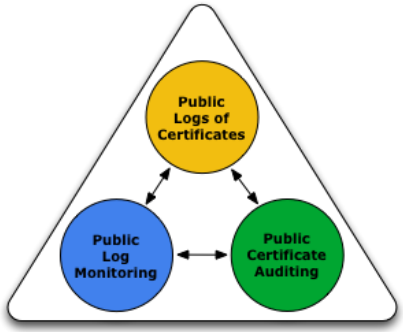
Aggregating Certificate Transparency Gossip Using Programmable Packet Processors

Rasmus Dahlberg, Tobias Pulls, Jonathan Vestin,
Toke Høiland-Jørgensen, and Andreas Kassler

Certificate Transparency—in short CT

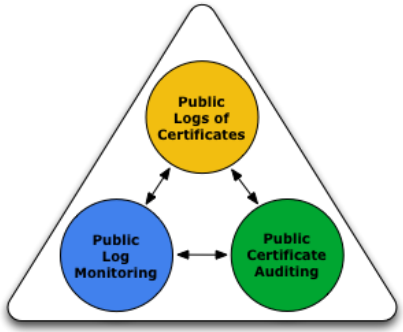
- Goal is to detect mis-issuance
- Publicly log all certificates
- Clients require proof of logging



https://www.certificate-transparency.org/what-is-ct

Certificate Transparency—in short CT

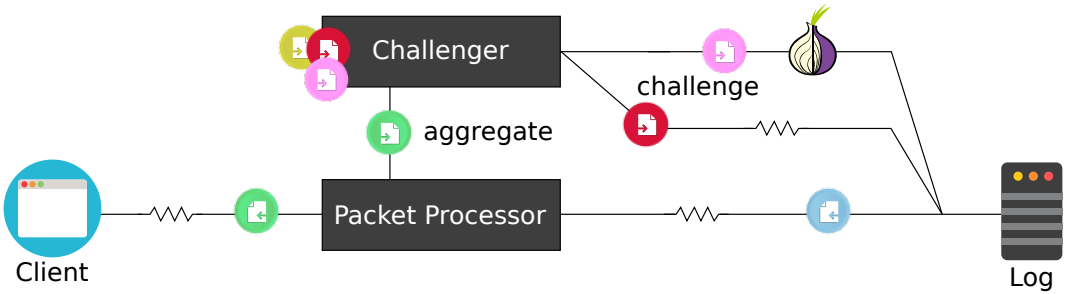
- Goal is to detect mis-issuance
- Publicly log all certificates
- Clients require proof of logging



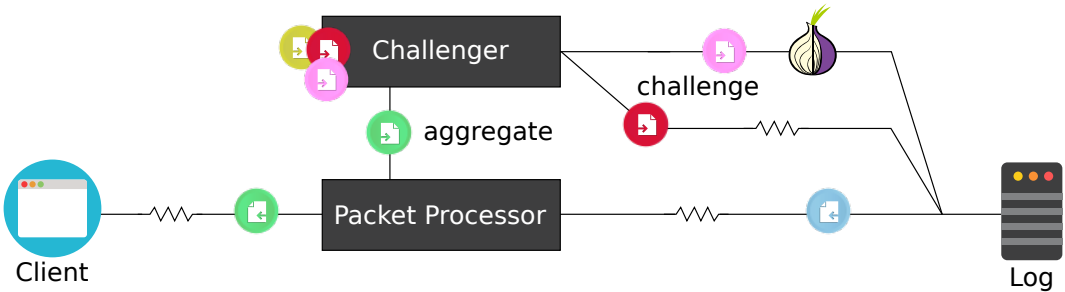
https://www.certificate-transparency.org/what-is-ct

How do you know if you see the same log?

