Certificate Transparency

Anupam Datta

Certificate Transparency

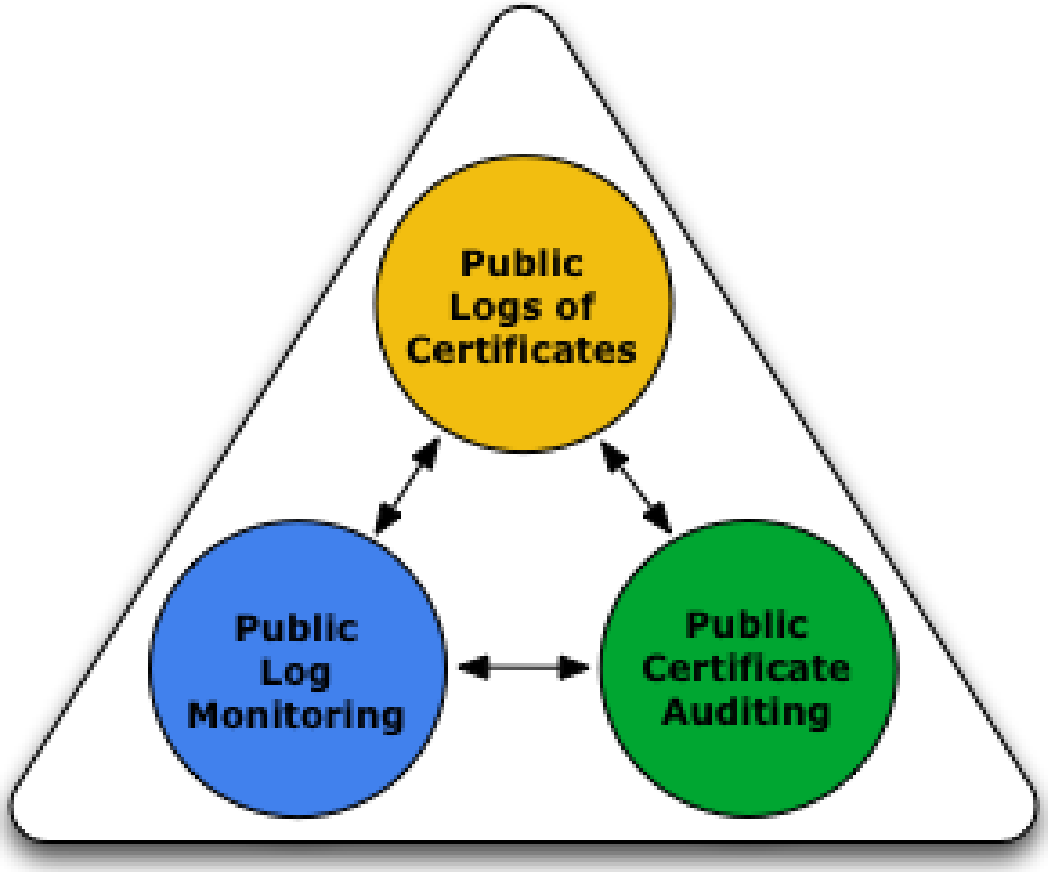
"Google's Certificate Transparency project fixes several structural flaws in the SSL certificate system, which is the main cryptographic system that underlies all HTTPS connections. … If left unchecked, these flaws can facilitate a wide range of security attacks, such as website spoofing, server impersonation, and man-in-the-middle attacks."

https://www.certificate-transparency.org/

Goals

- Make it impossible (or at least very difficult) for a CA to issue a SSL certificate for a domain without the certificate being visible to the owner of that domain.
- Provide an open auditing and monitoring system that lets any domain owner or CA determine whether certificates have been mistakenly or maliciously issued.
- Protect users (as much as possible) from being duped by certificates that were mistakenly or maliciously issued.

