Sicurezza delle reti[1]

Mattia Monga

Dip. di Informatica e Comunicazione
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

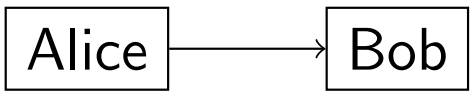
a.a. 2010/11

Lezione XIX: L'autenticazione in rete

Autenticazione

Autenticazione

Autenticare significa verificare l'identità di un soggetto (non necessariamente umano)



Tre modalità di base per l'autenticazione (di Alice) in rete:

1. tramite una password (ossia la conoscenza di un segreto)
2. basata sulla locazione (logica o fisica) da cui proviene la richiesta di autenticazione
3. per mezzo di operazioni crittografiche su dati forniti dall'autenticatore (Bob).

Pericoli

Alcune vulnerabilità sono intrinseche:

- Le password possono essere indovinate
- Le locazioni possono essere millantate
- I dati crittografici possono essere intercettati e riutilizzati (replay attack)

Queste minacce possono essere mitigate

- Aumentando la cardinalità delle password possibili
- Controlli di coerenza
- Crittografia a chiave pubblica e protocolli articolati

L'autorizzazione conseguita con l'autenticazione dura un intervallo temporale detto sessione.

Password guessing

- Una password può essere scelta in maniera prevedibile (anziché del tutto casuale) nell'insieme possibile.
- *Online guessing*: l'attaccante prova tutte le password possibili (brute force); si limitano i tentativi e/o si rallenta il feedback
- *Offline guessing*: l'attaccante accede all'elenco dei segreti (generalmente crittati con hash) e prova elenchi di parole (dictionary attack); si salano gli hash

Utente	salt	stored password
Alice	42	$\mathrm{hash}(42 \vert \mathrm{password}_{\mathrm{Alice}})$

- Le password possono essere intercettate
- Una password può essere utilizzata in occasioni differenti
- Un problema classico è anche la distribuzione iniziale delle credenziali; si fanno scadere al primo accesso

Credenziali generalizzate

Alice può provare la sua identità mostrando

- qualcosa che sa (password tradizionale)
- qualcosa che ha (authentication token)
- qualcosa che è (biometria)

È naturalmente possibile (e spesso desiderabile) avere autenticazioni a piú fattori.

Client inaffidabili

La diffusione del malware ha reso spesso inaffidabili i client
Chi garantisce che la schermata di login non sia un *cavallo di Troia* capace di memorizzare/rubare le credenziali?
(Si noti che la protezione del "tastierino" che cambia ad ogni login è puramente apparente.
Un esempio di falsa sicurezza, che tra l'altro impedisce all'utente di utilizzare meccanismi automatici di memorizzazione delle password)

