11 - Cryptographic Techniques for Authentication

Per essere sicuri che la password sia sicura anche quando il canale di comunicazione tramite il quale viene trasmessa non è sicuro è necessario **on inviare mai** la password in plain-text. Si può fare uso di OTP (one time password).

Challenge-response general scheme

- L'utente U dichiara la sua intenzione di comunicare con l'host
- l'host sceglie una "*challenge*" e la manda a U
- U computa una "*risposta*" alla challenge la manda indietro all'host
- L'host compara la risposta ricevuta da U con quella aspettata per la challenge inviata
- Se sono uguali allora l'accesso è garantito, altrimenti no
- Questo è uno schema OTP visto che la "risposta" è unica per la challenge e può essere usata una volta sola (la "challenge" cambia sempre)

Challenge-response con crittografia simmetrica

- Utente U e host condividono una chiave K (password)
- U dichiara la sua intenzione di accedere all'host
- l'host genera una stringa random ***chal***, la salva e la manda ad U
- U calcola $\text{resp} = C_K(\text{chal})$ e invia il risultato all'host come response alla challenge
- L'host compara **chal** con $D_K(\text{resp})$
- Se sono uguali, accesso garantito, altrimenti no
- Visto che solo U (e host) conoscono K, l'autenticazione è assicurata

Challenge-response con crittografia asimmetrica

- l'host possiede un file con tutti le chiavi pubbliche degli utenti
- L'utente U dichiara la sua intenzione di accedere all'host
- l'host genera una stringa random ***chal***, la salva e la manda ad U
- U firma la challenge e la manda indietro all'host: $\text{resp} = \text{Sign}(\text{chal})$
- L'host verifica la firma $\text{Verify}(\text{resp})$
- se è valida- Se sono uguali, accesso garantito, altrimenti no
- le proprietà della firma digitale assicurano l'autenticazione

One-time-password: using “One-Way Hash” Function

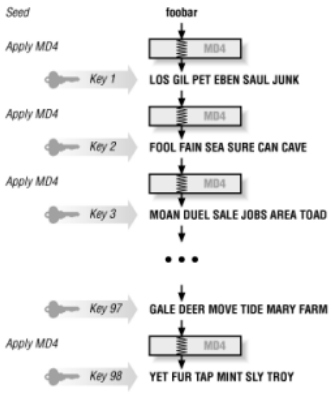
- L'host genera un numero casuale R per l'utente U
- l'host calcola $x_0 = R, x_1 = f(x_0), x_2 = f(x_1), x_3 = f(x_2) \dots$ dove f è la one way hash function
- U porta con se $x_0, \dots, x_{99}$, l'host salva (in chiaro) x_{100}
- Per accedere all'host, U deve mandare il suo nome e x_{99} (in chiaro)
- Host riceve (U, y) e calcola $f(y)$ e la compara con il valore salvaato per U (che è x_{100})
- se sono uguali, accesso garantito (host deve ricevere x_{99} altrimenti l'acesso è negato)
- U cancella x_{99} dalla propria lista, host rimpiazza x_{100} con x_{99}

Password Card.

E' la tessera, delle dimensioni di una carta di credito, contenente **40 password (codici numerici) monouso** necessarie per confermare le sue operazioni. Una volta immessa la password, l'operazione è autorizzata e la password in questione è automaticamente eliminata dal sistema (sono password "usa e getta"). La Password Card contiene 40 codici: prima che si esauriscano, sarà cura della Banca inviare una nuova Card, valida solo dopo l'esaurimento della precedente.

Esempi in pratica

- S/Key è un applicazione di questa idea che genera chiavi come digest di parole pronunciabili, che le rende più facili da leggere e scrivere



If the server knows *Key* 98, for example, then S/Key prompts the user for *Key* 97, takes user's response, and runs MD4 over it; if the user has provided the correct *Key* 97, the result will be *Key* 98 that the server knows. The server lets the user in, and remembers *Key* 97 for the next time.

- OTP possono essere fisici security tokens
  - Basati su implicite "challenge", normalmente real time (minuti)
  - Il token calcola la "risposta" come $f(t|k)$ dove f è la one way hash function, t è il tempo reale (in minuti) e k è una secret key buit in nel token (che è associato con l'account bancario dell'utente)

Strong authentication

Un'autenticazione forte è quando si combinano due tra:

- qualcosa che si conosce
- qualcosa che si ha
- qualcosa che si è
  Normalmente sono i primi due.
- Un oggetto fisico come il bancomat, cellulare o security token
- assieme al PIn o password (quclosa che si conosce)

OTP più login/password crea una forma di "strong authentication" conosciuta anche come "2-step-verification" o 2FA