Big Picture


Public
Logs of
Certificates
Public
Log
Monitoring
Public
Certificate
Auditing

Components (1)

- Certificate logs
  - Network service for cryptographically assured, publicly auditable, append-only records of certificates
  - Certificates submitted by CAs or anyone else
  - Run by Google, DigiCert, Symantec,…

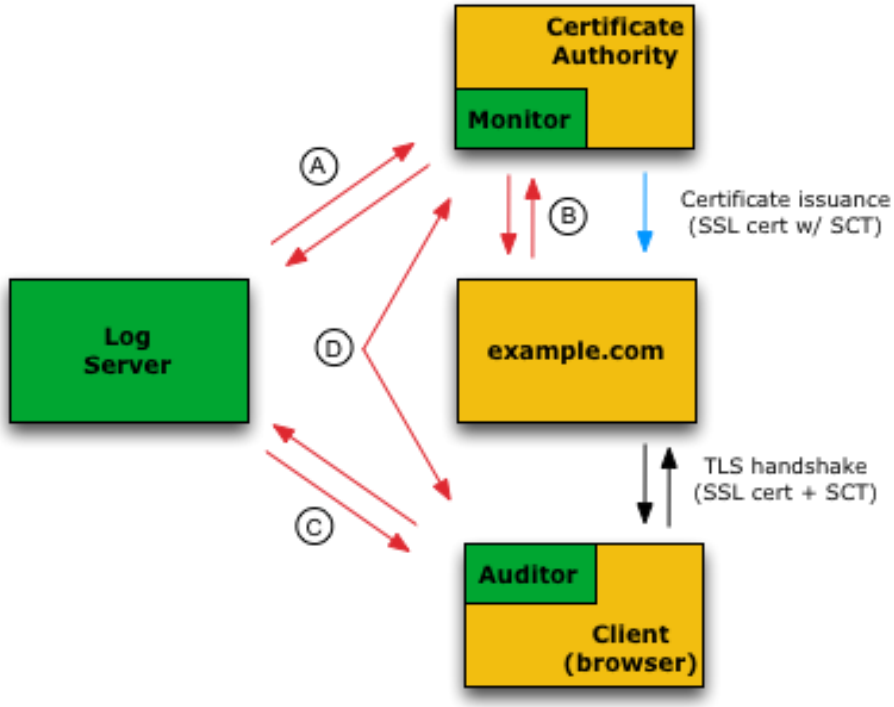
Components (2)

- Monitors
  - Analogy: Credit-reporting alert to website owner
  - Servers that periodically contact all of the log servers and watch for suspicious certificates
    - illegitimate or unauthorized certificates, unusual certificate extensions, or certificates with strange permissions (for example, CA certificates)
  - Verify that all logged certificates are visible in the log.
  - Run by website owners or subscription service for website owners

Components (3)

- Auditors
  - Lightweight software component typically on TLS client
  - Verify logs behaving correctly and cryptographically consistent.
  - Verify that a particular certificate appears in a log
  - Part of browser's TLS client, standalone service, or a secondary function of a monitor

Typical Configuration



Ⓐ Monitors watch logs for suspicious certs and verify that all logged certs are visible.

Ⓑ Certificate owners query monitors to verify that nobody has logged illegitimate certs for their domain.

Ⓒ Auditors verify that logs are behaving properly; they can also verify that a particular cert has been logged.

Ⓓ Monitors and auditors exchange information about logs to help detect forked or branched logs.

