Next steps for draft-mouris-cfrg-mastic

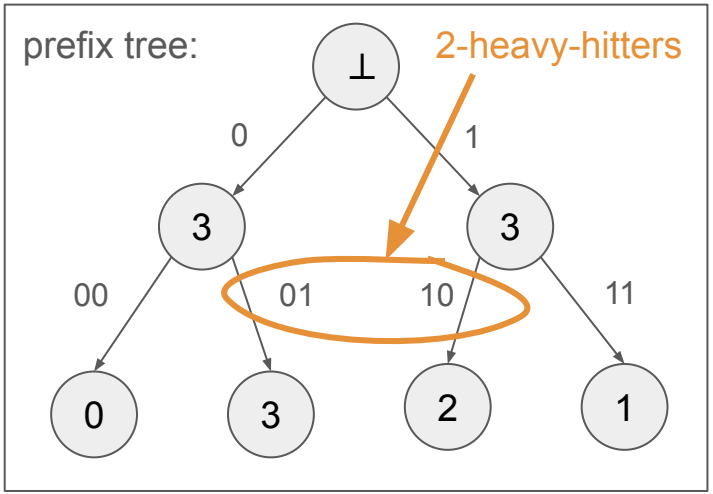
CFRG – IETF 119

Christopher Patton

Background

- PPM wants to solve the ***t*-heavy-hitters problem**: Each Client uploads a bit string and the Collector wants to learn the subset of strings uploaded at least *t* times
- Solved by Poplar1 (draft-irtf-cfrg-vdaf):
  - IDPF: Each Aggregator computes a secret share of the **prefix tree:**
    - each path is labeled by a candidate **prefix**
    - each node is labeled by the **prefix count**: the number of strings that begin with the prefix that labels the path from the root
  - For each string and level of the tree, the Aggregators compute an arithmetic sketch to check for malicious Client behavior (e.g., double counting)

strings: 00, 01, 01, 10, 11, 10, 01



Mastic, a VDAF for *t*-heavy-hitters

- Additional use cases:
  - ***w*-weighted-heavy-hitters**: each string has an associated **weight** and the Collector wants to know the strings with total weight at least *w*
  - **attribute-based metrics**: Prio3-style metrics grouped by Client properties (e.g., software version or geolocation) without reducing the anonymity set
- More efficient and simpler design than Poplar1:
  - IDPF → VIDPF: lightweight, implicit one-hotness guarantee
  - Arithmetic sketch → FLP: used in Prio3; more general; requires one round of communication instead of two
- Mitigates a known footgun for Poplar1: early termination is not safe.

Status of draft

- Since IETF 118 (draft-mouris-cfrg-mastic-01) we have:
  - Begun security analysis (ia.cr/2024/221): proofs of privacy and robustness for composition of primitives; still need concrete security bounds for the primitives themselves (mostly implicit in prior work)
  - Begun implementation (rust)
- Goal for next draft (draft-mouris-cfrg-mastic-03): Align the reference implementation with security analysis and rust code
  - **The next draft should be ready for RG adoption call.**

Questions for the RG

- **Q1**: Shall we replace Poplar1 with Mastic in draft-irtf-cfrg-vdaf?
  - Consensus so far: adopt draft-mouris-cfrg-mastic and consider removing Poplar1 from draft-irtf-cfrg-vdaf once we're more comfortable with it
    - Arguments against:
      - Editorial overhead: Mastic and Poplar1 have many features in common
      - CFRG shouldn't recommend two solutions for the same problem
      - If we remove Poplar1, then we could consider reducing generality of VDAF syntax ⇒ reduced protocol complexity for users (e.g., draft-ietf-ppm-dap)

Questions for the RG

- **Q2**: How shall we govern development and review of new VDAFs?
  - Many protocols fit into the VDAF framework:
    - Prio3 – draft-irtf-cfrg-vdaf (RG item)
    - Poplar1 – draft-irtf-cfrg-vdaf (RG item)
    - Mastic – draft-mouris-cfrg-mastic
    - PINE - draft-chen-cfrg-vdaf-pine
  - We expect other protocols to fit into this framework: they don't all necessarily need to be endorsed by CFRG
  - VDAF is a subset of MPC ("multi-party computation"): there may (soon!) be a need to develop other MPC techniques