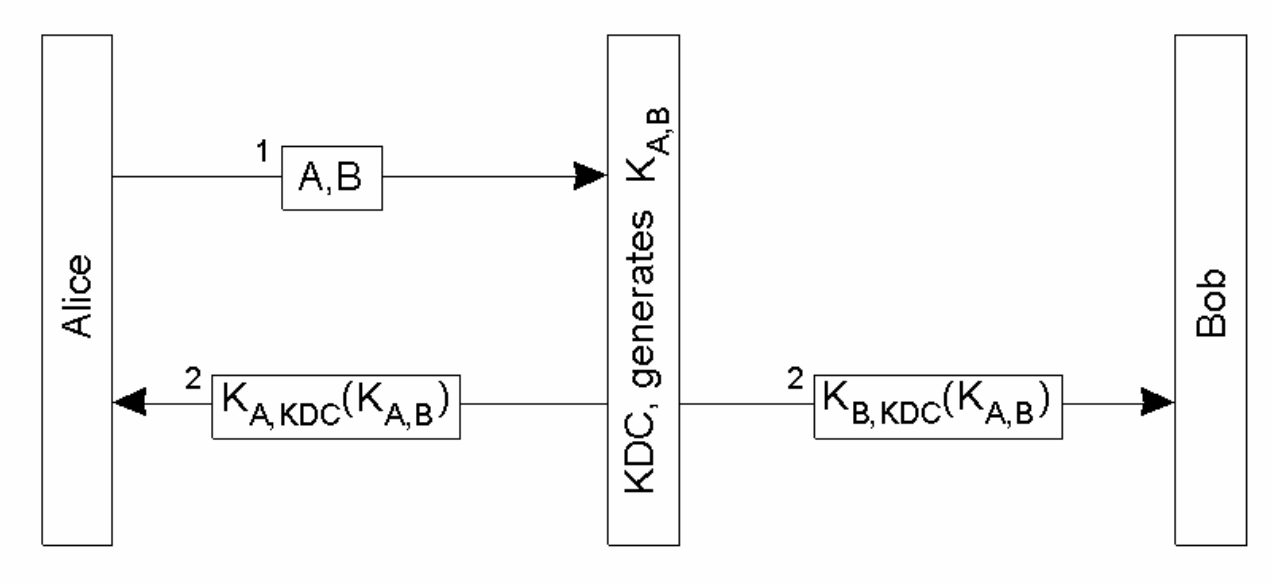
Il **Reflection Attack** fa fallire il protocollo a 3 messaggi.



Per risolvere questo problema **devono essere usati *challenge* differenti**.

Autenticazione con Uso di un Key Distribution Center (1)