Figure 3

Key Idea:
Log Proofs

Logs with Merkle Tree Hash

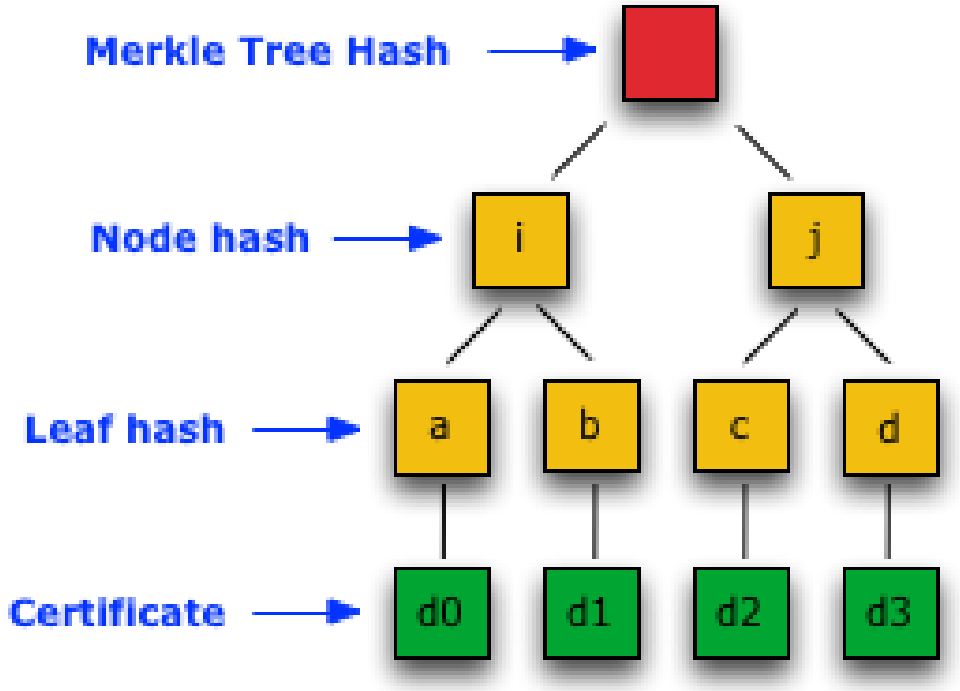


Figure 1

Log server signs Merkle Tree Hash

Adding Certificates to Log

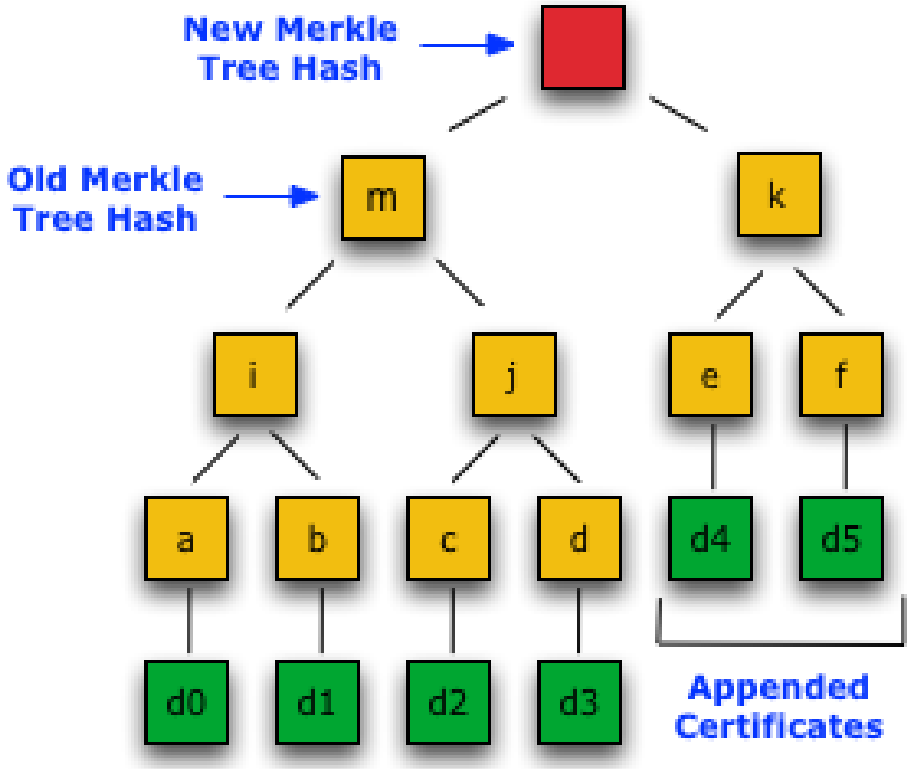


Figure 2

Merkle Consistency Proofs

Later version of log includes everything in the earlier version, in the same order, and all new entries come after the entries in the older version (used by monitors + auditors)

