Batch signatures
CFRG IETF 118

Higher throughput via Merkle Trees

The problem

E.g. load balancers, reverse proxies, etc have to deal with many incoming signing requests per second.

Post-quantum signatures have much higher latency than ECC/RSA. This has the potential to cause performance issues when signing threads get clogged.

Can we make post-quantum signatures more performant?

The idea

Construct Merkle tree from message transcripts

Sign the root with a base signature algorithm

Signature is the Merkle tree sibling path (to reconstruct the root) plus the root signature

Sibling path varies for each signature

Same root signature for all in same batch

TLS-specific idea **first proposed** in:

[draft-davidben-tls-batch-signing-00]

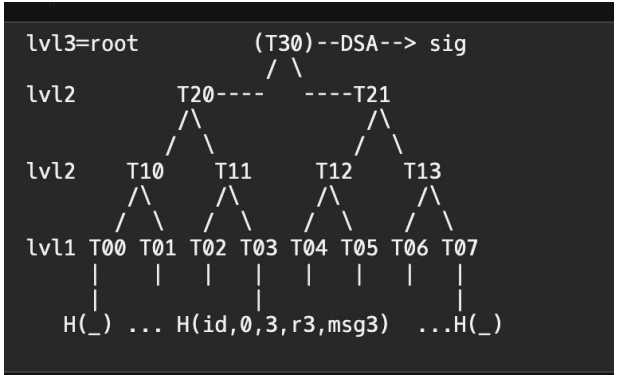

Figure 1: Merkle tree batch signature construction

Updates

In Batch Signatures, Revisited, the authors provide security proofs for the construction, including privacy notions

We based security on Target Collision Resistance, instead of Collision resistance. This enables **halving of the sibling path length.** Technique from SPHINCS+

Development in PQC (and acknowledgement of its performance problems) makes high throughput signing increasingly appealing

Different use case and applications to Merkle tree certs, however the TCR/CR update would be applicable there.

Draft

Internet-Draft: https://datatracker.ietf.org/doc/draft-joseph-tls-batch-signatures/

Github Repo: https://github.com/PhDJsandboxaq/draft-joseph-batch-signatures