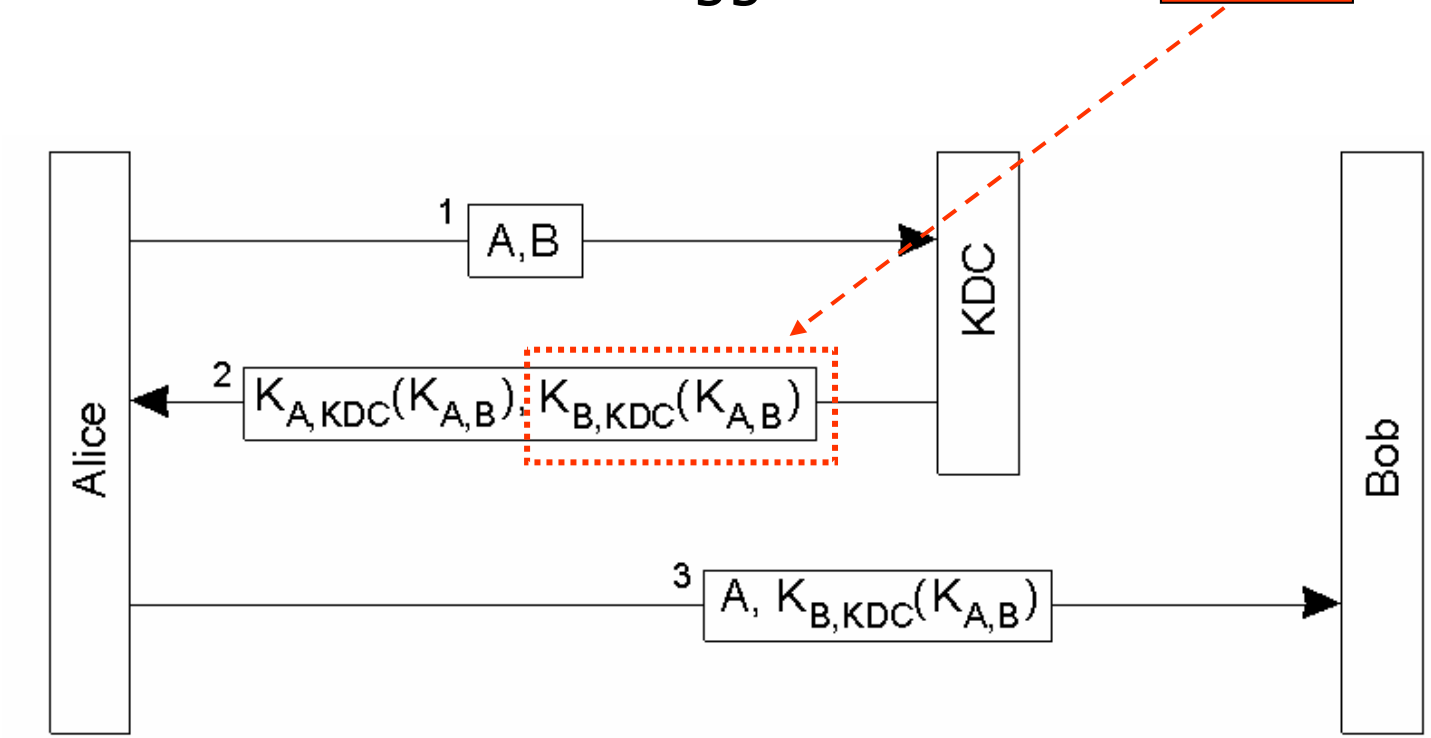
- In un sistema distribuito composto da **N** host, ogni host condivide **N-1** chiavi e globalmente sono necessarie **N(N-1)/2** chiavi segrete.
- In questo caso può essere usato un approccio centralizzato
  => il **Key Distribution Center (KDC)** gestisce solo **N** chiavi.



Il principio di uso di un KDC.

Autenticazione con Uso di un Key Distribution Center (2)

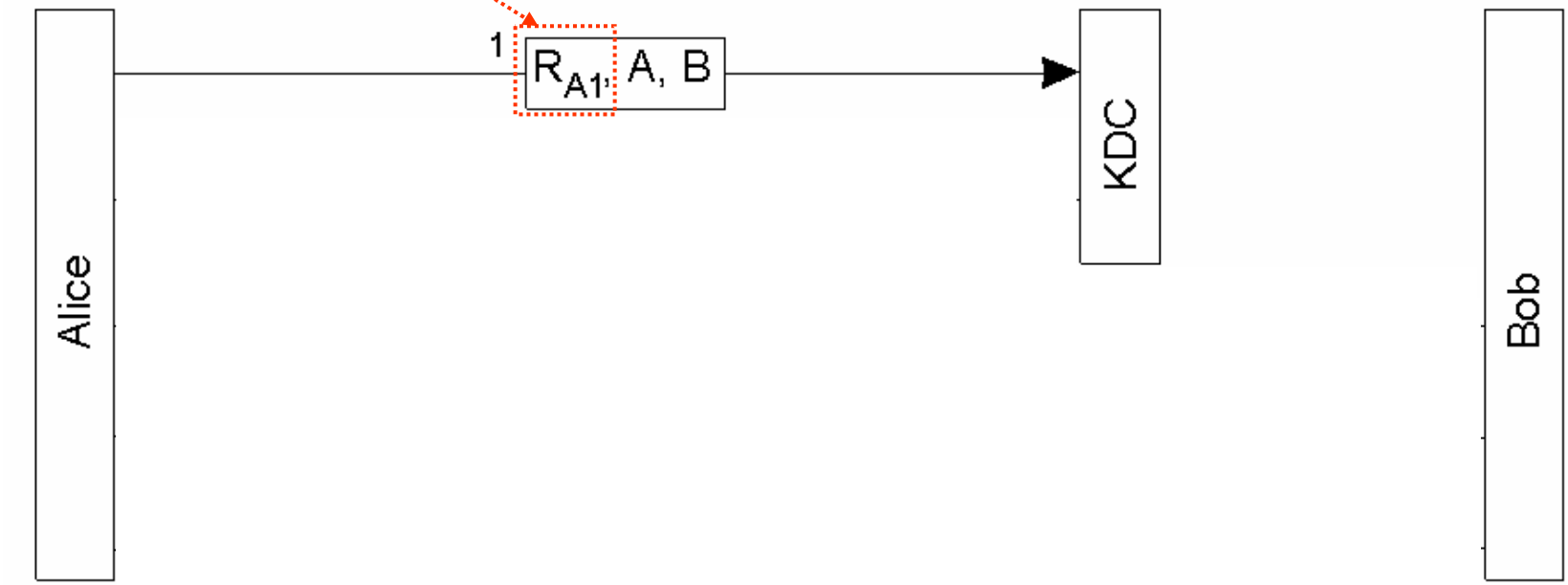
- Il KDC può passare $K_{B,KDC}(K_{A,B})$ ad ***A*** affida ad ***A*** il compito di connettersi a ***B***. Il messaggio è chiamato **ticket**.