Anti-malware

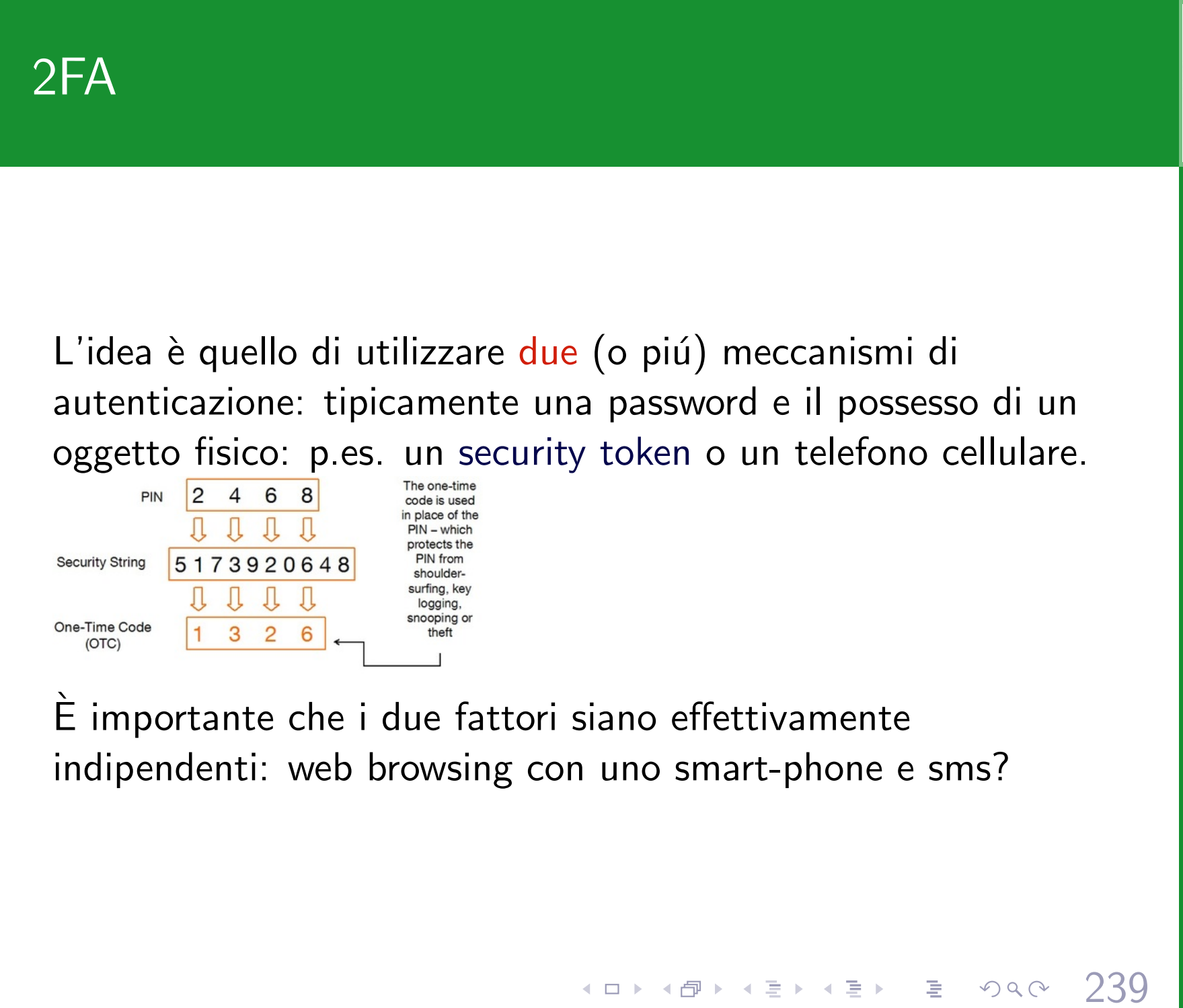
Contro client alterati è difficile proteggersi, ma protezioni piú efficaci sono:

- In ogni sessione viene comunicata solo parte della password
- Two-factor authentication (2FA)
- One-time password (OTP)

Nessuna di queste misure protegge da:

- Man in the Browser il client alterato non si limita ad intercettare le credenziali, ma è in grado di manipolare i dati della transazione
- Session hijacking l'attaccante è in grado di manipolare una sessione già in corso

2FA



L'idea è quello di utilizzare due (o piú) meccanismi di autenticazione: tipicamente una password e il possesso di un oggetto fisico: p.es. un security token o un telefono cellulare.

È importante che i due fattori siano effettivamente indipendenti: web browsing con uno smart-phone e sms?

Security token

I piú diffusi si basano su synchronous dynamic password: la password cambia ad intervalli regolari, per esempio ogni minuto.
La sincronizzazione per periodi lunghi è critica (e intervalli di accettazione piú lunghi introducono una vulnerabilità): alcuni permettono la risincronizzazione tramite collegamento fisico.

One Time Password

In generale si tratta di password utilizzate una volta soltanto e pertanto non riutilizzabili da un eventuale intercettore.
L'effettivo possesso di un security token è generalmente verificato tramite una one time password generata dal token stesso o richiesta *out of band*.

Lo schema di Lamport

Leslie Lamport nel 1981 ha proposto uno schema per autenticazione tramite OTP che non prevede la necessità di sincronizzazione temporale.
Si basa sull'esistenza di una funzione di hash H sicura (non invertibile).

