Data Ethics, Security and Privacy

Jennifer Helsby, Eric Potash

Computation for Public Policy

Lecture 9: February 2, 2016

computationforpolicy.github.io

Announcements

- Homework 3 out on Thursday
- Final project
  - Description up on website

`https://computationforpolicy.github.io/assignments/final.html`

  - Proposal due on Friday, Feb 12th at 5pm

Today

- Privacy and security 101
  - Protecting user data
  - Keeping systems secure
- Data ethics issues

Why we care

- Developing web applications
- Collecting data from third party websites e.g. via scraping
- Collecting and using personal data
- Setting up servers and deploying code
- Social science experiments involving humans

tl;dr

Don’t access other’s data, computers, or networks without permission

Don’t violate people’s privacy

Computer Fraud and Abuse Act (CFAA)

- US federal anti-hacking law
- Very broad scope, notoriously unclear
- Can violate CFAA by:
    - trespassing into a protected computer
    - exceeding authorized access

I am not a lawyer, this does not constitute legal advice

Example: Web Scraping

- Rapid web scraping (especially if done across multiple machines) can put a burden onto a site, especially a smaller provider

Time of Troubles:

- Rapid scraping
  - Denial of Service attack
- Circumventing blocking
  - People can get quite upset, potential CFAA violation
- Publishing scraped content
  - Copyright implications

Security

Security?

- Security: Developing robust systems in the presence of adversaries
- “A system is secure if it behaves precisely in the manner intended - and does nothing more” - Ivan Arce
- **Security is not a boolean value**

No such thing as absolute security!

- Consider your home. Do you lock your home door?
- Security Mindset: Rational paranoia

Why security is hard

- People love new features
- Huge number of lines of code in modern applications and OSes
  - e.g. ~10 M lines of code in the Linux Kernel
- Exploding complexity of systems
  - Complexity is the enemy of security
- High security applications can be difficult to use
- Few people care
  - Until something happens

What are we trying to protect?

- User data
- Infrastructure

Information Security Properties

- Confidentiality
  - Information is secret; only authorized people can access it
- Authenticity
  - Information comes from who you think it came from
- Integrity
  - Information has not been tampered with
- Reliability
  - Systems remain up
- Also: anonymity (actions not linked to identity), ...

Who are you trying to protect against?

Adversaries

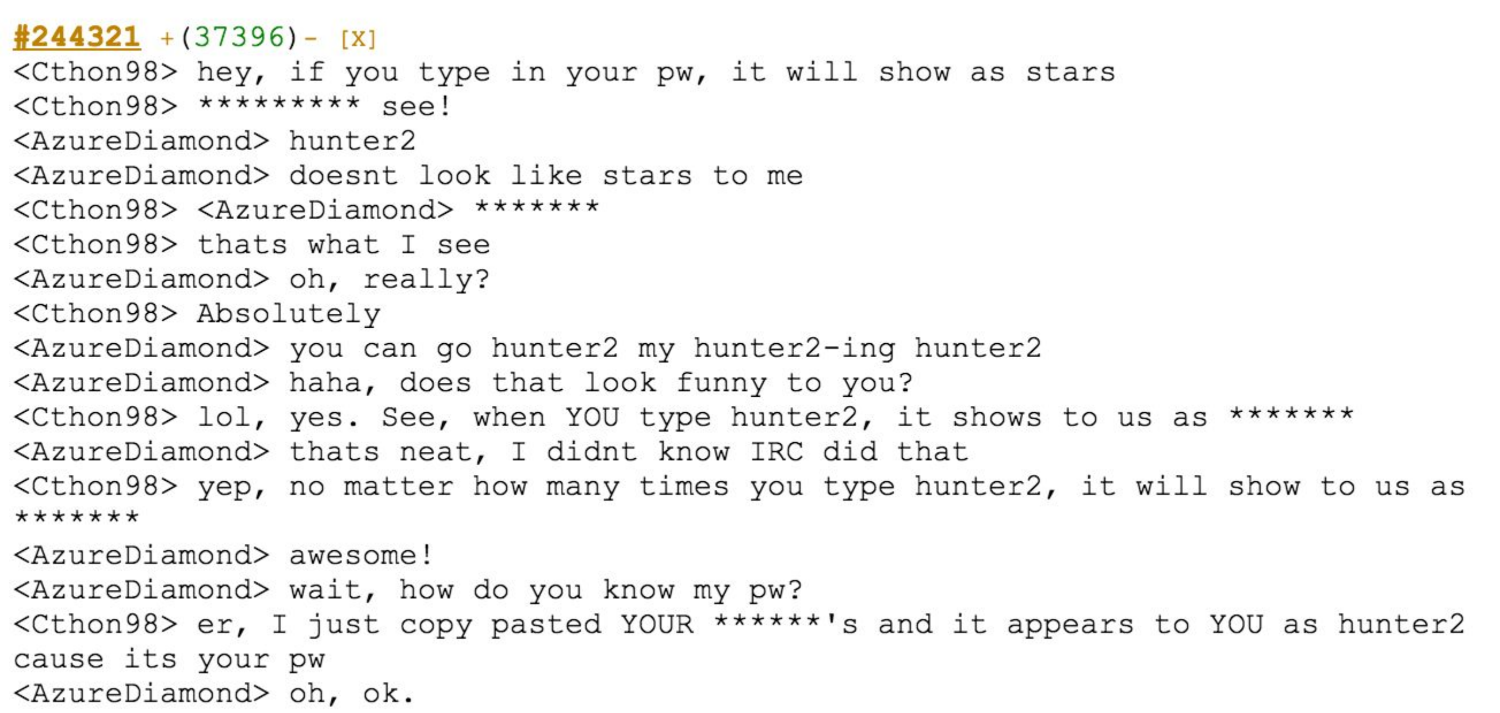
- Security researchers (industry)
  - Goal: Fix problems, notoriety, cash
- Security researchers (academia)
  - Goal: Fix problems, papers
- Individuals
  - Goal: lulz, notoriety, political
- Criminal groups
  - Goal: cash
- Governments
  - Goal: espionage, control of population

