Standardizing MPC for Privacy Preserving Measurement

RWC 2022

Tim Geoghegan
Eric Rescorla

Christopher Patton
Christopher Wood

ISRG
moz://a

CLOUDFLARE Research

Takeaways from the previous talk

- Operating a service sometimes requires measuring the system's users ⇒ In many cases, only aggregates are needed

Takeaways from the previous talk

- Operating a service sometimes requires measuring the system's users ⇒ In many cases, only aggregates are needed
- Measurements used in aggregates are often privacy-sensitive (even if indirectly) ⇒ Improve privacy by distributing the computation among multiple servers (ENPA/Prio)

Takeaways from the previous talk

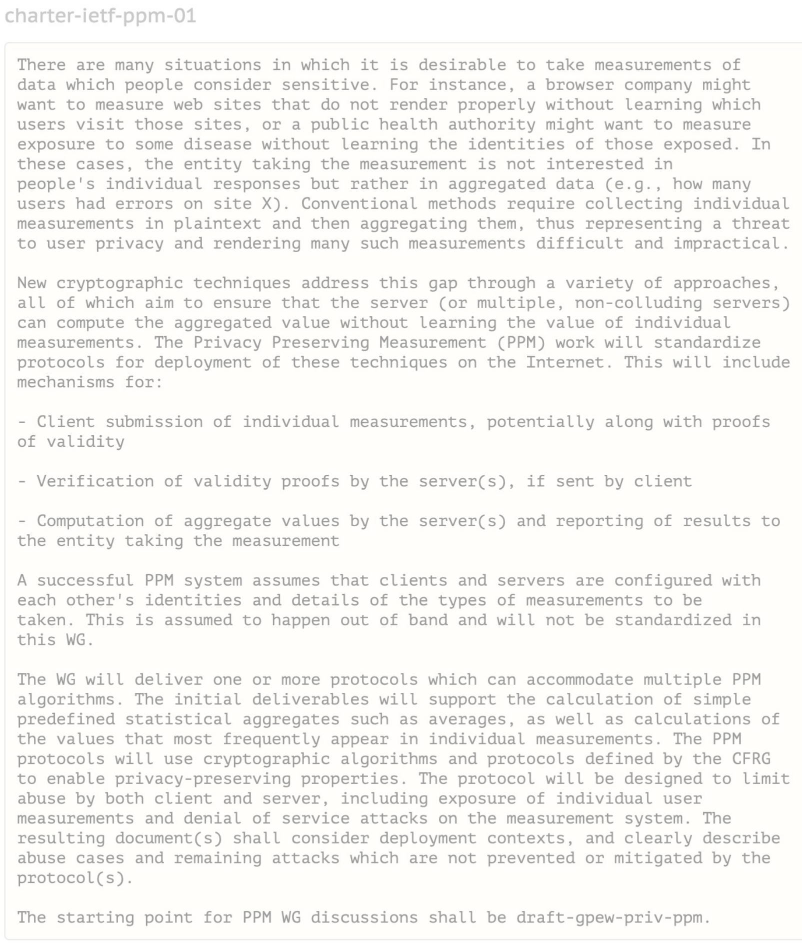
- Operating a service sometimes requires measuring the system's users ⇒ In many cases, only aggregates are needed
- Measurements used in aggregates are often privacy-sensitive (even if indirectly) ⇒ Improve privacy by distributing the computation among multiple servers (ENPA/Prio)
- ENPA designed to solve a specific use case ⇒ What to do about aggregates for which ENPA/Prio is not well-suited?

Takeaways from the previous talk

- Operating a service sometimes requires measuring the system's users ⇒ In many cases, only aggregates are needed
- Measurements used in aggregates are often privacy-sensitive (even if indirectly) ⇒ Improve privacy by distributing the computation among multiple servers (ENPA/Prio)
- ENPA designed to solve a specific use case ⇒ What to do about aggregates for which ENPA/Prio is not well-suited?
- Cryptography is advancing (lots of MPC protocols in the literature for various aggregation functions) ⇒ Lacks a clear roadmap for deployment

What's next?

- The IETF has formed a working group for standardizing *privacy-preserving measurement*.



charter-ietf-ppm-01

There are many situations in which it is desirable to take measurements of data which people consider sensitive. For instance, a browser company might want to measure web sites that do not render properly without learning which users visit those sites, or a public health authority might want to measure exposure to some disease without learning the identities of those exposed. In these cases, the entity taking the measurement is not interested in people's individual responses but rather in aggregated data (e.g., how many users had errors on site X). Conventional methods require collecting individual measurements in plaintext and then aggregating them, thus representing a threat to user privacy and rendering many such measurements difficult and impractical.

New cryptographic techniques address this gap through a variety of approaches, all of which aim to ensure that the server (or multiple, non-colluding servers) can compute the aggregated value without learning the value of individual measurements. The Privacy Preserving Measurement (PPM) work will standardize protocols for deployment of these techniques on the Internet. This will include mechanisms for:

- Client submission of individual measurements, potentially along with proofs of validity

- Verification of validity proofs by the server(s), if sent by client

- Computation of aggregate values by the server(s) and reporting of results to the entity taking the measurement

A successful PPM system assumes that clients and servers are configured with each other's identities and details of the types of measurements to be taken. This is assumed to happen out of band and will not be standardized in this WG.

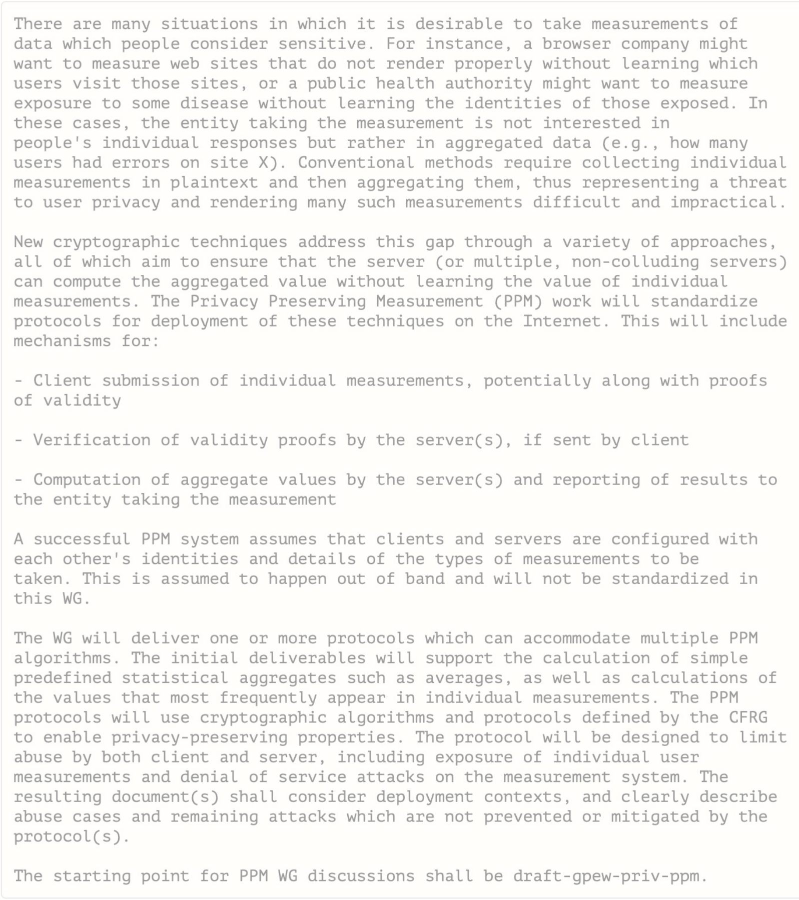
The WG will deliver one or more protocols which can accommodate multiple PPM algorithms. The initial deliverables will support the calculation of simple predefined statistical aggregates such as averages, as well as calculations of the values that most frequently appear in individual measurements. The PPM protocols will use cryptographic algorithms and protocols defined by the CFRG to enable privacy-preserving properties. The protocol will be designed to limit abuse by both client and server, including exposure of individual user measurements and denial of service attacks on the measurement system. The resulting document(s) shall consider deployment contexts, and clearly describe abuse cases and remaining attacks which are not prevented or mitigated by the protocol(s).

