Public Key Infrastructure

- CEG 6430/4430 Cyber Network Security
- Junjie Zhang
- junjie.zhang@wright.edu
- Wright State University

Public Key Infrastructure

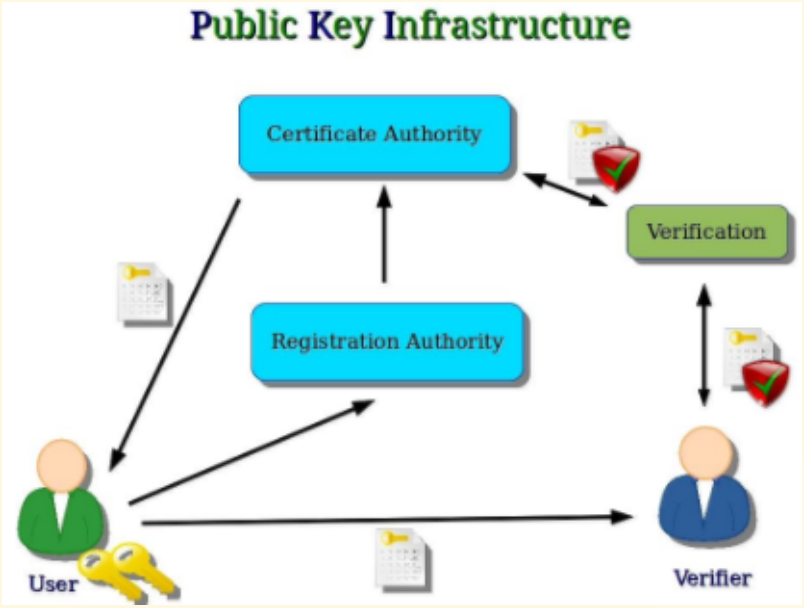
A PKI is to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

Why PKI?

- Question: There is a public key that claims itself belongs to *that* Alice. How do I know it actually belongs to Alice?
- A Potential Solution: Contact a trusted authority, who stores the public key for *that* Alice.
- Challenge: It does not scale up since this authority will need to serve requests from all Internet users.

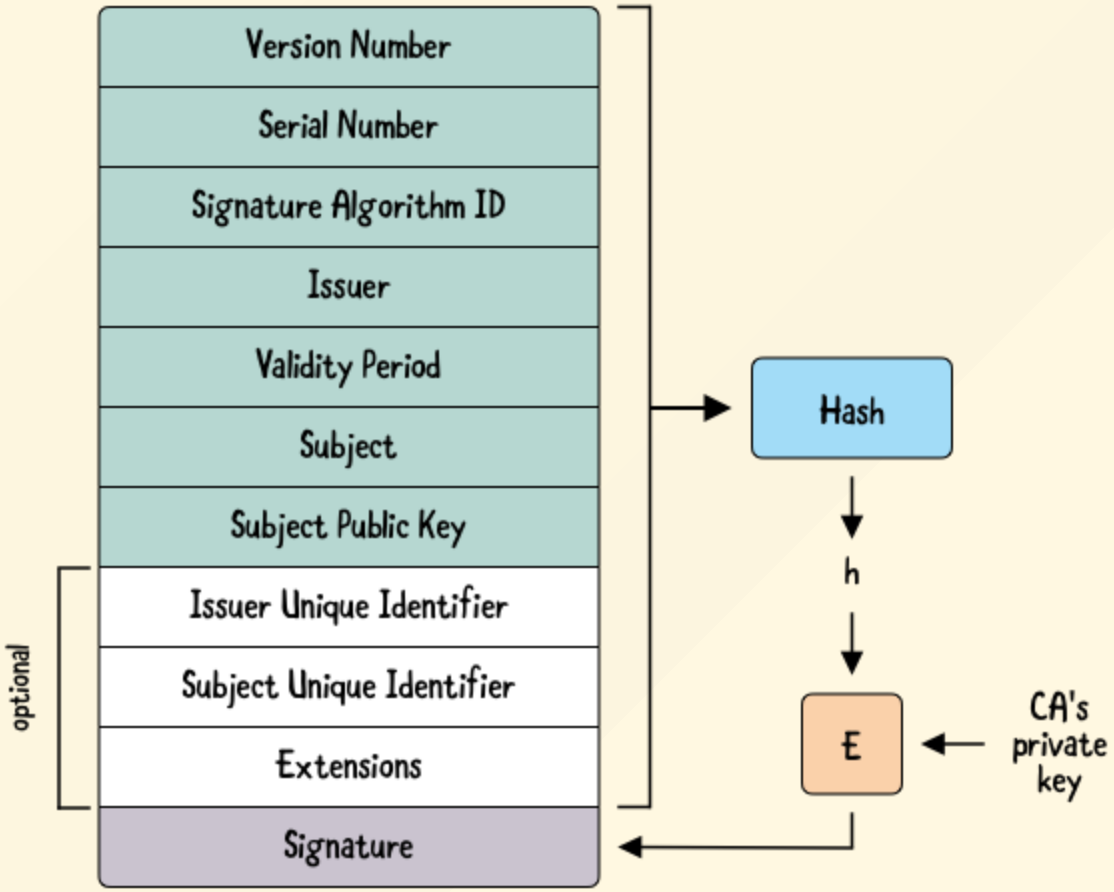
Why PKI?

