Autenticazione

Autenticazione

- L’autenticazione è uno degli elementi più critici nella sicurezza
- Essa permette ad un’entità (una persona o un sistema) di dichiarare la propria identità ad un’altra entità
- Una buona infrastruttura di autenticazione già protegge dalla maggior parte degli attacchi
- Autenticarsi significa disporre di **credenziali**
- Di solito l’entità che vuole autenticarsi deve dimostrare la conoscenza di un segreto all’altra

Tipi di autenticazione

- Autenticazione dei **dati**
  - prova dell'origine dei dati
  - normalmente associata all'integrità

- Autenticazione dei **peer**
  - prova dell'identità dell'altro estremo della comunicazione
  - autenticazione singola o mutua
  - autenticazione del client (ad esempio username + password)
  - autenticazione del server (meno frequente ma molto importante!)

- AuthN: Authentication
- AuthZ: Authorization
- Accounting: tengo traccia di tutto quello che succede (log)

Message authentication code (MAC)

- Metodo usato per verificare l'integrità e l'autenticità di un messaggio (ottimo se il messaggio si corrompe nel "canale", accidentalmente)
- Può anche verificare l'identità del mittente (diversamente dal solo hash, visto che l'attaccante potrebbe mettersi in mezzo, essendo la funzione hash pubblica)
- Richiede l'utilizzo di una chiave segreta simmetrica (diversamente dal solo hash)
- Si può basare sulla combinazione di una funzione hash con una chiave segreta
- Oppure può sfruttare un cifrario simmetrico (confidenzialità + digest cifrato) e si parla di **authenticated encryption** (utile se Eve altera i messaggi)

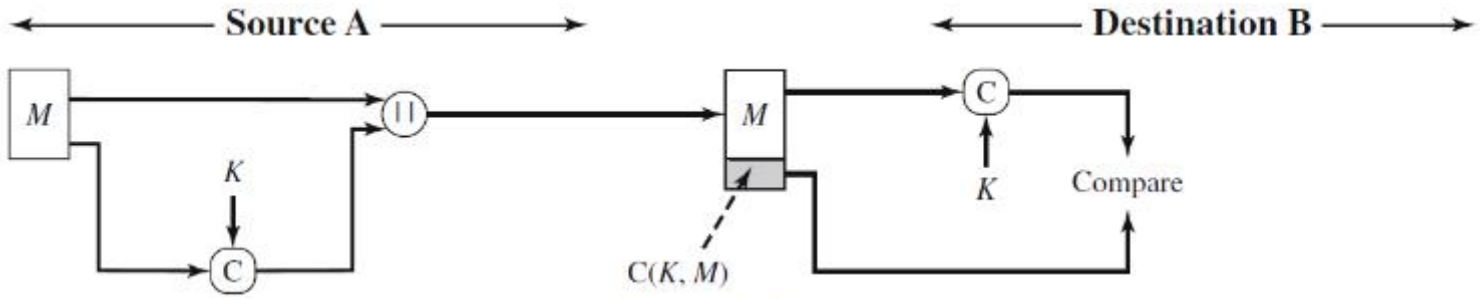
Message authentication code (MAC) - 2

- Le due parti (A e B) condividono una chiave segreta (K)
- Quando A deve inviare un messaggio (M) a B, ne calcola il MAC:

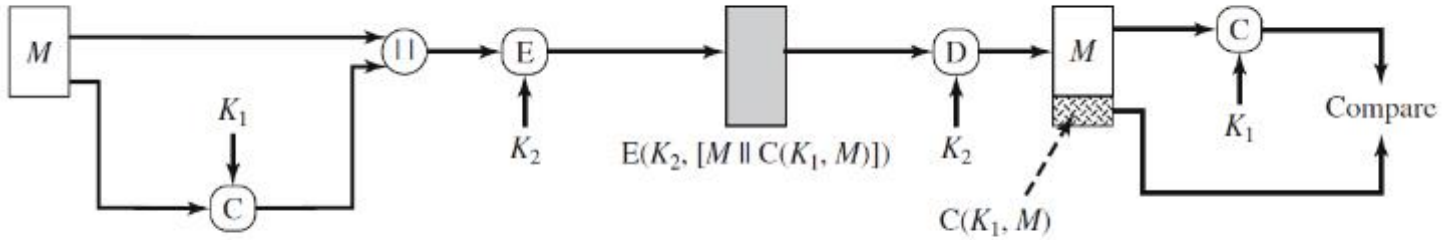
$$MAC = C(K, M)$$

  - C = funzione MAC
  - MAC = valore del MAC
- Il risultato viene trasmesso a B insieme al messaggio M
- B ripete il calcolo e verifica che il suo valore del MAC coincida con quello ricevuto

