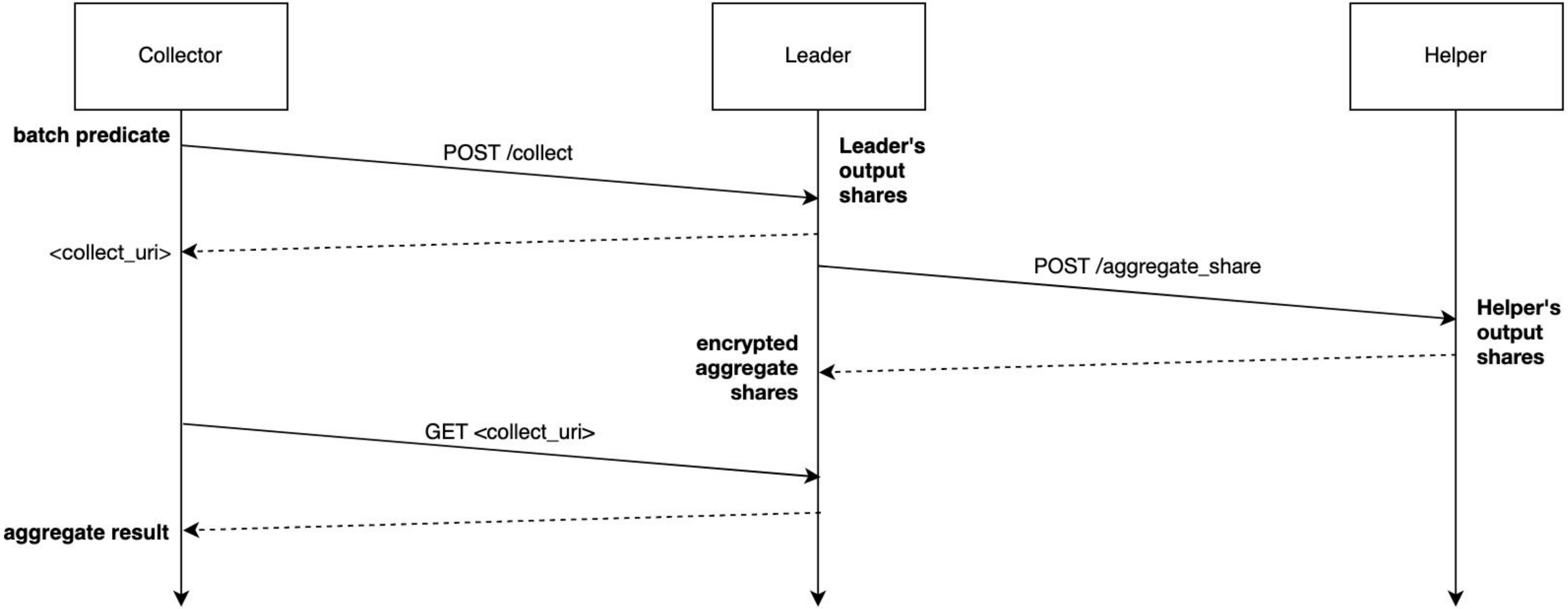
Collect Privacy

PPM - IETF 114 - Philadelphia

Collect Sub-Protocol

Collect sub-protocol: Collector chooses a "**batch predicate"**. Leader and Helper aggregate the output shares of all reports that satisfy the predicate.



Query Validation

Existing constraints oriented around “time-series” batch predicates

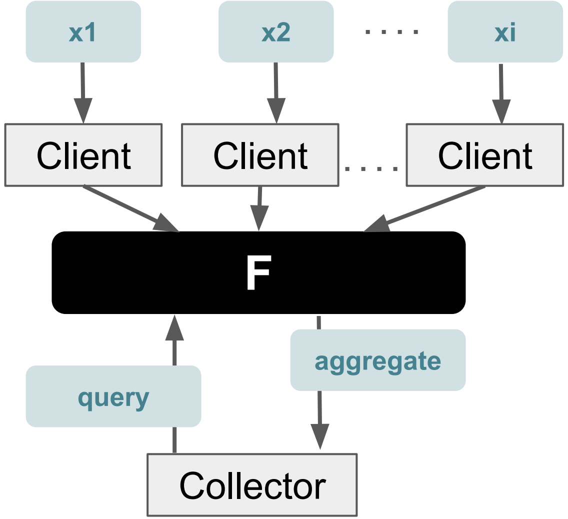
1. Number of reports MUST be at least **min_batch_size**
2. A report in the batch has not been collected more than **max_batch_lifetime** times
3. Batch predicate interval MUST align with batch window boundaries
4. Batch predicate interval MUST NOT overlap with any prior batch predicate

… this is not flexible (no group-based or chunk-based collection)