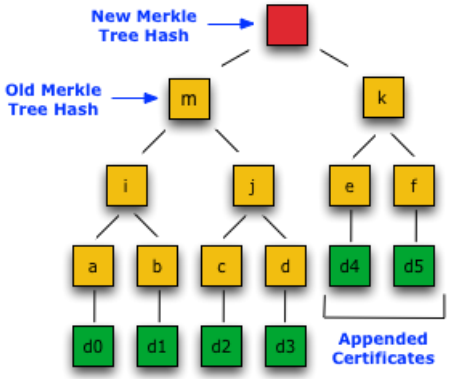


Figure 2

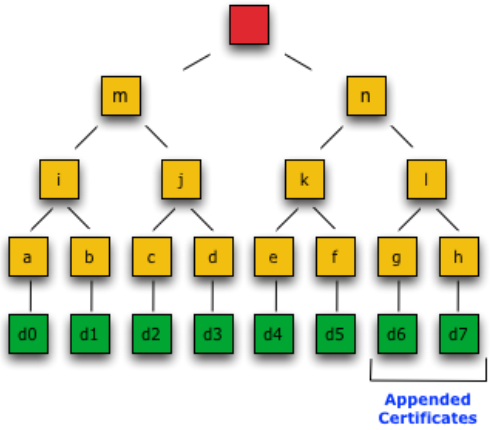


Figure 3

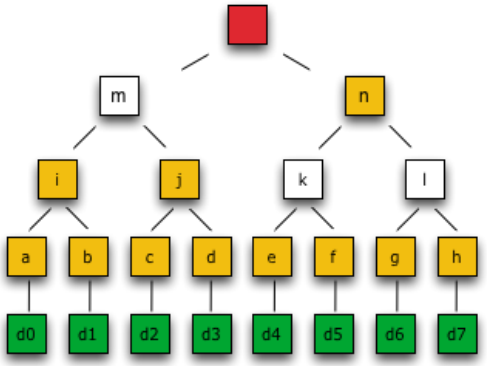


Figure 4

Merkle Audit Proofs

Verify that a specific certificate has been included in a log (used by auditors)

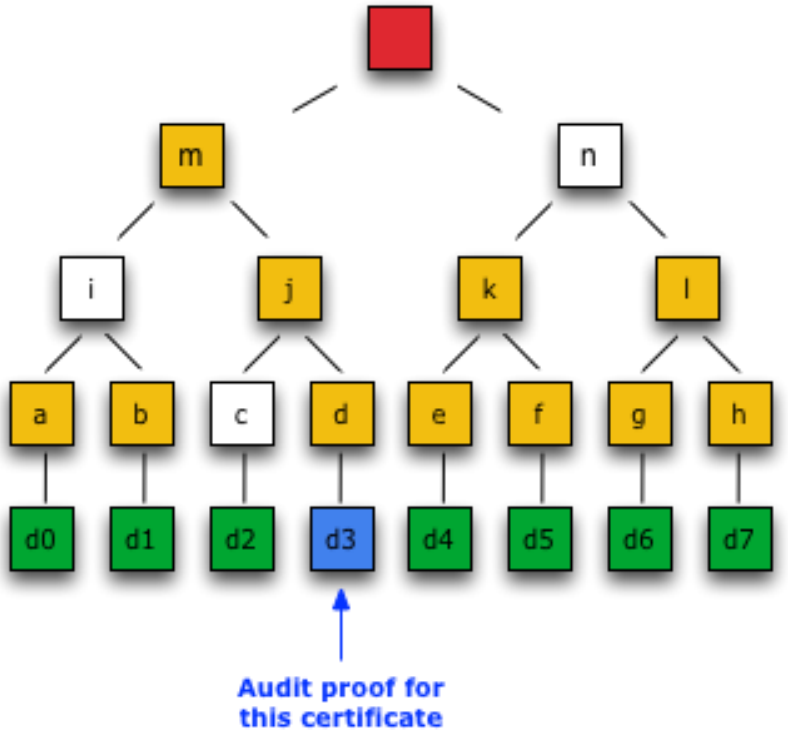


Figure 5

Efficient Proof Checking

A log containing 10 million certificates would require a consistency proof that has only 24 node hashes. If you increase the log size to 20 million certificates the number of node hashes in the consistency proof goes to 25.

