Distributed Aggregation Protocol
Draft and implementation updates

Tim Geoghegan
PPM - IETF 116 - Yokohama
Draft updates: [draft-irtf-cfrg-vdaf-04, -05]

- Security analysis of Verifiable Distributed Aggregation Functions, by Davis et al.: [ia.cr/2023/130]
- Summary of protocol implications on the CFRG list
- SHA-3 based PRNG to replace AES
draft-ietf-ppm-dap-04: catching up with VDAF-05

- Report ID is used as nonce during input sharding
  - Restricts DAP to VDAFs with 16 byte nonces

- VDAF verification key requirements
  - MUST NOT be revealed to Clients
  - Aggregators MUST commit to a key before processing reports for a task
  - Suffices for Leader to choose the key and distribute it to Helper(s)
  - Verification key cannot rotate independently of a DAP task!
  - Task negotiation remains out of scope for DAP

- VDAF aggregation parameter validation
  - VDAF requirements vary, so DAP needs a generic mechanism
  - Vdaf.is_valid(curr_agg_param, previous_agg_params) -> Bool
Collection resource representation now includes interval of time spanned by constituent reports
  ○ Particularly valuable for fixed-size tasks
  ○ May be smaller than the interval in the request for time-interval tasks

New HTTP API
  ○ Removes need for message parsing hacks
  ○ Request idempotence for robust error recovery
  ○ [Discussed at IETF 115](https://www.ietf.org/meetings/115)
  ○ [And on GitHub](https://github.com)
Implementations

- **libprio-rs**
  - Implements Prio3 and Poplar1 VDAF families and VDAF abstraction
  - VDAF-04 in `prio-0.11.x`, VDAF-05 in `prio-0.12.y`
  - We'd love to see more implementations – Go would be great
- **Daphne**
  - Helper implementation targeting Cloudflare Workers
  - DAP-04 implementation is underway
  - Docker images available!
- **Janus**
  - Client, Leader, Helper, Collector implementations
  - DAP-04 implementation is complete (Prio3 only)
- **divviup-ts**
  - Typescript Client implementation
  - DAP-04 implementation mostly complete (Prio3 only)
- **Firefox**
  - DAP-04 client implementation underway
Daphne/Firefox/Janus interoperability test

- December 2022
- DAP-02, VDAF-03
- Prio3Sum
- 1% of Firefox nightly installs (~400 clients) all uploading the value 3
Daphne/Firefox/Janus interoperability test

- It worked!
- Scale too small to learn much about performance
- Automated task provisioning is vital
  - draft-wang-ppm-taskprov is one way forward
Future goals

● Poplar1
● Simplicity
  ○ Cut non-essential features and error codes
● Clarification
  ○ Sequence and block diagrams
  ○ Consistency and concision in text
● Central/server differential privacy
  ○ Generic mechanisms in conjunction with VDAF for applying noise to aggregate shares
● Security considerations
● Decoupling the aggregation parameter from aggregation jobs (#405)
● Specialize to one aggregator?
## Backup: New HTTP API

<table>
<thead>
<tr>
<th>Resource</th>
<th>Supported by…</th>
<th>Required methods</th>
<th>Relative path</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPKE configuration</td>
<td>Leader, helper</td>
<td>GET</td>
<td>/hpke_config[?task_id={task-id}]</td>
</tr>
<tr>
<td>Report</td>
<td>Leader</td>
<td>PUT</td>
<td>/tasks/{task-id}/reports</td>
</tr>
<tr>
<td>Aggregation job</td>
<td>Helper</td>
<td>PUT, POST, DELETE</td>
<td>/tasks/{task-id}/aggregation_jobs/{aggregation-job-id}</td>
</tr>
<tr>
<td>Aggregate shares</td>
<td>Helper</td>
<td>POST, DELETE</td>
<td>/tasks/{task-id}/aggregate_shares</td>
</tr>
<tr>
<td>Collections</td>
<td>Leader</td>
<td>PUT, POST, DELETE</td>
<td>/tasks/{task-id}/collections/{collection-job-id}</td>
</tr>
</tbody>
</table>