What can they do?

Classes of attacks

- Social engineering
  - Manipulating humans into revealing sensitive information
  - e.g. Call someone up, pretend to be sysadmin, ask them their password

Social Engineering



```
#244321 +(37396)- [X]
<Cthon98> hey, if you type in your pw, it will show as stars
<Cthon98> ********* see!
<AzureDiamond> hunter2
<AzureDiamond> doesnt look like stars to me
<Cthon98> <AzureDiamond> *******
<Cthon98> thats what I see
<AzureDiamond> oh, really?
<Cthon98> Absolutely
<AzureDiamond> you can go hunter2 my hunter2-ing hunter2
<AzureDiamond> haha, does that look funny to you?
<Cthon98> lol, yes. See, when YOU type hunter2, it shows to us as *******
<AzureDiamond> thats neat, I didnt know IRC did that
<Cthon98> yep, no matter how many times you type hunter2, it will show to us as
*******
<AzureDiamond> awesome!
<AzureDiamond> wait, how do you know my pw?
<Cthon98> er, I just copy pasted YOUR ******'s and it appears to YOU as hunter2
cause its your pw
<AzureDiamond> oh, ok.
```

bash.org

What can they do?

Classes of attacks

- Social engineering
  - Manipulating humans into revealing sensitive information
  - e.g. Call someone up, pretend to be sysadmin, ask them their password
- Software with malicious intent - “malware”
  - Gather sensitive info, extort user, disruption e.g. Cryptolocker, Stuxnet

What can they do?

Classes of attacks

- Social engineering
    - Manipulating humans into revealing sensitive information
    - e.g. Call someone up, pretend to be sysadmin, ask them their password
- Software with malicious intent - “malware”
    - Gather sensitive info, extort user, disruption e.g. Cryptolocker, Stuxnet
- Web application attacks
    - Vulnerabilities that often occur due to improper checking (“sanitizing”) of user input
- Phishing
    - Attempt to get sensitive information by pretending to be a trusted entity e.g. Nigerian prince

What can they do?

Classes of attacks

- Social engineering
  - Manipulating humans into revealing sensitive information
  - e.g. Call someone up, pretend to be sysadmin, ask them their password
- Software with malicious intent - “malware”
  - Gather sensitive info, extort user, disruption e.g. Cryptolocker, Stuxnet
- Web application attacks
  - Vulnerabilities that often occur due to improper checking (“sanitizing”) of user input
- Phishing
  - Attempt to get sensitive information by pretending to be a trusted entity e.g. Nigerian prince
- Network attacks
  - Passive: e.g. Packet sniffing - capturing traffic as it crosses the network
  - Active: Man in the middle (MitM)
- OS/application attacks
  - Due to software bugs that can lead to unintended outcomes with security implications

What happens if they succeed?