- The authority (i.e., the certificate authority) now only audits the registration and creates the certificate.
- The owner/user distributes the certificate.
- The receiver/verifier verifies the certificate using CA's public key.

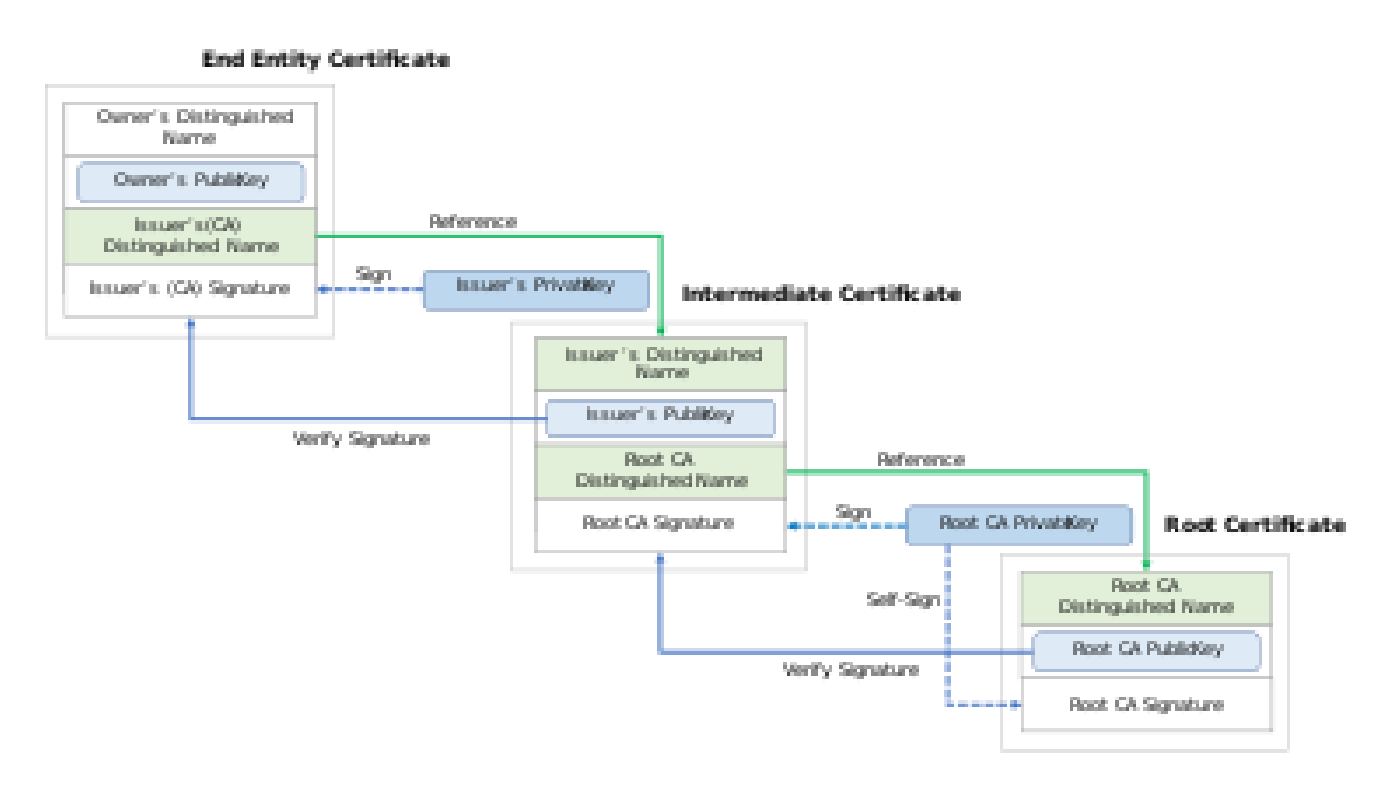


PKI - The Registration and Verification Process

X.509 Certificate Format



PKI - Chain of Trust

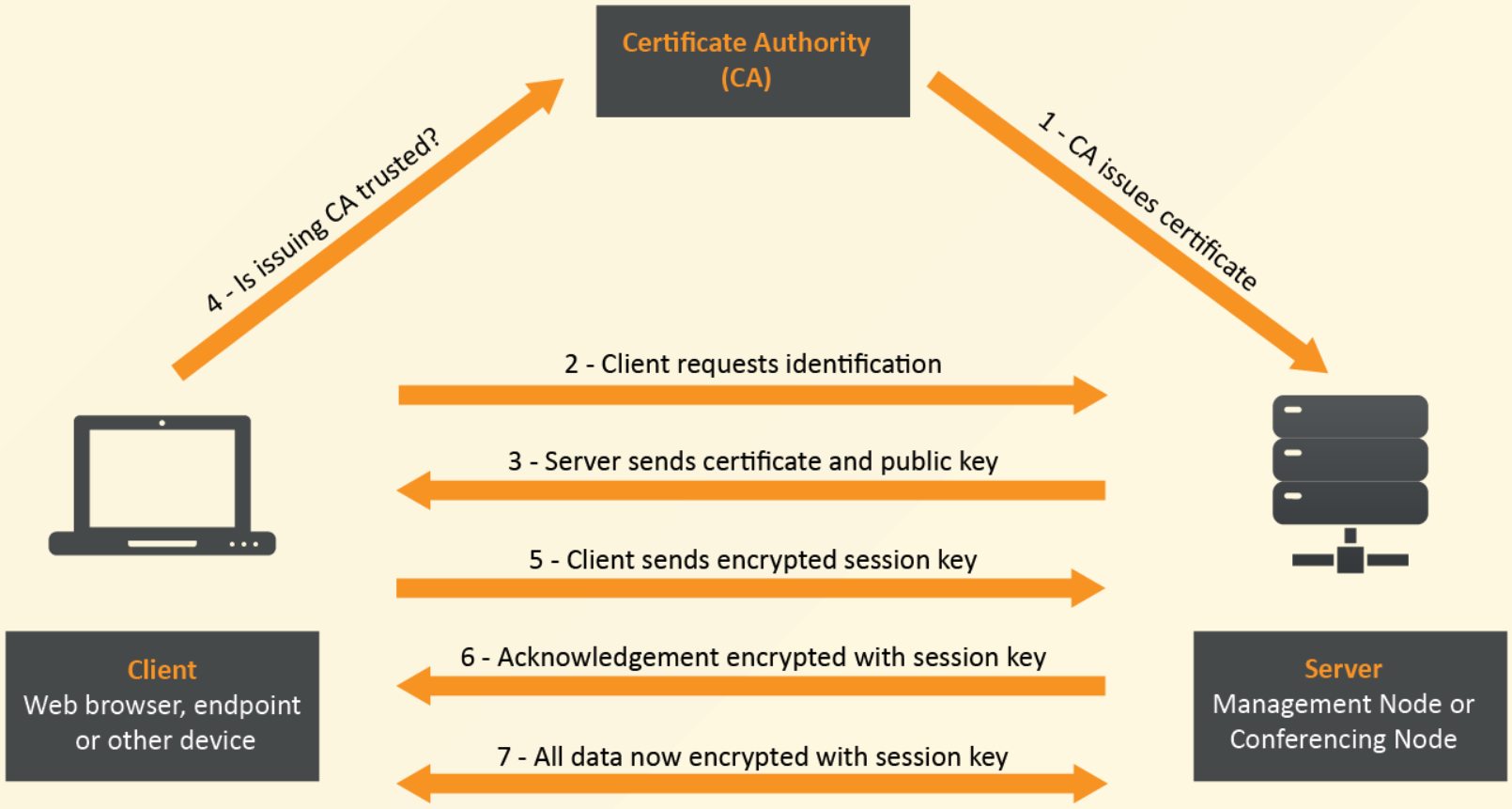


The CA Market