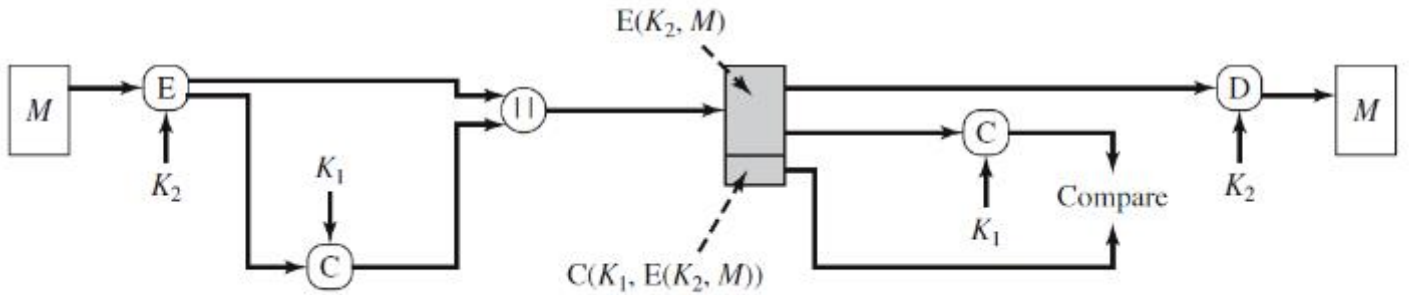
Message authentication code (MAC) - 3



(a) Message authentication



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

Bob deve lavorare in serie

Bob può lavorare in parallelo

Hash con chiave: HMAC

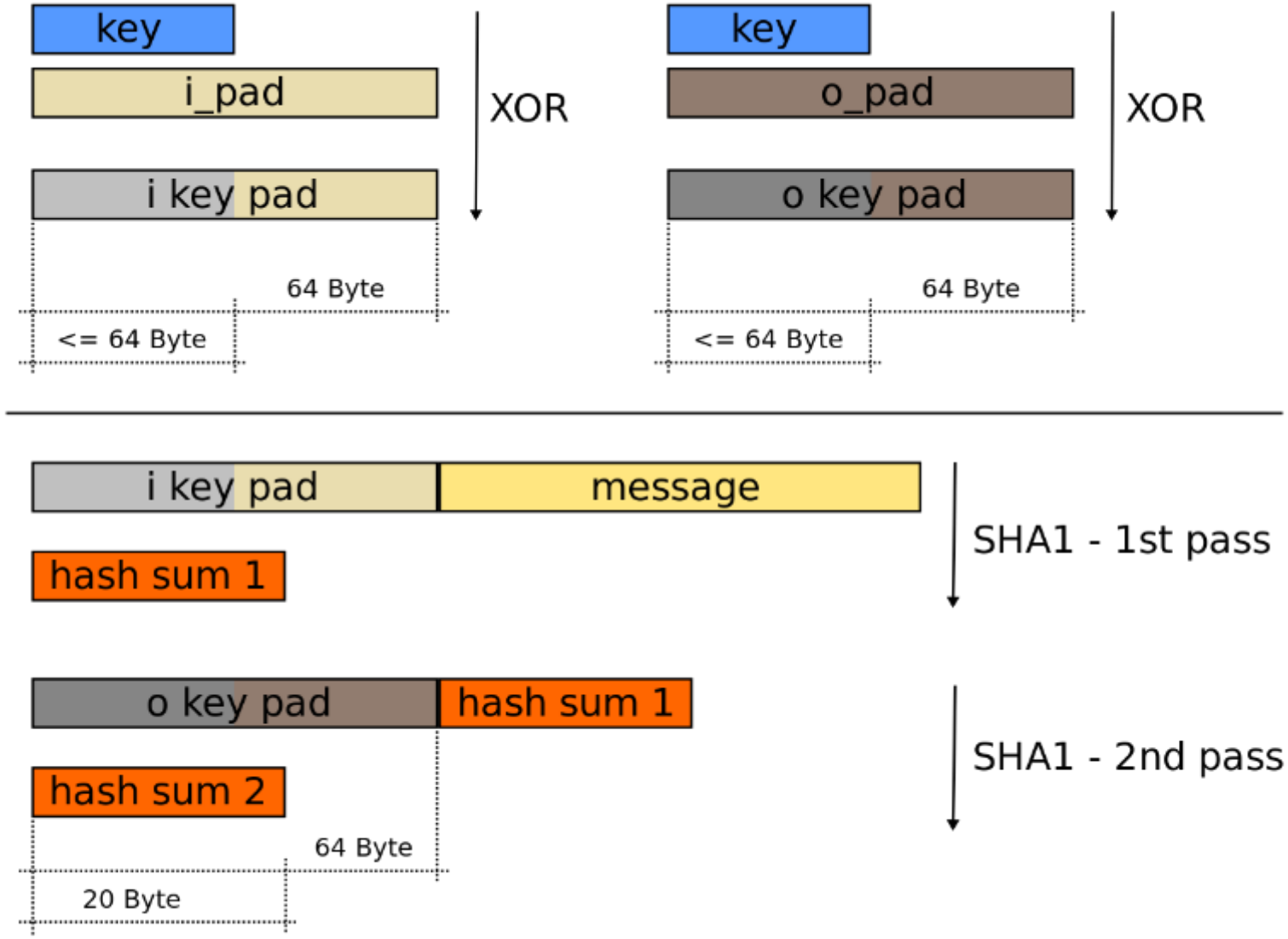
- Algoritmo MAC basato su una funzione hash
- Molte funzioni hash (come MD5, SHA-1 e SHA-2) usano uno stato interno che evolve in funzione dei blocchi del messaggio ed assume un valore finale che coincide col digest
- Questo le rende vulnerabili ad **attacchi basati su estensione** del messaggio:
  - Partendo dall'ultimo valore della variabile di stato, si può **prolungare** (senza modificare quanto già c'è) il messaggio e continuare il calcolo del digest
  - Ciò potrebbe consentire ad Eve di calcolare un HMAC valido per un messaggio esteso pur non conoscendo la chiave