Key Idea:
Signed Certificate Timestamp

Log Operation

- Signed Certificate Timestamp
  - When someone submits a valid certificate to a log, the log responds with a signed certificate timestamp (SCT), which is simply a promise to add the certificate to the log within some time period.
- Enables distributed implementation of log server for resilience and availability
- TLS server delivers SCT with certificate

Delivering SCT

- X.509v3 Extension

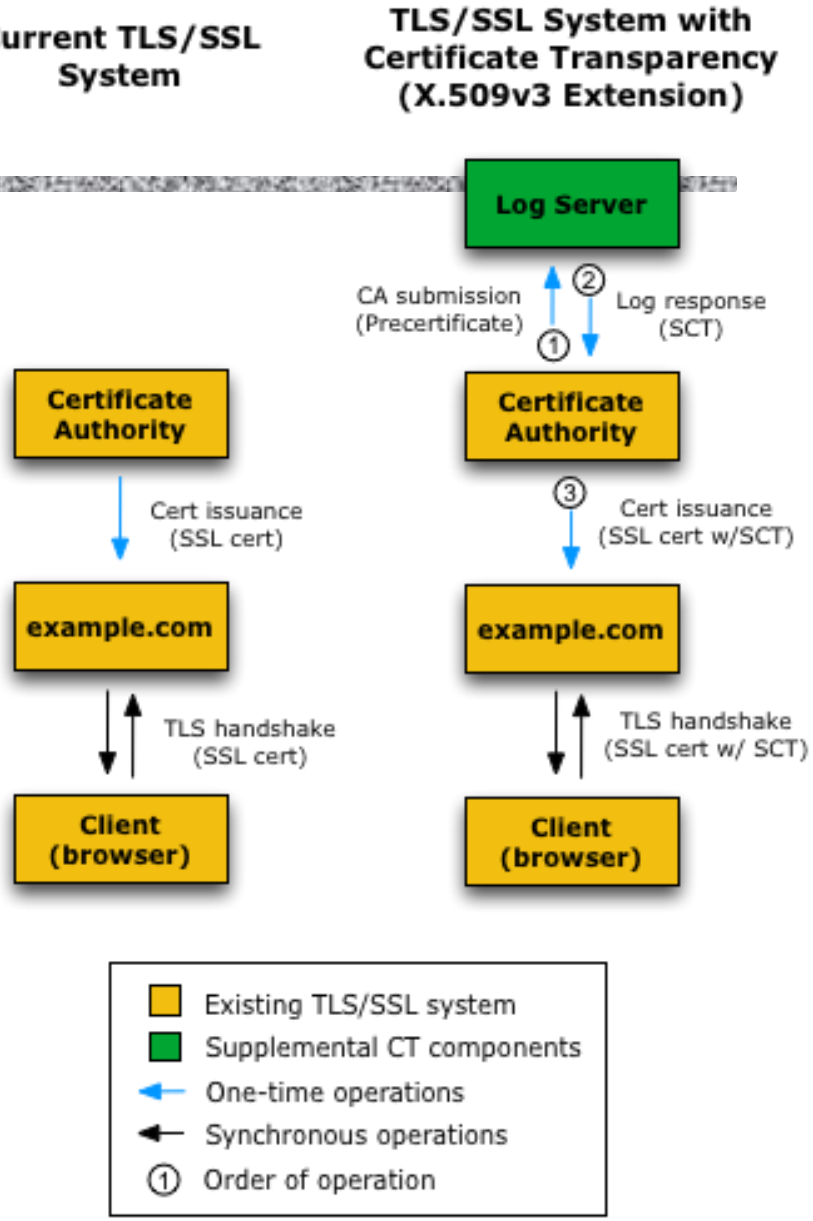


Figure 1

