CSE 484/M584: Computer Security (and Privacy)

Spring 2025

David Kohlbrenner
dkohlbre@cs

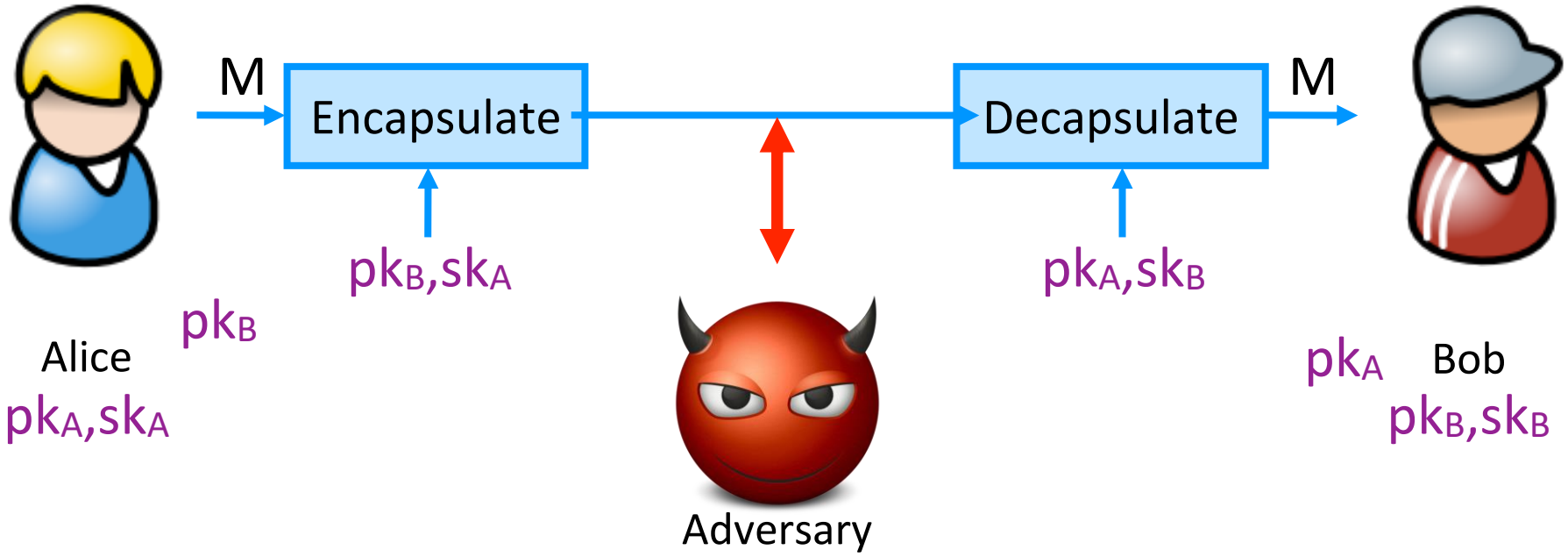
UW Instruction Team: David Kohlbrenner, Yoshi Kohno, Franziska Roesner, Nirvan Tyagi. Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials

Admin

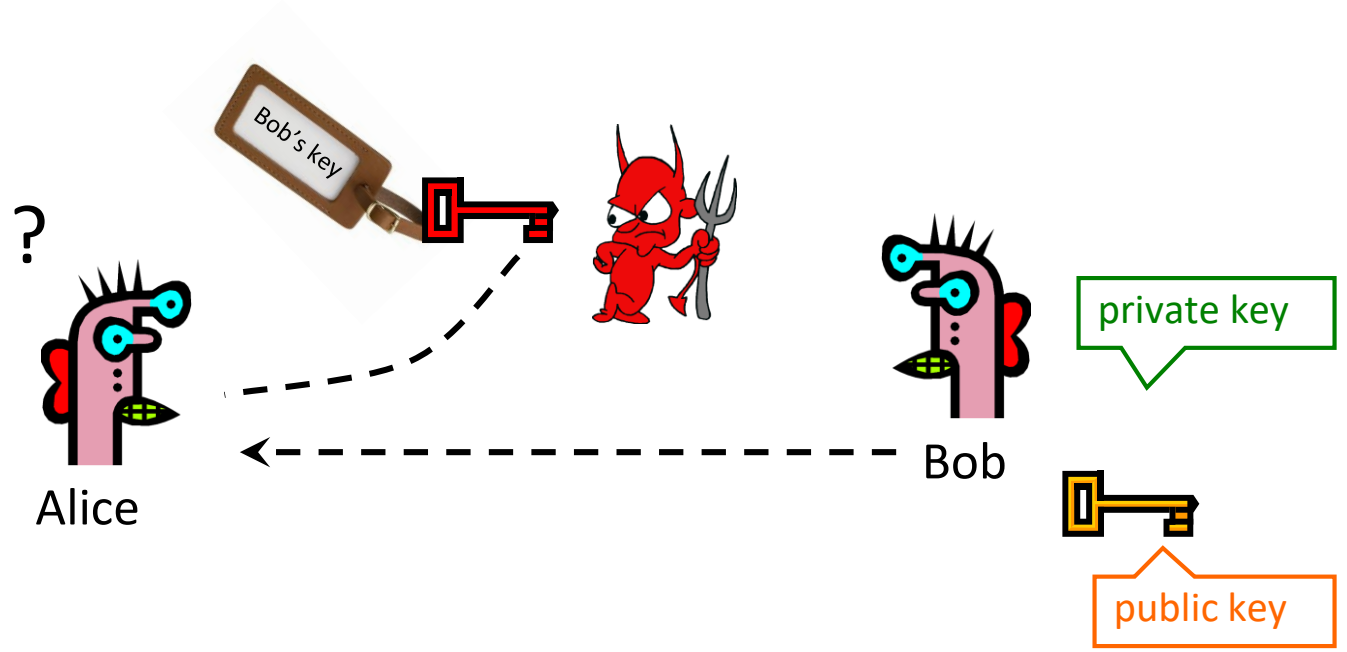
- Lab 2 (Cryptolab) this Wednesday

Asymmetric Setting

Each party creates a public key pk and a secret key sk.