Hash con chiave: HMAC (2)

- Per evitare tali attacchi, HMAC usa una doppia applicazione della funzione hash

- La chiave segreta viene inizialmente usata per derivare due altre chiavi:
  - la **chiave interna** viene usata dal primo passaggio dell'algoritmo per produrre un hash interno partendo dal messaggio
  - la **chiave esterna** viene usata dal secondo passaggio dell'algoritmo per calcolare il valore di HMAC partendo dal risultato del primo passaggio

HMAC basato su SHA-1



Autenticazione dei peer

- Un utente può avere **credenziali** di diverso tipo:
  - Quello che sa (password, pin…)
  - Quello che ha (badge, smartcard…)
  - Quello che è (impronte digitali, caratteristiche vocali, analisi della retina…)
  - Combinazioni delle precedenti

- Autenticazione a **singolo fattore**
  - Usa un solo tipo di credenziali per autenticare l'utente. Garantisce un livello di sicurezza minimo ed è sconsigliata per applicazioni sensibili (ad es. finanziarie).

- Autenticazione a **più fattori**
  - Usa due o più tipi di credenziali e fornisce livelli di sicurezza maggiori.

Authentication, Authorization, Accounting

- Dopo avere verificato l'identità del soggetto, il sistema deve determinare i suoi diritti e privilegi: questo è il processo di **autorizzazione**.
- Il sistema dovrebbe inoltre tenere traccia degli eventi relativi alle autenticazioni e autorizzazioni avvenute, tramite la raccolta di file di **log**.
- Tali log servono per ragioni di «contabilità» (**accounting**), ma sono anche utili per alcune funzioni di sicurezza, come il rilevamento di intrusioni.