1. Alice e Bob concordano un segreto W
2. Bob conserva $H(\dots H(H(W))\dots) = H^n(W)$ e n
3. Autenticazione
   1. Alice comunica la propria *username*
   2. Bob risponde con il numero n
   3. Alice comunica $x = H^{n-1}(W)$
   4. Bob verifica che $H(x) = H^n(W)$ e decrementa n

Lo schema funziona n volte, poi bisogna cambiare W.

S/KEY

Lo schema di Lamport è stato implementato da Neil Haller Phil Karn e John Walden in S/KEY.

Usa numeri a 64 bit + 2 bit di parità. Poiché stringhe casuali di 8 caratteri potrebbero essere difficili da utilizzare per un utente umano, è prevista una mappatura su 2048 parole da 1 a 4 caratteri: i 66 bit diventano una sequenza di 6 parole (`TAG SLOW NOV MIN WOOL KENO` ↔ `0x3F3BF4B4145FD74B`).

```
{
"A",    "ABE",  "ACE",  "ACT",  "AD",   "ADA",  "ADD",
"AGO",  "AID",  "AIM",  "AIR",  "ALL",  "ALP",  "AM",   "AMY",
"AN",   "ANA",  "AND",  "ANN",  "ANT",  "ANY",  "APE",  "APS",
"APT",  "ARC",  "ARE",  "ARK",  "ARM",  "ART",  "AS",   "ASH",
"ASK",  "AT",   "ATE",  "AUG",  "AUK",  "AVE",  "AWE",  "AWK",
"AWL",  "AWN",  "AX",   "AYE",  "BAD",  "BAG",  "BAH",  "BAM",
"BAN",  "BAR",  "BAT",  "BAY",  "BE",   "BED",  "BEE",  "BEG",
"BEN",  "BET",  "BEY",  "BIB",  "BID",  "BIG",  "BIN",  "BIT",
"BOB",  "BOG",  "BON",  "BOO",  "BOP",  "BOW",  "BOY",  "BUB",
"BUD",  "BUG",  "BUM",  "BUN",  "BUS",  "BUT",  "BUY",  "BY",
"BYE",  "CAB",  "CAL",  "CAM",  "CAN",  "CAP",  "CAR",  "CAT",
"CAW",  "COD",  "COG",  "COL",  "CON",  "COO",  "COP",  "COT",
"COW",  "COY",  "CRY",  "CUB",  "CUE",  "CUP",  "CUR",  "CUT",
"DAB",  "DAD",  "DAM",  "DAN",  "DAR",  "DAY",  "DEE",  "DEL",
"DEN",  "DES",  "DEW",  "DID",  "DIE",  "DIG",  "DIN",  "DIP",
"DO",   "DOE",  "DOG",  "DON",  "DOT",  "DOW",  "DRY",  "DUB",
"DUD",  "DUE",  "DUG",  "DUN",  "EAR",  "EAT",  "ED",   "EEL",
"EGG",  "EGO",  "ELI",  "ELK",  "ELM",  "ELY",  "EM",   "END",
"EST",  "ETC",  "EVA",  "EVE",  "EWE",  "EYE",  "FAD",  "FAN",
"FAR",  "FAT",  "FAY",  "FED",  "FEE",  "FEW",  "FIB",  "FIG",
"FIN",  "FIR",  "FIT",  "FLO",  "FLY",  "FOE",  "FOG",  "FOR",
"FRY",  "FUM",  "FUN",  "FUR",  "GAB",  "GAD",  "GAG",  "GAL",
"GAM",  "GAP",  "GAS",  "GAY",  "GEE",  "GEL",  "GEM",  "GET",
"GIG",  "GIL",  "GIN",  "GO",   "GOT",  "GUM",  "GUN",  "GUS",
"GUT",  "GUY",  "GYM",  "GYP",  "HA",   "HAD",  "HAL",  "HAM",
"HAN",  "HAP",  "HAS",  "HAT",  "HAW",  "HAY",  "HE",   "HEM",
"HEN",  "HER",  "HEW",  "HEY",  "HI",   "HID",  "HIM",  "HIP",
"HIS",  "HIT",  "HO",   "HOB",  "HOC",  "HOE",  "HOG",  "HOP",
"HOT",  "HOW",  "HUB",  "HUE",  "HUG",  "HUH",  "HUM",  "HUT",
"I",    "ICY",  "IDA",  "IF",   "IKE",  "ILL",  "INK",  "INN",
"IO",   "ION",  "IQ",   "IRA",  "IRE",  "IRK",  "IS",   "IT",
"ITS",  "IVY",  "JAB",  "JAG",  "JAM",  "JAN",  "JAR",  "JAW",
"JAY",  "JET",  "JIG",  "JIM",  "JO",   "JOB",  "JOE",  "JOG",
"JOT",  "JOY",  "JUG",  "JUT",  "KAY",  "KEG",  "KEN",  "KEY",
"KID",  "KIM",  "KIN",  "KIT",  "LA",   "LAB",  "LAC",  "LAD",
"LAG",  "LAM",  "LAP",  "LAW",  "LAY",  "LEA",  "LED",  "LEE",
"LEG",  "LEN",  "LEO",  "LET",  "LEW",  "LID",  "LIE",  "LIN",
"LIP",  "LIT",  "LO",   "LOB",  "LOG",  "LOP",  "LOS",  "LOT",
"LOU",  "LOW",  "LOY",  "LUG",  "LYE",  "MA",   "MAC",  "MAD",
"MAE",  "MAN",  "MAO",  "MAP",  "MAT",  "MAW",  "MAY",  "ME",
```

