Week 8: Authentication: CAPTCHA
Combined Authentication & Key Exchange
Goals

• Authenticate principals
• Distribute a session key to both securely
• Principals can communicate only if they are properly authenticated

Authentication relies on proving you know your secret key
Symmetric Key Authentication & Key Exchange

- We use a trusted third party (Trent) who knows all the keys

I’d like to talk with Bob

Request a session key

Get encrypted key + ticket
We use a trusted third party (Trent) who knows all the keys.

Send the ticket (key):

Let's talk:
Guard against replay attacks

- **Needham-Schroeder: add nonces in encrypted messages**
  - Random numbers will be different with different sessions
  - Messages from old sessions will be rejected

Guard against attacker who knows an old session key

- **Add timestamps in encrypted messages**
  - Attacker's replayed messages will have an older timestamp – and be rejected

- **Add IDs (sequence numbers) in encrypted messages**
  - Attacker's replayed messages will have the wrong number – and be rejected
Public Key Authentication & Key Exchange

- No need for a third party – public keys can reside in X.509 certificates
- Prove you have your private key

Bob, can you encrypt this random number with your private key?

Bob's Certificate

$D_B(r_1) \equiv r_1$

Alice is convinced Bob has Bob's private key
Public Key Authentication – mutual authentication

- No need for a third party – public keys can reside in X.509 certificates
- Prove you have your private key

Alice, can you encrypt this random number with your private key?

Bob is convinced Alice has Alice's private key

Bob's Certificate

$D_A(r_2) \equiv r_2$
Public Key Authentication – key exchange

- Encrypt a session key with the other party's public key.

Here's a session key we can use

Let's talk
User Authentication
### Three Factors of Authentication

<table>
<thead>
<tr>
<th>Factor</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Ownership</strong></td>
<td>Key, card</td>
</tr>
<tr>
<td>Something you have</td>
<td></td>
</tr>
<tr>
<td><strong>2. Knowledge</strong></td>
<td>Passwords, PINs</td>
</tr>
<tr>
<td>Something you know</td>
<td></td>
</tr>
<tr>
<td><strong>3. Inherence</strong></td>
<td>Biometrics (face, fingerprints)</td>
</tr>
<tr>
<td>Something you are</td>
<td></td>
</tr>
</tbody>
</table>
User authentication protocols

• **Password Authentication Protocol (PAP)**
  – User: \{ name, password \}
  – Server: $lookup(name) \neq password$

• **Hashed password storage**
  – User: \{ name, password \}
  – Server: $lookup(name) \neq hash(password)$

• **Hashed passwords with salt**
  – User: \{ name, password \}
  – Server: $lookup(name) \Rightarrow salt, stored_{password} \quad hash(stored_{password}) \neq hash(salt || password)$
One-time passwords

• **Sequence-based**
  - **S/key:**
    - $P_1 = \text{hash}(R)$, $P_2 = \text{hash}(P_1)$, $P_3 = \text{hash}(P_2)$, $P_4 = \text{hash}(P_3)$, …
  - User: { name, $P_n$ }
  - Server:
    - $\text{lookup(name)} \equiv \text{hash}(P_n)$
    - update database: name.password = $P_n$

• **Challenge-Handshake Authentication Protocol (CHAP)**
  - **Server**: challenge
  - **Client**: hash(challenge, secret)
  - Server hash(challenge, stored_secret) \equiv client_response
One-time passwords

- **Time-based One-Time Password**
  - User: \{ name, client_password=${hash}(secret, time) \}
  - Server:
    - \( hash(lookup(name).secret, \text{time}) \triangleq \text{client_password} \)

- **Hash-based One-Time Password**
  - User: \{ name, client_password = $hash$(secret, token_id, counter) \}
  - Server:
    - Server: \( lookup(name) \Rightarrow \text{stored_secret, stored_token_id, stored_counter} \)
    - \( hash(\text{stored_secret, stored_token_id, stored_counter}, \text{time}) \triangleq \text{client_password} \)
    - update database: name.counter = name.counter + 1
Biometric Authentication

• **Pattern matching**
  – Set thresholds to determine if the match is close enough