… and we’ve not adequately motivated this rigidity

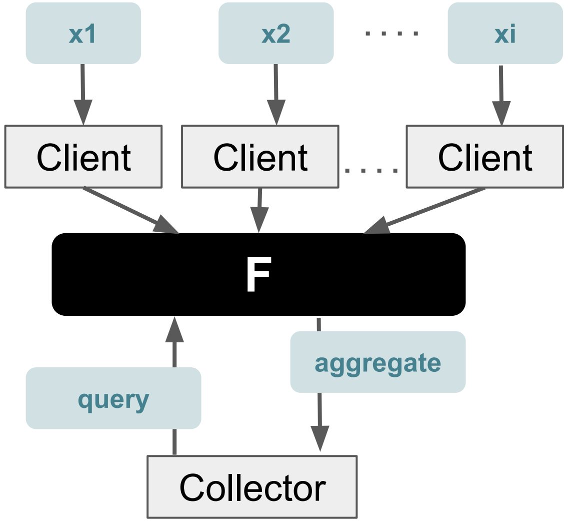
Privacy in DAP

DAP is an MPC protocol for computing some aggregate function **F(query, x1, …, xi)** over client inputs **(x1,..., xi)**



Privacy in DAP

DAP is an MPC protocol for computing some aggregate function **F(query, x1, …, xi)** over client inputs **(x1,..., xi)**



Privacy means that the protocol leaks nothing about honest client inputs beyond what is revealed through **F** when evaluated over honest client inputs

Privacy[1] Threat Model

There is a limited number of malicious clients

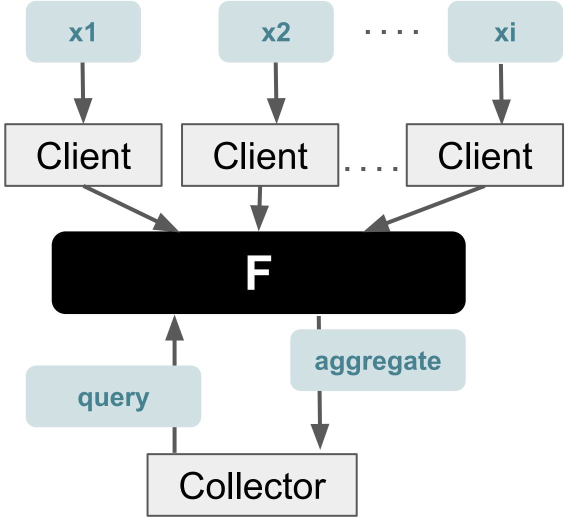
Some clients provide honest inputs, others don't

At least one of the aggregators is honest

Each aggregator assumes the others are malicious

Collector is malicious

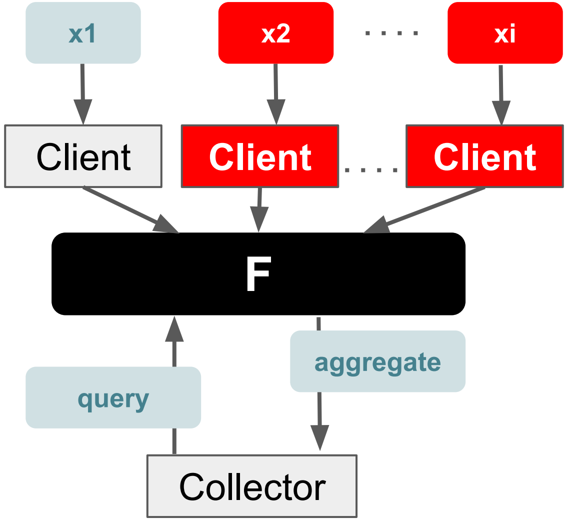
Collect queries are adaptively constructed to learn information about individual client inputs



