Autenticazione Utente: PAP, CHAP e One-Time Password (OTP)

Introduzione all'Autenticazione Utente

L'autenticazione ha lo scopo di dimostrare che si è effettivamente chi si dichiara di essere. Non va confusa con l'identificazione, che è semplicemente la presentazione di un'identità digitale, come un ID o una email. L'autenticazione richiede di provare il controllo di un segreto associato a quell'identità, per esempio una password di cui solo l'utente è a conoscenza.

Secondo la definizione del NIST (SP 800-63-3), l'autenticazione digitale è il processo di stabilire la fiducia nelle identità utente presentate elettronicamente a un sistema informativo.

Categorie di Autenticazione

Le forme di autenticazione comuni si basano su:

- **Qualcosa che conosci**: password, PIN, chiave segreta o risposta a domanda segreta.
- **Qualcosa che possiedi**: smart card, token fisici, dispositivi hardware unici (PUF - Physically Unclonable Function).
- **Qualcosa che sei**: dati biometrici statici come impronte digitali, retina, volto (con considerazioni sulla privacy).
- **Qualcosa che fai**: biometria comportamentale, ad esempio riconoscimento vocale, modalità di digitazione o movimenti del mouse.

Altri fattori includono parametri contestuali come la posizione o l'ora d'accesso.

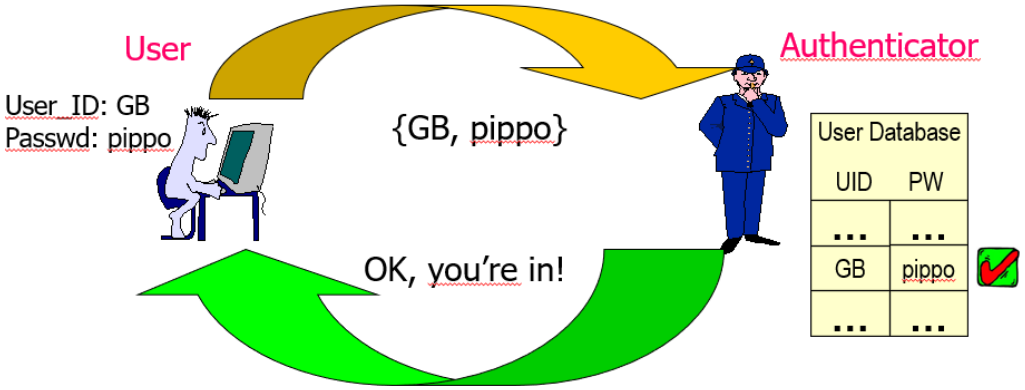
Prova della Conoscenza di un Segreto

Dimostrare di conoscere una password non significa necessariamente mostrarla in chiaro. Esistono vari protocolli con livelli diversi di esposizione:

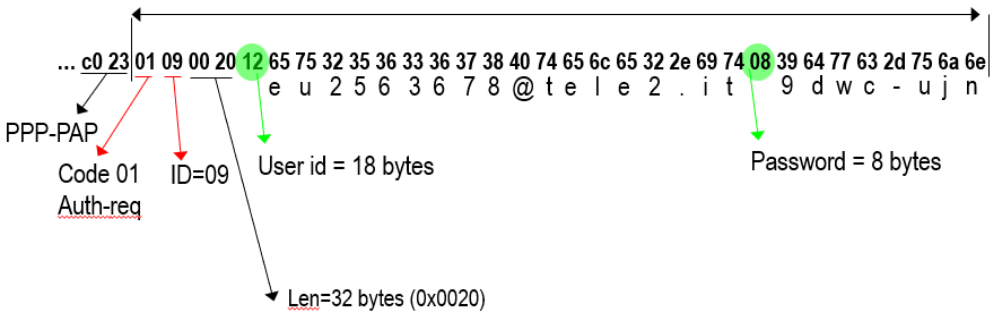
- **PAP (Password Authentication Protocol)**: la password viene inviata in chiaro al server. È il metodo più semplice, ma altamente insicuro in presenza di intercettazioni.
- **CHAP (Challenge Handshake Authentication Protocol)**: non si trasmette mai la password in chiaro. Il server invia una sfida (challenge), il client risponde con un valore calcolato in base alla password e alla sfida, tipicamente usando una funzione hash.
- **Zero-Knowledge Proof (ZKP)**: protocolli avanzati che permettono di dimostrare la conoscenza del segreto senza rivelarlo minimamente.

PAP: Vantaggi e Limitazioni

PAP è semplice: invia la password in chiaro. Ciò significa che se il canale di comunicazione è vulnerabile (es. reti wireless non protette), un attaccante può intercettarla facilmente. Non offre protezione dai replay attack (rilanci di messaggi) né da tentativi di forza bruta effettuati direttamente sul canale, poiché il controllo della frequenza e del tempo delle richieste è affidato interamente al peer autenticato.