Uso di un ticket per permettere ad Alice di connettersi a Bob.

Autenticazione con Uso di un Key Distribution Center (3)

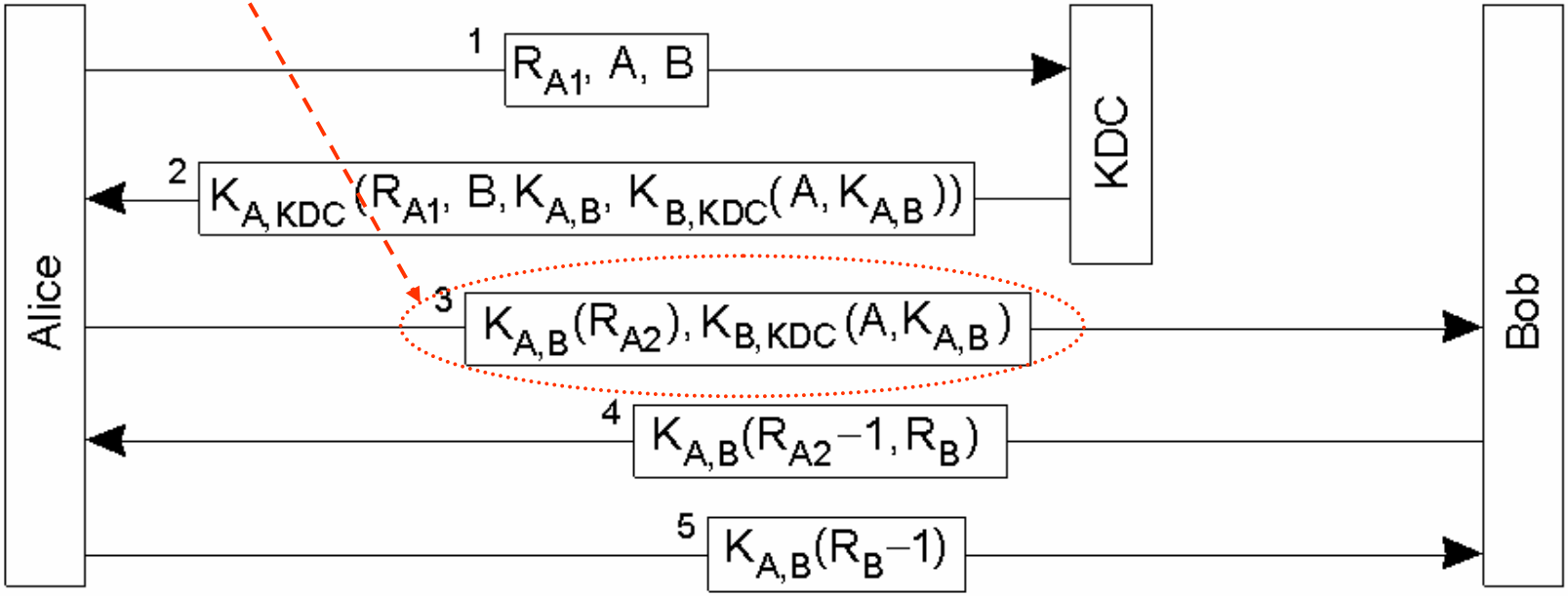
Nonce: numero random usato solo una volta.



Il protocollo di autenticazione di **Needham-Schroeder**.

