Public Key Infrastructures

Ralph Holz

Network Architectures and Services
Technische Universität München

November 2014

Part 3: Proposals to enhance or replace X.509

Can X.509 be reinforced? Or replaced?

No ‘silver bullet’ known that would resolve all issues

- Attacker model of SSL/TLS + X.509 $\approx$ protect credit card numbers
- State-scale attacks were not in scope back in the 1990s

For the exam:

- Your reader will include two chapters summarising X.509 history and these proposals
- Feel free to ask—the following has not made it to regular text books yet
- We include it because we want you to be able to deal with technologies that may be ubiquitous and relevant in a few years

Several recent proposals

Group by technology/mechanism:

- Hardening certification
- Pinning: store client-local information about a site
- Store information in the DNS, use DNSSEC
- Notary principle
- Public logs

A word of warning

Some of these concepts are relatively new

- Only a few have reached IETF
  - E.g. DNS-based authentication of names entities (DANE), Certificate Transparency
- Others may yet enter an IETF track:
  - E.g, TACK
- Some are available in code, but have no standard

The underlying ideas are very relevant

- Many concepts well-known, but never before applied to X.509
- The concepts can be applied to many security mechanisms

Background: attacker models

Always assess a concept against attacker definition

- We are not going to give such definitions here, but consider the differences between the following
- Weaker attacker:
  - E.g., on WiFi access point, or some local network gateway
  - May control DNS traffic, but cannot interfere with DNSSEC
- Regional attacker:
  - Controls all traffic of a country
  - Control over routing, control over DNS
  - Controls own top-level domain (DNSSEC!)
  - May compromise CA
- Supra-regional attacker. Same as above, plus:
  - ‘Cyber-war’: a state risking ‘digital military confrontation’
  - Attacks on global routing (BGP, possible)
  - Attacks on infrastructure to control DNSSEC

Hardening certification

Aim: make it harder to trick CAs into misissuance

- Idea: agree on a common set of rules for certification
- The CA/Browser Forum is a collaborative body of browser vendors and CAs
- Founded in 2005
- The members have so far released two ‘standards’:
  - Extended Validation (2010)
  - Baseline Requirements (2012)

CA/Browser Forum standards

Extended Validation (EV)

- CAs strictly require state-issued documents before certification
- Certificates have OID that browsers evaluate
- More expensive, rarely bought by customers
- Liability introduced (capped at 2000 USD)

Base Line Requirements

- Minimum requirements for validation, forbid less secure practices
- E.g. how to contact requester (email), length of keys etc.
- No liability prescribed

Discussion of these standards

The standards define common practices:

- Forbid insecure practices, require strong cryptography
- They do not add any new security concepts
- The ‘weakest link’ argument continues to hold

The standards are a pure agreement:

- There are no real sanctions if standards are violated
- CAs have repeatedly violated the standards agreed upon:
  - Certificates without revocation information
  - Certificates with keys that are too short
  - Certificates with expiry periods that are too long
- Maybe birth pangs; remains to be seen if situation improves
- Far from being a solution (in any attacker model)

Pinning

Aim: reassurance of a certificate's authenticity

- As a defence against rogue CAs issuing malicious certs
- Idea: client stores information about a host/Web site on first contact
- Most commonly: store the public key of a site
- Use this information to re-identify a site later
- E.g. if public key is suddenly different on next connect: warn user

Pinning assumes a secure first connection

- Thus also known as 'trust-on-first-use'
- Inherent bootstrapping problem

Two pinning variants

Static pinning

- Preloaded pins:
  Google Chrome, Mozilla Firefox (smallish number)
- User-driven pinning:
  add-ons for browsers that allow users to store and compare public keys of sites

Dynamic pinning

- Idea: communicate helpful information to aid clients with pinning

Issues to solve

Depending on the variant, pinning has shortcomings:

- For certain users, secure first contact may not be possible
  - E.g. dissidents in authoritarian countries
- Life-cycle problem
  - Servers may (legitimately) update/upgrade their keys
- Scalability
  - Browsers cannot come preloaded with pins of all sites, and keep them up to date

Public Key Pinning for HTTP

An IETF Internet draft to provide pinning for HTTPS

- Dynamic pinning
- Servers communicate life-time and hash value of their X.509 public key in the HTTP header
- In-band, i.e. secured with the same TLS/X.509 mechanisms
- Addresses short-comings of simple pinning:
  - Life-cycle management for X.509 key upgrade or compromise
  - Addressed with ‘backup pins’: second set of backup X.509 keys, whose pins are always communicated in addition to the primary ones
- Advantage: easy to deploy, no problems for clients that are not aware of the pinning
- Features reporting function: report key mismatches to a URL!