As of July 2022, analysis based on Alexa top 10 million:

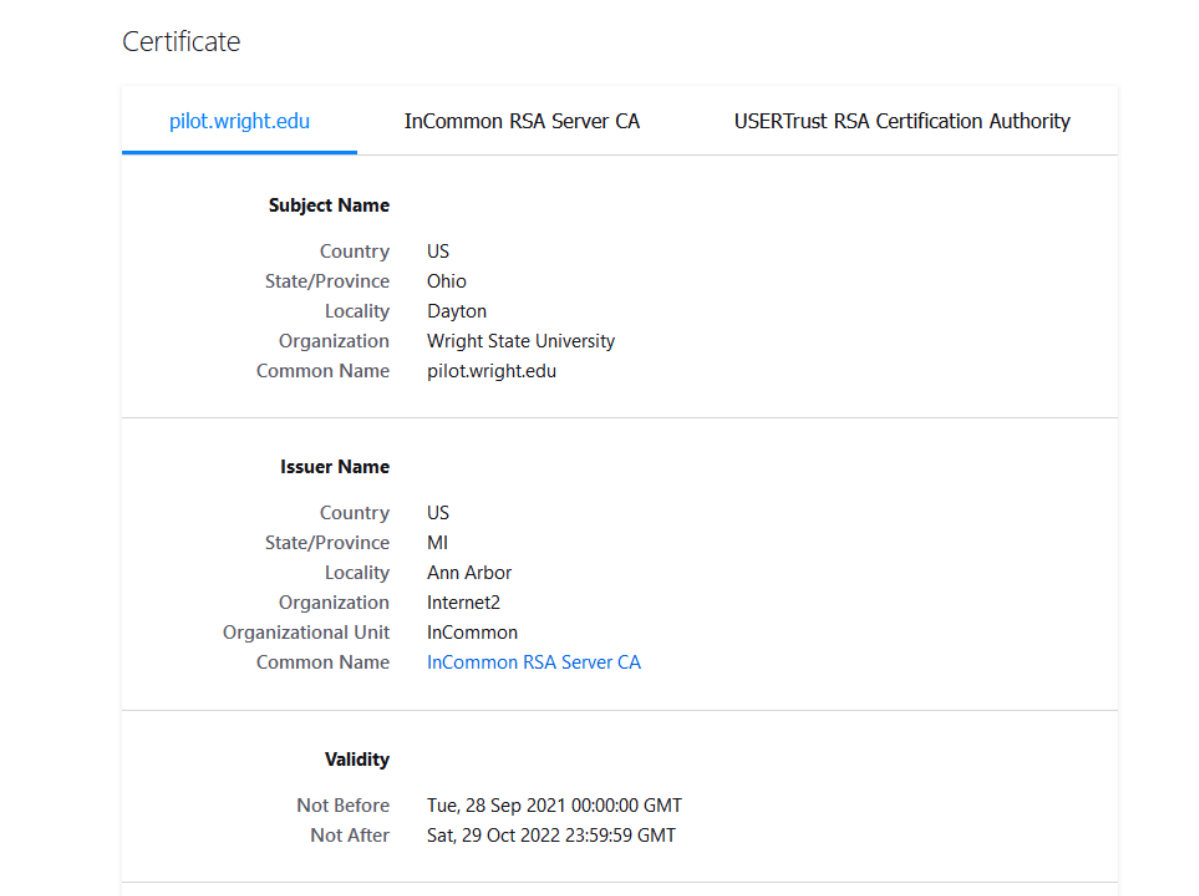
CA	Market Share
IdenTrust	48.9%
DigiCert	18.7%
Sectigo	15.5%
Let's Encrypt	8.2%
GoDaddy	6.1%
GlobalSign	2.7%

