

what software engineers should know about

Security

17-313 Fall 2025

Foundations of Software Engineering

<https://cmu-17313q.github.io>

Eduardo Feo Flushing

Sources:

- Some slides adapted from CMU 17-437/637 Web Application Development
- "*What software engineers should know about privacy*". MSE Seminar. Hana Habib. CMU
- "*Ethics, Fairness, Responsibility, and Privacy in Data Science*". CMSC 25900. U. Chicago

Learning goals

- Explain why software is vulnerable to attacks
- Use the right secure software terminology
- Discuss a wide range of security attacks that can target software systems and tools and techniques to identify, prevent, and mitigate them

Smoking Section

- Last **two** full rows



Security



What Do We Mean by Security?

Security is about protecting systems from unauthorized:

- Access (who can get in and what they can read)
- Modification (what can be changed)
- Disruption (what can be broken or stopped)

Threats can target:

- Data at rest
 - Data in transit
 - Execution environments (browsers, servers, mobile)
 - People (social engineering)
- 
- Today's focus**

Why Network Security Matters

- Networks are the weakest common denominator for nearly all applications.
- Networks are inherently untrusted.
- Anyone between sender/receiver **can potentially** observe, inject, or modify traffic.
- Modern apps rely heavily on remote APIs, cloud services, Wi-Fi, mobile networks.
- Most attacks begin with stealing or manipulating traffic.

Attacks can be expensive

FT Financial Times

UK regulator hits Equifax with £11mn fine over cyber breach

The Financial Conduct Authority has fined credit reporting agency Equifax just over £11mn for failing to protect the data of nearly 14mn UK...



The Guardian

Uber fined \$148m for failing to notify drivers they had been hacked

Failure to report 2016 data breach 'one of the most egregious cases we've ever seen', says Illinois attorney general.



CBS News

PlayStation Network breach has cost Sony \$171 million

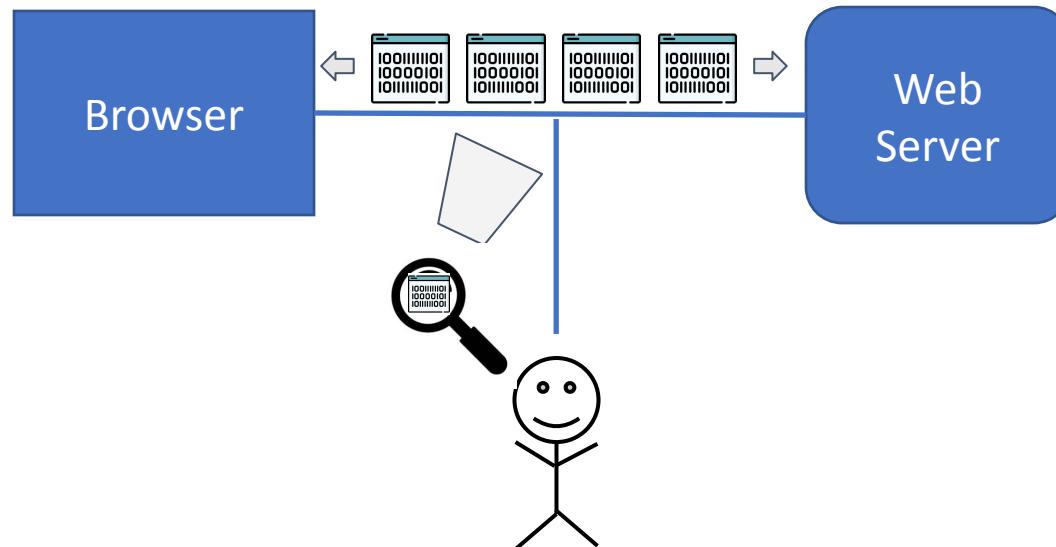
(CBS/AP) TOKYO - Sony has spent 14 billion yen, which translates to roughly \$171 million, to cover the costs of the massive security breach...



May 24, 2011

Common Network Attacks

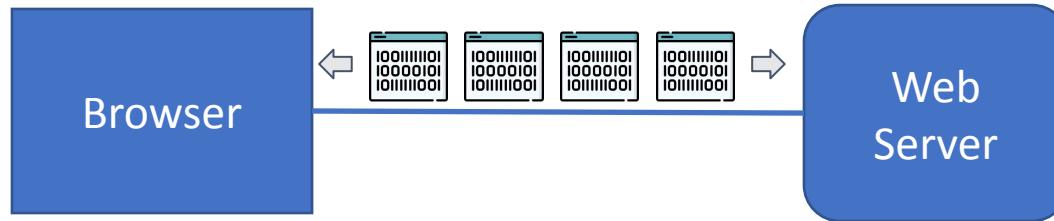
Sniffing (Eavesdropping)



What harm can an attacker cause just by seeing your data in transit?