More complex variant

Trust Assertions for Certificate Keys (TACK)

- Servers have separate public/private key pair: TACK keys
  - Servers send so-called tacks in a TLS extension in the handshake:
    - Signed information about currently valid X.509 key plus an expiry time
    - Tacks have ‘generations’ (form of sequence numbers): this creates a window of currently valid tacks
    - Tacks allow to communicate X.509 key roll-over and compromise as well as roll-over of TACK keys
    - Clients pin to the tacks
- Advantage: highly flexible, out-of-band mechanism
- Disadvantage: need for TLS extension (server upgrade necessary!)

Pinning

Extremely strong if assumption of secure first connection holds

- Attacker can only attack client or server, but there is no other Trusted Third Party to compromise
- Practical usefulness first demonstrated by Google:
  - Google pins all Google sites in their browser (static pinning)
  - This was how the DigiNotar incident was detected!
- This concept can hold up to any attacker who cannot compromise either client or server

Public log schemes

Idea of a public log

- Public logs store some information publicly and append-only
- They sign every new entry and establish a ‘history’ of entries
- Public logs are neutral. Their only role is observe and assert their observations by signing them.
- **Certificate Transparency (CT):** logs for X.509
  - **Aim:** make **transparent** who issued certificates to whom, and when
  - **Anyone** can verify logs’ content and/or their correct operation
  - Enables detecting rogue CA issuing certificates for a domain
  - Proposes 30+ logs around the globe, run by different parties
  - **After-the-fact** solution; no direct defence for clients