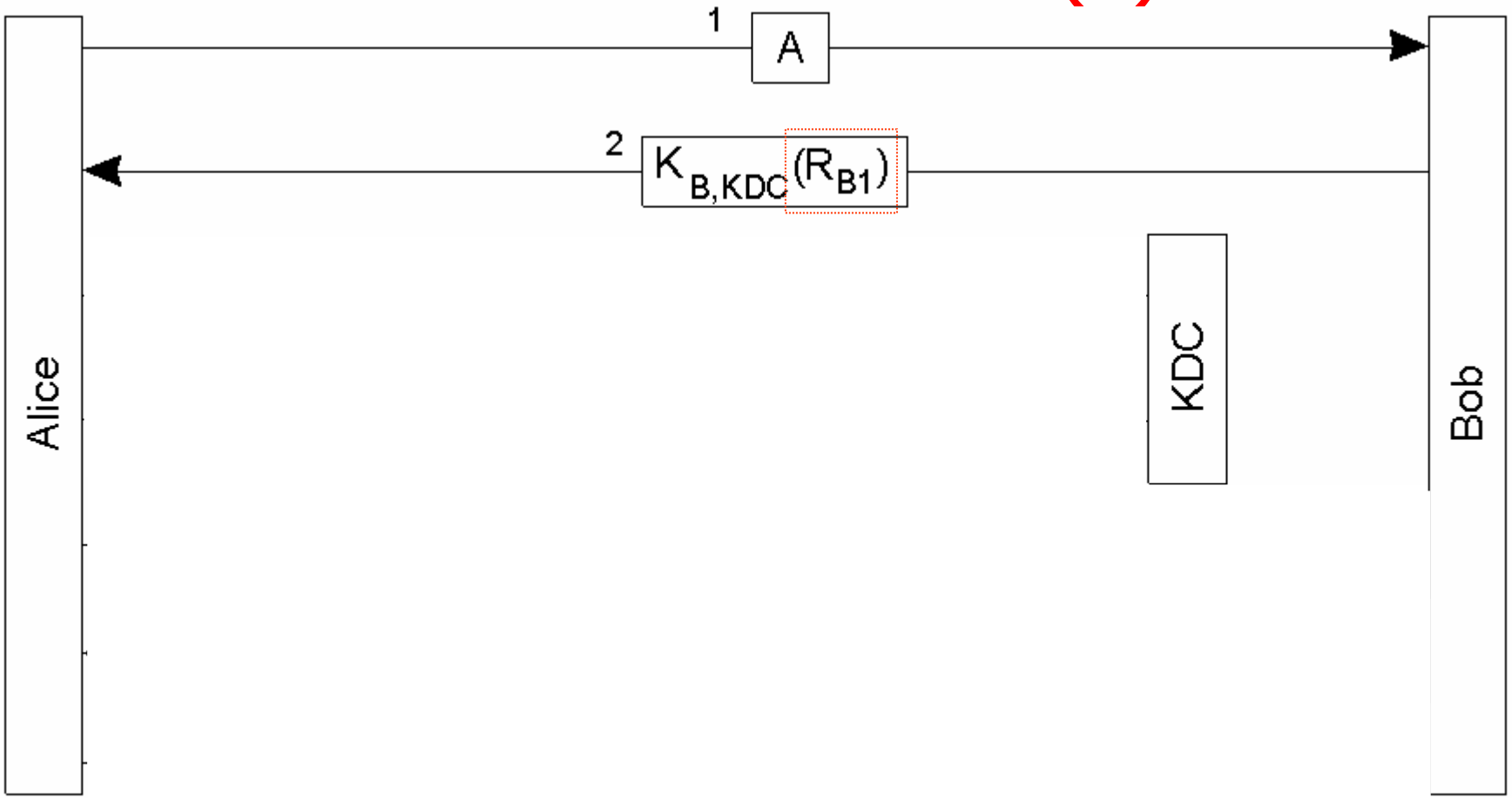
Autenticazione con Uso di un Key Distribution Center (4)

- Il messaggio 3 deve essere correlato al messaggio 1 per evitare l'intrusione di un'altra entità.
- Soluzione: Un nonce inviato inizialmente da Bob deve essere usato nel messaggio di Alice.



Il protocollo di autenticazione di **Needham-Schroeder**.