Vulnerabilità dello schema di Lamport

Anche questo schema non protegge da Man in the Browser o Session Hijacking. Un problema ancora piú grave è l'attacco con n piccolo

1. Bob conserva il numero n
2. Mallory impersona Bob e manda ad Alice un $n' < n$
3. Alice risponde con $H^{n'}(W)$
4. Mallory potrà usare $H^{n'}(W)$ per sostituirsi ad Alice quando Bob arriverà a conservare n'

Il pericolo può essere mitigato rendendo Alice edotta su n attuale, in modo che possa insospettirsi per eventuali $n' << n$

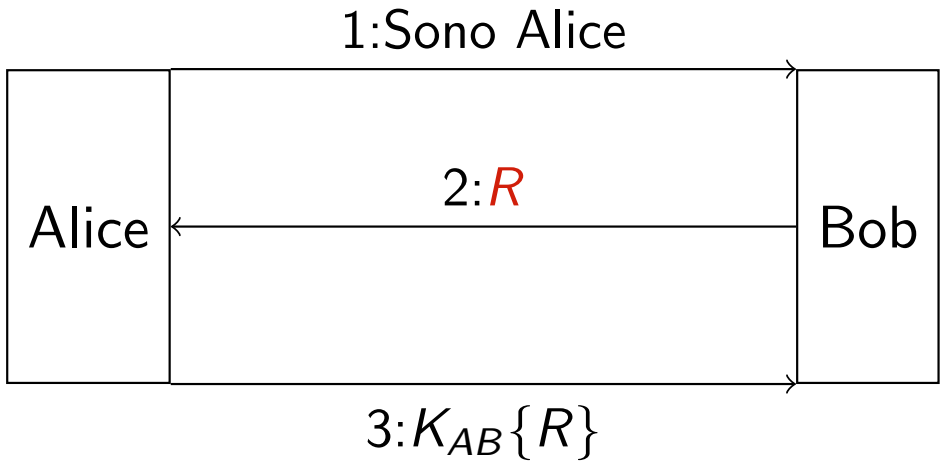
Paper Lamport

Lo schema di Lamport però funziona bene con modalità "carta e penna".