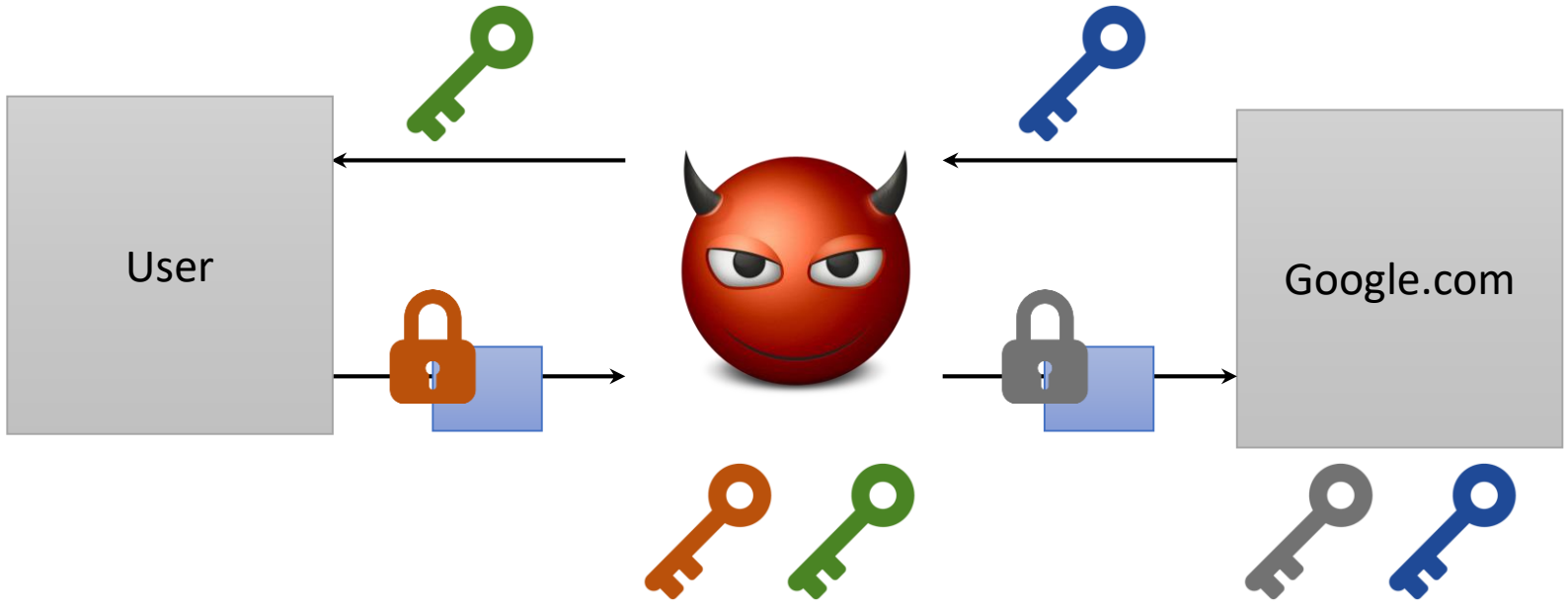


Authenticity of Public Keys



Problem: How does Alice know that the public key they received is really Bob's public key?

Person-in-the Middle/On-path-attacker



Distribution of Public Keys

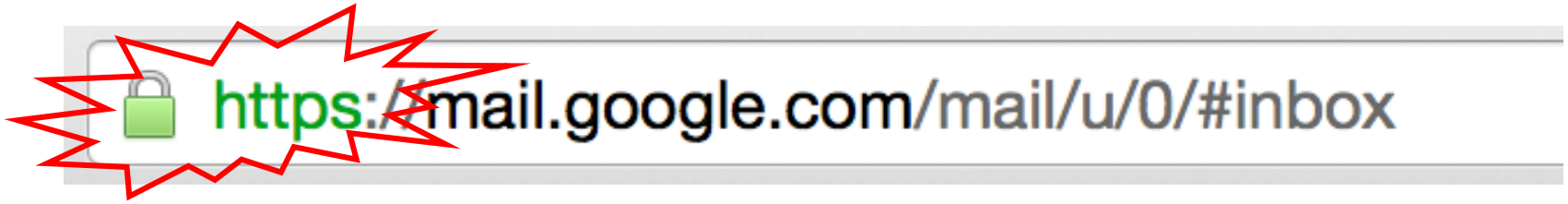
- **Public announcement or public directory**
  - Difficult to validate, expensive to host.

Distribution of Public Keys

- **Public announcement or public directory**
  - Difficult to validate, expensive to host.

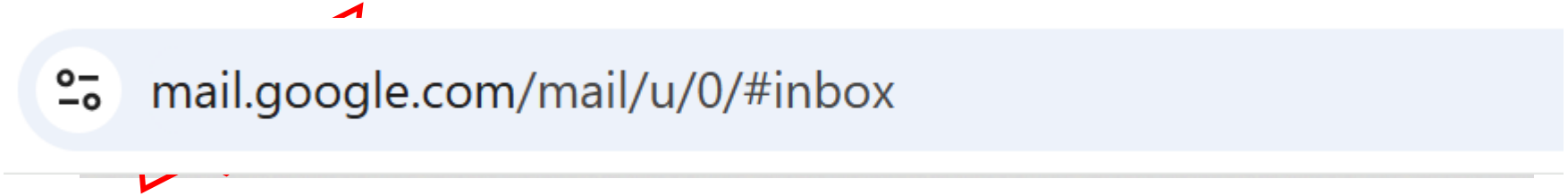
- **Public-key certificate**
  - Signed statement specifying the key and identity
  - $\text{Sign}(\text{“Bob”}||\text{PKB}, \text{SKCA})$
  - Additional information often signed as well (e.g., expiration date)

