Cybersecurity: Cryptographic Techniques for Authentication

Ozalp Babaoglu

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

Identification through insecure channels

- What if the communication channel through which the user inserts a password (Internet) is not secure?
- Either make sure that password is never transmitted in the clear (as plain-text)
- Or use "One-time-password" (OTP) (also know as "One-shot-password") scheme whereby a password becomes useless after it has been used once
- Usually based on symmetric or asymmetric cryptography

Challenge-Response: General scheme

- User U declares her intention to access the host
- Host selects a "challenge" and sends it to U
- U computes a "response" to the challenge and sends it back to the host
- Host compares the response received from U with the response "expected" for the challenge it sent
- If they match, access granted, otherwise no
- This is an OTP scheme since "response" is unique for the challenge and can be used only once (because the "challenge" changes each time)

Challenge-Response: using Symmetric Cryptography

- User U and host share a secret key K (password)
- User U declares her intention to access the host
- Host generates a random string $chal$ and sends it to U (and remembers $chal$)
- U computes $resp = C_K(chal)$ and sends it to the host as the response to the challenge
- Host compares $chal$ with $D_K(resp)$
- If they match, access granted, otherwise no
- Since only U (and host) knows K, authentication is assured

Challenge-Response: using Asymmetric Cryptography

- Host keeps a file of every user's public key
- User U declares her intention to access the host
- Host generates a random string *chal* and sends it to U
- U signs the challenge and sends it back to host as the response: $resp=Sign(chal)$
- Host verifies the signature: $Verify(resp)$
- If it is valid, access granted, otherwise no
- Property of digital signatures assures authentication

One-time-password: using "One-Way Hash" Function

- Host generates a random number R for user U
- Host computes $x_0 = R$, $x_1 = f(x_0)$, $x_2 = f(x_1)$, $x_3 = f(x_2)$, … where f is a one-way hash function
- U takes with her $x_0, \ldots, x_{99}$, the host stores (in the clear) x_{100}
- To access host, U sends her name and x_{99} (in the clear)
- Host receives (U, y) and computes $f(y)$ and compares with the value stored for user U (which is x_{100})
- If they match, access granted (host ***must*** have received x_{99}), otherwise access denied
- U crosses off x_{99} from her list, host replaces x_{100} with x_{99}

One-Time Passwords in Practice

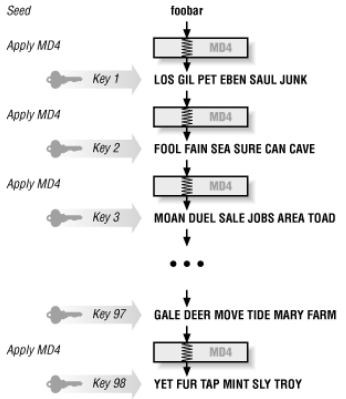
- Dal sito UniCredit Banca SpA

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.

One-Time Passwords in Practice

- S/Key is an implementation of this idea that generates keys as the digest of pronounceable words, so they are easier for a user to read and type



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.

One-Time Passwords in Practice

- From facebook.com

One-Time Passwords

What's a one-time password and how do I get one?

You can use a one-time password to log into your account anytime you feel uncomfortable entering your real password on Facebook (ex: in a library or internet cafe). Here's how:

1. If you're in the US, send a text message to 32665 with the message **otp**. If you're not in the US, check out this list to see which mobile carriers support this feature and what number you should use.
2. If your mobile number is already linked to your Facebook account, we'll reply with a unique, 8-character temporary password. If you haven't added this mobile number to your account, we'll send you an email with instructions on how to add it and collect your code.
3. Once you get your code, just enter it in the **Password** section of the Facebook login page.

Your one-time password will be valid for 20 minutes and can't be reused.

One-Time Passwords in Practice

- One-time passwords can be packaged as physical "security tokens"
- Based on implicit "challenge", usually real time (in minutes)
- Token computes "response" as $f(t \mid k)$ where f is a one-way hash function, t is the real time (in minutes) and k is a (secret) key built into the token (which is associated with user's account when the bank issues the token)

One-Time Passwords in Practice

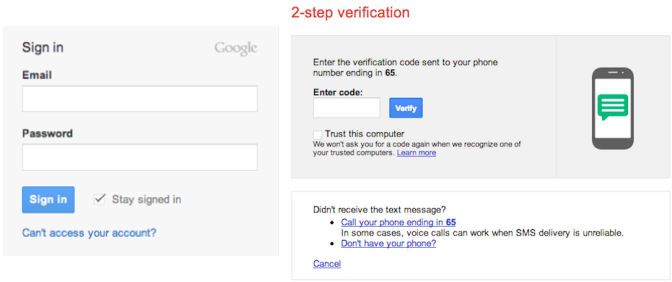
- Today, physical security tokens are often replaced with "Mobile Token" apps running on smart phones

Strong Authentication

- *Strong authentication* combines any two of:
  - Something you know
  - Something you have
  - Something you are
- A combination of the first two is the most common
  - A physical object such as Bancomat, cell phone (with a given SIM) or "security token" (something you have)
  - Augmented with a PIN or password (something you know)

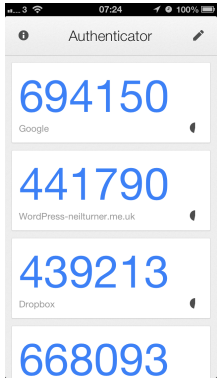
2-Step Verification

- OTP used in combination with login/password to create a form of "strong authentication" is known as "2-step verification" (or "2-factor authentication")
- Google example: OTP sent as a text message (SMS)

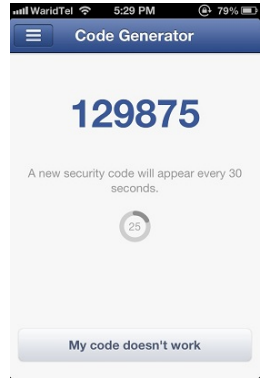


2-Step Verification

- OTPs can also be generated using "Authenticator" apps



Google



Facebook

"Something that you have" Physical Security Keys

- Look like USB dongles (memory sticks)
- Act much like keys to physical doors/locks
- Render stolen passwords worthless
- Example: *Google Physical Keys* as part of its *Advanced Protection Program*