ECS 89

5/30

Announcements

- Next assignment due Tu June 3
- Final in this room, Wds June 11, 8AM

- Today:
  - Sound
  - Security

Getting sounds

- The Audio HTML element holds a sound clip, just like the Img element holds a picture.
- Like Canvas, this is new in HTML5.
- So to use this, you need an audio clip.
- One place to look – SoundBible.com
- Formats - .wav, .mp3. I used .mp3 but I think .wav would have worked.

An Audio object in Javascript

```
var bell1 = new Audio("Bell.mp3");
```

- The variable bell1 contains an Audio object.
- This is a special case of an HTMLMediaElement, which also includes video!
- One method: play!

```
bell1.play()  // rings the bell
```

Issue in Explorer

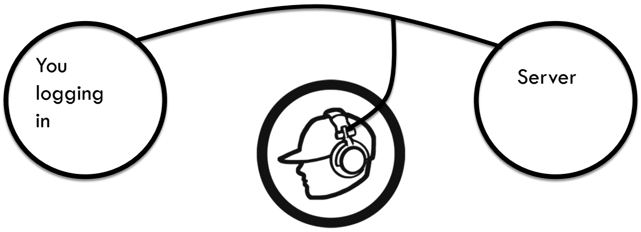
- What is the problem?
- How to fix?

Security

- So far nothing we have done is secure
- Anybody can go onto our Web sites and put information into our databases; we are only checking that the format is correct
- If we cared about our server data (eg. users private data, financial data, a service we are trying to sell…) we need to control access

Login

- We need to get users to log in before allowing them access to server data.
- Eavesdropper attack: a computer "listening" to the login process can learn your password.



Encryption

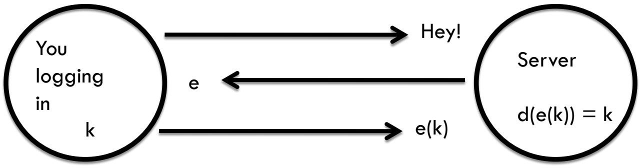
- HTTPS – the S is for Secure
- All communication between you and the server is encrypted
- Over-simplified encryption example: add k to the unicode for every letter. So if k = 3 and my password was "abc", I would send "def"
- Eavesdropper sees "def", not my password
- Server decrypts by subtracting k=3, getting "abc"

The session key k

- Very important that I know k, and the server knows k, but the eavesdropper does not!
- Need to establish k **before** the log-on process
- Keep k until session is over, eg. until browser is closed
- "Handshake" protocol when first accessing the server over HTTPS to establish k

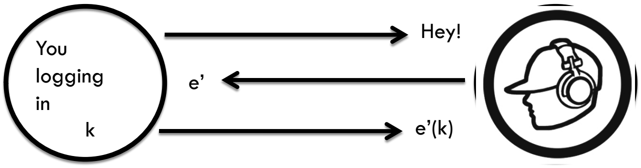
How to establish k?

- Use public-key encryption.
- Scheme with two keys, e for encryption and d for decryption. Server keeps d secret, but not e.
- Idea: (WAY oversimplified!)



Complication

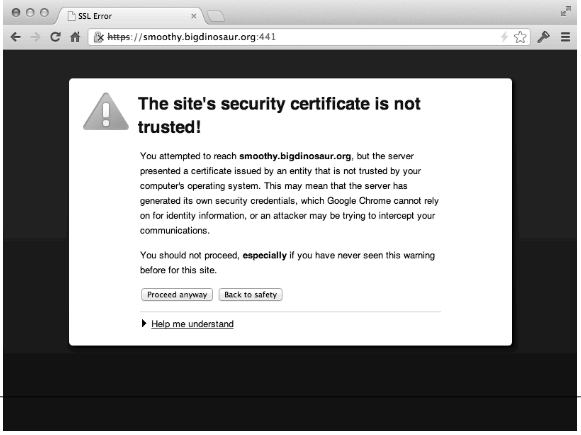
- Someone could pretend to be the server and hand out bogus e' keys
- And then you give them your password…



Certificates

- A Certification Authority publishes guarantees that the public key of the server is indeed the right one for that server
- Server has to pay for this service!
- Browsers have a list of Certification Authorities that they trust

Invalid Certificate Web page



The site's security certificate is not trusted!

You attempted to reach **smoothy.bigdinosaur.org**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Google Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

You should not proceed, **especially** if you have never seen this warning before for this site.

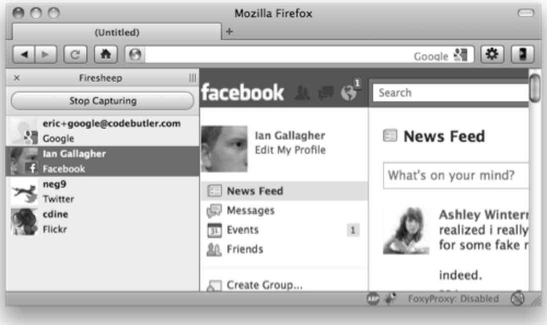
Common model

- HTTPS is clearly needed for login
- Banks, purchases, etc. then use the private key for the rest of the session
- Some Websites – including Facebook in its default settings - then use regular HTTP for subsequent transactions
- Cookie stored in browser is sent with every message to let the server know which session this is

Firesheep

- Install this Firefox app, and visit your local coffee shop
- Steals cookies as they go by

FIresheep



Issues holding back more HTTPS

- Cost of certificates
- Virtual hosting
- Disables caching in the network, which can slow things down
- Makes servers run slower
- More complication in general (why we did not do it)

- Any server transaction assuming privacy should be using HTTPS