Autenticazione singola o mutua

- **Autenticazione singola**: il client autentica il server oppure il server autentica il client
- **Autenticazione mutua**: il client autentica il server e allo stesso tempo il server autentica il client
- Esempio di autenticazione singola: consultazione di un sito informativo (Wikipedia)
  - Il client deve essere certo che il server è autentico
  - Il server non ha bisogno di autenticare il client
- Esempio di autenticazione mutua: consultazione della posta elettronica
  1. Il client deve essere certo che il server è autentico
  2. Il server deve accertarsi dell'identità del client

Requisiti dell'autenticazione

- **Non trasferibilità**: Alice (authenticator) non dovrebbe essere in grado di riutilizzare la prova fornita da Bob (client), cioè Alice non dovrebbe essere in grado di impersonare Bob – **L'authenticator** è l'intermediario dell'autenticazione tra il client e il server in un'infrastruttura di autenticazione completa e dovrebbe solo inoltrare messaggi senza appropriarsene

- **Non sostituibilità**: Eve che osserva gli scambi tra Alice e Bob non dovrebbe essere in grado di impersonare né Alice né Bob

- Tali requisiti devono essere soddisfatti anche quando Alice e Bob eseguono autenticazioni multiple e Eve può fare un numero infinito di tentativi di autenticazione

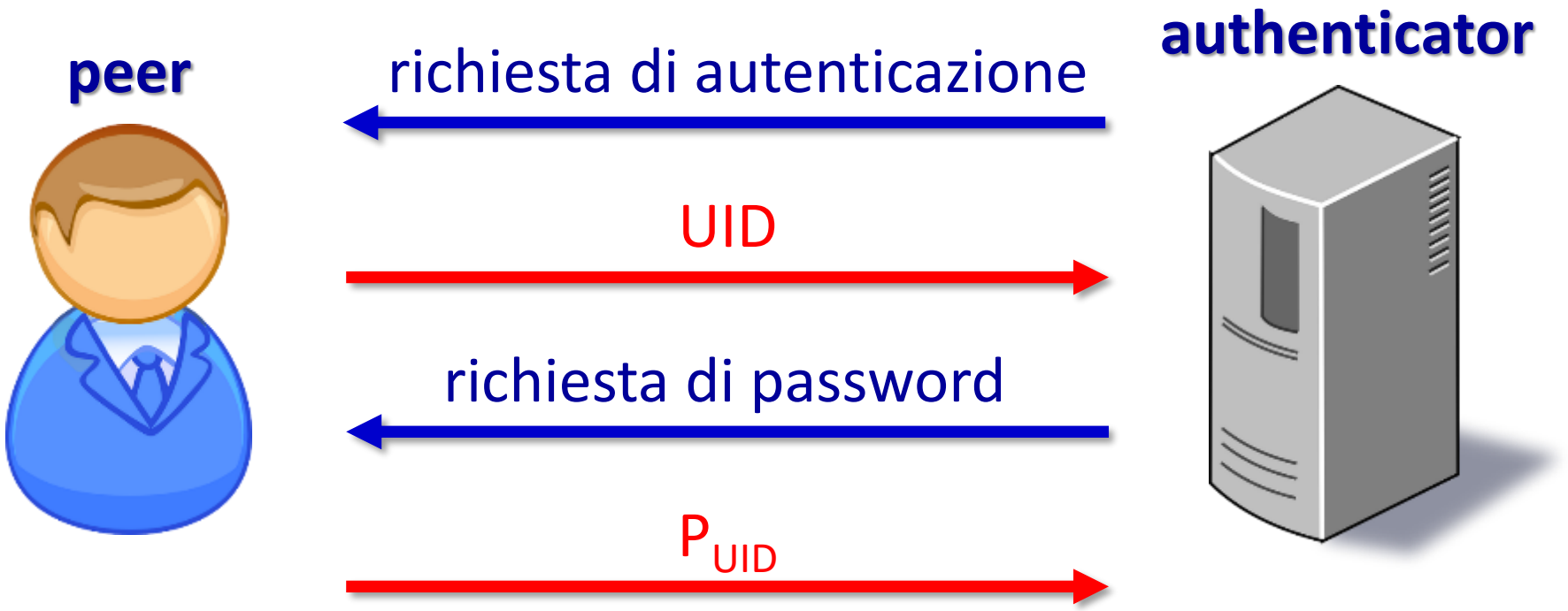
Protocolli di Autenticazione

- Lo scambio delle credenziali deve essere sicuro
- È fondamentale avere dei solidi meccanismi per gestire l'autenticazione
- Questi meccanismi sono chiamati **protocolli di autenticazione**
- Esistono numerosi protocolli di autenticazione che presentano diversi livelli di sicurezza…quelli standard e quelli proprietari:
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Response Handshake Authentication Protocol)
  - MS-CHAP v1/v2 (variante Microsoft del CHAP)
  - EAP (Extensible Authentication Protocol)

