Engineering Secure Software

APPLIED CRYPTOGRAPHY PART 2

Recap

- Symmetric key:
  - Benefit: fastest, mathematically the strongest
  - Drawback: distributing the keys
- Public key:
  - Benefit: Easier to distribute the keys
  - Drawback: Trusting public keys is tricky

SSL: Secure Sockets Layer

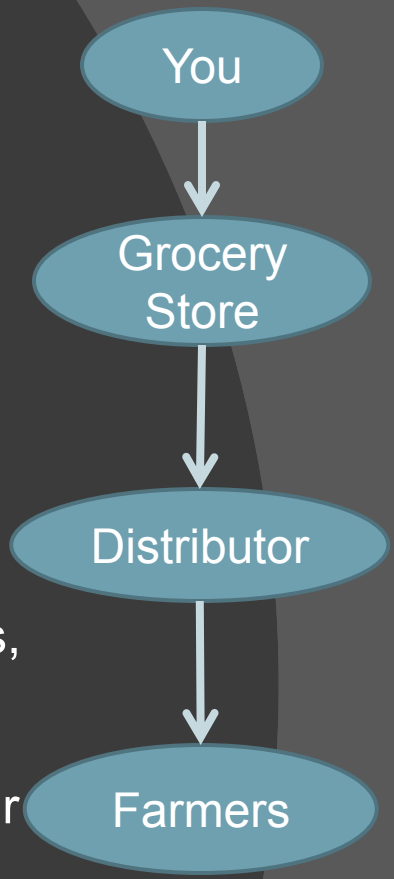
- SSL (and TLS) are the public-key encryption standards today
  - Protocols suffixed with “s” : https, ftps, etc.
  - Another algorithm implementation best left to the experts
- Untrusted public keys?
  - For ~$30/year, you too can get your public key signed!!
    - Seriously, this is how it works
    - e.g. Verisign & GoDaddy are “ceritificate authorities” (CA)
    - Thus, trust the public key != trust the website
  - Self-signed certificate?
    - Not usually a good idea to accept them, but…
    - If the key changes, you will be alerted
    - So you only need to trust the server *once*

Pretty Good Privacy

- An open protocol created in 1991
  - Primarily used for email encryption today
  - Very popular in open source culture
- Combines symmetric-key and public-key cryptography
  - Symmetric is much faster and harder to crack than public-key
  - Use public-key to distribute the symmetric key
  - Untrusted recipient now has your symmetric key?
    - One-time symmetric key only
    - Use a secure PRNG to generate symmetric keys

PGP Web of Trust

- How do you trust PGP public keys?
  - There are no PGP “Certificate Authorities”
  - Public key databases are open
- How do you know that the food you’re eating is disease-free?
  - You trust the grocery store, who trusts the distributors, who trust the farmers
  - FDA is also a trusted third party
  - But, when you trust the farmers directly, you trust their food more
- In the same way, PGP incentivizes short trust chains
  - Each person can “sign” someone else’s key, connecting you to them in the web of trust
  - Each “hop” diminishes the trust of a given public key



PGP Mean Shortest Distance

- How trusted should this key be?
  - Geodesic paths (shortest paths)
  - Compare the mean geodesic distance to the entire network mean
  - “Closeness” in social network analysis
- Relatively trusted by the community?
  - Many will trust you
  - You are trusted by people who trust you
- Low MSD? Not as relatively trusted
  - Fewer people trust you
  - Then less-trusted people trust you

- *Alt text: if you want to be extra safe, check that there's a big block of jumbled characters at the bottom.*
- http://xkcd.com/1181/

Cryptanalysis

- Definition: “the analytic investigation of an information system with the goal of illuminating hidden aspects of that system” [NSA.gov]
- In other words: breaking cryptography
- Comes in many forms
  - Brute force attacks
  - Theoretical/Algorithmic weaknesses
  - Side-channel attacks

Side Channel Attacks

- Side channel
  - Information emitted from a *physical implementation* of a cryptosystem
- Side channel vulnerabilities are mutually exclusive from algorithmic vulnerabilities
  - Although coding vulnerabilities can lead to side channel attacks
- e.g. Password fields obscure the text to prevent someone from looking over your shoulder
- e.g. Keeping the sticky on your monitor

Timing attacks

- Use the timing of an operation to gain information
- e.g. computing large prime numbers for SSL
  - Constant concern for OpenSSL
    CVE-2013-0169
  - “Square and multiply” algorithm
- e.g. timing for checking for a password
- e.g. cache-hit vs. cache-miss on a sensitive record

Data Remanence

- Deleted data is not always deleted
  - Hard drives release the memory, but it's not necessarily overwritten
  - Magnetic fields can remain even after it's been overwritten
- Many, many creative ways to do this…
  - Freezing RAM with liquid nitrogen
  - Hibernation files
  - Core dumps

So many more...

- Power monitoring attacks
  - Can predict which branch of an if-statement was taken by monitoring power
  - Particularly nasty on embedded devices
  - Even AES can be broken this way
- Acoustic analysis of hard drive sounds
- “Chatter” - even the known existence of encrypted communication can be useful information

Lessons from Side Channels

- Okay, so what?
  - Can we even do anything about this?
  - What must software engineers do?
- Lesson 1: Identify your side channels
  - Network chatter, timing, power, etc.
- Lesson 2: You have not identified all of your side channels
- Lesson 3: Better testing
  - Realistic production environments
  - Third-party testers with security experience

Keeping Up

- Networking & crypto algorithms are constantly changing
  - New networking protocols, new models
  - Broken crypto algorithms
- You will need to keep up with the news on algorithms
  - Organizations: CWE, OWASP
  - Bloggers & Researchers
    - Bruce Schneier: http://www.schneier.com/
    - Steve Gibson: http://www.grc.com/news.htm
    - Gary McGraw: www.cigital.com, IEEE Privacy & Security