Public log: a Merkle Hash Tree

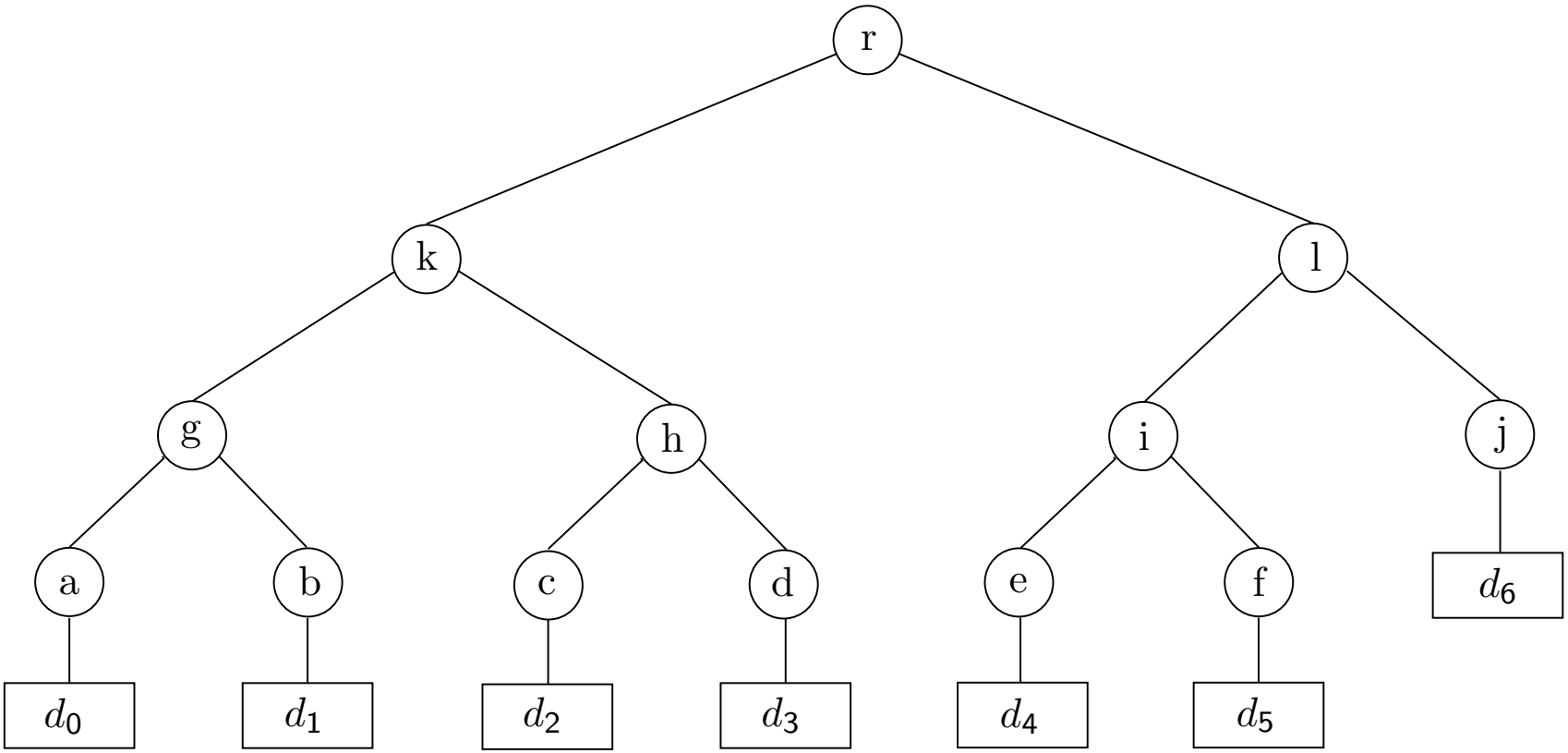


Figure : Log is a Merkle tree, d_i are new certificate chains.

Proving properties of Merkle Hash Trees

The tree structure is beneficial for proving certain conditions are met

- These proofs do not require full copies of the tree—a subset, logarithmic in size, is enough
- The algorithms to determine the subsets, and how to carry out the proofs, are described in RFC 6962
- The logs must allow to retrieve the necessary subset for any given certificate in the tree
- So-called monitors and auditors are entities that continously watch the operation of logs and use these proofs to determine the logs are well-behaving
- This is a form of '**cross-validation**': watching the watchers

Proofs

Consistency

- Prove the append-only property
- Prove that no certificate was removed from the tree, or some certificate injected in the wrong position
- Works by obtaining subset of nodes needed to prove that tree from a certain moment t_0 on always adhered to the append-only property
- In other words: the logs cannot fake the logged history once they have started logging

Inclusion (audit path proof)

- Prove that a certificate has been included in the tree

Watchers: monitors

Computationally powerful entities tracking the operation of several logs

- Primary function: continously verify the *append-only* property (consistency checks)
- Act on behalf of less powerful entities, e.g. browsers or domain owners
- Possible parties fulfilling this role: ISPs, CAs. But anyone is free to set up a monitor.
- Secondarily, they may also keep copies of logs
- This enables them to search for violating certificate issuances:
  - E.g. they have a list of domains to ‘protect’
  - They may watch continously if a second certificate for a domain appears, which the domain owner never authorised

Watchers: auditors

Auditors are computationally less powerful entities

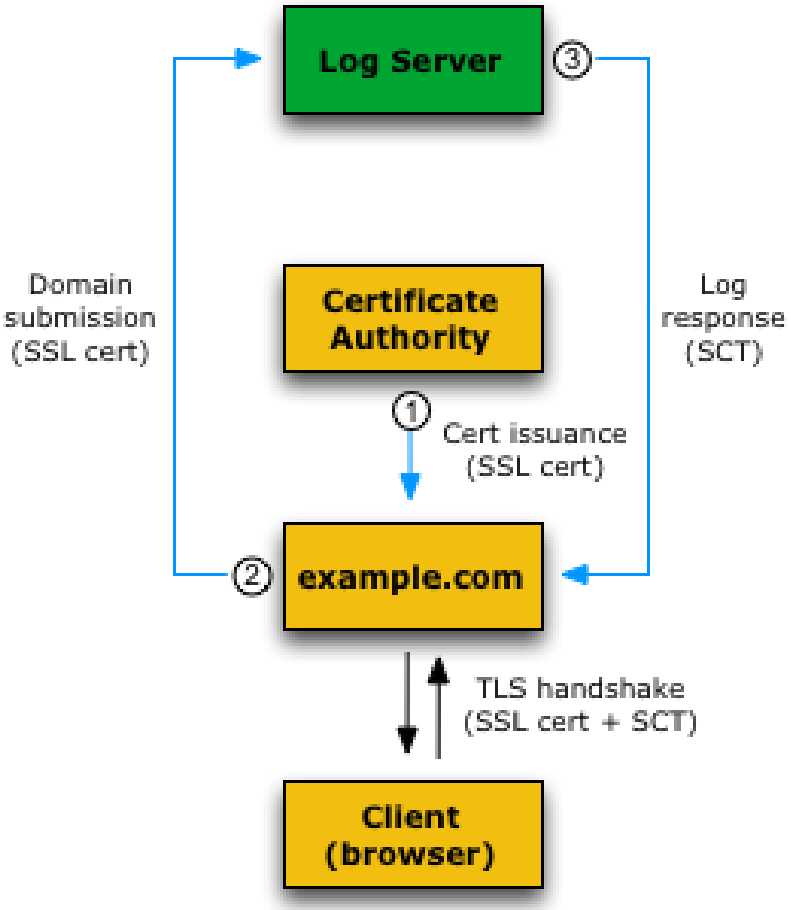
- Typically, they do not keep copies of the logs
- Typical parties fulfilling this role: browsers
- Auditors may check either consistency (like monitors, but without having copies of the logs)
- They may also do inclusion checks