- Alice conserva una lista cartacea di password $(H^n(W), H^{n-1}(W), \dots)$
- Una volta usata la prima della lista la cancella

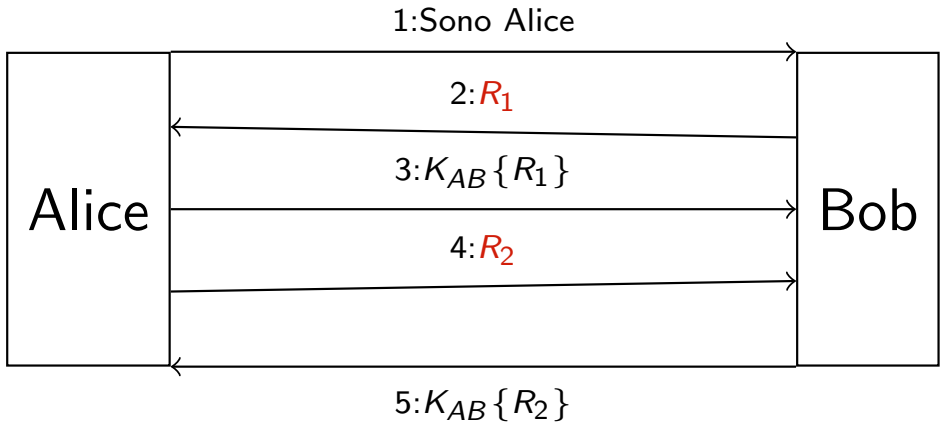
Questo schema non è suscettibile all'attacco di n piccolo, ma la lista è naturalmente un punto critico.

Challenge/Response



- L'autenticazione non è reciproca: qualcuno potrebbe sostituirsi a Bob.
- Offline-guessing di K_{AB} è possibile intercettando R e $K_{AB}\{R\}$

Challenge/Response con mutua autenticazione

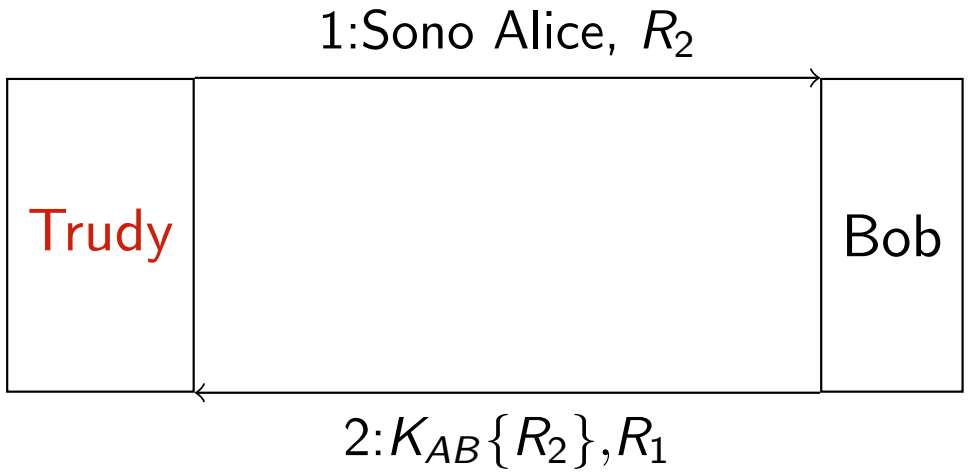


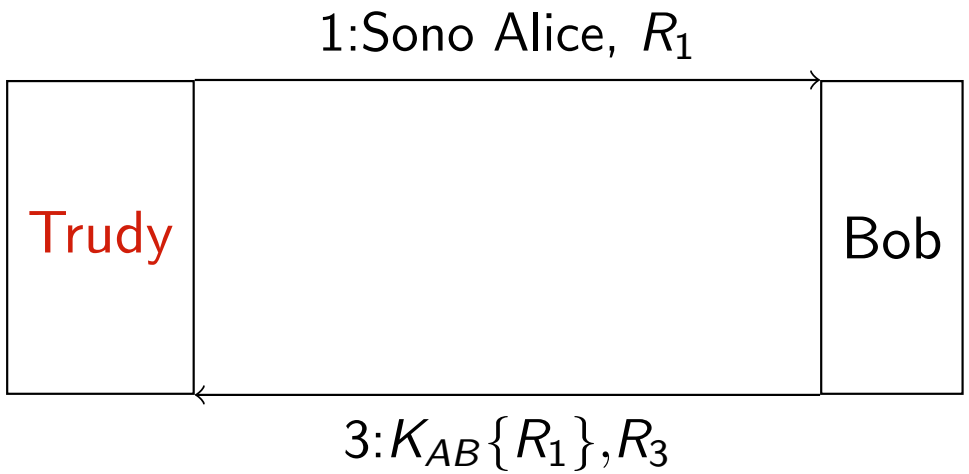
Sono necessari ben 5 scambi: si può rendere piú efficiente?