Impacts

- Financial
- Data
- Privacy
- Downtime
- Identity theft
- Legal implications

What can we do to stop them?

Countermeasures

- Use *cryptography*: use codes to communicate securely in the presence of a 3rd party
- Apply regular *security updates*:
  - Known vs. 0day
- Use *strong passwords*:
  - Manage with password manager
- Use *two-factor authentication* (2FA) where possible:
  - Makes it more difficult to compromise

What can we do to stop them?

Countermeasures

- Use *cryptography*: use codes to communicate securely in the presence of a 3rd party
- Apply regular *security updates*:
  - Known vs. 0day
- Use *strong passwords*:
  - Manage with password manager
- Use *two-factor authentication* (2FA) where possible:
  - Makes it more difficult to compromise
- *Security Theater*: Countermeasures that make people “feel” secure but offer little or no security benefit
  - e.g. TSA

Privacy

Privacy Matters

- Important consideration when handling user data
- *Privacy*: The right to protect your personal information
- Respecting privacy gives users control

Personally Identifiable Information (PII)

- Any information that can be used to identify a person
  - Name, social security number, birthdate, address, …
  - Biometrics (face, fingerprint, gait), genetic information, …
- In some contexts, legal requirements for handling PII exist
  - FERPA, HIPAA
- Redaction, de-identification, anonymization
  - A tricky business: Netflix prize

Respecting Privacy

- *Security:* Do not collect data that you cannot keep secure!
- *Focus data collection:* Only collect data that you actually need
- *Respect context:* Do not re-purpose data
- *Transparency:* Allow users to understand how their data is used

Adapted from https://www.eff.org/deeplinks/2012/03/best-practices-respect-mobile-user-bill-rights

Cryptography 101

Nomenclature

- Unencrypted text: *Plaintext / cleartext*
- Encrypted text: *Ciphertext*
- *Encryption*: Scrambling
- *Decryption*: Unscrambling
- *Digital signature*: Cryptographic equivalent of traditional signatures

Cryptographic Hash Functions

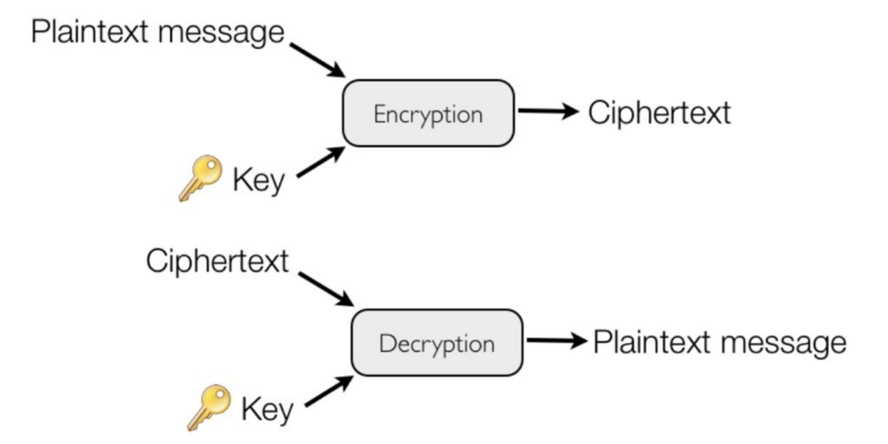
- A *hash function* takes arbitrary sized input and produces fixed sized output
- A *cryptographic hash function* is one that satisfies some additional properties
- Bad ones: MD5 (worst), SHA1 (bad)
- Good ones: SHA256, SHA512

File Integrity Checking

- MD5, SHA1 (bad), SHA256, … checksums
- Used to check if files have been modified or corrupted in transit
- Can also use PGP signatures (*.asc)
    - Stronger guarantee
    - A verified signature checks that the developer wrote the code you are going to run
    - apt-get checks these for you