The starting point for PPM WG discussions shall be draft-gpew-priv-ppm.

What's next?

- The IETF has formed a working group for standardizing *privacy-preserving measurement*.
    - In-scope: Any specific *aggregation function* (e.g., count, mean, variance, quantiles, heavy hitters, model training, …)

charter-ietf-ppm-01



There are many situations in which it is desirable to take measurements of data which people consider sensitive. For instance, a browser company might want to measure web sites that do not render properly without learning which users visit those sites, or a public health authority might want to measure exposure to some disease without learning the identities of those exposed. In these cases, the entity taking the measurement is not interested in people's individual responses but rather in aggregated data (e.g., how many users had errors on site X). Conventional methods require collecting individual measurements in plaintext and then aggregating them, thus representing a threat to user privacy and rendering many such measurements difficult and impractical.

New cryptographic techniques address this gap through a variety of approaches, all of which aim to ensure that the server (or multiple, non-colluding servers) can compute the aggregated value without learning the value of individual measurements. The Privacy Preserving Measurement (PPM) work will standardize protocols for deployment of these techniques on the Internet. This will include mechanisms for:

- Client submission of individual measurements, potentially along with proofs of validity

- Verification of validity proofs by the server(s), if sent by client

- Computation of aggregate values by the server(s) and reporting of results to the entity taking the measurement

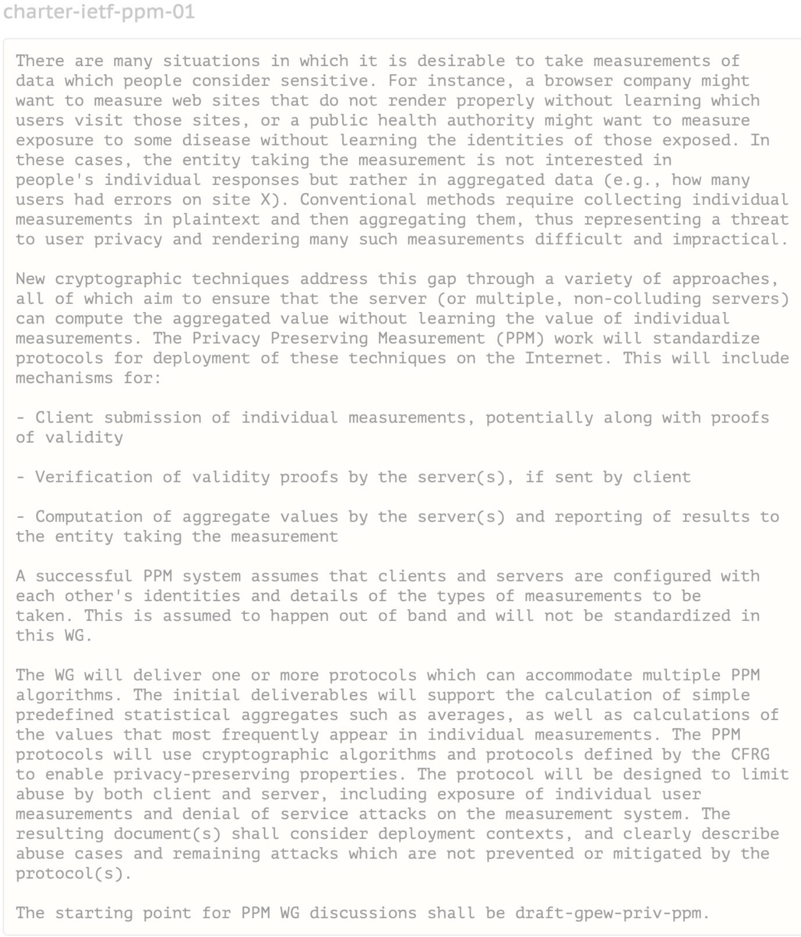
A successful PPM system assumes that clients and servers are configured with each other's identities and details of the types of measurements to be taken. This is assumed to happen out of band and will not be standardized in this WG.

The WG will deliver one or more protocols which can accommodate multiple PPM algorithms. The initial deliverables will support the calculation of simple predefined statistical aggregates such as averages, as well as calculations of the values that most frequently appear in individual measurements. The PPM protocols will use cryptographic algorithms and protocols defined by the CFRG to enable privacy-preserving properties. The protocol will be designed to limit abuse by both client and server, including exposure of individual user measurements and denial of service attacks on the measurement system. The resulting document(s) shall consider deployment contexts, and clearly describe abuse cases and remaining attacks which are not prevented or mitigated by the protocol(s).

The starting point for PPM WG discussions shall be draft-gpew-priv-ppm.

What's next?

- The IETF has formed a working group for standardizing *privacy-preserving measurement*.
  - In-scope: Any specific *aggregation function* (e.g., count, mean, variance, quantiles, heavy hitters, model training, …)
  - Out-of-scope: Data anonymization *without aggregation*



charter-ietf-ppm-01

There are many situations in which it is desirable to take measurements of data which people consider sensitive. For instance, a browser company might want to measure web sites that do not render properly without learning which users visit those sites, or a public health authority might want to measure exposure to some disease without learning the identities of those exposed. In these cases, the entity taking the measurement is not interested in people's individual responses but rather in aggregated data (e.g., how many users had errors on site X). Conventional methods require collecting individual measurements in plaintext and then aggregating them, thus representing a threat to user privacy and rendering many such measurements difficult and impractical.

New cryptographic techniques address this gap through a variety of approaches, all of which aim to ensure that the server (or multiple, non-colluding servers) can compute the aggregated value without learning the value of individual measurements. The Privacy Preserving Measurement (PPM) work will standardize protocols for deployment of these techniques on the Internet. This will include mechanisms for:

- Client submission of individual measurements, potentially along with proofs of validity

- Verification of validity proofs by the server(s), if sent by client

- Computation of aggregate values by the server(s) and reporting of results to the entity taking the measurement

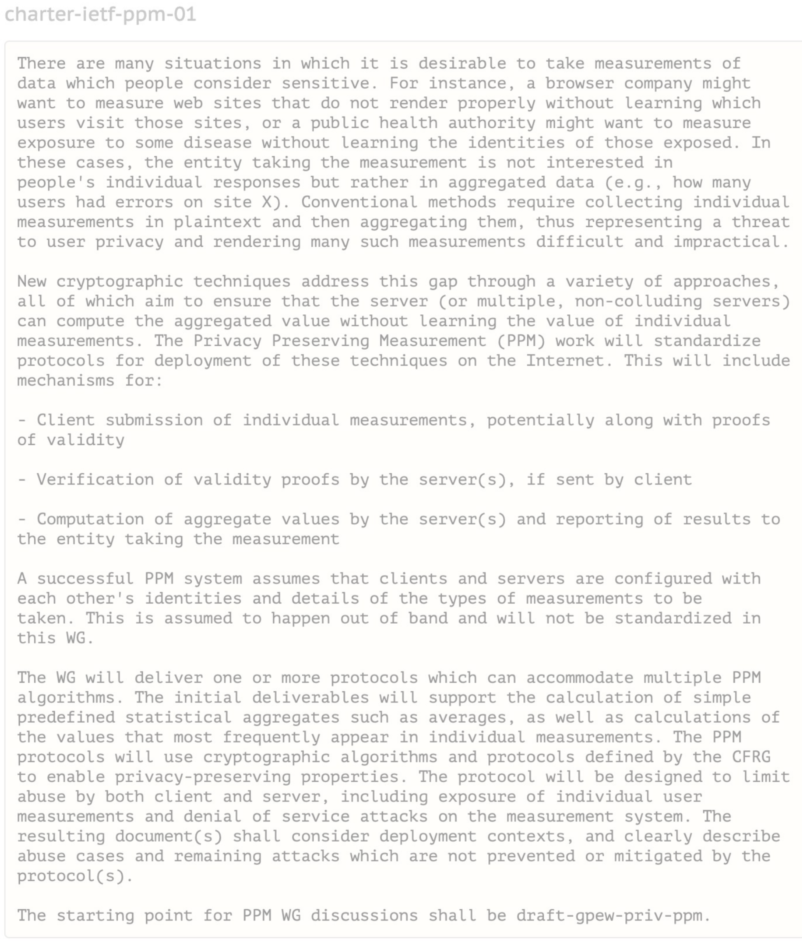
A successful PPM system assumes that clients and servers are configured with each other's identities and details of the types of measurements to be taken. This is assumed to happen out of band and will not be standardized in this WG.