Common Network Attacks

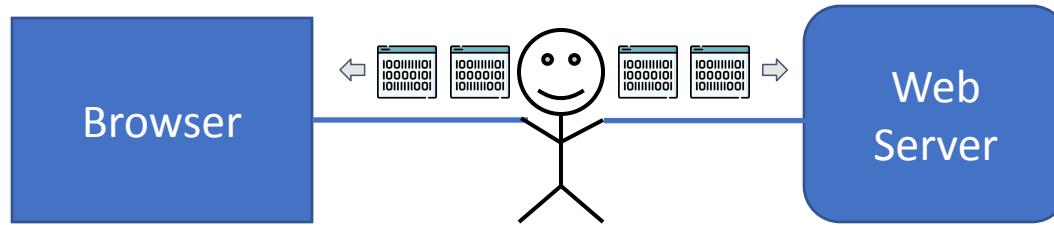
Person-in-the-Middle



What kinds of damage can an attacker cause by sitting quietly between you and the server, altering your traffic without you noticing?

Common Network Attacks

Person-in-the-Middle



What kinds of damage can an attacker cause by sitting quietly between you and the server, altering your traffic without you noticing?

Common Network Attacks

Spoofing

- Pretending to be someone you're not
- IP spoofing
 - Pretending to a "client" you're not (with a specific IP address)
- E-mail Spoofing
- DNS spoofing
 - Pretending to be a server that you're not
 - Fool a DNS server to give out incorrect IP addresses for DNS Names
- Note: also be careful of typos or similar characters attacks:
 - <http://mytimes.com>
 - <http://paypal1.com>

The Big Three (Core Security Properties)

Modern software security relies on protecting three fundamental properties:

- Authentication:
 - Verifies the identity of a user or system
- Authorization:
 - Controls access to actions and resources
- Confidentiality
 - Protects data from being viewed by unauthorized parties

Terms Defined

- Authentication
 - Knowing with whom you are communicating
 - User knowing the server and/or server knowing the user
 - Which is more important??
- Authorization
 - User having privilege to perform an operation on server
- Confidentiality
 - Communicating without others knowing what's been said
 - Intermediaries cannot change what was said
 - Typically includes protection from replay attack
 - (Typically does *not* provide secrecy of communication. Others can know communication occurred)

Poll

Authentication: Verifies the identity of a user or system

Authorization: Controls access to actions and resources

Confidentiality: Protects data from being viewed by unauthorized parties

- Which of the “big three” protect you from:
 - Sniffing? **Confidentiality**
 - Spoofing? **Authentication**
 - Person-in-the-middle Attack? **Authentication + Confidentiality**

Concepts **every SWE** need to know

To protect the Big Three (Authentication, Authorization, Confidentiality), software engineers rely on a small set of powerful crypto tools:

- One-way Hashing
- Secret Key Encryption
- Public Key Encryption
- Certificates

Hashing

(aka Message Digests, One-Way Hashing)

- A hash function is a one-way encoding of data
 - Same input, same output
 - Different output, different input
- Easy (relatively) to compute the hash function
- **Hard** to compute the hash function's inverse

Hashing (aka Message Digests, One-Way Hashing)

- A hash function is a one-way encoding of data
 - Same input, same output
 - Different output, different input
- Easy (relatively) to compute the hash function
- Hard to compute the hash function's inverse
- We only store hashed passwords on disk
 - To prevent passwords from being compromised if our servers are broken into

```
pbkdf2_sha256$180000$pP3DfkAYXSS1$L9JMQtFygrKbT246E/ZEaFScCTaX1p2v2ANNl4ryXY=
```

Algorithm

Iterations

Salt

Hashed Password

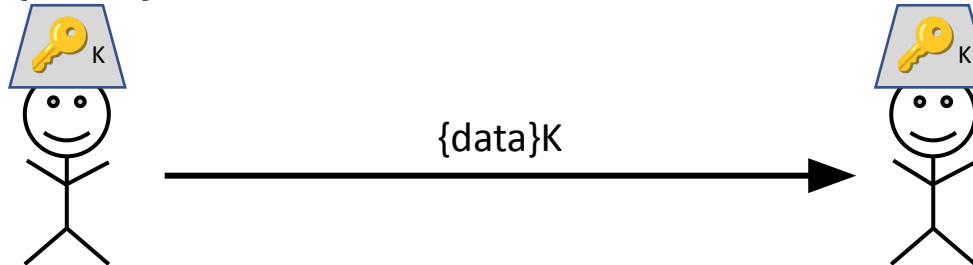
Secret Key Cryptography (aka Symmetric, Private Key Crypto)

- Like in the old movies and spy books
- One key (K)

