DAP: Allowing more than one Helper (or not)

Christopher Patton and Brandon Pitman

IETF 116 – PPM
DAP-04 aggregation flow: Leader commits first

Leader

input_share_0, verifier_share_0

encrypted input_share_1

commit the share to the (partial) aggregate

Helper

verifier_share_1

verifier

commit the share to the (partial) aggregate

Prio3 preparation

input_share_0

to prep

preparation

input_share_1

verifier_share_0

verifier_share_1

prep

prep

prep_shares_to_prep

verifier
DAP-04 envisions multiple Helpers

Prio3 preparation (three Aggregators)

Leader

Helper 1

Helper 2

encrypted input_share_1

encrypted input_share_2

verifier_share_1

verifier_share_2

verifier

commit

commit

commit
Alternative flow: Helper commits first

Leader

encrypted input_share_1

verifier_share_1

commit

Leader

Helper

encrypted input_share_1, verifier_share_0

commit

Helper

encrypted input_share_1, verifier_share_0

commit

Prio3 preparation

input_share_0

input_share_1

prep

prep

verifier_share_0

verifier_share_1

prep_shares_to_prep

verifier
Alternative flow: Helper commits first

- Main upside: Fewer HTTP requests → reduced latency, impact of network issues
  - 1-round VDAFs (e.g., Prio3) take one request instead of two
  - 2-round VDAFs (e.g., Poplar1) take two requests instead of three
- Main downside: Loss of generality: No support for multiple Helpers
  - WG decision: Shall we continue to support multiple Helpers in DAP or specialize the protocol for 1-Helper?
Consideration #1: Generality

- Use case (a): More Aggregators → Weaker trust model (It should be harder to collude if more organizations are involved)
  - Not all VDAFs support multiple Helpers (e.g., Poplar1)
- Use case (b): Robustness in the presence of a misbehaving Aggregator*
  - Idea [ia.cr/2019/188, ia.cr/2023/080]: Run a 2-Aggregator VDAF with each pair of 3 Aggregators; use majority vote to decide validity
    - If Leader acts as broadcast channel (as in DAP today), then we'd still have to trust it to not misbehave
- Use case (c): VDAF that requires three (or more) Aggregators to meet its security goals
  - No known examples of this (yet)
- Use case (d): MPC schemes other than VDAFs that require 3 or more Aggregators (e.g., the sorting scheme of [IPA])

*Reminder: We currently have privacy in the presence of a malicious Aggregator, but not robustness.
Consideration #2: Complexity

- Current draft is complex, due in part to generality of supporting multi-round, multi-Aggregator VDAFs.
- Complexity impedes adoption:
  - Undefined behavior in current draft
  - Harder to implement correctly
  - Harder to reason about security
Consideration #3: State of current deployments

- Latency improvement requires reworking the aggregation flow: Perhaps too late in the game for such a large change?
  - Open-source implementations:
    - **Janus** (all roles)
    - **Daphne** (Aggregator only)
    - **Firefox** (Client only)
  - Known deployments: 3-month trial in Firefox Nightly (with ISRG and Cloudflare Research)

- (Another angle) More deployment experience with current architecture would help inform whether the latency improvement is needed in practice.
Consideration #4: Scope of DAP spec

- PPM has a much broader mandate than specifying DAP
  - Other classes of MPC, STAR, and beyond: Different drafts for each, or one monolithic draft?
- Ship a spec now that we can deploy; leave more general behavior to future drafts
  - There are likely parts of the current DAP draft that we would want to re-use in future drafts, e.g., the API, security considerations, etc.
Proposal for 1-Helper DAP

- Modify aggregation protocol to take advantage of one Helper [PR#393]
  - In current DAP, only the Leader can merge verifier shares into a verifier because only the Leader has all of the shares.
  - In One-Helper DAP, either aggregator can merge verifier shares into a verifier.
  - Effectively, the Leader no longer needs to act as a broadcast channel; protocol modification takes advantage of this. Aggregators “take turns” merging shares.
  - Total count & order of VDAF operations is not changed.
  - Total count of transmitted verifiers / verifier shares is not changed. (direction of communication changes in some cases)
  - Upshot: total count of network round-trips to complete aggregation is reduced by about half.
## Proposal for 1-Helper DAP

<table>
<thead>
<tr>
<th></th>
<th>Current DAP</th>
<th>1-Helper DAP</th>
</tr>
</thead>
</table>
| **Aggregation Initialization** | Leader: input share  
Helper: verifier share | Leader: input share, verifier share  
Helper: verifier, next verifier share |
| **Comms (non-terminal)**      |                                                 |                                                          |
| **Aggregation Continuation**  | Leader: verifier  
Helper: verifier share | Leader: verifier, next verifier share  
Helper: verifier, next verifier share |
| **Comms (non-terminal)**      |                                                 |                                                          |
| **Network Round Trips**       | ROUNDS + 1                                      | ⌈(ROUNDS + 1) / 2⌉                                      |
Example: 1-round VDAF (e.g. Prio3)

Current DAP

Leader

init{helper input share}

continue{helper verifier share 0}

continue{verifier 0}

finish{}

Helper

1-Helper DAP

Leader

init{helper input share, leader verifier share 0}

finish{verifier 0}

Helper
Example: 2-round VDAF (e.g. Poplar1)

Current DAP

Leader

- init{helper input share}
- continue{helper verifier share 0}
- continue{verifier 0}
- continue{helper verifier share 1}
- continue{verifier 1}
- finish{}

Helper

1-Helper DAP

Leader

- init{helper input share, leader verifier share 0}
- continue{verifier 0, helper verifier share 1}
- continue{verifier 1}
- finish{}

Helper
Summary

- **WG decision:** Shall we continue to support multiple Helpers in DAP (needs work) or specialize the protocol for 1-Helper?
  - Pitch: the aggregation flow will take about half as many network round-trips.

- **Considerations:**
  - #1: Generality (change rules out some use cases for DAP)
  - #2: Complexity (simpler protocol → easier adoption)
  - #3: Current deployments (big change → wait until we have more experience to decide)
  - #4: Scope of DAP draft (one draft to rule them all, or not?)