Interactions: logs and other parties

Certification: CAs and logs

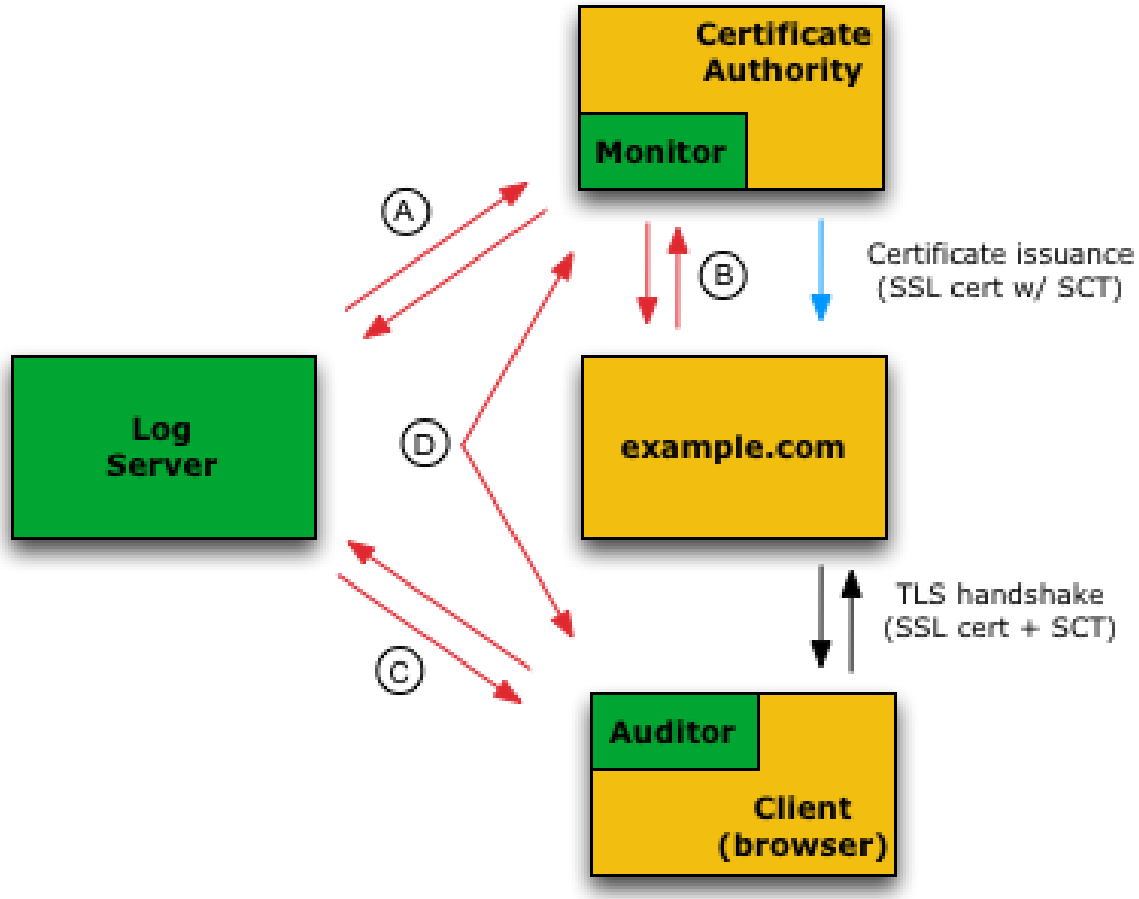
- When a CA issues a certificate, it must send it to at least two logs for incorporation
- A log returns a Signed Certificate Timestamp (SCT) proving it has accepted the cert
- This is forwarded to the actual domain operator
  - As part of OCSP Stapling, or
  - Using a TLS extension, or
  - Incorporated into the X.509 cert
- The OCSP Stapling solution is the current favourite; it is unclear yet which one will prevail
- The SCT is sent to any TLS client connecting to the domain so the client knows which logs track this cert