• **False Accept Rate (FAR)**
  – Non-matching pair of biometric data is *accepted* as a match

• **False Reject Rate (FRR)**
  – Matching pair of biometric data is *rejected* as a match

• **Balance security (low FAR) vs. convenience (low FRR)**
CAPTCHA: Detecting Humans
Gestalt Psychology (1922-1923)

• Max Wertheimer, Wolfgang Köler, Kurt Koffka

• Laws of organization
  – Proximity
    • We tend to group things together that are close together in space
  – Similarity
    • We tend to group things together that are similar
  – Good Continuation
    • We tend to perceive things in good form
  – Closure
    • We tend to make our experience as complete as possible
  – Figure and Ground
    • We tend to organize our perceptions by distinguishing between a figure and a background
Objects on Mars?

Elvis

Face

Female statue
Gestalt Psychology: text continuity
HELLO
Authenticating humanness

Battle the Bots
  – Create a test that is easy for humans but extremely difficult for computers

CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans Apart
  – Image Degradation
    • Exploit our limits in OCR technology
    • Leverages human Gestalt psychology: reconstruction

Origins
  – 1997: AltaVista – prevent bots from registering URLs with the search engine
  – 2000: Yahoo! and Manuel Blum & team at CMU
    • EZ-Gimpy: one of 850 words
  – Henry Baird @ CMU & Monica Chew at UCB
    • BaffleText: generates a few words + random non-English words
Microsoft

See captchas.net
They’re getting harder
Problems

• **Accessibility**
  - Visual impairment → audio CAPTCHAs
  - Deaf-blind users are left out

• **Frustration**
  - OCR & computer vision has improved a lot!
  - Challenges that are difficult for computers may be difficult for humans

• **Attacks**
  - Man in the middle attacks
    • Use human labor – CAPTCHA farms
  - Automated CAPTCHA solvers
    • Initially, educated guesses over a small vocabulary
Alternate approaches

• MAPTCHAs = math CAPTCHAs
  – Solve a simple math problem

• Puzzles, scene recognition
Alternate approaches
No premium user. Please enter the one that can NOT be created from the unfolded pattern. 29 seconds remain.
Qualifying question

Just to prove you are a human, please answer the following math challenge.

Q: Calculate:
\[
\frac{\partial}{\partial x} \left[ 6 \cdot \sin \left( x - \frac{\pi}{2} \right) + 3 \cdot \cos \left( 2 \cdot x - \frac{\pi}{2} \right) \right] \bigg|_{x=\pi}
\]

A: 

Note: If you do not know the answer to this question, reload the page and you'll (probably) get another, easier, question.
• Ask users to translate images of real words & numbers from archival texts
  – Human labor fixed up the archives of the New York Times

• Two sections
  – (1) known text
  – (2) image text
  – Assume that if you get one right then you get the next one correct
    • Try it again on a few other people to ensure identical answers before marking it correct

• Google bought reCAPTCHA 2009
  – Used free human labor to improve transcription of old books & street data

2014: Google found that AI could crack CAPTCHA & reCAPTCHA images with 99.8% accuracy
NoCAPTCHA reCAPTCHA

Just ask users if they are a robot

- Reputation management
  - “Advanced Risk Analysis backend”
  - Check IP addresses of known bots
  - Check Google cookies from your browser
  - Considers user’s engagement with the CAPTCHA: before, during, and after
    - Mouse movements & acceleration, precise location of clicks

- Newest version: invisible reCAPTCHA
  - Don’t even present a checkbox
If risk analysis fails,
- Present a CAPTCHA
- For mobile users, present an image identification or labeling problem
Other approaches: Text/email verification

- **Text/email verification**
  - Ask users for a phone # or email address
  - Similar to two-factor authentication but we're not authenticating the user
  - Service sends a message containing a verification code
    - Still susceptible to spamming & automation
    - Makes the process more cumbersome
    - Requires users to disclose some information

- **Measure form completion times**
  - Users take longer than bots to fill out and submit forms
  - Measure completion times
    - Bots can program delays if they realize this is being done
The End.