- Shared Secret  K

- Used to encrypt and decrypt

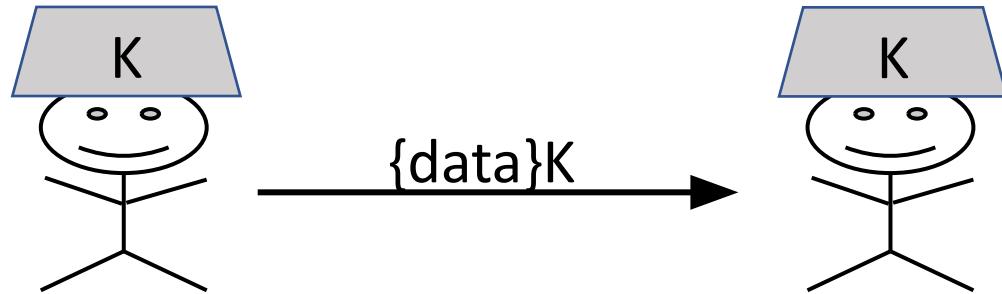
- Notation: $\{ \text{data} \}K$



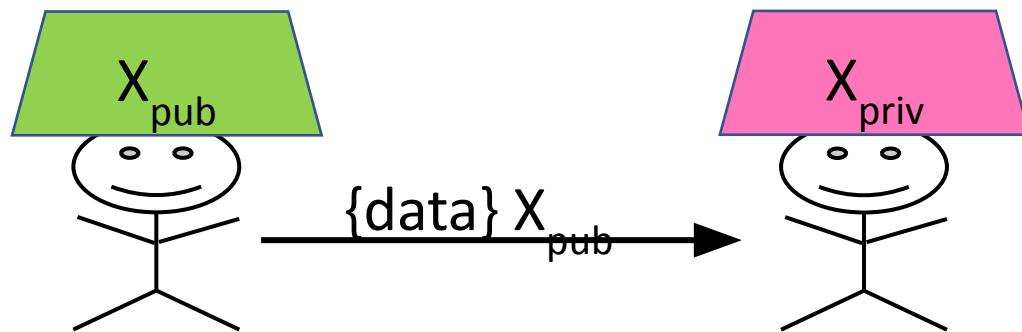
Public Key Cryptography (aka Asymmetric Key Crypto)

- Key Pair (key 1 & key 2)
 - Either key can be used to encrypt (key 1 or key 2)
 - You can only decrypt using the “other key” (key 2 or key 1)
 - One key is given out (**the public key**)
 - The other key is kept secret (**the private key**)
 - Notation: For entity X, we have keys X_{pub} & X_{priv}
- A public key can be given out freely to
 - Encrypt data sent to the holder (X) of the private key
 - Notation: $\{\text{data}\}X_{\text{pub}}$

Secret Key
Crypto

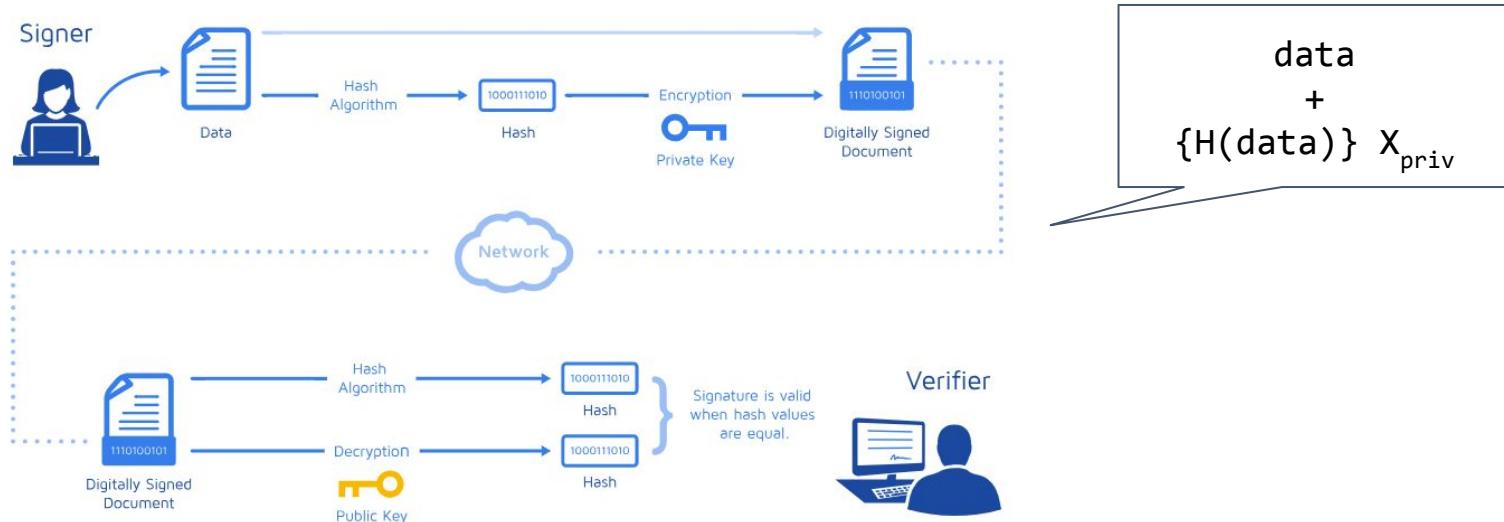


Public Key
Crypto



Public key cryptography: Authentication

- Digital signature on a message gives a recipient confidence that the message came from a sender known to the recipient.