Figure: logs and TLS/X.509



Source: certificate-transparency.org

Figure: logs, auditors, monitors



Source: certificate-transparency.org

Gossiping

Problem: split-horizon attacks

- Monitors and auditors cannot prevent logs from keeping ‘alternate histories’, where one history is the real one, shown some parties, and the other is a fake one, shown to other parties
- With considerable effort, such split-horizon attacks can be used by attackers to bypass the cross-validation system and trick clients
- Thus: gossiping between auditors, monitors, TLS clients
- Gossiping is not yet specified, but here are the main ideas:
  - Clients, auditors and monitors should notify domains which tree head they see
  - This means that logs showing alternate histories to some clients will be ultimately detected

Discussion of Certificate Transparency

Advantages

- Adds transparency to X.509 in the hope of detecting malicious behaviour early
- If deployed correctly, CT may have strong change of being a serious reinforcement to X.509, thwarting even state-level attackers

Potential issues

- No direct, immediate help for clients
- Very complex setup
- Very expensive operation
- Needs changes on the side of CAs

Background: DNSSEC

DNSSEC is an addition to the classic DNS

- DNS organised in a tree, consisting of zones holding domain names
- Idea: create PKI along this tree (ideal structure!)
- Root zone (top node): signed by one global trusted key
- A zone is signed by its parent zone, and signs its subzones
- Resource records (domain information) are signed
- Problems:
  - User clients do not validate—task left to resolvers
  - Countries control their top-level domain: can hijack subdomains
  - Complex technology, little deployment
- This is enough to understand the following; more later

Certificate Authority Authorization (CAA)

CAA resource record: store which CA is responsible for a domain

- E.g. Google may add value `symantec.com` to resource records for `google.com`
  - Try it: `dig +short -t TYPE257 google.com`
- The value is a unique identifier for a CA
- Before issuing a certificate for a domain, a CA is supposed to query the CAA record
- The CAA record also allows to define a URL where one can report violations, e.g. if you find a certificate that is not from the CA defined in the CAA record
- CAA does **not** mandate DNSSEC

Discussion of CAA

Advantages

- Very simple and cheap concept (just DNS entries and queries)
- Reinforcement to the issuance process—CAs can quickly query if the domain owner **wants** them to be responsible
- Avoiding DNSSEC reduces complexity considerably
- The URL for reporting is a very valuable addition

Issues

- No DNSSEC means well-positioned attacker can interfere with DNS query (even weakest attacker we discussed)
- No direct protection for **clients**
- No defence at all when a CA is compromised

DANE: DNS-based authentication of named entities

TLSA record stores a so-called ‘trust anchor’ for TLS-based communication with domain

- Store certificate or public key of domain
- Alternatively, store certificate or public key of CA instead
- No need to have CA-issued certificate: supports self-signed certificates, too
- Instead of storing certificates or public keys: can also store their hash values
- DANE-TLSA mandates use of DNSSEC (even forbids connection on mismatch between X.509 certificate in connection and the values in the record)