Delivering SCT

- TLS Extension
- OCSP Stapling

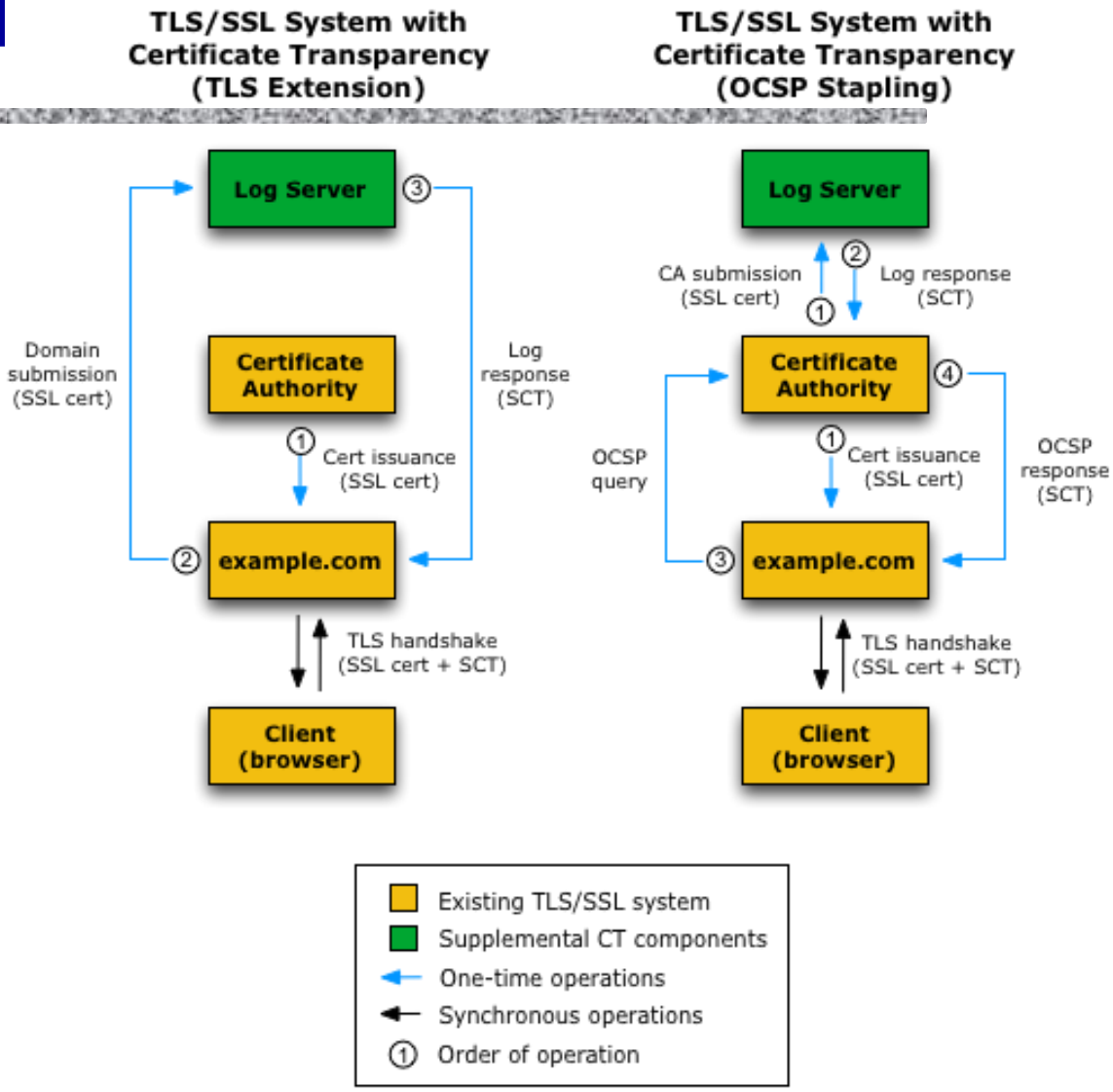


Figure 2

Key Idea:
Gossip protocol

Gossip protocol

- Log proofs allow an auditor or a monitor to verify that their view of a particular log is consistent with their past views
- They also need to verify that their view of a particular log is consistent with other monitors and auditors.

- To facilitate this verification, auditors and monitors exchange information about logs through a gossip protocol.

