# DAP Task Enforcement & Configuration

PPM - IETF 114 - Philadelphia

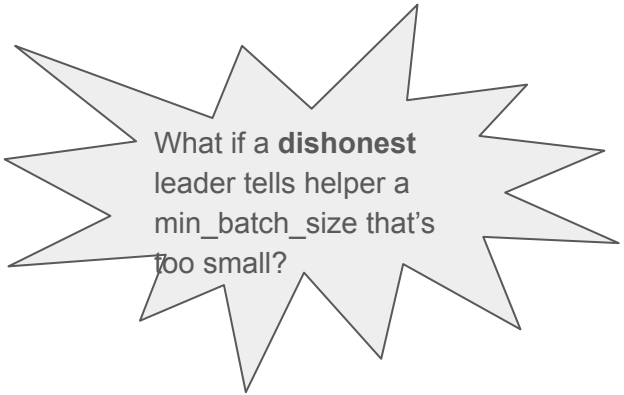
# Task Parameters

- `aggregator_endpoints`: A list of URLs relative to which an aggregator's API endpoints.
  `max_batch_lifetime`: The maximum number of times a batch of reports may be used in collect requests.
- `min_batch_size`: The minimum number of reports that appear in a batch.
- `min_batch_duration`: The minimum time difference between the oldest and newest report in a batch.
- A unique identifier for the VDAF instance used for the task, including the type of measurement associated with the task.
- `collector_config`: Collector hpke-config.
- `vdaf_verify_key`: The VDAF verification key shared by the aggregators.

Task Configuration: the process of creating a task with parameters prior to receiving report. Is handled **out of band**, no detailed spec in DAP yet.

# Potential (Privacy) Problems

How to make sure privacy essential parameters, e.g. min_batch_size, are configured faithfully?




More privacy parameters will be needed, e.g. Differential Privacy. These can be VDAF specific.

How does DAP earn client's trust if the parameters are configured out of band, and hence out of sight?

## Transparency

User should know what privacy guarantee they get when participating in a DAP task.

## Enforcement

Aggregators should enforce the privacy parameters they have are the same as the ones user have seen.

# In Band Task Enforcement (Issue #271)

Let client know what task parameters are being used.

- Better Auditing
- Can even hardcode in client

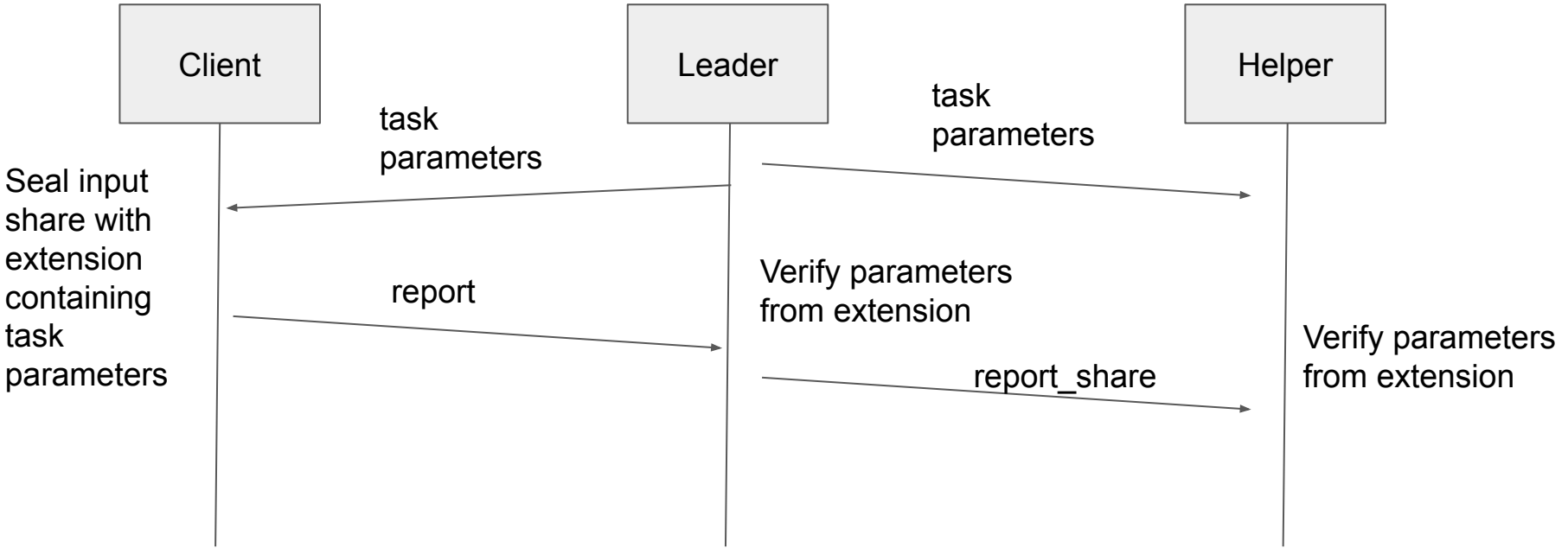
Client sends task parameters back to server in band, in Report.Extension.