Discussion of DANE-TLSA

Advantages

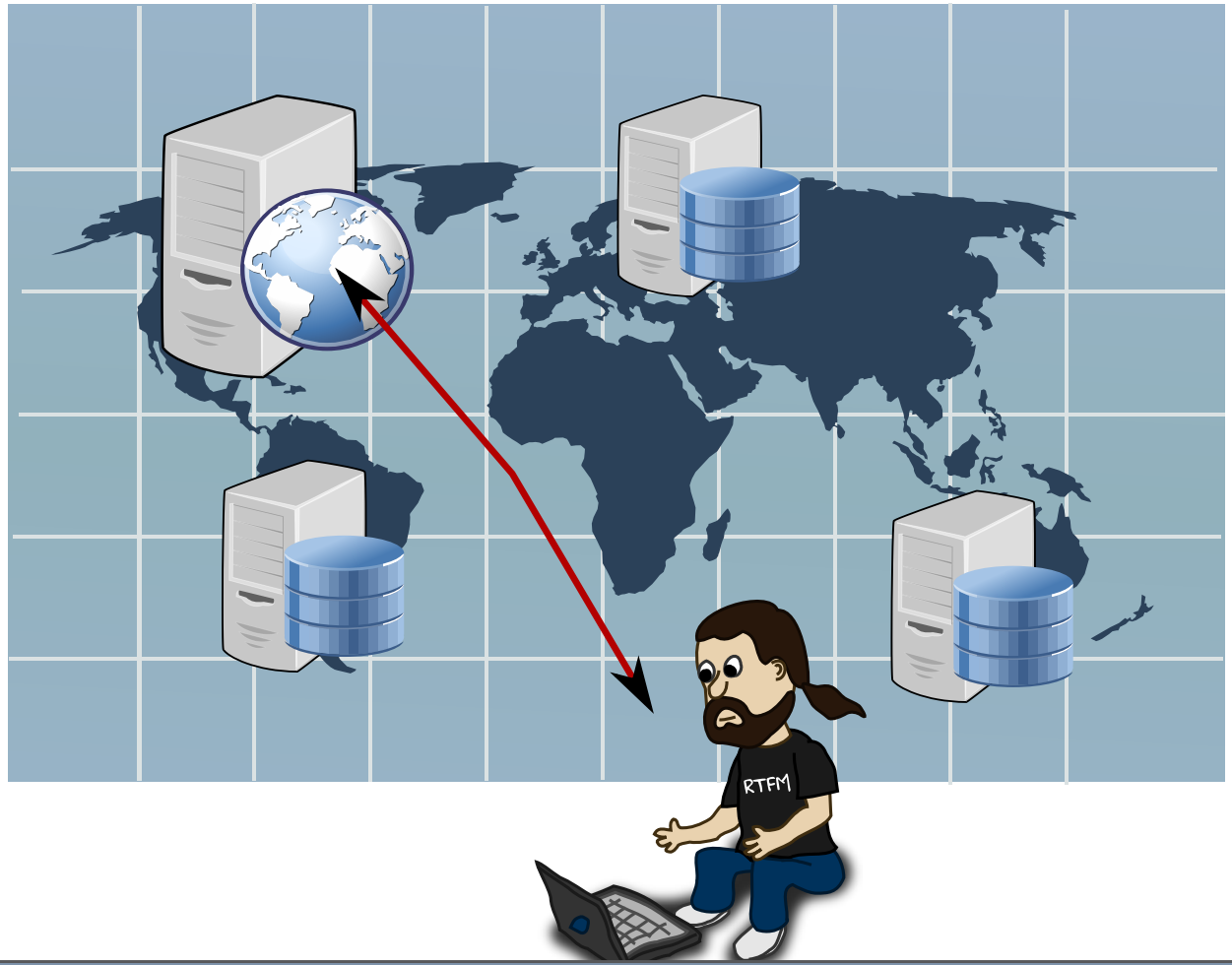
- Out-of-band mechanism with strong reassurance on certificate validity
- Protects completely against our weaker, local attacker

Potential issues

- DNS operators need to become PKI operators—same level of assurance like CA checks?
- Countries are often in control of their TLDs—think of `bit.ly`. This enables state-level attacks:
  - Regional attacker: can modify TLSA records of his zone
  - Global attacker: may be able to modify some other zones, too