Gossip with SCT Feedback

- Goal: Transport SCTs and certificate chains from HTTPS clients to CT auditors/monitors via HTTPS servers

Gossip with SCT Feedback

- Steps
  1. HTTPS clients store SCTs and certificate chains they see and later send them to the originating HTTPS server by posting them to a well- known URL

  2. HTTPS servers store SCTs and certificate chains received from clients and later share them with CT auditors by either posting them or making them available on a well-known URL

1. HTTPS client to server

- HTTPS client visits server S:
  - Stores SCT's from known logs + certificate chain
- HTTPS client visits S again
  - Stores new SCT's and sends to server SCT's in its store that it received from other sources
- SCTs and corresponding certificates are POSTed to the originating HTTPS server at the well-known URL:

https://<domain>/.well-known/ct/v1/sct-feedback

HTTPS client to server

- Privacy protection
  - An SCT MUST NOT be sent to any other HTTPS server than one serving the domain that the certificate signed by the SCT refers to
  - Otherwise leaks information about sites visited by clients to servers, auditors, monitors

HTTPS server to auditors

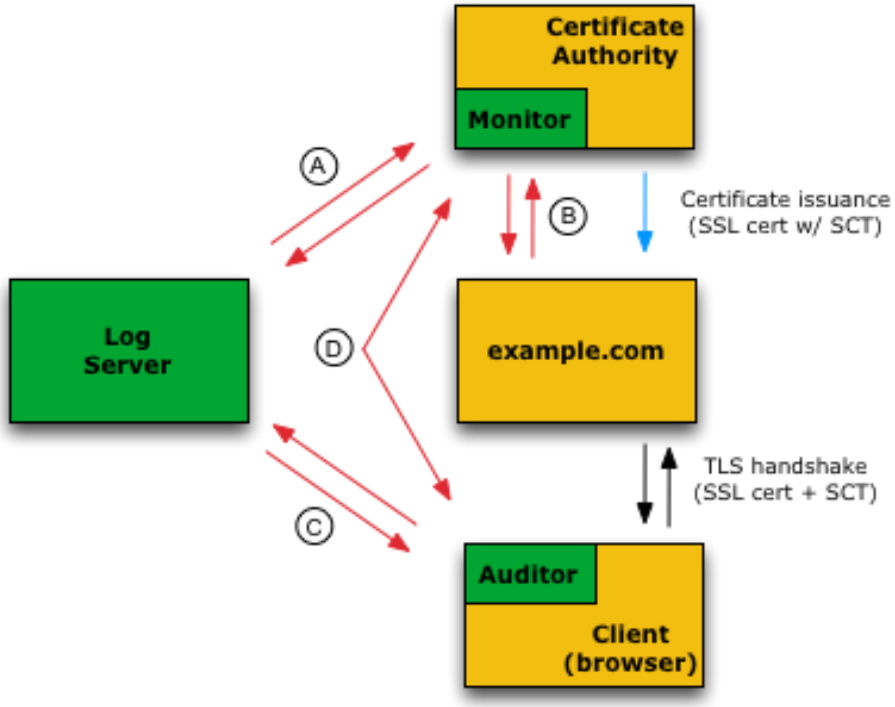
- HTTPS servers receiving SCTs from clients SHOULD share SCTs and certificate chains with CT auditors by either
  - providing the well- known URL: https://<domain>/.well-known/ct/v1/collected-sct-feedback or
  - by HTTPS POSTing them to a number of preconfigured auditors.

HTTPS server to auditors

- Auditors SHOULD provide the following URL accepting HTTPS POSTing of SCT feedback data: https://<auditor>/ct/v1/sct-feedback

- Auditors SHOULD regularly poll HTTPS servers at the well-known collected-sct-feedback URL.

Typical Configuration



Ⓐ Monitors watch logs for suspicious certs and verify that all logged certs are visible.

Ⓑ Certificate owners query monitors to verify that nobody has logged illegitimate certs for their domain.

Ⓒ Auditors verify that logs are behaving properly; they can also verify that a particular cert has been logged.

Ⓓ Monitors and auditors exchange information about logs to help detect forked or branched logs.

Figure 3

Deployment

- Log servers
  - Google, DigiCert, Symantec,…
  - https://www.certificate-transparency.org/known-logs
- Client implementations
  - Apple's new App Transport Security for iOS 9 and Mac OS X 10.11 includes support for Certificate Transparency
  - Chrome 41, released in March 2015, began enforcing the CT requirement for all EV certificates issued after 1 Jan 2015.
- IETF Standardization
  - https://datatracker.ietf.org/wg/trans/charter/