The WG will deliver one or more protocols which can accommodate multiple PPM algorithms. The initial deliverables will support the calculation of simple predefined statistical aggregates such as averages, as well as calculations of the values that most frequently appear in individual measurements. The PPM protocols will use cryptographic algorithms and protocols defined by the CFRG to enable privacy-preserving properties. The protocol will be designed to limit abuse by both client and server, including exposure of individual user measurements and denial of service attacks on the measurement system. The resulting document(s) shall consider deployment contexts, and clearly describe abuse cases and remaining attacks which are not prevented or mitigated by the protocol(s).

The starting point for PPM WG discussions shall be draft-gpew-priv-ppm.

What's next?

- The IETF has formed a working group for standardizing *privacy-preserving measurement*.
    - In-scope: Any specific *aggregation function* (e.g., count, mean, variance, quantiles, heavy hitters, model training, …)
    - Out-of-scope: Data anonymization *without aggregation*
    - Out-of-scope: General-purpose MPC



charter-ietf-ppm-01

```
There are many situations in which it is desirable to take measurements of
data which people consider sensitive. For instance, a browser company might
want to measure web sites that do not render properly without learning which
users visit those sites, or a public health authority might want to measure
exposure to some disease without learning the identities of those exposed. In
these cases, the entity taking the measurement is not interested in
people's individual responses but rather in aggregated data (e.g., how many
users had errors on site X). Conventional methods require collecting individual
measurements in plaintext and then aggregating them, thus representing a threat
to user privacy and rendering many such measurements difficult and impractical.

New cryptographic techniques address this gap through a variety of approaches,
all of which aim to ensure that the server (or multiple, non-colluding servers)
can compute the aggregated value without learning the value of individual
measurements. The Privacy Preserving Measurement (PPM) work will standardize
protocols for deployment of these techniques on the Internet. This will include
mechanisms for:

- Client submission of individual measurements, potentially along with proofs
of validity

- Verification of validity proofs by the server(s), if sent by client

- Computation of aggregate values by the server(s) and reporting of results to
the entity taking the measurement

A successful PPM system assumes that clients and servers are configured with
each other's identities and details of the types of measurements to be
taken. This is assumed to happen out of band and will not be standardized in
this WG.

The WG will deliver one or more protocols which can accommodate multiple PPM
algorithms. The initial deliverables will support the calculation of simple
predefined statistical aggregates such as averages, as well as calculations of
the values that most frequently appear in individual measurements. The PPM
protocols will use cryptographic algorithms and protocols defined by the CFRG
to enable privacy-preserving properties. The protocol will be designed to limit
abuse by both client and server, including exposure of individual user
measurements and denial of service attacks on the measurement system. The
resulting document(s) shall consider deployment contexts, and clearly describe
abuse cases and remaining attacks which are not prevented or mitigated by the
protocol(s).

The starting point for PPM WG discussions shall be draft-gpew-priv-ppm.
```

Overview

Part I

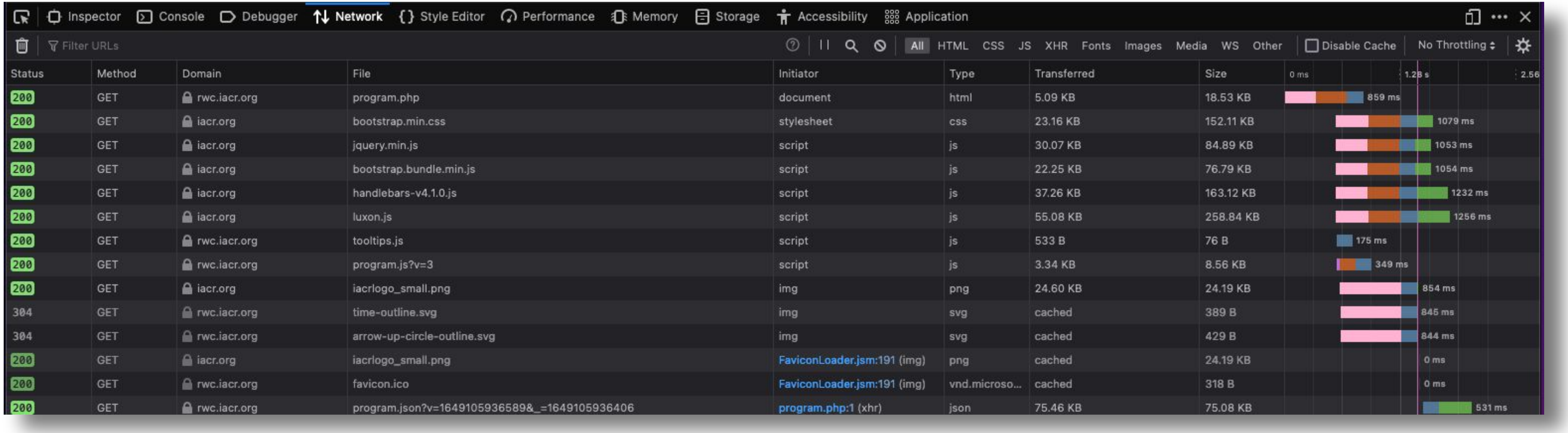
- Prio [CGB17]
- Poplar [BBCG+21]
- Other candidates

Part II

- Verifiable Distributed Aggregation Functions (VDAFs)
- The Privacy-Preserving Measurement (PPM) protocol

Prio [CGB17]

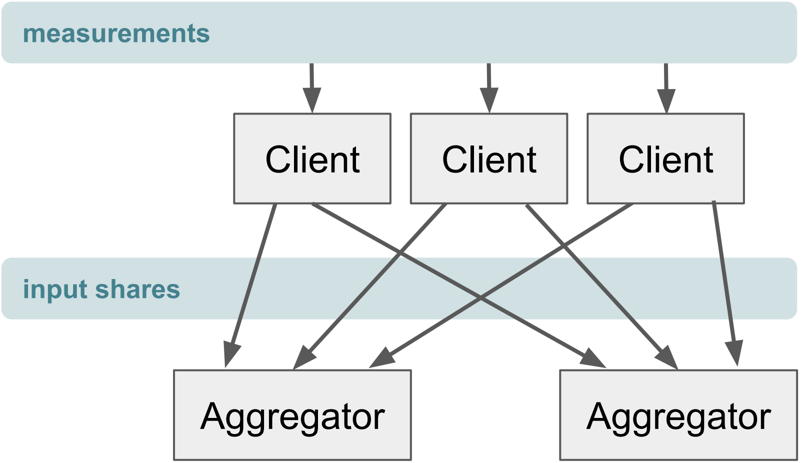
E.g.: *Are users of my website experiencing high-latency?*



Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms 1.28 s 2.56
200	GET	rwc.iacr.org	program.php	document	html	5.09 KB	18.53 KB	859 ms
200	GET	iacr.org	bootstrap.min.css	stylesheet	css	23.16 KB	152.11 KB	1079 ms
200	GET	iacr.org	jquery.min.js	script	js	30.07 KB	84.89 KB	1053 ms
200	GET	iacr.org	bootstrap.bundle.min.js	script	js	22.25 KB	76.79 KB	1054 ms
200	GET	iacr.org	handlebars-v4.1.0.js	script	js	37.26 KB	163.12 KB	1232 ms
200	GET	iacr.org	luxon.js	script	js	55.08 KB	258.84 KB	1256 ms
200	GET	rwc.iacr.org	tooltips.js	script	js	533 B	76 B	175 ms
200	GET	rwc.iacr.org	program.js?v=3	script	js	3.34 KB	8.56 KB	349 ms
200	GET	iacr.org	iacrlogo_small.png	img	png	24.60 KB	24.19 KB	854 ms
304	GET	rwc.iacr.org	time-outline.svg	img	svg	cached	389 B	845 ms
304	GET	rwc.iacr.org	arrow-up-circle-outline.svg	img	svg	cached	429 B	844 ms
200	GET	iacr.org	iacrlogo_small.png	FaviconLoader.jsm:191 (img)	png	cached	24.19 KB	0 ms
200	GET	rwc.iacr.org	favicon.ico	FaviconLoader.jsm:191 (img)	vnd.microso...	cached	318 B	0 ms
200	GET	rwc.iacr.org	program.json?v=1649105936589&_=1649105936406	program.php:1 (xhr)	json	75.46 KB	75.08 KB	531 ms

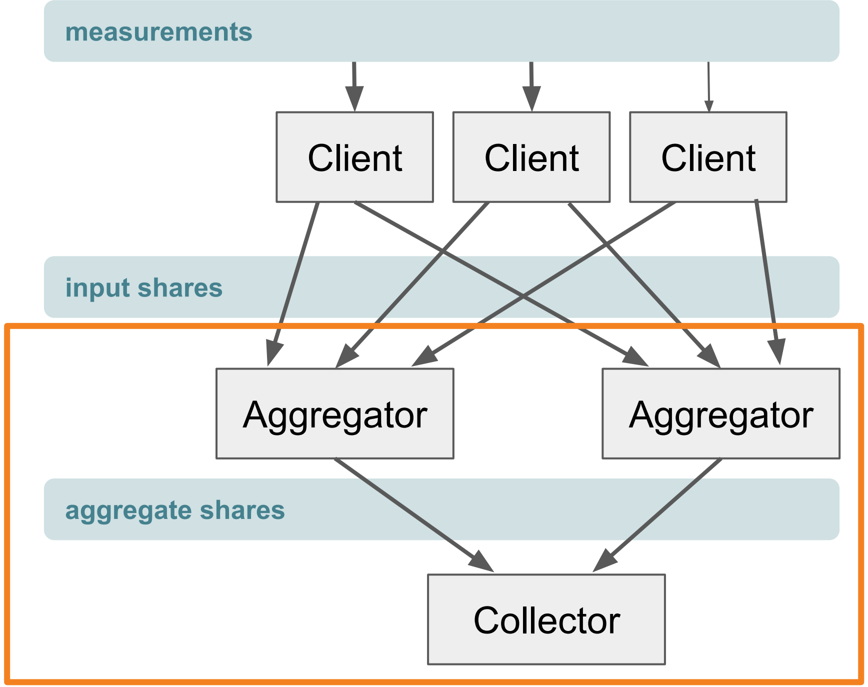
Prio [CGB17]

E.g.: *Are users of my website experiencing high-latency?*



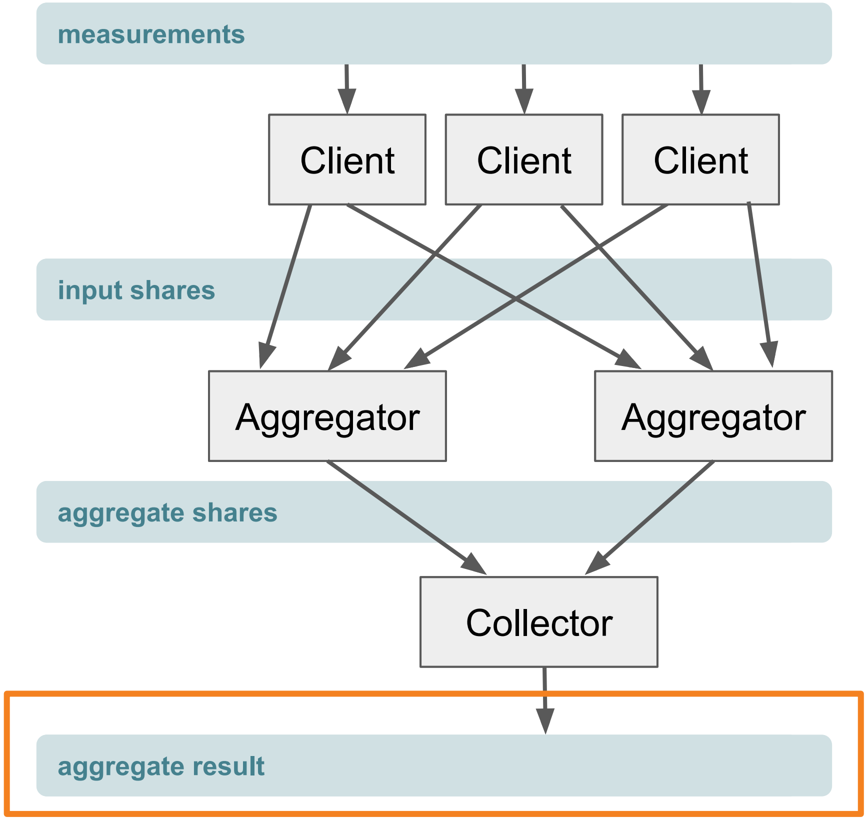
Prio [CGB17]

E.g.: *Are users of my website experiencing high-latency?*



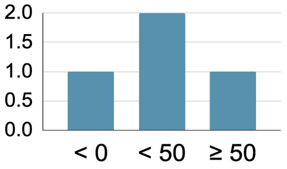
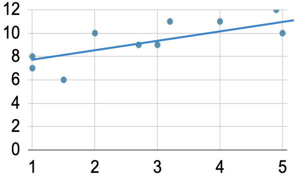
Prio [CGB17]

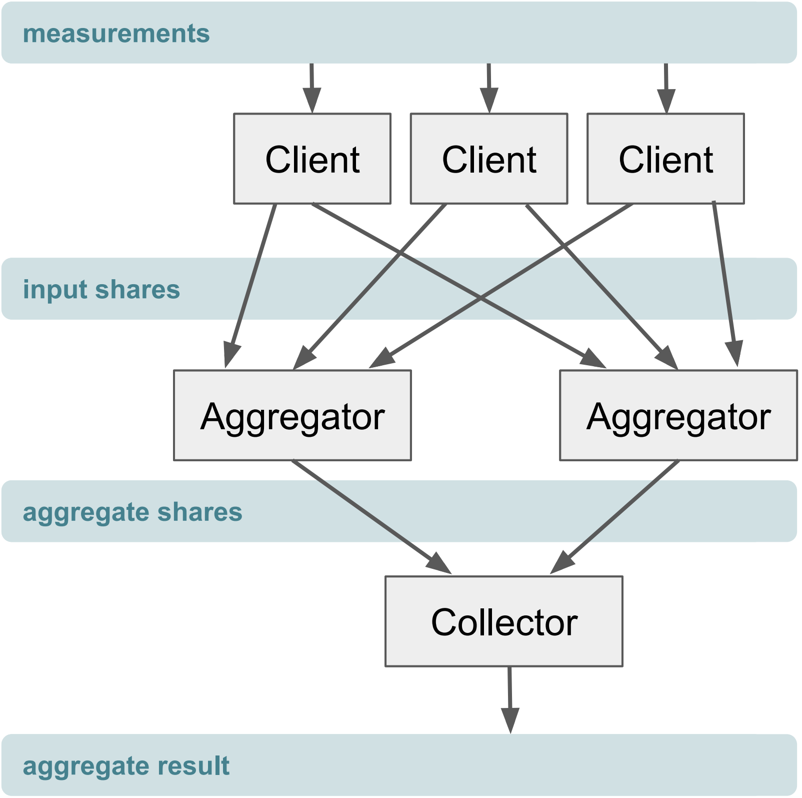
E.g.: *Are users of my website experiencing high-latency?*



Prio [CGB17]