[1] The robustness threat model is different

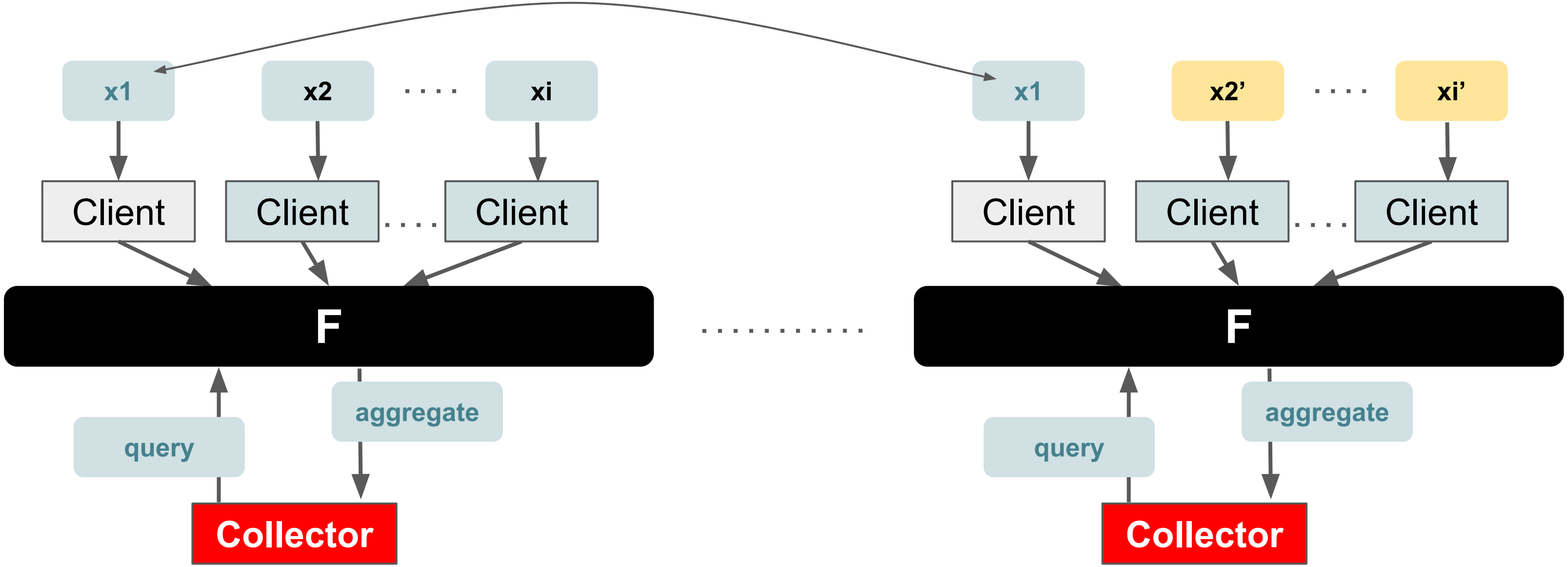
Relevant Attacks

Stuffing attack: Attacker (**Leader, Helper, or compromised Client(s)**) injects malicious reports into the system to skew the resulting aggregate and learn honest inputs



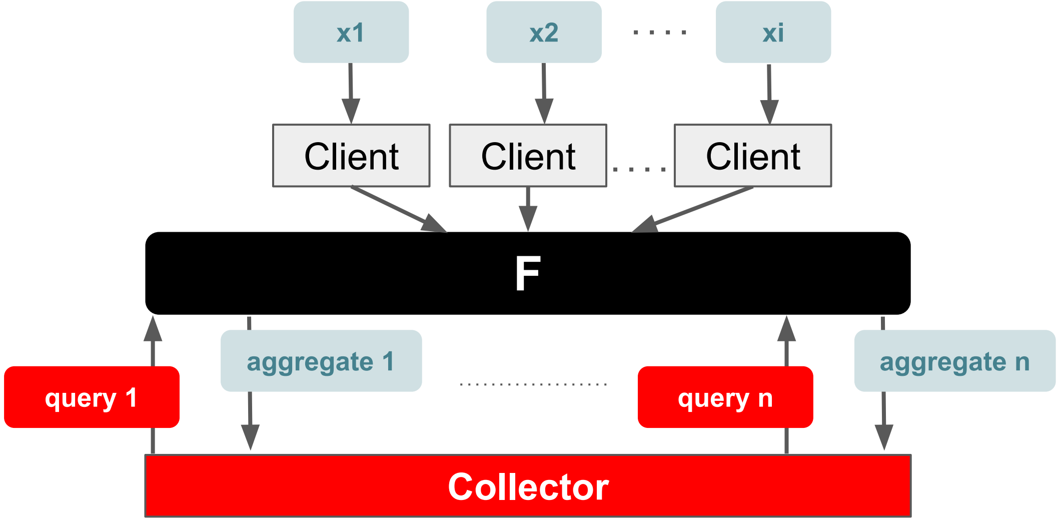
Relevant Attacks

Over sample: Attacker (**Collector**) learns output aggregate based on repeated honest client contributions (due to misconfigured clients / "misuse" of DAP)



Relevant Attacks

Intersection attack: Attacker (**Collector**) learns aggregate result for too many distinct batches and combines them to reconstruct honest inputs



Protocol Mitigations

How can we mitigate these issues?

Stuffing attack

Deployment-specific problem; dealt with in a variety of ways including, e.g., client upload authentication or differential privacy

Over sample

Client implementation-specific problem; dealt with in a variety of ways including differential privacy

Intersection attack

Limit what are valid collect requests while balancing query flexibility

First Class Intersection Mitigation (Collect Constraints)

Informally: Given any sequence of queries, it must not be possible for the Collector to compute an aggregate result based on some subset of reports of size less than **min_batch_size**

Enforcement intuition: Given a query and sequence of preceding queries, each Aggregator ensures that the size of all possible batch subsets that can be computed based on the combination of batches is at least **min_batch_size**

Open question: How do we express queries such that this privacy goal is met while being maximally useful?

Questions

1. Is the threat model clear?
2. Are there attacks on privacy we have not considered that the protocol should address?
3. Do folks agree with the selection of attacks (**intersection attack**) that can be mitigated in DAP?

Collect Privacy

PPM - IETF 114 - Philadelphia