F-PKI: Enabling Innovation and Trust Flexibility in the HTTPS Public-Key Infrastructure

Laurent Chuat*, Cyrill Krähenbühl*, Prateek Mittal†, Adrian Perrig*
*ETH Zürich, †Princeton University
Web PKI

• Essential building block for security on the Internet
• Basis of TLS, HTTPS, DoH, DoT, ...
• Myriad of improvements and extensions
  • OCSP (stapling)
  • Certificate Transparency
  • ACME
  • ...
Web PKI is Too Rigid

Equal Trust placed into a fixed set of CAs

Heterogeneous global society requires more flexibility!

I trust CA X more than CA Y

I trust CA Y more than CA X

I trust CA X but only for .ch domains

I trust CA Y for .ch domains and CA X for all other domains
Problems in the Web PKI

Weakest Link Security

No Trust Flexibility

Limited Control for Domain Owners

Lack of Innovation

“Certificate Transparency in the Wild”, Li et al., CCS ’19
Lack of Innovation

• All CAs must implement a new security measure
  • Lack of incentives to be the first one to innovate!

• Trust root changes cause collateral damage
  • Removal of CAs leads to unavailable (secure) websites

“Experiences Deploying Multi-Vantage-Point Domain Validation at Let’s Encrypt”, Birge-Lee et al., USENIX Security ‘21
Flexible PKI (F-PKI)

- Fix for weakest link security in Web PKI
- Flexible notion of trust
- Increased control over certificates for domain owners
- Incremental deployability
- No server-side modifications in HTTPS
- Leverages existing CT infrastructure
Web PKI
Web PKI: Weakest Link Security
Fix Weakest Link Security

Validate certificates from all CAs ⇒ detect misbehaving CA

1. How do we fetch all certificates?
2. What are conflicting certificates?
   • Different public keys?
   • Different Issuers?
Domain Owner Defines Conflicts

F-PKI Policies:
• Allowed Issuers
• Allowed Subdomains
• Allow Wildcards
• Maximum Lifetime
• ...

Owner of bank.com

CA

CSR

Policy: No subdomains

Policy: No subdomains
Web PKI Validation

Relying Party

Domain

HTTPS

X.509 Cert

Web PKI Validation

Accept

Reject
Policies signed by the CAs CA_X, CA_Y, and CA_Z are considered, i.e., these CAs are said to be "highly trusted".
Use Strongest Possible Policy

Policies:

- Allowed Issuers (intersection)
  \[ CA_1, CA_2 \cap CA_1, CA_2 \cap CA_2, CA_3 = \{CA_2\} \]

- Allowed Subdomains (intersection)
  \[ \{a-z\}.example.com \cap \star \cap b.example.com = b.example.com \]

- Allow Wildcards (logical conjunction)
  \[ \text{Allow} \land \text{-} \land \text{Disallow} = \text{Disallow} \]

- Maximum Lifetime (minimum)
  \[ \min(10 \text{ years} , 1 \text{ year} , 3 \text{ months}) = 3 \text{ months} \]
Enable Trust Flexibility

Highly trust CAs using multi-vantage point ACME
User-Dependent Trust

Highly trust CAs using multi-vantage point ACME

Highly trust CAs located in my own country

Who is highly trusted?

{US CAs}

{Swiss CAs}

D_1
D_2
...
D_m

Policy

Policy

Policy

Policy
User-Dependent Trust

Highly trust CAs using multi-vantage point ACME

Highly trust CAs located in my own country

I highly trust US CAs

f() = \{US CAs\}

I highly trust Swiss CAs

f() = \{Swiss CAs\}
Domain-Dependent Trust

Highly trust CAs using multi-vantage point ACME

Highly trust CAs located in my own country

Highly trust Google CAs for google.com subdomains

Highly trust US CAs for .gov Domains

How can I define f()?
Domain-Dependent Trust

Highly trust CAs using multi-vantage point ACME

Highly trust CAs located in my own country

Highly trust Google CAs for google.com subdomains

Highly trust US CAs for .gov Domains

How can I define f()?  
f(*) is defined over the validated domain!

f(*.google.com) = \{Google CAs\}

f(*.gov) = \{US CAs\}

![Diagram showing domain trust]

---

Network Security Group

27.04.2022
What is our Policy Oracle?

Map Servers!

• Fetches certificates from CT log servers
• Provides mapping from domain to all existing certificates
• Uses a sparse MHT to store certificates and verify correct operation
• Provides cryptographic proof of the (non-)existence of a certain domain to certificate set mapping
Certificate and Proof Retrieval

1. (certificate request)
2. (log submission)
3. (issuance)
4. (cert. upload)
5. 6. (TLS)

Domain Owner → Certification Authority

Web Server

Option a. (stapling)

Map Servers

Option a. (stapling)
Option b. (fetching)

Browser

Highly Trusted
CA₁, CA₂, ...

Trusted
CA₃, CA₄, ...

Certificate
Name(s): ...
Public key: ...
Issuer: ...
Policies: ...
Signature: ...

Network Security Group
Certificate and Proof Retrieval

1. (certificate request)
2. (log submission)
3. (issuance)
4. (cert. upload)

Domain Owner

Web Server

Option a. (stapling)

5. 6. (TLS)

Option b. (fetching)

Browser

Map Servers

Highly Trusted

CA1, CA2, ...

Trusted

CA3, CA4, ...

Certification Authority

Certificate
Name(s): ...
Public key: ...
Issuer: ...
Policies: ...
Signature: ...

Network Security Group 27.04.2022
Certificate and Proof Retrieval

DNS

Stapling
Conclusion

• F-PKI enables innovation and trust flexibility in the Web PKI
• F-PKI extends CT and is incrementally deployable
• Working proof-of-concept implementation

Thank you for your attention!

Cyrill Krähenbühl
Network Security Group
ETH Zürich
cyrill.kraehenbuehl@inf.ethz.ch