Overview—in-line aggregation and off-path verification

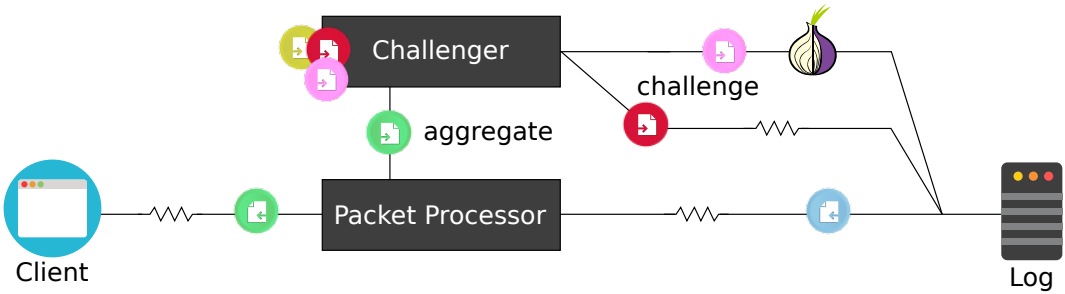


Overview—in-line aggregation and off-path verification



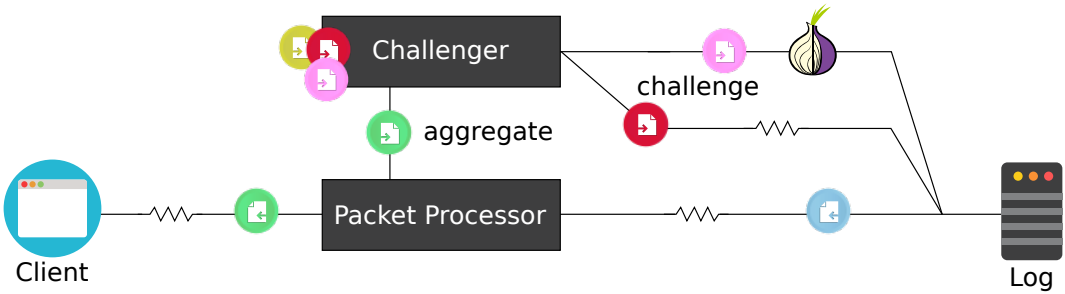
Security notion—aggregation indistinguishability

Overview—in-line aggregation and off-path verification



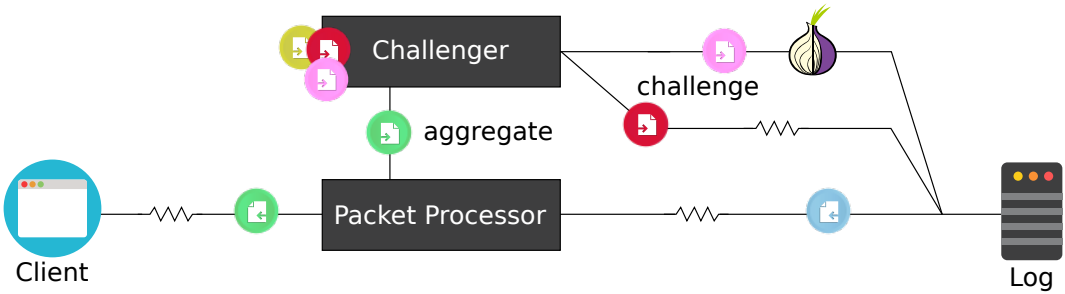
Security notion—aggregation indistinguishability
Intended attacker—distant

Overview—in-line aggregation and off-path verification



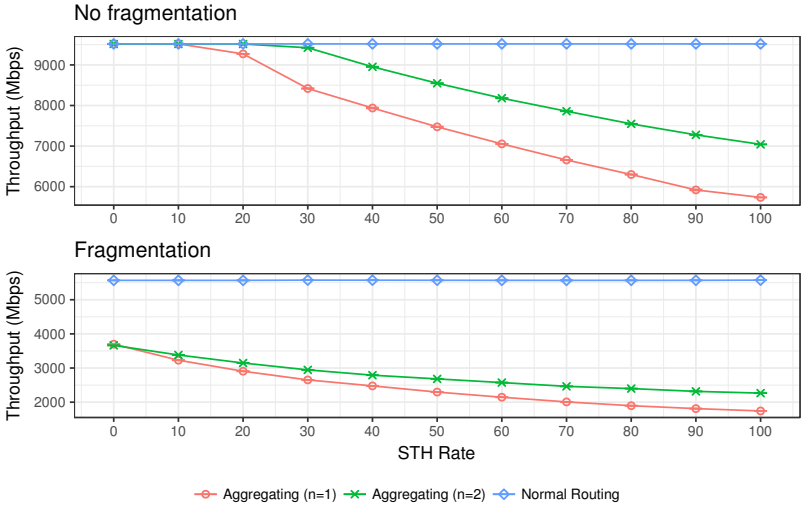
Security notion—aggregation indistinguishability
Intended attacker—distant
(Multi)path fragmentation