PKI Used in SSL/TLS



Get Certificate From a Real Server

You can get a certificate from your browser.



Get Certificate From a Real Server Using openssl

```
# establish an interactive SSL connection with a server using an HTTPS client.
# type Q or EOF to end this connection.
openssl s_client -connect www.wright.edu:443

# use the -showcerts option to get the complete certificate chain.
openssl s_client -showcerts -connect www.wright.edu:443

# extract certificates
openssl s_client -showcerts -connect  www.wright.edu:443 </dev/null | sed -n -e '/-.BEGIN/,/-.END/ p' > certifs.pem
```

Where is **USERTrust RSA**? It is in Firefox.

- It is a self-signed certificate preloaded to Firefox.
- www.wright.edu:443 does not send this certificate.
- The **InCommon** one is singed by **USERTrust RSA**

Inspect a Certificate

```
#You can manually save a certificate into the certifs.pem file.

#-noout omits the output of encoded information.
openssl x509 -in certifs.pem -text -noout

# extract a specific field of the public key
x509 -in certifs.pem -pubkey -noout
```

Weakness of PKI

Compromised and Misbehaved CAs

- CAs can be compromised. Attackers can therefore steal the private key and issue certificates on behalf of the CA.
  - DigiNotar Hacking
- CAs can forge certificates for questionable purposes.
  - trustwave forging certificates
  - Google bans cnnic certificates

Single Direction

Any CA can issue a certificate for any domain name without the owner's permission.

- Solution 1: Public Key Pinning: Pinning is the process of associating a host with their expected X509 certificate or public key.
- Solution 2: Certificate Transparency

No Trust Agility

- Trust v.s. not-trust, no middle ground.
- CA can be too large to fail.

Weak Domain Validation

- How domain is validated for certificate
- Using DNS poisoning attacks to bypass domain validation

Revocation Challenges

- There is a delay in propagating revocation information to each system (about 10 days).
- A *soft-fail* policy implemented in all current browsers: attempt to obtain the revocation information but ignore all failures.
  - An active network attacker can suppress OCSP reqeusts, and therefore make it possible to use a revoked (problematic) certificate.

User Failures

- Warnings are presented to users who do not know how to respond.
- They usually just ignore warnings and therefore invalidate PKI entirely.

Additional Readings

- The SSL Landscape
- Analysis of the HTTPS Certificate Ecosystem
- Web PKI