Activity: Try public key cryptography

- Individual or groups 2-3 students
- The challenge:
 - Send me an encrypted message using **my public key** and post it on slack **#lecture-security**, tagging the bot @pgp-demo
 - Format : Hello <andrewids> <optional a joke>
- To encrypt the message, use my public key shared on the channel
- You can any tool to encrypt. We recommend 8gwifi.org
- The message that you encrypt should start with your andrew id
 - The bot will reply “Hello <andrew ids>!” if you did it correctly

-----BEGIN PGP MESSAGE-----
Version: OpenPGP.js v.1.20130420
Comment: http://openpgpjs.org

wUwDPg1yJu9L0nkBAf9SpEoknt7ryM9kobfXB/8fdus2AHx2C6b5Fdes1+jx
wn4rkqganSC6c7DNKtZ+hSRp8JLRI6u483DkpX/UOfky001EBw17vF9t+/cLj
U+1z+QpvUjp/FBK1fGmQ+mMSvDSWl+0wd+DckoRHNJPZIXjUIzTTGRKqxuo
0JOHT7VLPTFeEQIgTueqBxDJd+uQ0Gex5E=
=duzj
-----END PGP MESSAGE-----

Notes:

My public key looks like this:

Make sure you copy and paste the full PGP block:

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: OpenPGP.js v.1.20130420
Comment: http://openpgpjs.org
...
-----END PGP PUBLIC KEY BLOCK-----

Some considerations

- Who can read the messages?
- Do we have confidentiality?
- How can I respond to your message if needed?
- How can I confirm the sender's (your) identity?

Activity: Try public key cryptography (Part 2)

- Go to slack and find an encrypted message I just posted on Slack
 - I used my private key to encrypt it
- Got to webencrypt.org/openpgpjs/
- Decrypt the message using my public key

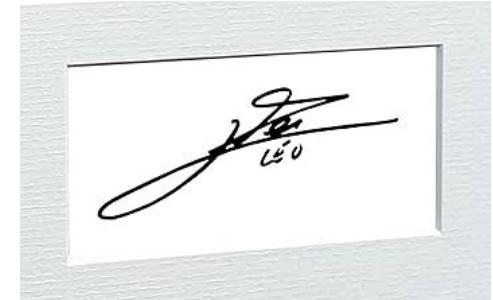
Public-key cryptography: Other Applications

How do I know that the sender is really who they say they are?

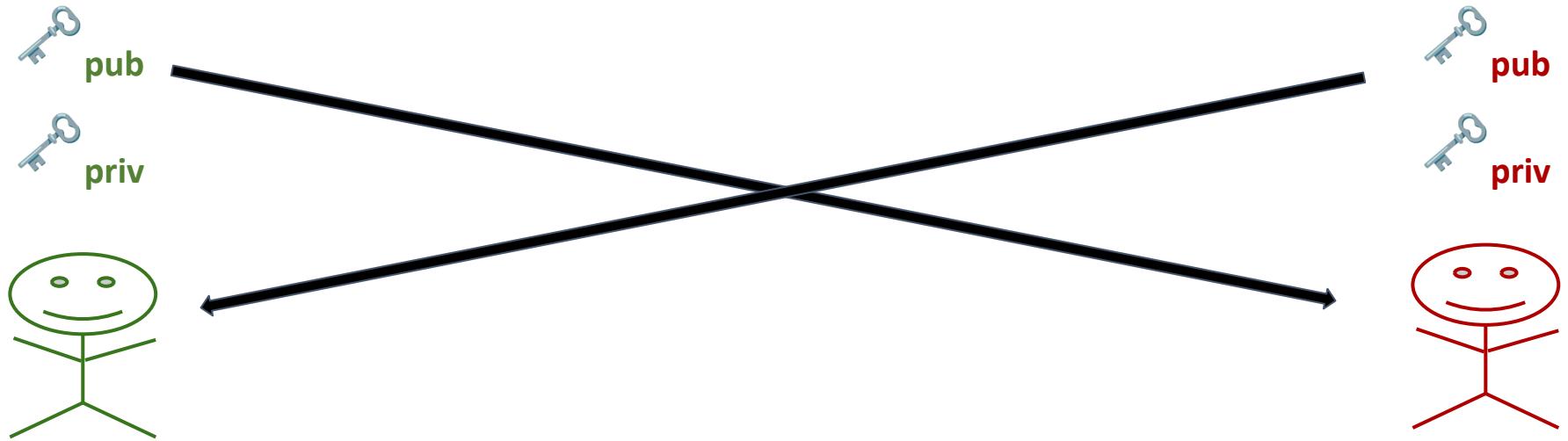
What we need

A way for the sender to produce something that:

- Only the real sender could have created
- Can be verified by anyone else
- Works even over an untrusted network



Authentication with public-key cryptography

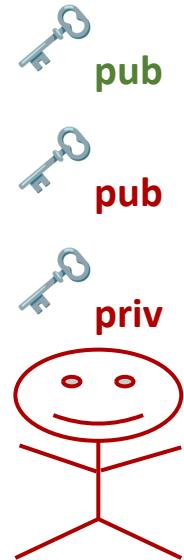


Authentication



{data}
+
hash({data})  pri

Sender signs the message with
their **private key**



Can the receiver confirm that
only the **private-key holder**
could have created this
signature?

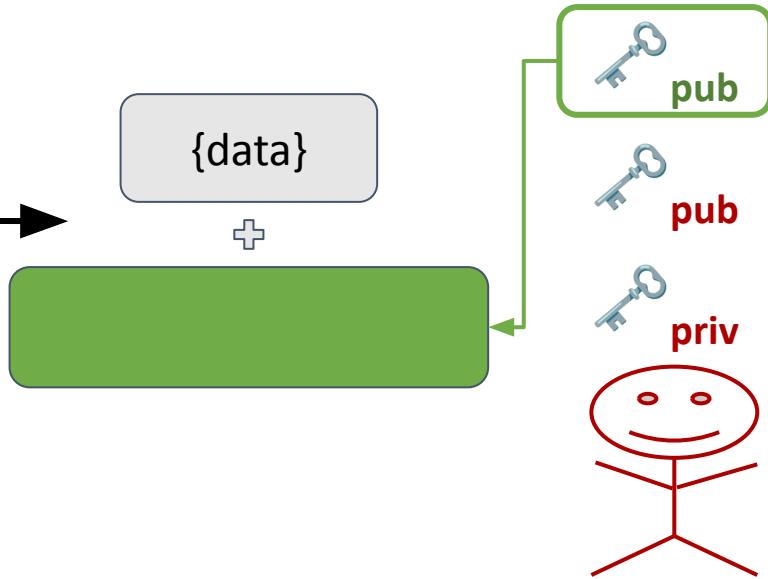
Authentication



{data}
+
hash({data})
v
Sender signs the message with
their **private key**

pri

v



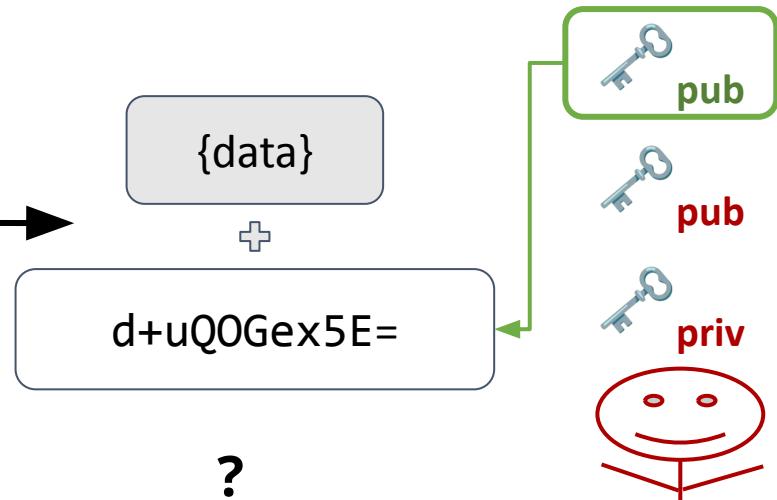
Can the receiver confirm that
only the **private-key holder**
could have created this
signature?

Authentication



{data}
+
hash({data})
v
Sender signs the message with
their **private key**

pri



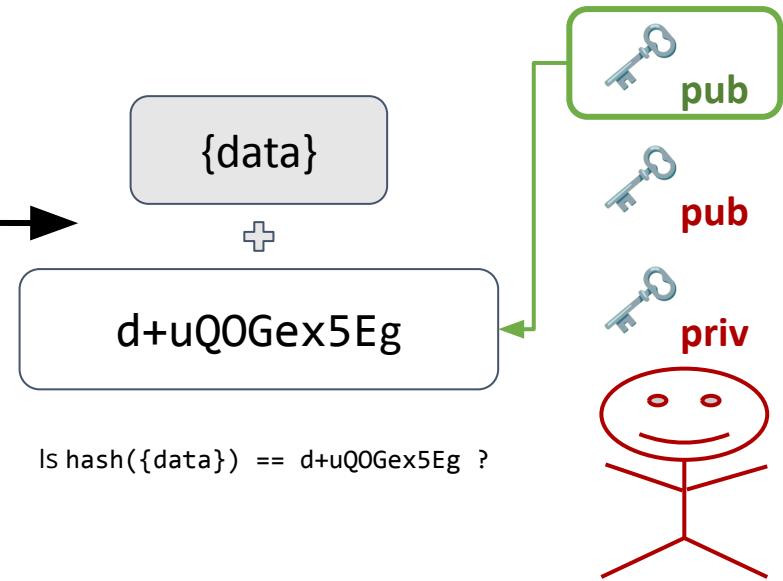
Can the receiver confirm that
only the **private-key holder**
could have created this
signature?