PAP (Password Authentication Protocol)

- È la forma di autenticazione più semplice
- Richiede soltanto nome utente e password che vengono trasferiti sulla rete e confrontati con una tabella delle coppie consentite residente sul server
- È definito in RFC 1334
- La password attraversa la rete in chiaro
- PAP offre scarsa sicurezza

PAP


peer
authenticator
richiesta di autenticazione
UID
richiesta di password
P_{UID}

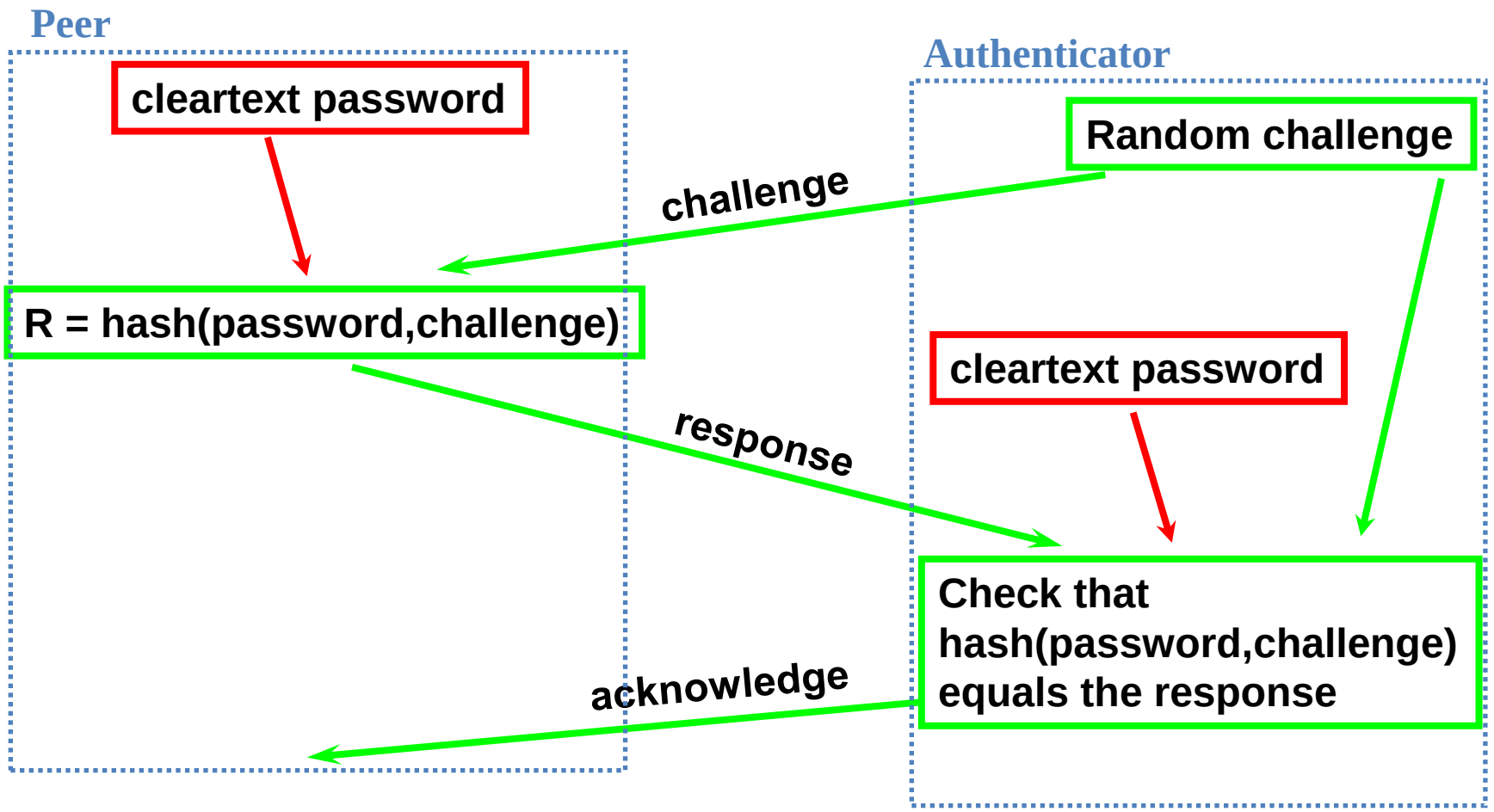
Memorizzare le password

- Non possono mai essere memorizzate in chiaro
- Per memorizzarle cifrate serve che il server memorizzi anche la chiave di cifratura, e allora…
- È più sicuro memorizzare un digest della password
- Esistono però gli attacchi del dizionario
- Per contrastarli serve usare il **salt**, che si **concatena** al messaggio quando si calcola il digest, così contrastando attacchi a dizionario paralleli su tutti gli utenti