Challenge/Response con mutua autenticazione

Challenge/Response con mutua autenticazione

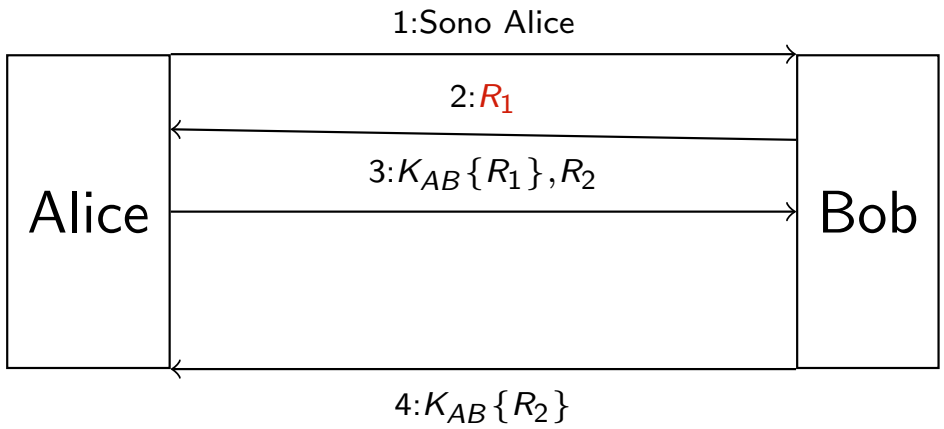
Purtroppo cosí si presta ad un reflection attack





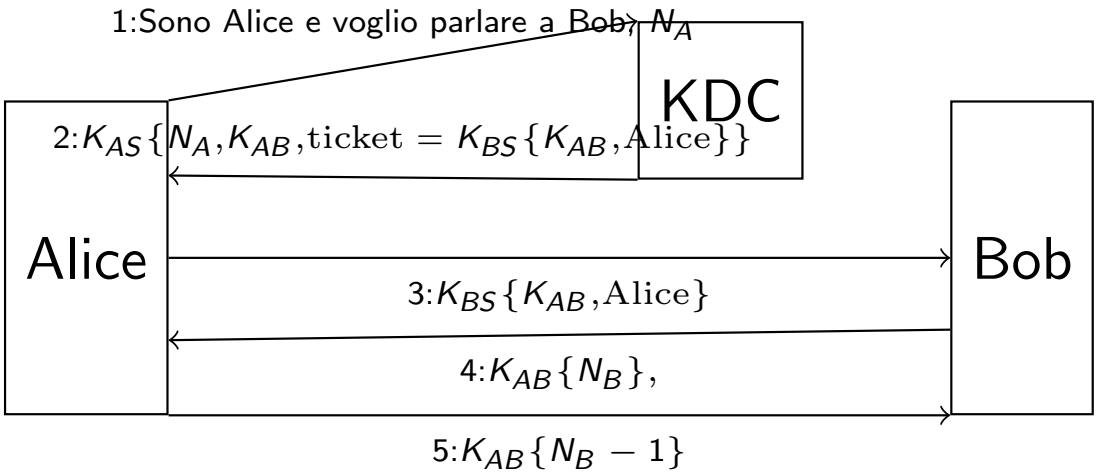
Challenge/Response con mutua autenticazione

Cosí va meglio:



Needham-Schroeder

Per semplificare la gestione dei segreti condivisi, possono essere introdotti dei Key Distribution Center (KDC). Uno dei protocolli classici è Needham-Schroeder [1978]



Qualora K_{AB} sia compromesso (p.es. accedendo alla macchina di Alice) è possibile un replay attack del *ticket*, quindi occorre complicarlo ulteriormente introducendo dei timestamp, in modo che i ticket non possano essere riutilizzati. I numeri N sono detti nonce.