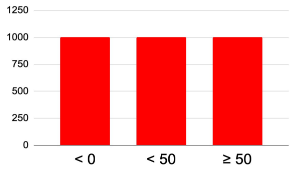
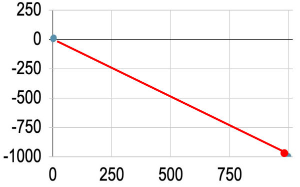
E.g.: *Are users of my website experiencing high-latency?*

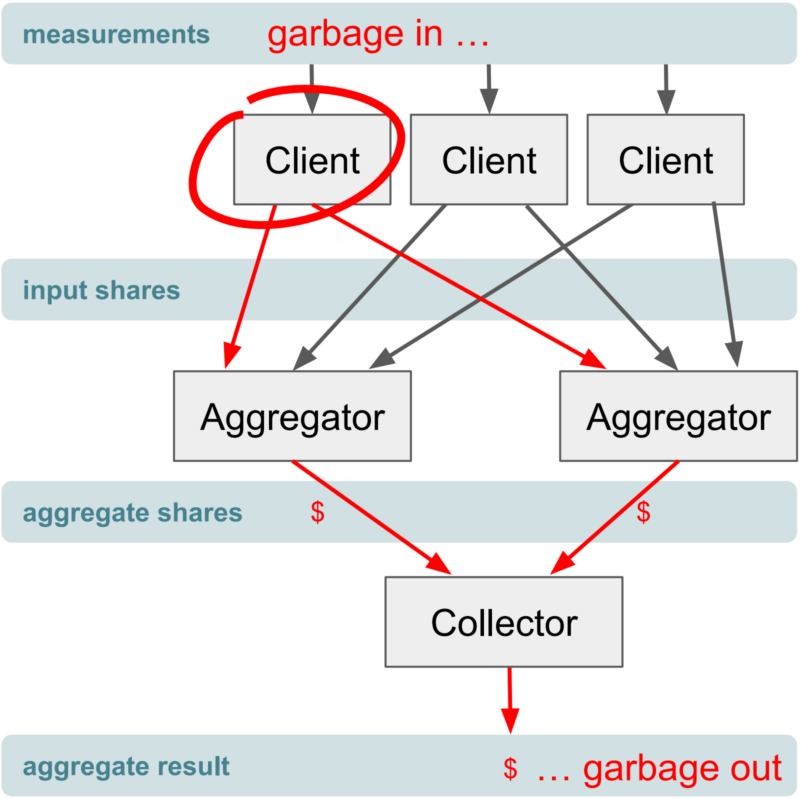
type	**measurements**	**aggregate result**
Count (ENPA) *(high latency?)*	1, 1, 0, 1, 0, 1	5
Mean, standard deviation *(of load time)*	182, 160, 190, 170, 175	175, 11
Histogram *(estimating distribution of load time)*	-7 ⇒ [1, 0, 0] 23 ⇒ [0, 1, 0] 45 ⇒ [0, 1, 0] 59 ⇒ [0, 0, 1]	
Linear regression *(load time as a function of no. of hops from client to server)*	(1, 7), (2, 10), (3, 9), (4, 11), …, (5, 10)	



Prio [CGB17]

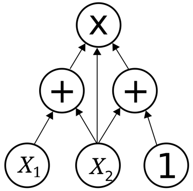
E.g.: *Are users of my website experiencing high-latency?*

type	measurements	aggregate result
Count (ENPA) *(high latency?)*	1, 1, 0, 1, 0, 999	1002
Mean, standard deviation *(of load time)*	182, 160, 190, 170, 999	340, 368
Histogram *(estimating distribution of load time)*	-7 ⇒ [1, 0, 0] 23 ⇒ [0, 1, 0] 45 ⇒ [0, 1, 0] [999, 999, 999]	
Linear regression *(load time as a function of no. of hops from client to server)*	(1, 7), (2, 10), (3, 9), (4, 11), …, (999, -999)	

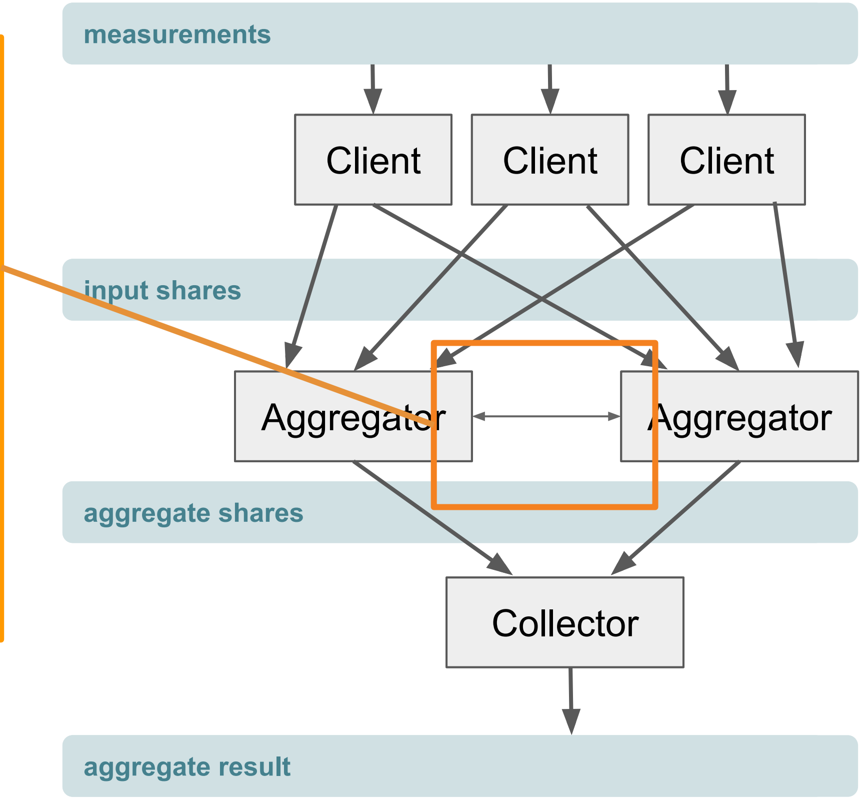


Prio [CGB17]

- Each measurement type specifies an **arithmetic circuit** C that recognizes valid inputs
- Each client generates a *fully linear proof (FLP)* [BBCG+19] of its input's validity
  - Proof shares allow Aggregators to jointly evaluate C on the secret shared input



source: Wikipedia



Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Poplar [BBCG+21]

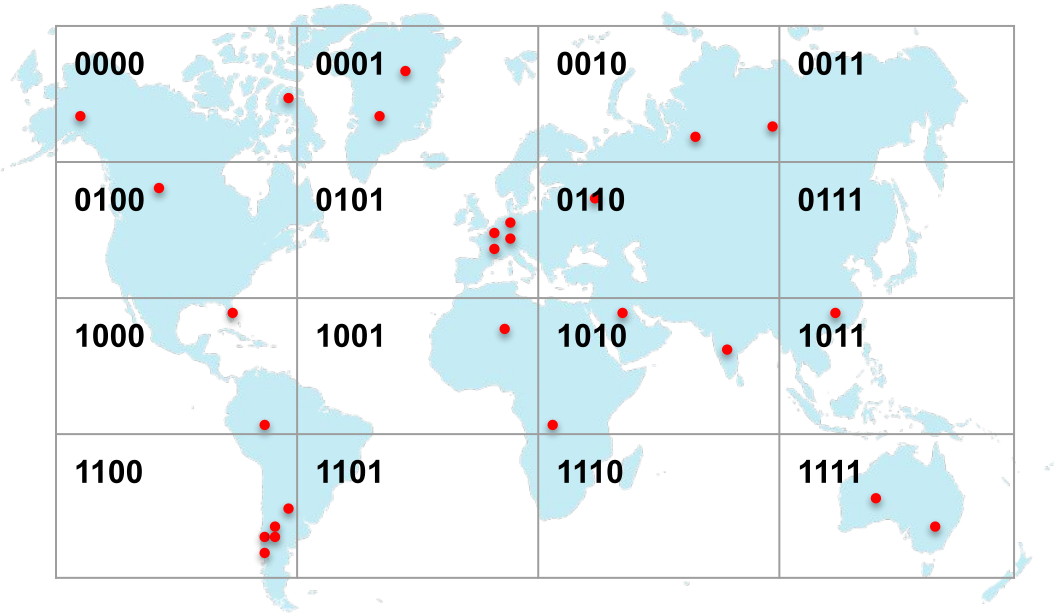
- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]