Symmetric Cryptography

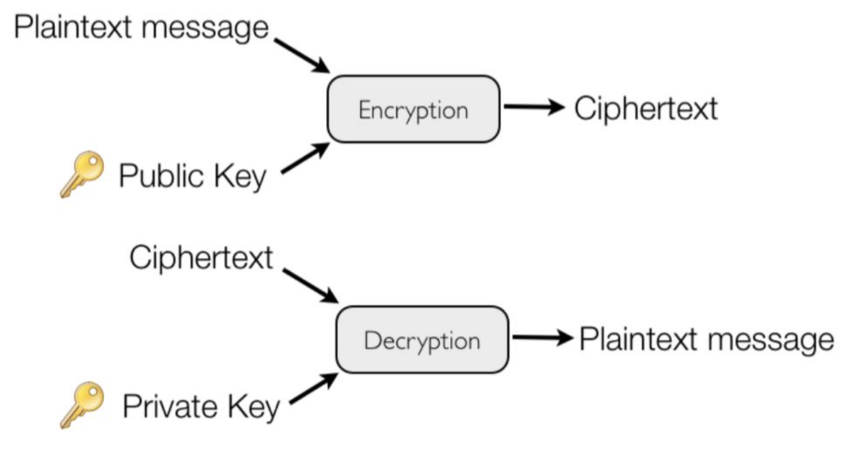
- Uses the same key both to encrypt and decrypt



- Problem: Our adversary is listening to our communications. How can we share both the message and the key?

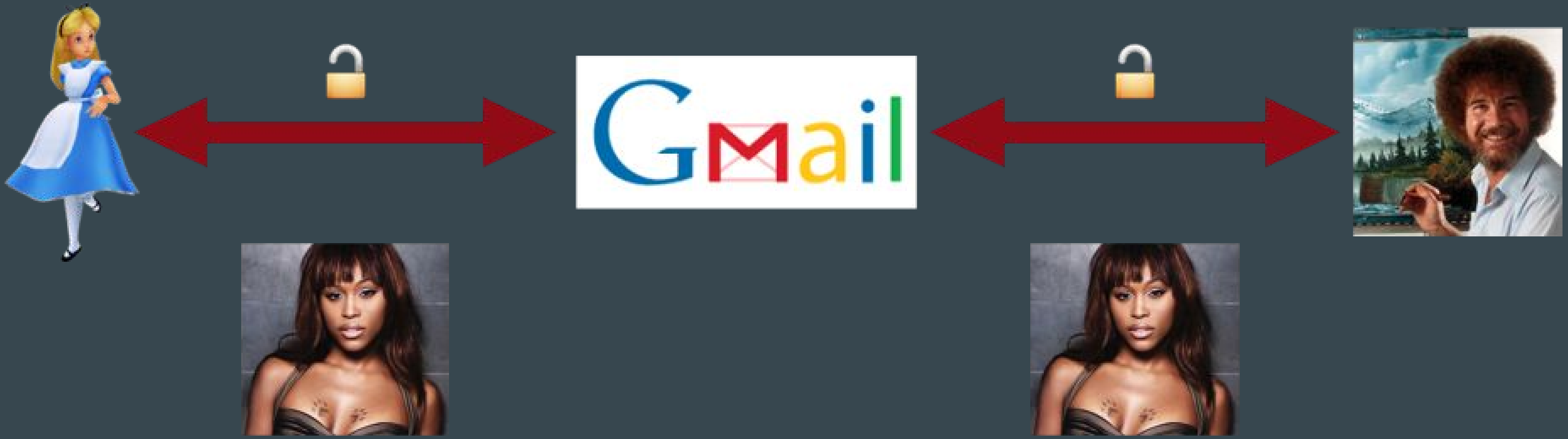
Asymmetric Cryptography

- Solution: Public key, or asymmetric encryption. Use a pair of keys:
  - private key - for you only
  - public key - for everyone

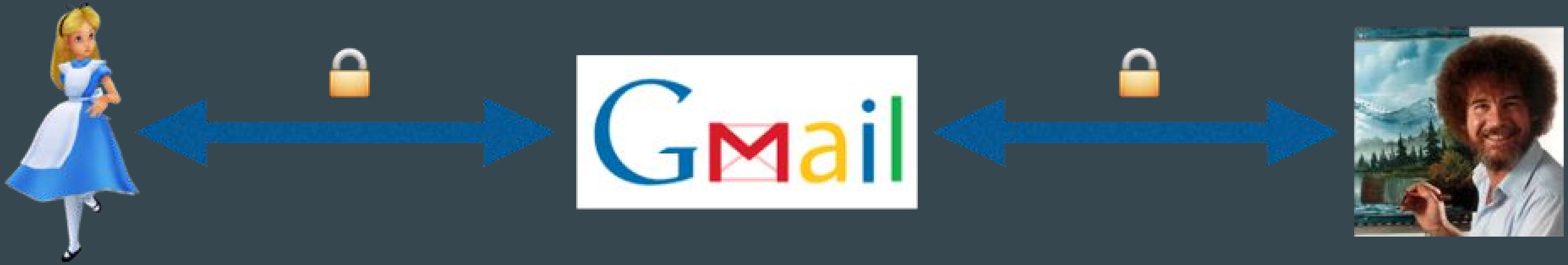


- Most systems use asymmetric systems (2 keys) to exchange a symmetric key (1 key) -> **Fast** and **secure**