You encounter this every day…



SSL/TLS: Encryption & authentication for connections

You encounter this every day…



SSL/TLS: Encryption & authentication for connections

homes.cs.washington.edu/~dkohlbre/

David Koh

I am an Assistant Prof
Engineering at the Un
Previously I was a pos
PhD with Hovav Shac

Google Scholar profile

Contact: dkohlbre [at]
If you are a UW stude
(david.kohlbrenner [at

Office: CSE2 310

Research

My research interests
I put special focus on

Active projects include
Next generation micro
FPGA analog side-cha
The Keystone TEE fra

Teaching

Certificate Viewer: *.cs.washington.edu

General | Details

Issued To

Common Name (CN)	*.cs.washington.edu
Organization (O)	University of Washington
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	InCommon RSA Server CA 2
Organization (O)	Internet2
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Saturday, February 22, 2025 at 4:00:00 PM
Expires On	Monday, February 23, 2026 at 3:59:59 PM

SHA-256 Fingerprints

Certificate	41790008543f52c5f3b97bb5b62adeaa664525700d922b74e08f1e6463e9f511
Public Key	b6deb9fe86e5ce50840d964be61da32c8dcac5f0c1da00544790dcbf2582c3d9

Distribution of Public Keys

- **Public announcement or public directory**
  - Difficult to validate, expensive to host.

- **Public-key certificate**
  - Signed statement specifying the key and identity
  - Additional information often signed as well (e.g., expiration date)

- **The approach: certificate authorities (CAs)**
  - Company/agency responsible for certifying public keys.
  - Provides certificates to anyone proving their identity.
  - $\text{Sign}(\text{“Bob”}||\text{PKB}, \text{SKCA})$
  - Certificate can then be handled *by the public key owner.*
  - Every computer is pre-configured with CA’s public key(s)

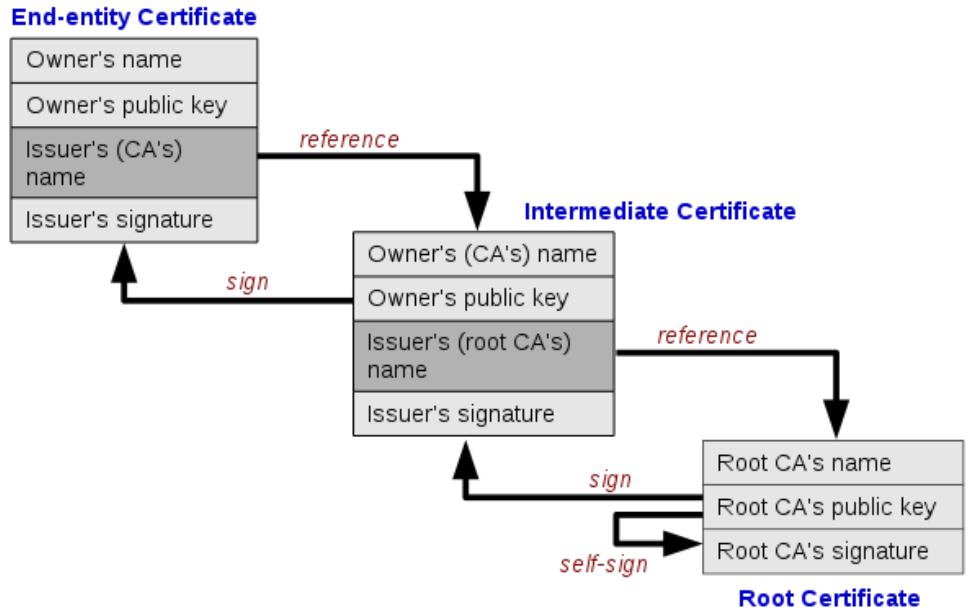
Certificate Authorities

- Many CAs
- Landscape has changed in the past decade
  - Old model: pay for certificates.
  - New model: certificates are free.

- How do they validate the owner of a domain?

Certificate Chains

- Single CA is impractical
- Instead, use *root* CAs who can delegate.
- Everybody must know the root's public key.
- Then use a certificate *chain*
  - sigVerisign("AnotherCA", PKAnotherCA), sigAnotherCA("Alice", PKA)
- Not shown in figure but important:
  - Each cert says "is this cert granting the ability to sign more certs?"