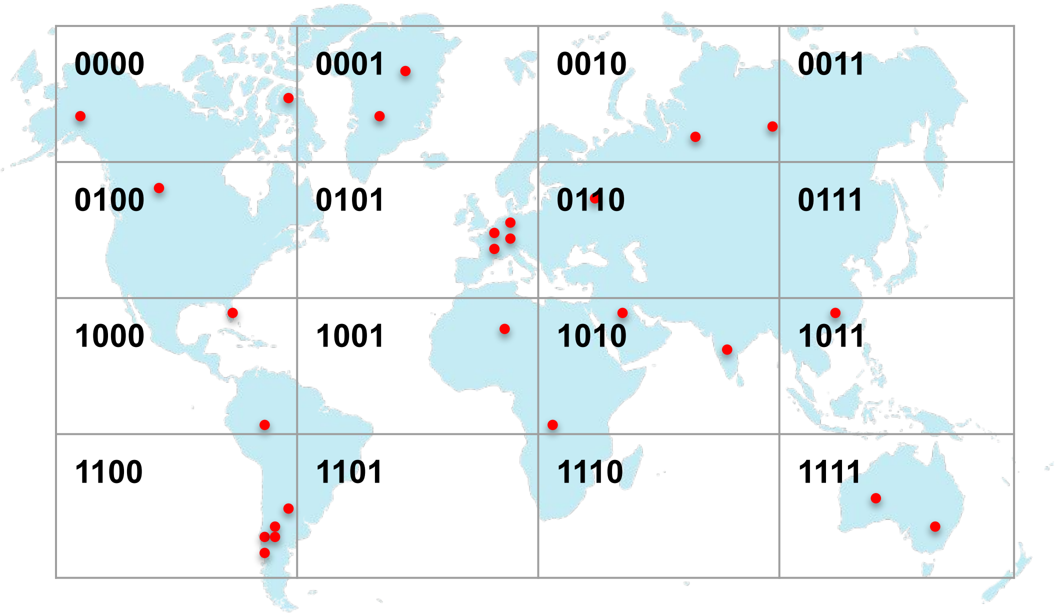
E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*



Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

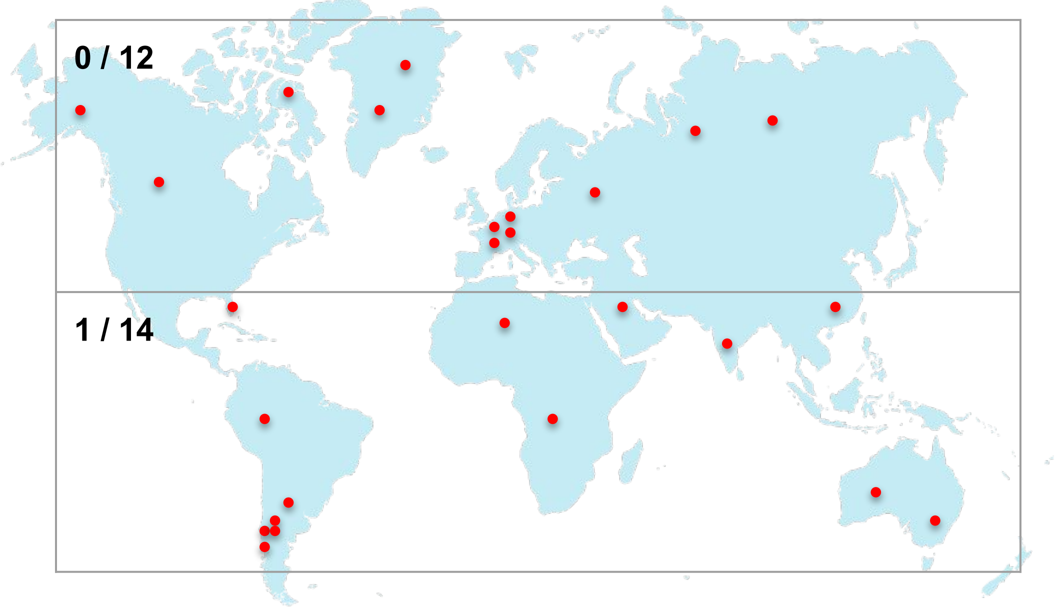


Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **0**, **1** / threshold: **4**

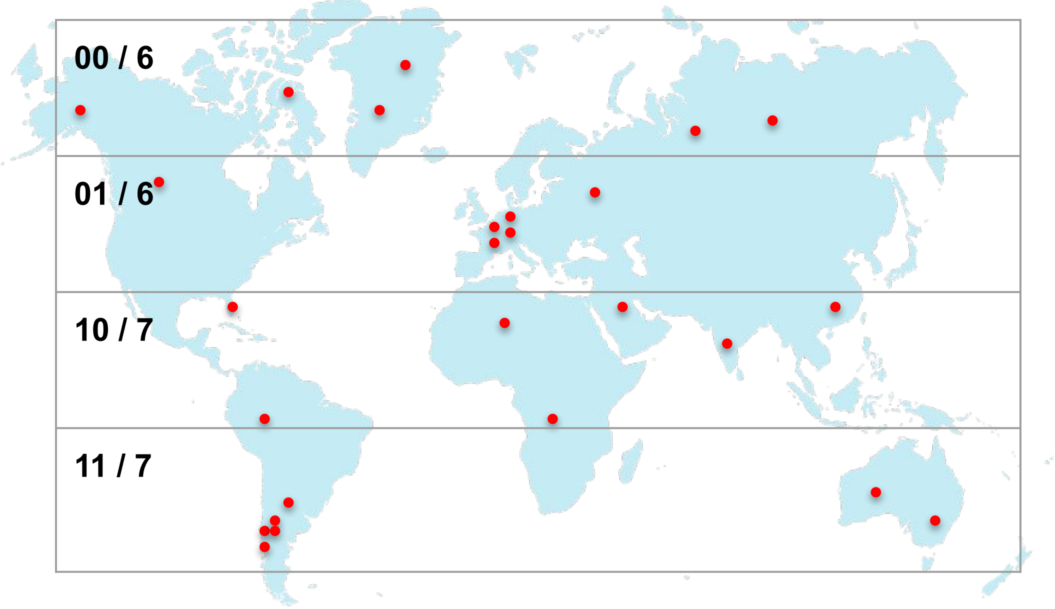


Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **00**, **01**, **10**, **11** / threshold: **4**

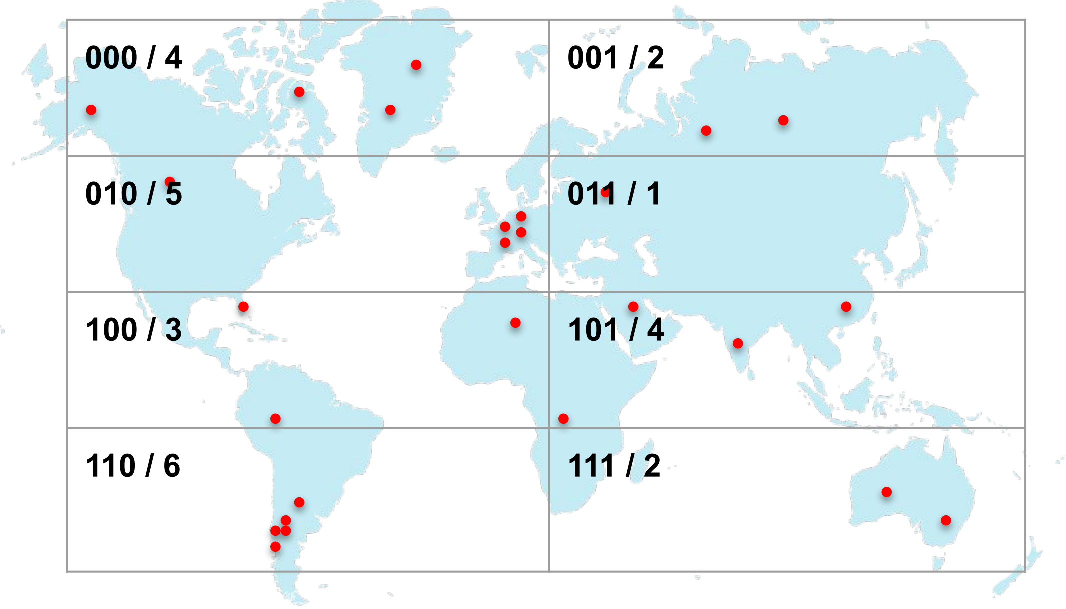


Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **000**, **001**, **010**, **011**, **100**, **101**, **110**, **111** / threshold: **4**