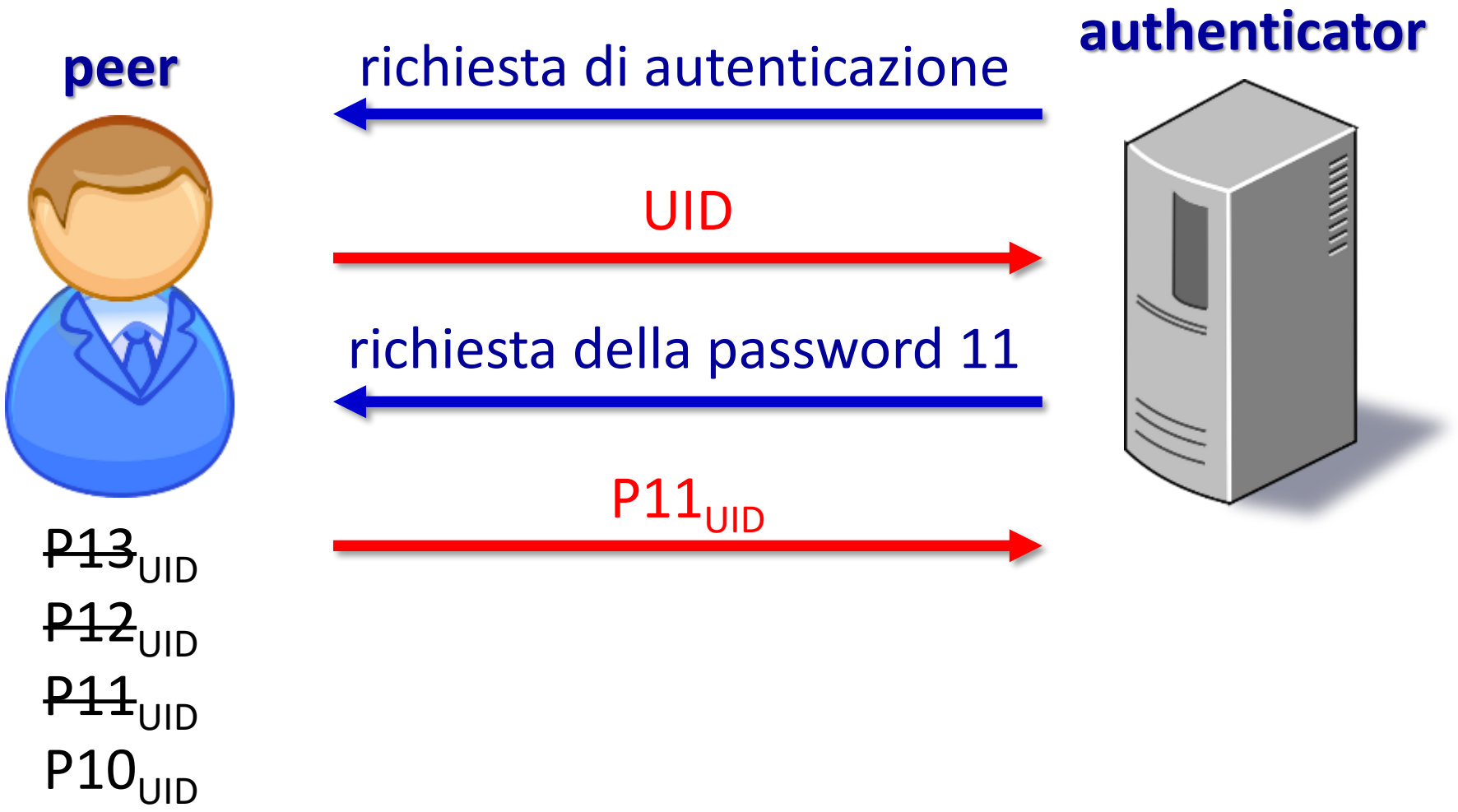
CHAP (Challenge-Handshake Authentication Protocol)