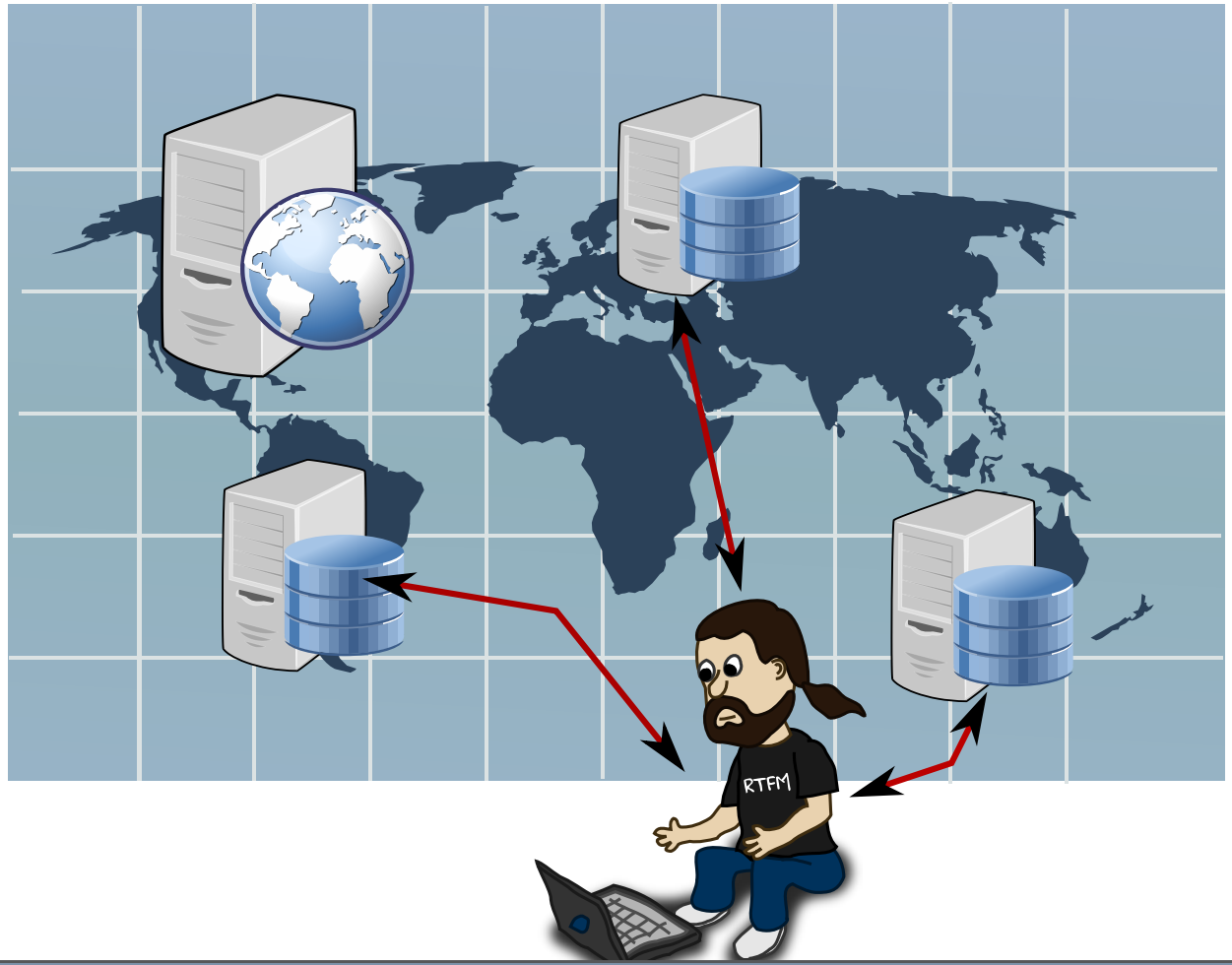
Notary-based systems

When connecting to a host and receiving the TLS certificate...



Notary-based systems

... connect to some special notaries elsewhere and double-check



Perspectives: a notary system

Principle of operation:

- Assumption: no attacker can control all paths through the Internet
- A number of notary systems are distributed around the globe, run by independent operators
- Notaries **scan** a list of domains regularly. Store and sign which certificates they see, at which time.
- Each notary also **shadows** a number of other notaries:
  - Downloads their observations and signs and stores them, too
  - Checks for inconsistencies: no contradicting entries
  - Defence against misbehaving or compromised notaries
- When clients connect to a domain, they receive a certificate. They double-check with 1-2 notaries **and** their shadows.

Discussion: Perspectives

Nota bene:

- The security of the system depends **entirely** on the attacker's capability to compromise notaries and on his position in the network

Some discussion points

- Shadowing concept is powerful variant of **cross-validation**: entities assess each other's compliance with rules of the system
- For it to work well, an attacker must not be able to predict which notaries and which shadows a client is going to use (otherwise the attacker can focus on compromising those notaries/shadows only)

Discussion: Perspectives

... continued

- There need to be many notaries—otherwise the attacker can compromise all of them and the system is broken
- Given enough notaries, one can make a stochastic argument that the effort for the attacker to break all notaries/shadows becomes quickly unreasonable
- An attacker sitting on ‘the last hop’ to a server can trick all notaries, however
- Big advantage: changes only necessary on client-side, no consequences for servers or CAs