SSL/TLS High Level

- SSL/TLS consists of two protocols
  - Familiar pattern for key exchange protocols
- Handshake protocol
  - Use public-key cryptography to establish a shared secret key between the client and the server
- Record protocol
  - Use the secret symmetric key established in the handshake protocol to protect communication between the client and the server

Corporate CAs?

- Many corporations require that all company machines have an additional **Root Certificate** installed, owned and controlled by the company IT.
- This would allow the company to create a certificate for any website, service, etc. they want and have it trusted by any company machine. (But not by anyone else's).

- What does this let corporate IT do?
- Why might they want to do that?

Many Challenges…

- Weak security at CAs
  - Allows attackers to issue rogue certificates
- Users don’t notice when attacks happen
  - We’ll talk more about this later in the course
- How do you revoke certificates?

Rogue Certs

- In Jan 2013, a rogue *.google.com certificate was issued by an intermediate CA that gained its authority from the Turkish root CA TurkTrust
  - TurkTrust accidentally issued intermediate CA certs to customers who requested regular certificates
  - Ankara transit authority used its certificate to issue a fake *.google.com certificate in order to filter SSL traffic from its network
- This rogue *.google.com certificate was trusted by every browser in the world

Bad CAs

- DarkMatter (https://groups.google.com/g/mozilla.dev.security.policy/c/nnLVNfqgz7g/m/TseYqDzaDAAJ and https://bugzilla.mozilla.org/show_bug.cgi?id=1427262)
  - Security company wanted to get CA status
  - Questionable practices

- Symantec! (https://wiki.mozilla.org/CA:Symantec_Issues)
  - Major company, regular participant in standards
  - Poor practices, mismanagement 2013-2017
  - CA distrusted in Oct 2018

- Recall: How can we trust the CAs? What happens if we can't?

Certificate Revocation

- Revocation is very important

Certificate Revocation

- Revocation is very important

- Many valid reasons to revoke a certificate
  - Private key corresponding to the certified public key has been compromised
  - User stopped paying their certification fee to this CA and CA no longer wishes to certify them
  - CA's private key has been compromised!

How do we revoke a certificate?

- Scenario:
  - Web browsers connect to website X, get certificate from X.
  - They validate that the certificate was signed by some CA C.
  - C wants to revoke the certificate, X is not necessarily responsive.
  - Consider both security and efficiency…

  Gradescope!

Certificate Revocation Mechanisms

- Certificate revocation list (CRL)
  - CA periodically issues a signed list of revoked certificates
    - Credit card companies used to issue thick books of canceled credit card numbers
  - Can issue a “delta CRL” containing only updates
- Online revocation service
  - When a certificate is presented, recipient goes to a special online service to verify whether it is still valid
    - Like a merchant dialing up the credit card processor

Modern revocation (for TLS certificates)

- Give up.

Modern revocation (for TLS certificates)

- Give up.

- New CABForum rules, passed recently:

- Expand Section 4.2.1 to detail the allowed data reuse periods for validation data (both for domains/IPs and for everything else in Section 3.2)
  - Eventual reduction of non-SAN validation data reuse from **825 to 398 days**
  - Eventual reduction of SAN validation data reuse from **398 days to 10 days**
- Expand Section 6.3.2 to detail a schedule for reducing Public TLS certificate maximum validity periods in coming years
  - Eventual reduction of maximum validity period from **398 days to 47 days**
- These reductions are proposed to occur starting in March 2026 and concluding in March 2029

Attempt to Fix CA Problems:

Certificate Transparency

- **Problem:** browsers will think nothing is wrong with a rogue certificate until revoked
- **Goal:** make it impossible for a CA to issue a bad certificate for a domain *without the owner of that domain knowing*
- **Approach:** auditable certificate logs
  - Certificates published in public logs
  - Public logs checked for unexpected certificates

www.certificate-transparency.org

Next Major Topic!

Web+Browser Security