Authentication



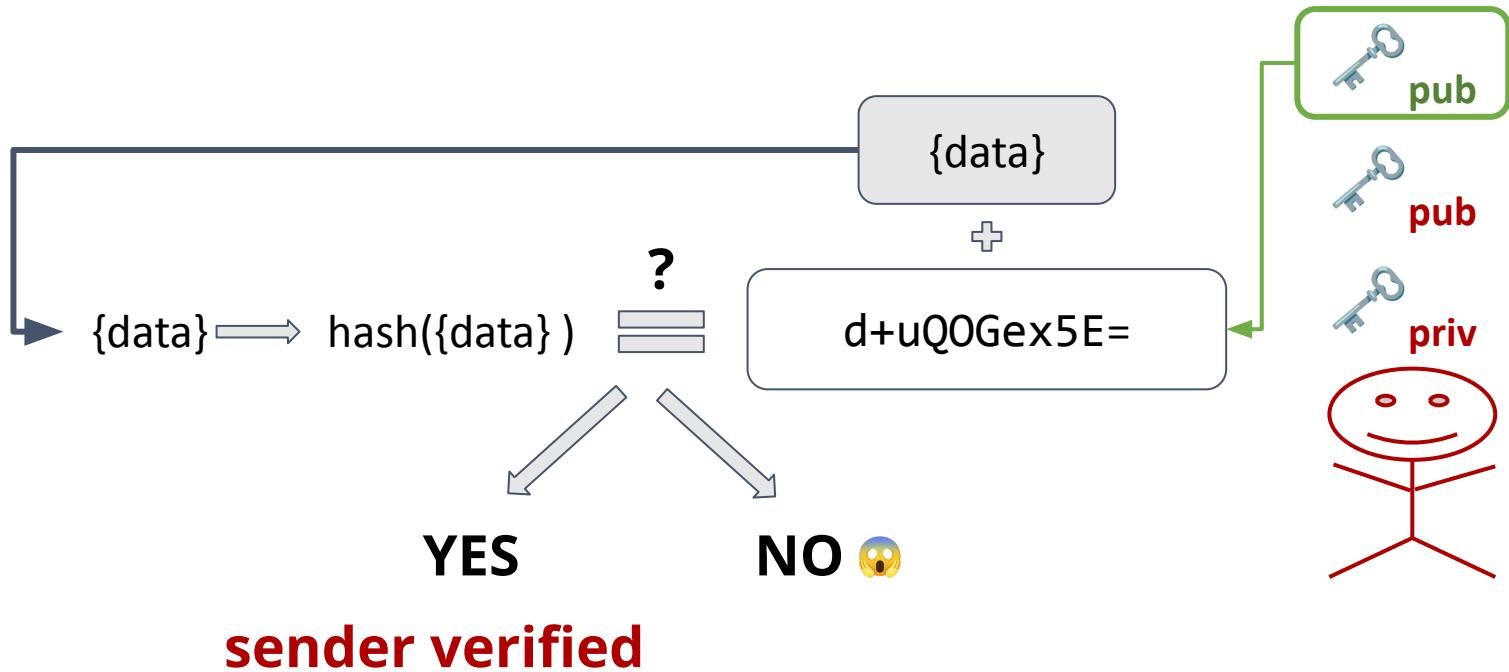
{data}
+
hash({data})
v
Sender signs the message with
their **private key**

pri



Can the receiver confirm that
only the **private-key holder**
could have created this
signature?