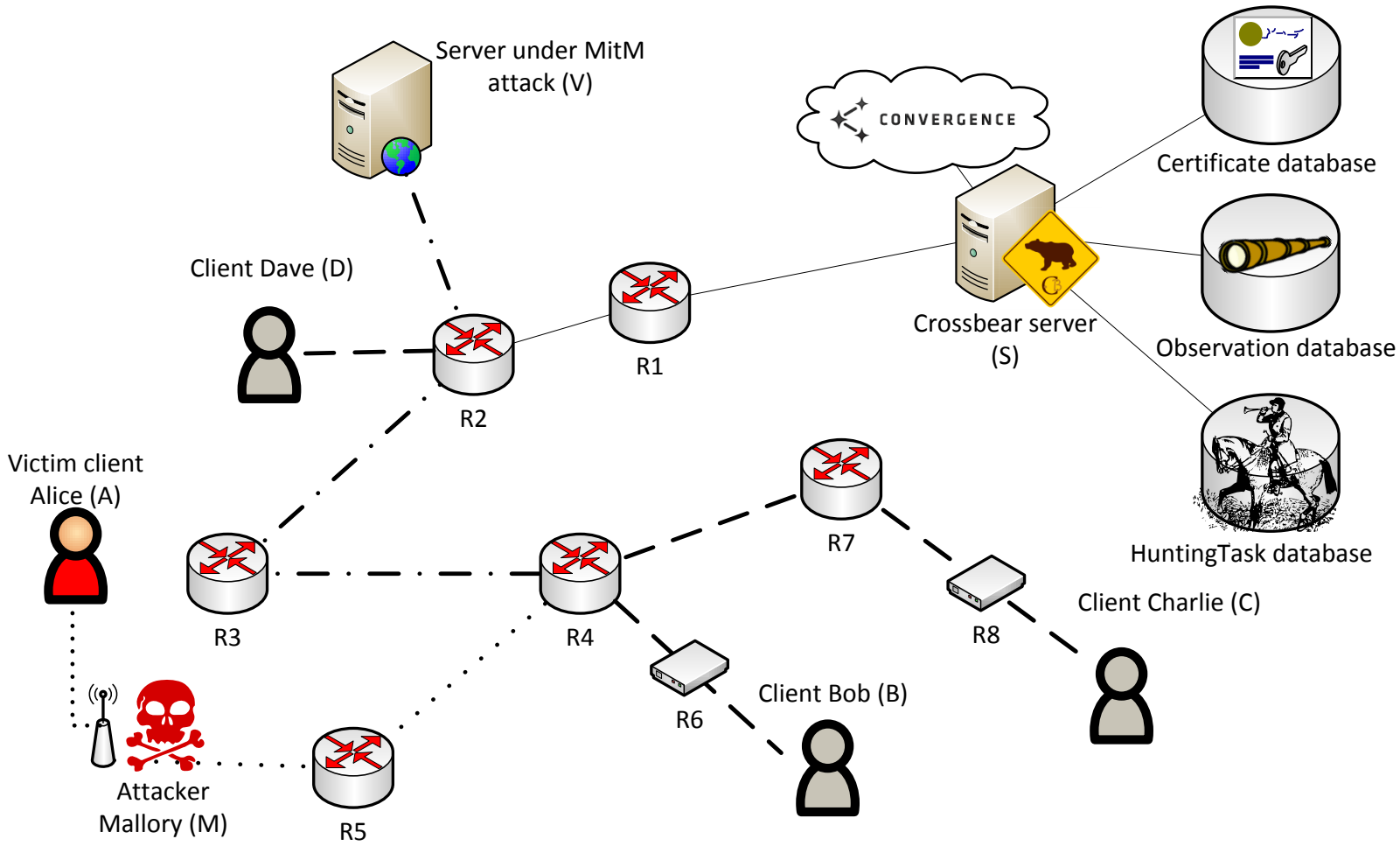
Notary-based systems

Examples

- Perspectives (CMU, 2009): browser plug-in
  - In operation
  - But shadow concept never implemented
  - Few notaries—project cannot guarantee their benign intentions
- Convergence (Marlinspike, 2011): browser plug-in, discontinued
- Crossbear (Holz, 2011):
  - Different goal: detect attacks by finding mismatches between notaries and clients
  - Interpret a mismatch as potential attack, try to determine position of attacker

Crossbear

Goal is *detection and localisation*



Attempt: summary of proposals

There is no candidate that solves all issues. Here is a biased view:

- Certificate Transparency is gaining support and stands a good chance to make the X.509 PKI transparent to investigations
- Pinning is very powerful, but probably only an addition to other concepts
- DANE-TLSA has support, but little deployment so far (DNSSEC)—this remains to be seen
- Notary concepts are relatively complex and have little to no support so far