- Dopo aver stabilito la connessione, l'autenticatore invia una sfida al client che chiede di essere autenticato
- Il client prova di conoscere un segreto condiviso rispondendo alla sfida
- Può essere utilizzato più volte all'interno di una sessione per verificare se questa è stata dirottata
- È definito in RFC 1994
- Non supporta nativamente la mutua autenticazione
- Richiede la disponibilità in chiaro del segreto condiviso

CHAP (Challenge-Handshake Authentication Protocol)

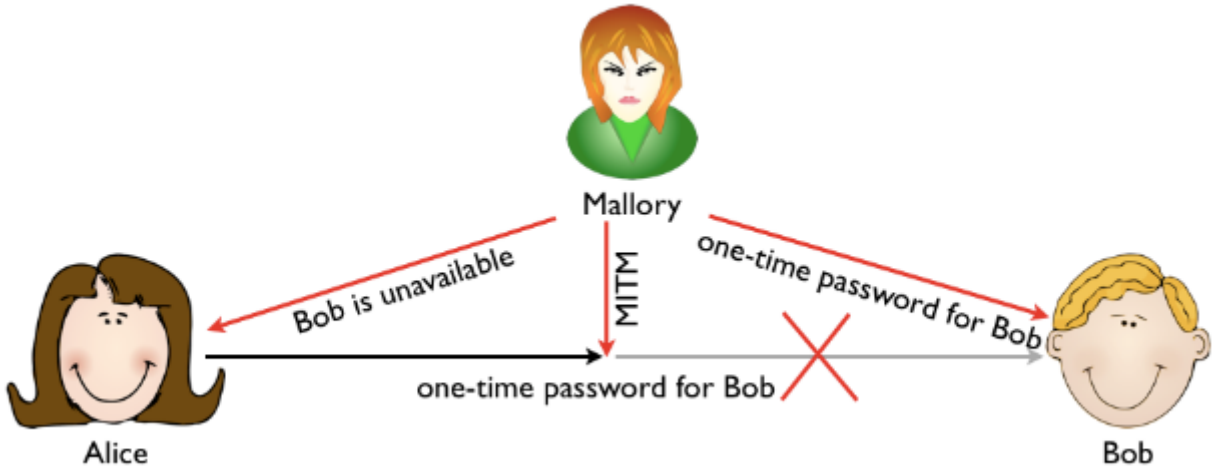


One-Time Password (OTP)



One-Time Password (OTP) - 2

- Ciascuna password è valida solo per un'esecuzione del protocollo di autenticazione
- Immune ad intercettazione (la password non si riusa)
- Soggetta ad attacchi MITM (senza protezione delle credenziali)