Authentication



Comparison

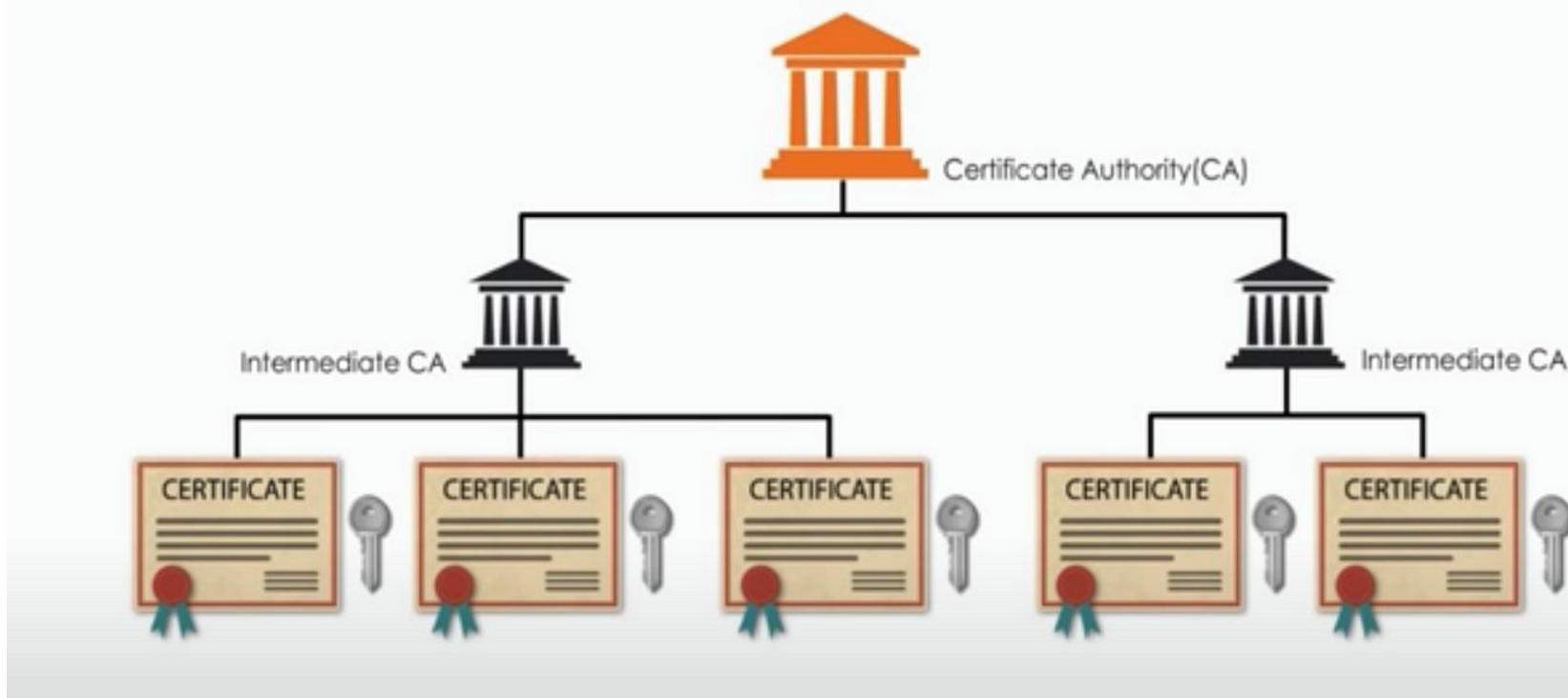
- Public & secret key crypto are (both) very secure
 - Unless the keys are compromised
- Public key crypto is computationally expensive
 - Secret key crypto is relatively fast
- It's hard to distribute the secret key between communicating parties
 - This is why we like public key cryptography
 - We just use public key to distribute secret keys which are then used

How to distribute the public keys?

Certificate Authority

- A Certificate Authority (CA) confirms an entity's public key
 - Usually this will be a server's public key
- Companies get paid to do this
 - They "check out" the requestor
 - Now-a-days domain registrars provide this service
 - They issue a "certificate" with the information
 - Certificates are signed with the **CA's private key**
- **CA's public key** is "well-known"
 - It's in an additional certificate
 - Pre-installed or added to your configuration