HTTP



HTTPS

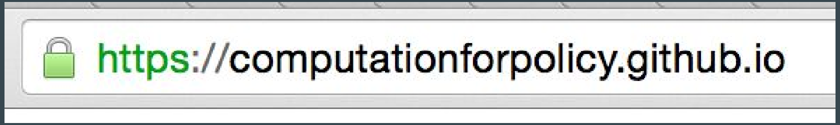


HTTPS: What’s going on

https://computationforpolicy.github.io

- When you visit an HTTPS site:
  - We check that the public key presented to use has been signed by a valid Certificate Authority: checking the **Certificate**
  - We use that public key to set up an asymmetric encrypted channel
  - We share/construct a symmetric key

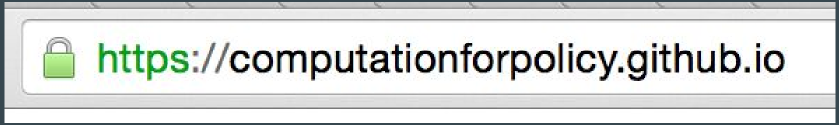
HTTPS: What’s going on



https://computationforpolicy.github.io

- When you visit an HTTPS site:
  - We check that the public key presented to use has been signed by a valid Certificate Authority: checking the **Certificate**
  - We use that public key to set up an asymmetric encrypted channel
  - We share/construct a symmetric key
- How do we know that bankofamerica.com is run by Bank of America?
  - It presents a valid certificate (signed by a valid CA)
  - The CA has checked that the URL is authentic and is in the control of Bank of America
  - We use the CA’s public key (baked into our browsers) to check the signature

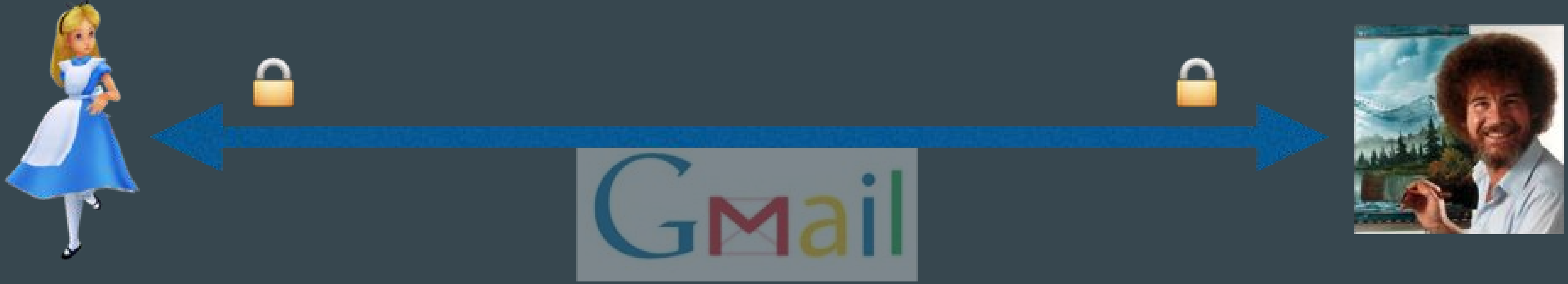
HTTPS: What’s going on



https://computationforpolicy.github.io

- When you visit an HTTPS site:
    - We check that the public key presented to use has been signed by a valid Certificate Authority: checking the **Certificate**
    - We use that public key to set up an asymmetric encrypted channel
    - We share/construct a symmetric key
- How do we know that bankofamerica.com is run by Bank of America?
    - It presents a valid certificate (signed by a valid CA)
    - The CA has checked that the URL is authentic and is in the control of Bank of America
    - We use the CA’s public key (baked into our browsers) to check the signature
- All this relies on trusting the CAs