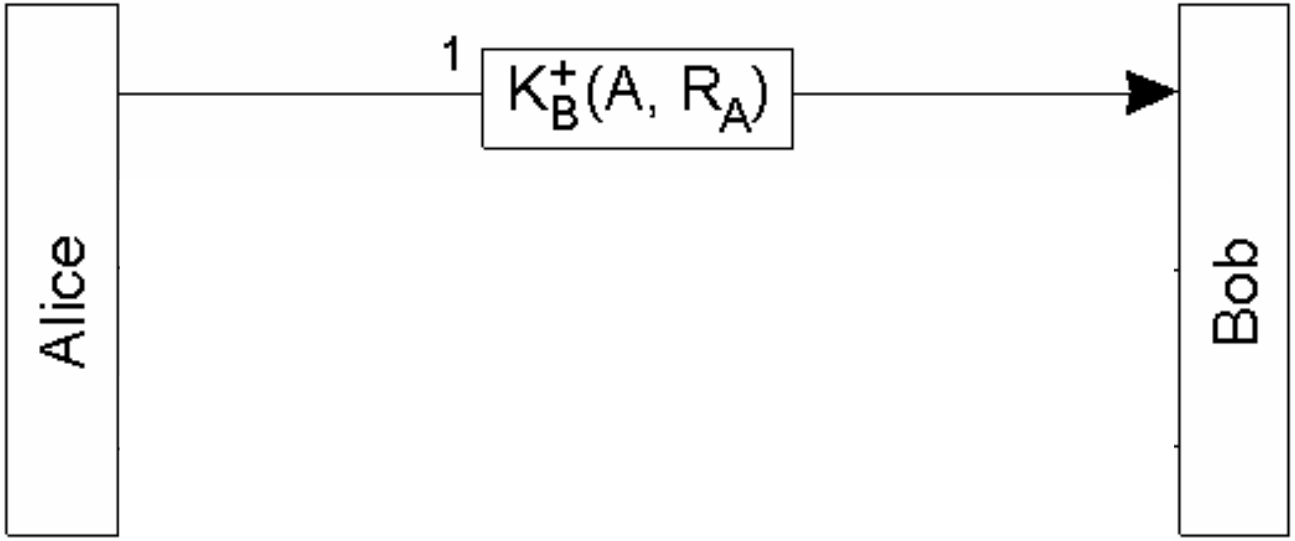
Autenticazione con Uso di un Key Distribution Center (5)



Protezione da riuso scorretto di una chiave generata nella sessione precedente nel protocollo di Needham-Schroeder.

Autenticazione con Crittografia a Chiave Pubblica

- Alice vuole definire un canale sicuro con Bob e ambedue posseggono la chiave pubblica dell'altro.
- Bob genera una session key usata per le successive comunicazioni.



Mutua autenticazione in un sistema a chiave pubblica senza un KDC.

Messaggi: Integrità e Confidenzialità

- Oltre all'autenticazione, un canale sicuro deve guarantire:
  - **confidenzialità** contro l'intercettazione
    usando ***crittografia***
  - **integrità dei messaggi** contro le modifiche
    usando ***firme digitali***

Esempio: Kerberos (1)

- **Kerberos** è un sistema sviluppato al MIT per supportare l'implementazione della sicurezza in sistemi distribuiti. In particolare, Kerberos assiste i clienti si stabilire canali sicuri con un server.

- Kerberos è basato sul protocollo di autenticazione di **Needham-Schroeder**.