Esempio:

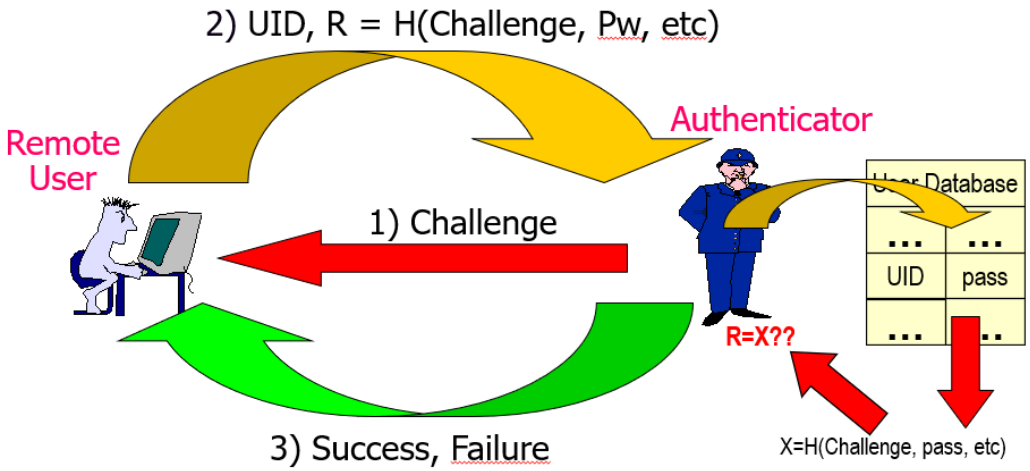


CHAP: Autenticazione con Sfide

CHAP migliora la sicurezza introducendo una *sfida* (challenge) casuale inviata dal server. Il client risponde con un valore hashato della sfida e della password:

$$\text{Risposta} = H(\text{SFIDA}||\text{password})$$

Questa strategia evita di trasmettere la password in chiaro e garantisce protezione contro l'intercettazione diretta.



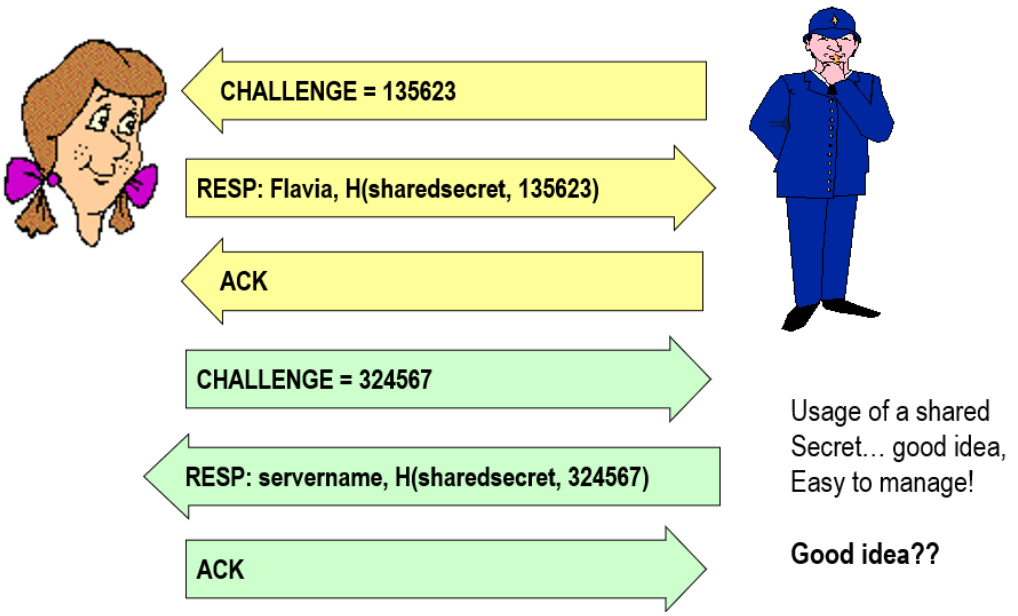
Punti chiave di CHAP:

- La sfida deve essere univoca e non ripetuta (non deve ripetersi mai la stessa sfida).
- Il server può ripetere la sfida più volte durante una sessione per assicurare sicurezza continua.