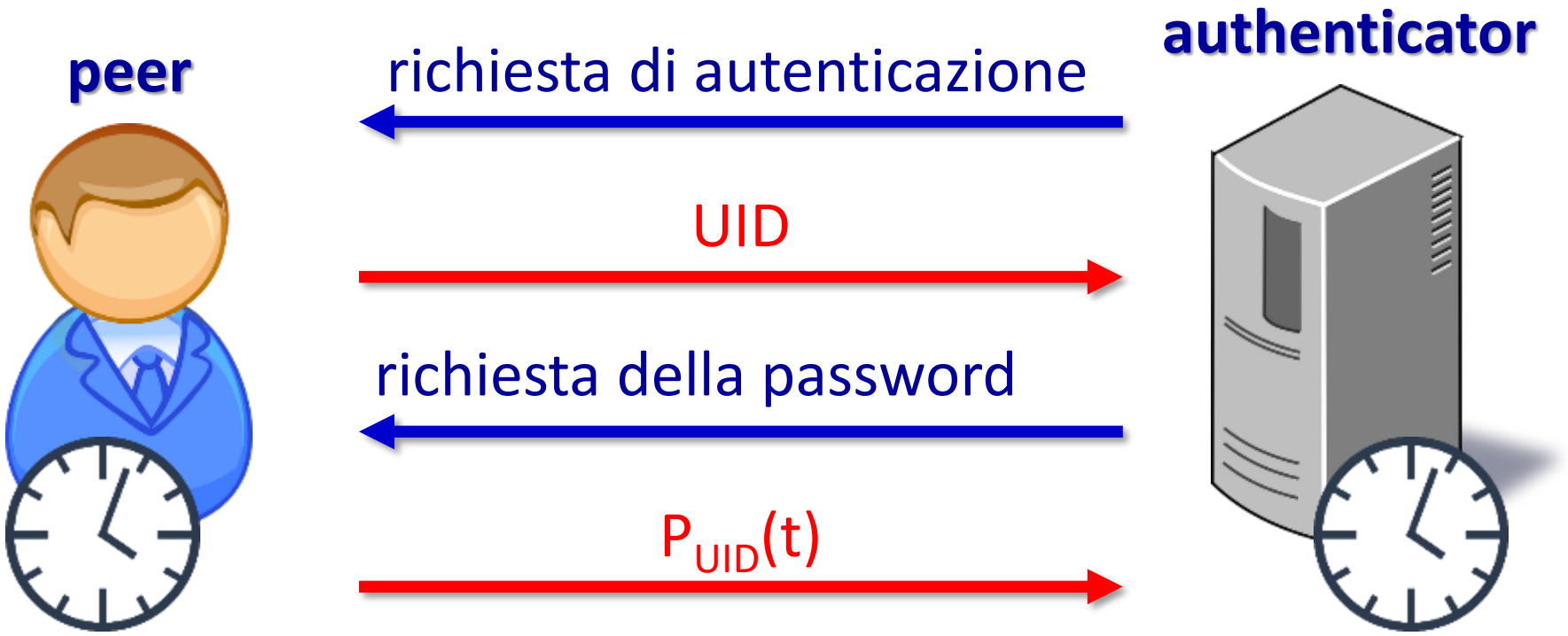


One-Time Password (OTP) - limiti

- Distribuzione delle password onerosa:
  - quantità di password
  - possibilità di esaurimento delle password

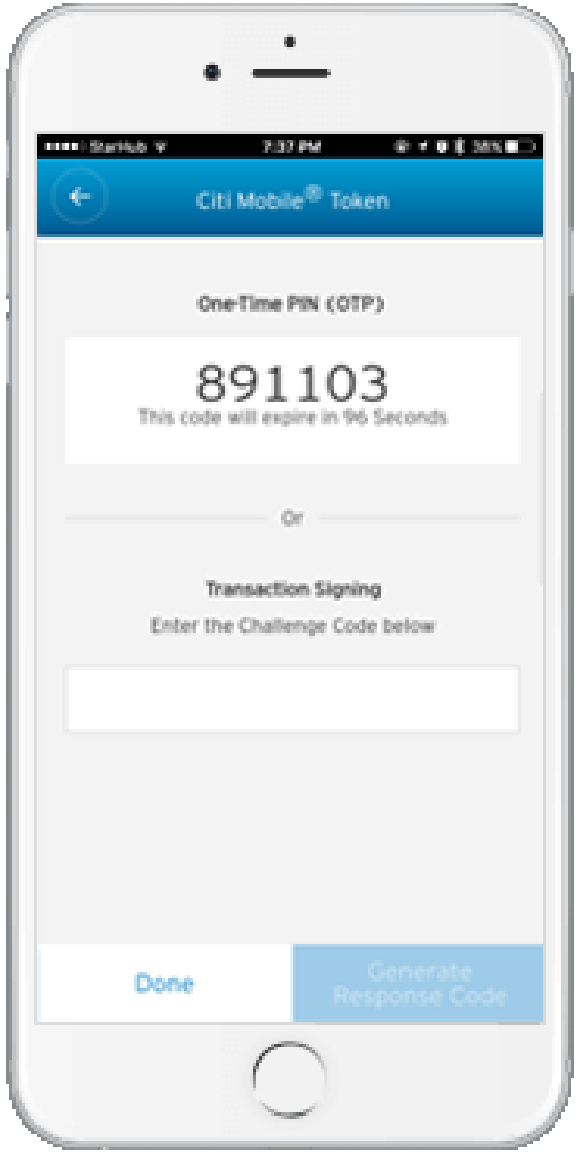
- **Soluzione**: generare le OTP tramite una funzione (Sistema S/KEY, con catena di Hash applicate sequenzialmente, che non approfondiamo), che però non risolve tutti i problemi…

OTP basata sul tempo



Il tempo può essere sostituito da un contatore

TOTP


Citi Mobile® Token
One-Time PIN (OTP)
891103
This code will expire in 96 Seconds
Or
Transaction Signing
Enter the Challenge Code below
Done
Generate Response Code


OTP c100
641966
ONE TIME PASSWORD