- Used in AAD, malicious aggregator cannot change them
- Extensible for VDAF

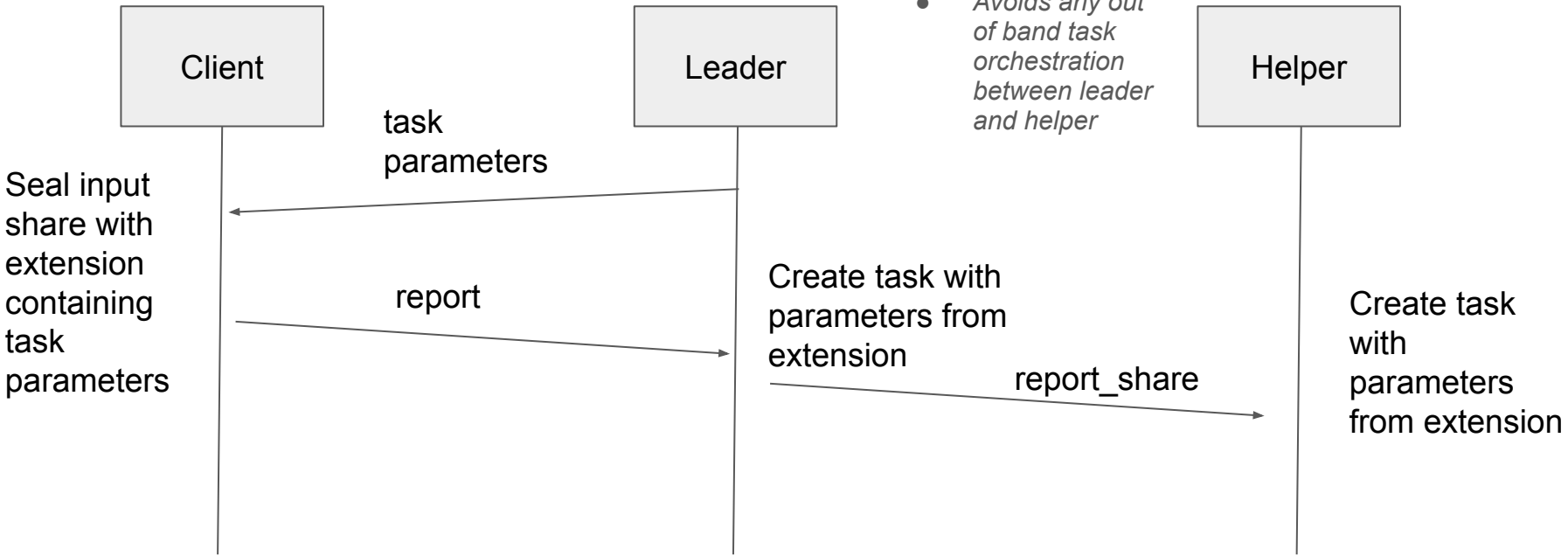
Aggregators check the task parameters in Extension match what they stored for that task.

- Malicious client cannot pollute aggregation

# Example: task enforcement



# One step further: In band task configuration (Issue #290)




- *Avoids any out of band task orchestration between leader and helper*

- *Malicious client cannot pollute a good batch: reports with different parameters will be aggregated in different batches*

# Task Identity

- Currently task is identified by a UUID

- Task enforcement means task is identified by a tuple of (task_id, task_parameters)

- task_id can be redefined to represent the tuple. e.g.

*task_id = hash(“shared info” || extension)*