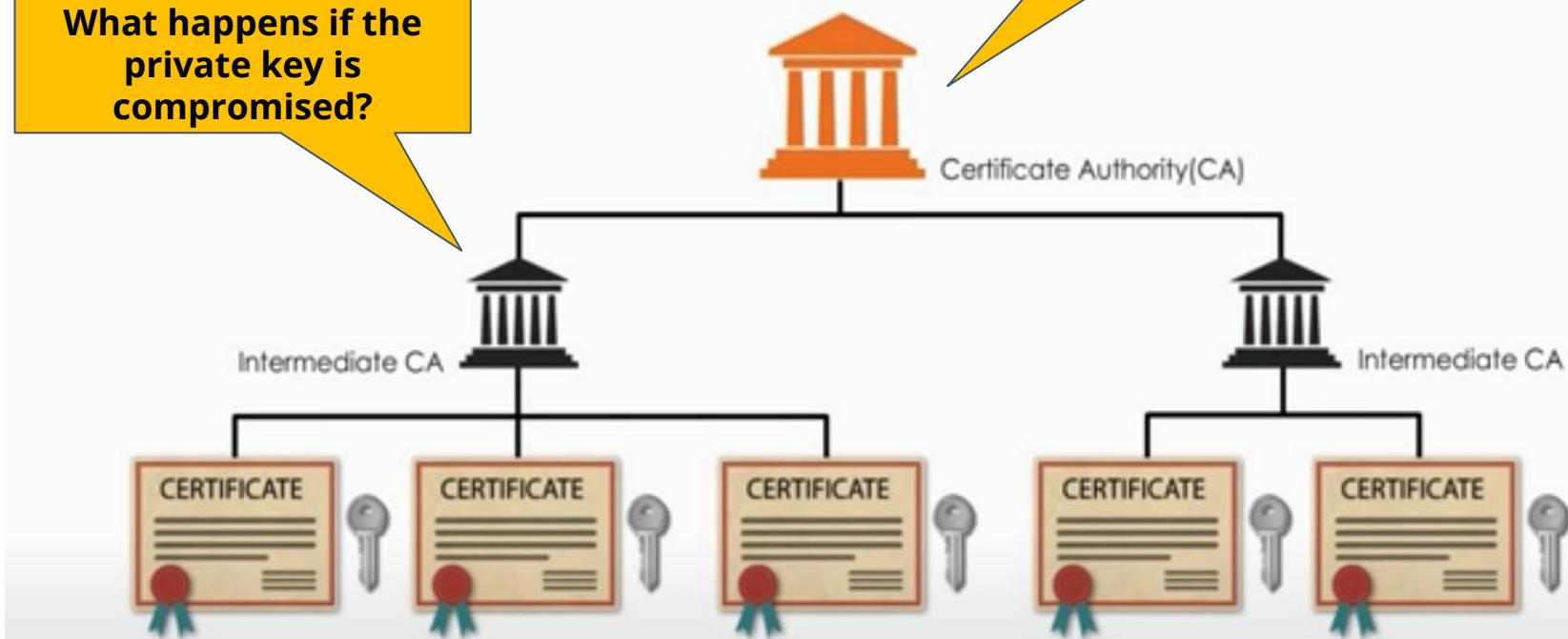
Certificate Chain of Trust



Certificate Chain of Trust

What happens if the private key is compromised?

What happens if the private key is compromised?



Generating keys for a Root Certificate Authority



SSL: Combining two ciphers

- The expensive public-key cipher
 - Consists of two keys: one public, one private
 - These are each typically 1024-bit or 2048-bit keys
 - But has great key distribution properties
- The inexpensive symmetric cipher
 - These are typically 128-bit or 256-bit keys
 - Need to distributed the symmetric key
 - SSL uses public-key encryption to distribute the symmetric key

What does SSL Give You?

- SSL can be used for any TCP/IP communication
- Once you have SSL
 - You have confidentiality
 - You have server authentication
- User authentication can be done using
 - Your own userids and passwords

User authentication using passwords?



Other factors that can lead to security breaches

- Injection Flaws, XML External Entities (XXE) Attacks
- Deployment misconfigurations
 - Default accounts with unchanged passwords
 - Unnecessary features enabled
 - Improperly configured permissions on cloud services
- Cross-Site Scripting (XSS)
 - Allows attackers to execute scripts in the victim's browser
- **Using dependencies with known vulnerabilities (CVE)**
- **Insufficient Logging and Monitoring**

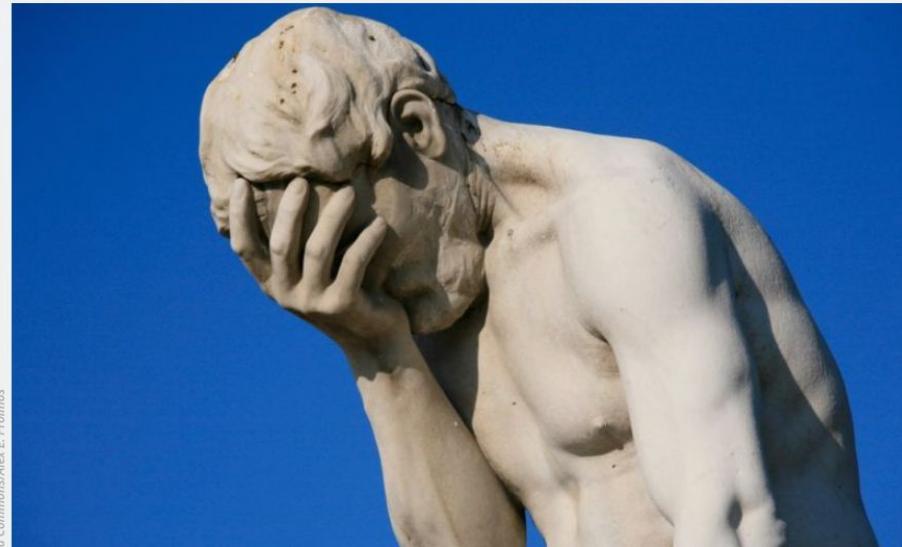
Common Vulnerabilities and Exposures (CVE)

BIZ & IT —

Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/13/2017, 11:12 PM



Common Vulnerabilities and Exposures (CVE)



OpenCVE

CVE Alerting Platform



Dependabot

Logging and Monitoring

- Early Detection of Security Incidents
- Forensic Analysis and Compliance
- Proactive Threat Hunting
- Audit Trails for Accountability