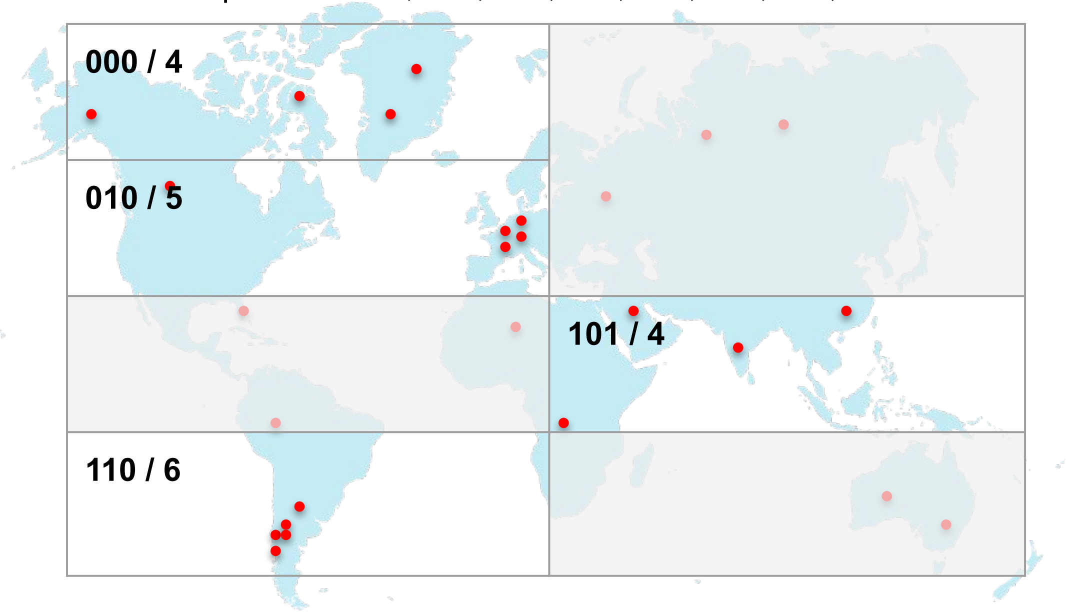
000 / 4	**001 / 2**
010 / 5	**011 / 1**
100 / 3	**101 / 4**
110 / 6	**111 / 2**

Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **000**, **~~001~~**, **010**, **~~011~~**, **~~100~~**, **101**, **110**, **~~111~~** / threshold: **4**

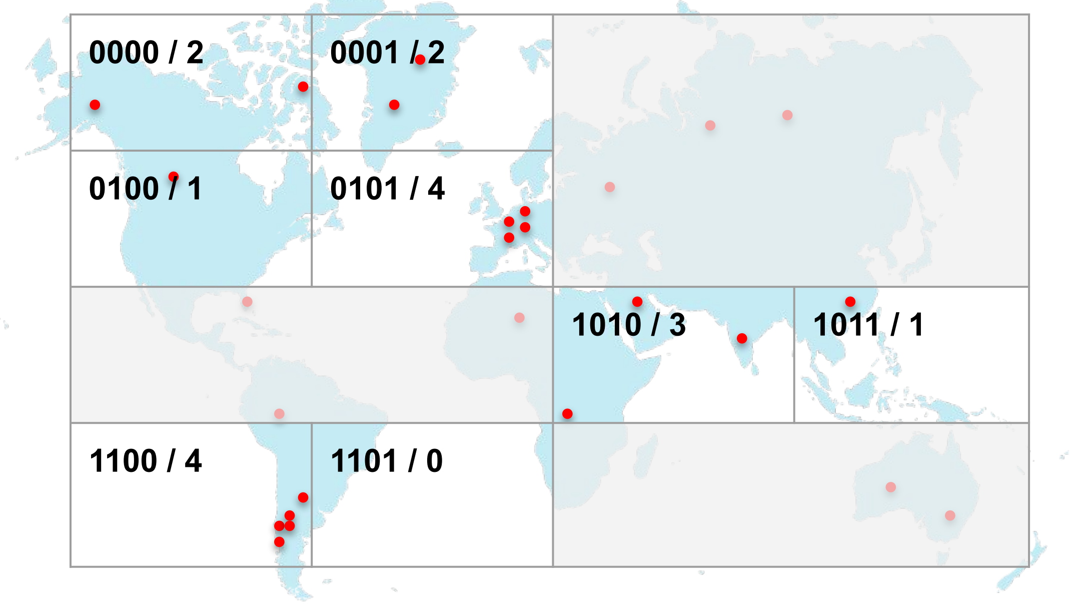


Poplar [BBCG+21]

- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **0000**, **0001**, **0100**, **0101**, **1010**, **1011**, **1100, 1101** / threshold: **4**

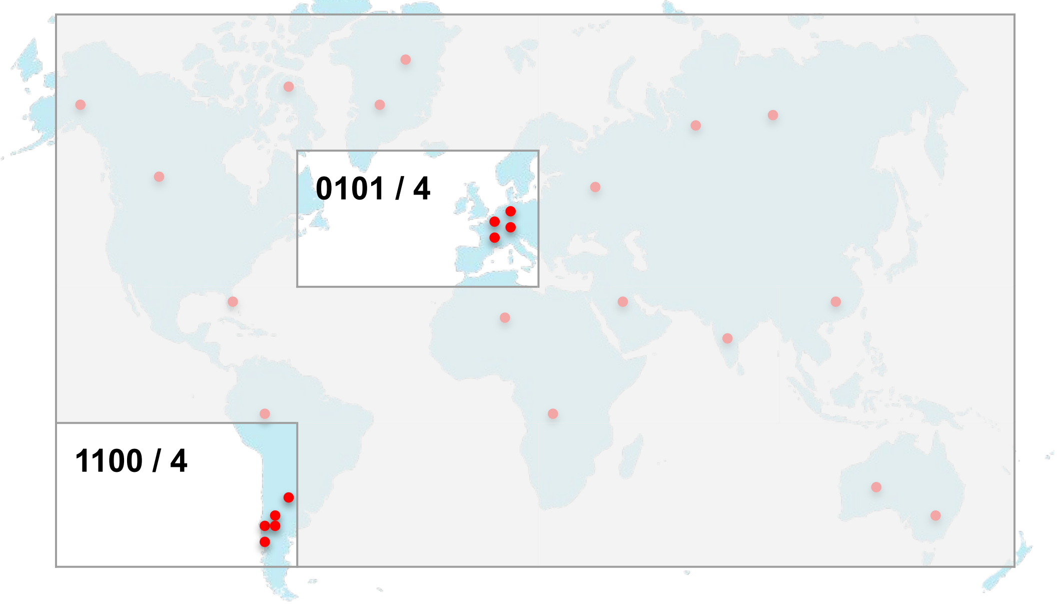


Poplar [BBCG+21]