End-to-end (e2e) Encryption

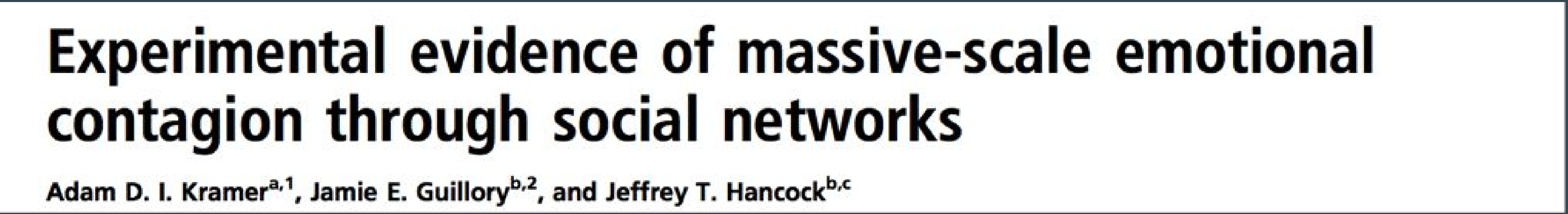


For Developers

- Good practice to have support for HTTPS
  - Can do by putting your site on Github pages
  - Can buy certs through e.g. gandi.net
  - Or get them for free through Let's Encrypt https://letsencrypt.org/
- Never blindly roll your own cryptography
  - Use pre-existing well-tested systems, e.g. TLS
  - Popular library: NaCl https://nacl.cr.yp.to/
- Consider security auditing and bug bounties

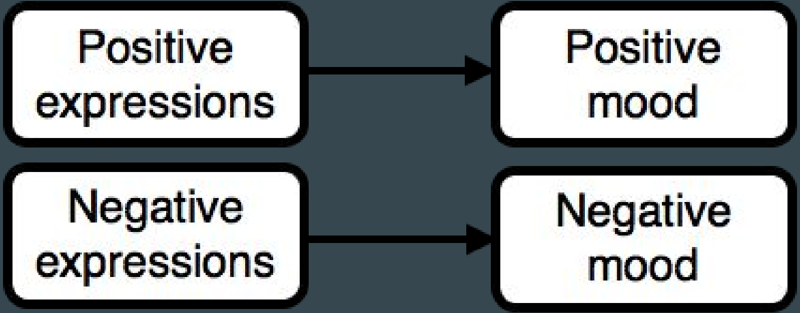
Ethics

Example: Facebook Emotional Manipulation



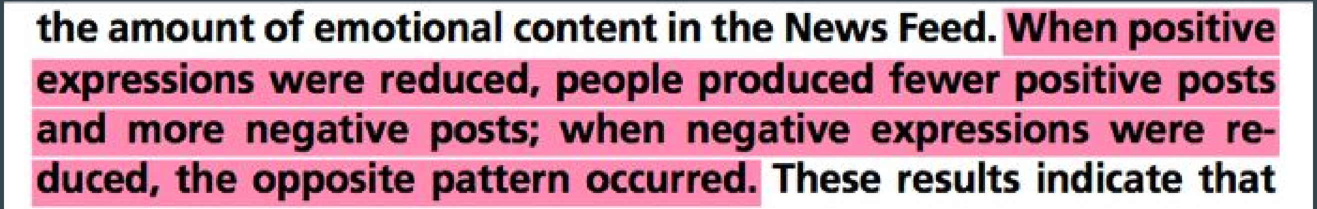
Experimental evidence of massive-scale emotional contagion through social networks

Adam D. I. Kramer[a,1], Jamie E. Guillory[b,2], and Jeffrey T. Hancock[b,c]



Positive expressions → Positive mood

Negative expressions → Negative mood



the amount of emotional content in the News Feed. When positive expressions were reduced, people produced fewer positive posts and more negative posts; when negative expressions were reduced, the opposite pattern occurred. These results indicate that

Data Ethics Pledge

I am
responsible for what I design and code.

I will do my best to be
inquisitive and not condescending,
ever mindful of privacy and security,
and
to combat technological determinism,
to remember that artifacts have politics,
to beware the power of defaults,

https://ctsp.berkeley.edu/an-ischool-pledge-of-ethics

Ethics Guidelines

- Talk to your advisor, determine if an IRB is necessary for experimentation
- Ensure that the subjects have informed consent if conducting social science experimentation
- Consider the implications of what you are building: who does it empower?

Further References

Cryptography Engineering by Ferguson, Schneier, and Kohno

Security Engineering by Anderson

The Tangled Web: A Guide to Securing Modern Web Applications by Zalewski