Overview—in-line aggregation and off-path verification

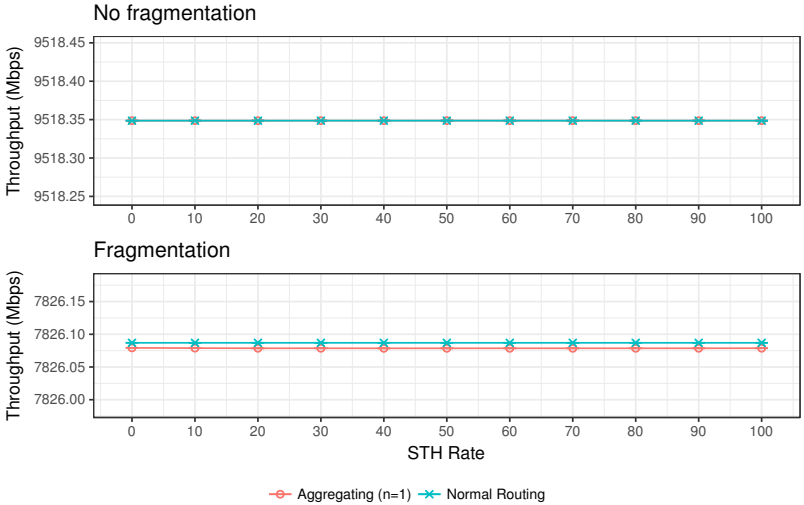


Security notion—aggregation indistinguishability
Intended attacker—distant
(Multi)path fragmentation
Implementation? XDP, P4, ...

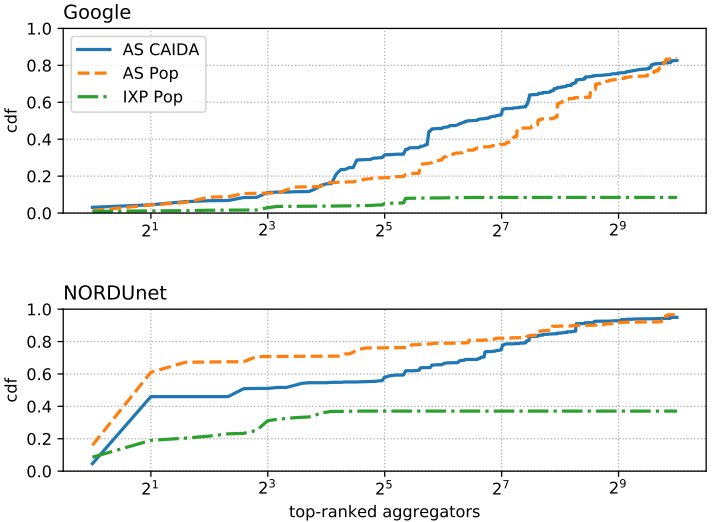
Performance and aggregation indistinguishability—XDP



Performance and aggregation indistinguishability—P4

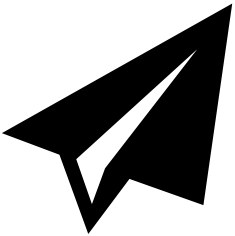


Network measurements—split-view protection against Google and NORDUnet



Take away

- Program the network to gossip ‘as a service’
- Easily deployed, not much opt-in needed



Any questions?