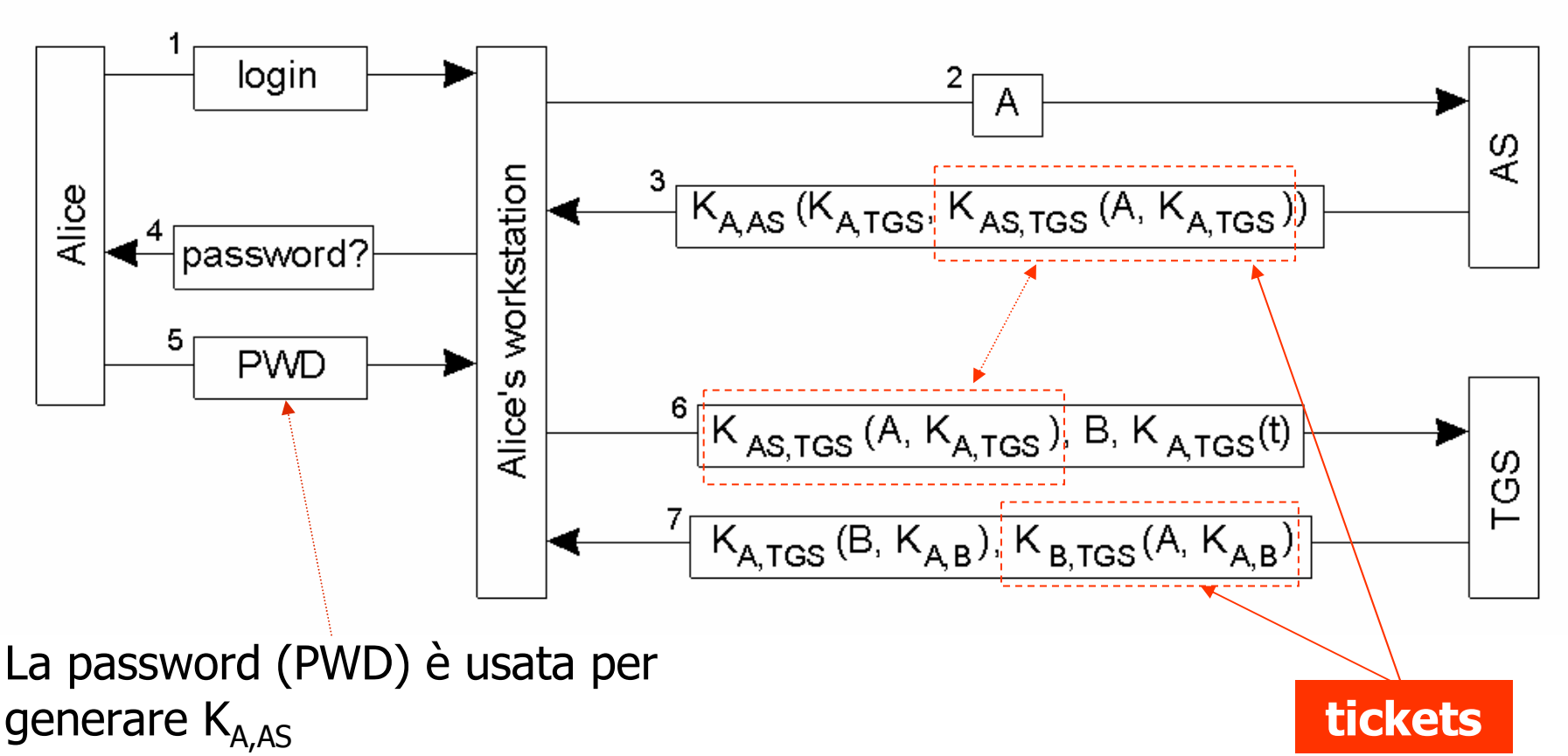
- Kerberos usa due componenti principali:
  - L'Authentication Server (AS)
  - Il Ticket Granting Service (TGS).

Esempio: Kerberos (2)

- L'**Authentication Server** (AS) autentica un utente e fornisce una chiave da essere usata per rendere sicuri i canali con un server.

- Il **Ticket Granting Service** (TGS) stabilisce canali sicuri con un server usando ticket (chiavi segrete crittografate).

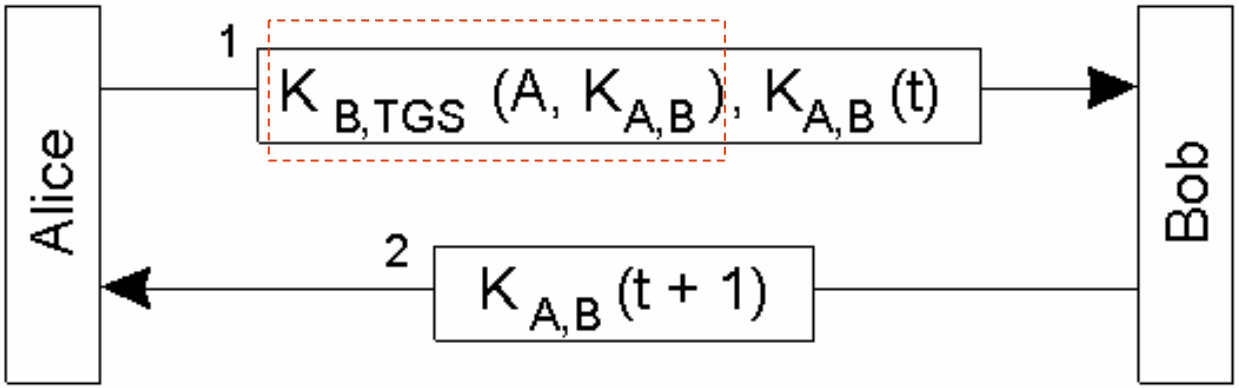
Esempio: Kerberos (3)



La password (PWD) è usata per generare $K_{A,AS}$

Autenticazione in Kerberos.

Esempio : Kerberos (4)



Creazione di un canale sicuro in Kerberos.