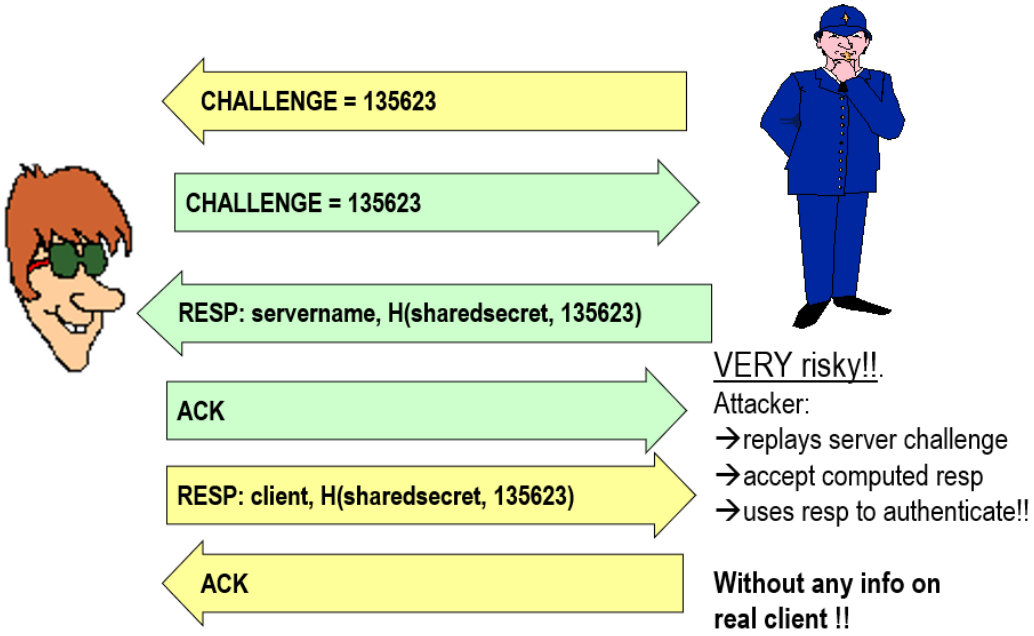
- CHAP richiede che il server conservi la password in forma chiara o reversibile, risultando vulnerabile a compromissioni backend.

Autenticazione Mutua

In alcuni scenari è importante che non solo il client si autentichi al server, ma anche il server si autentichi al client (autenticazione mutua). Ciò si ottiene coinvolgendo entrambe le parti in una serie di sfide e risposte.



Tuttavia, l'implementazione ingenua di questa procedura può permettere attacchi di *riflessione*, in cui un attaccante riutilizza messaggi di sfida per ingannare una delle parti. Le contromisure includono la variazione e l'ordine controllato delle sfide e risposte, e la dipendenza delle sfide stesse l'una dall'altra tramite concatenamento.

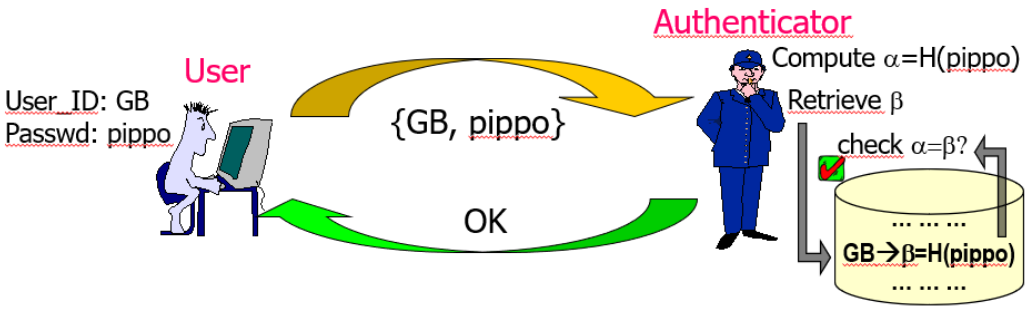


Password Hashed e Salatura

Conservare le password in chiaro è rischioso. La memorizzazione di hash delle password migliora la sicurezza. Usando un algoritmo di hash crittografico H si memorizza:

$$H(\text{password})$$

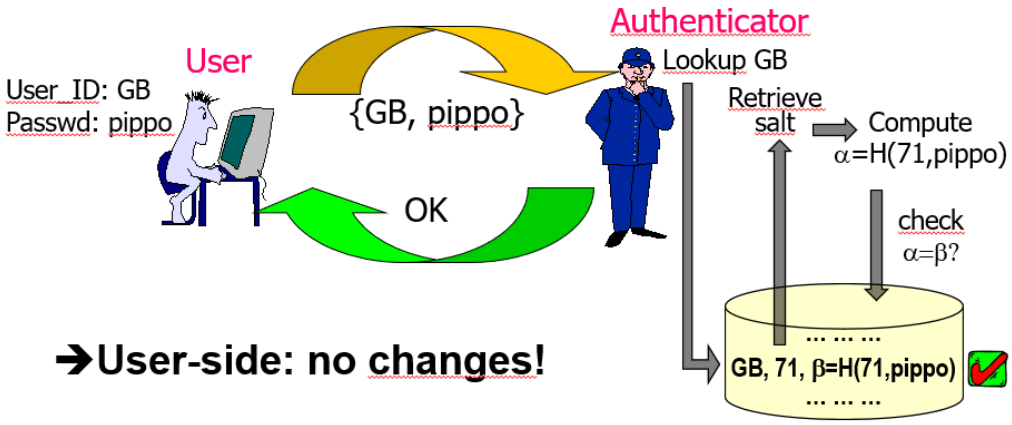
Tuttavia, gli hash semplici sono vulnerabili ad attacchi basati su tabelle precalcolate (*rainbow tables*).



Per renderli più resistenti si utilizza la *salatura* (*salt*), ovvero un valore casuale aggiunto alla password prima dell'hash:

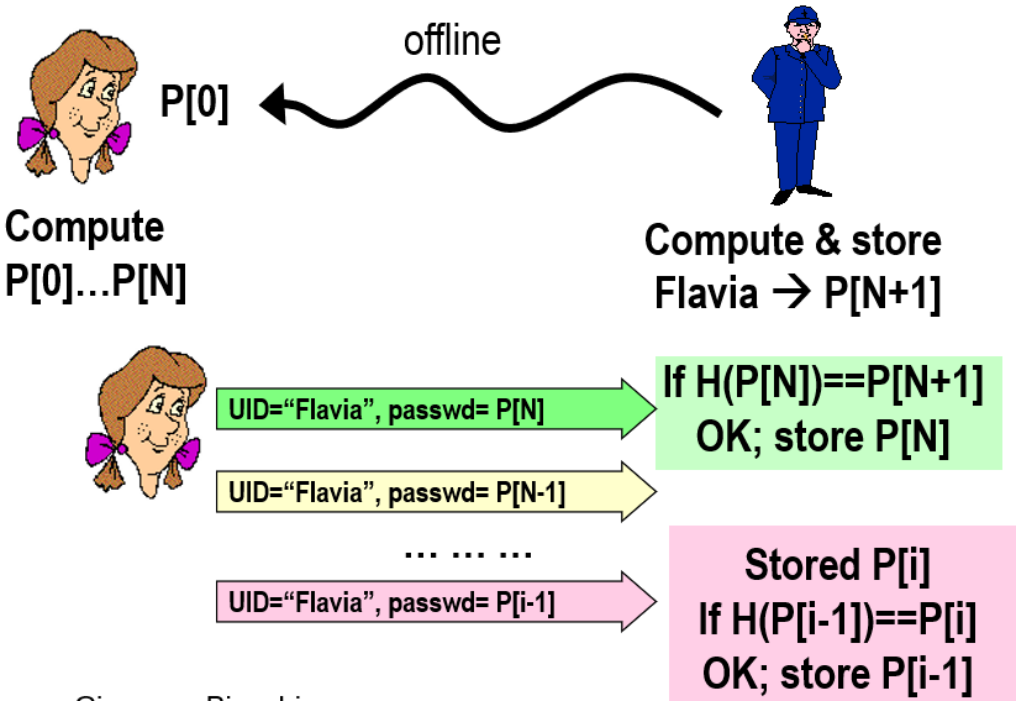
$$H(\text{salt}||\text{password})$$

Ciò rende inefficaci gli attacchi basati sul precomputo e consente di usare la stessa password con salt diversi.



One-Time Password (OTP)

Le password monouso (OTP) evitano il problema dei replay attack generando una nuova password a ogni autenticazione, spesso con un dispositivo esterno o un'app.



Le due varianti principali sono:

- **HOTP (HMAC-based One-Time Password)**: si basa su un contatore incrementale condiviso tra client e server.
- **TOTP (Time-based OTP)**: usa il tempo corrente come contatore, rendendo la password valida solo per brevi periodi.

Le OTP migliorano la sicurezza riducendo la finestra temporale in cui una password valida può essere usata da un attaccante.

Considerazioni Finali

Nessun protocollo di autenticazione è perfetto per tutti gli scenari. La scelta tra PAP, CHAP, OTP o altri metodi dipende da:

- La sicurezza del canale di comunicazione.
- Il livello di protezione desiderato contro attacchi client e server.
- Le risorse di calcolo e la complessità accettabile.

La sicurezza può essere migliorata combinando più fattori di autenticazione (multi-factor authentication) e proteggendo adeguatamente tutti i componenti del sistema.