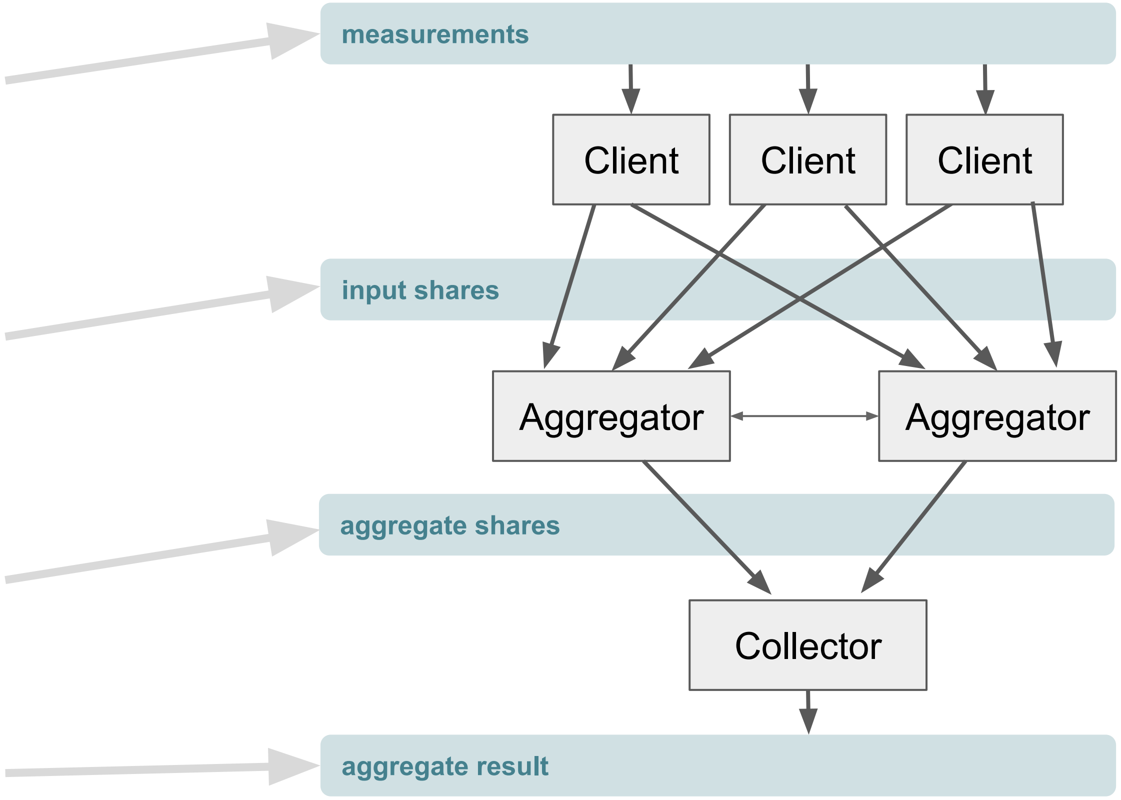
- **Problem** – securely aggregate the *heavy hitters*
  - Measurements: Arbitrary, bit strings
  - Aggregate result: Strings with at least T *hits*
- **Solution** – *Incremental Distributed Point Functions (IDPFs)* [BBCG+21]
  - Counts the number of strings beginning with a given *prefix*

E.g.: *From which ASes ("Autonomous Systems") are users experiencing high latency?*

Candidate prefixes: **~~0000~~**, **~~0001~~**, **~~0100~~**, **0101**, **~~1010~~**, **~~1011~~**, **1100**, **~~1101~~** / threshold: **4**



Poplar [BBCG+21]



Clients' ASes

secret-shared ASes

secret-shared hit count of each candidate AS prefix

hit count of each candidate AS prefix

Poplar [BBCG+21]

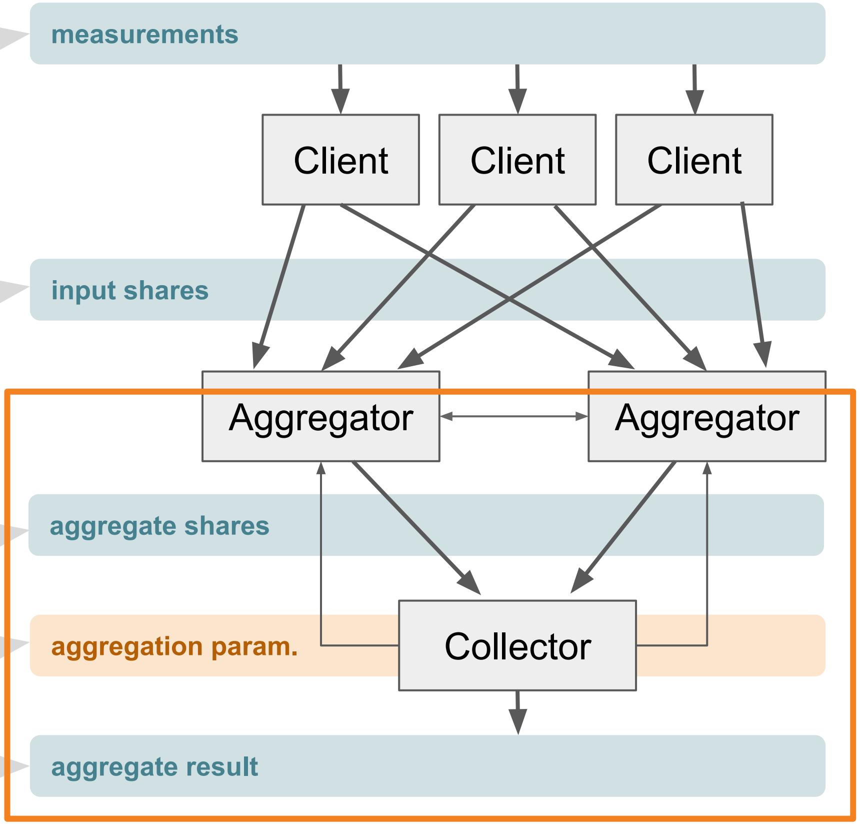
Clients' ASes

secret-shared ASes

secret-shared hit count of each candidate AS prefix

candidate AS prefixes

hit count of each candidate AS prefix



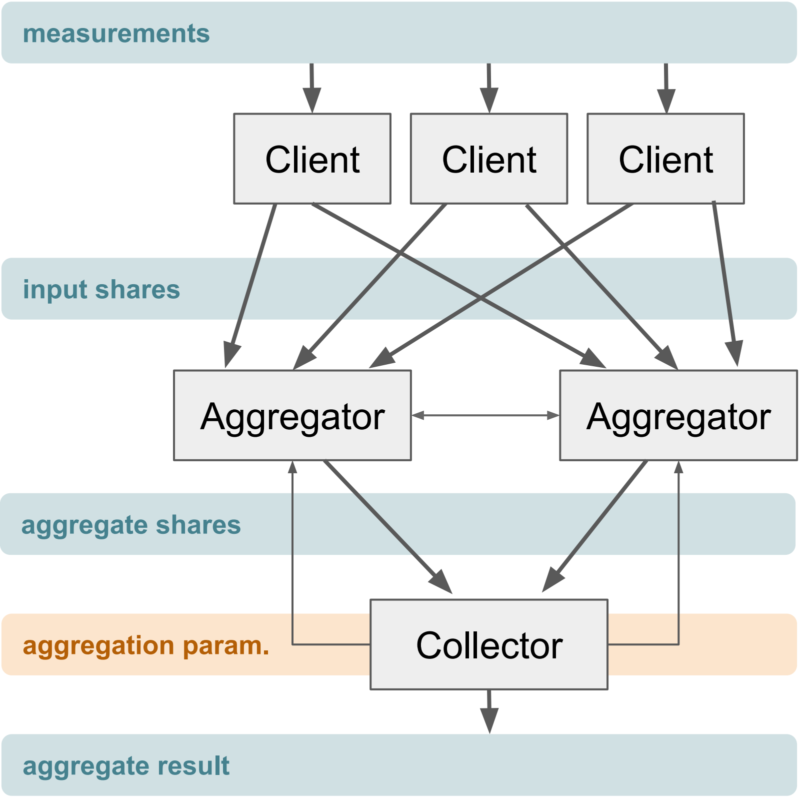
Other candidates

aggregation function

***F*(aggregation parameter,**

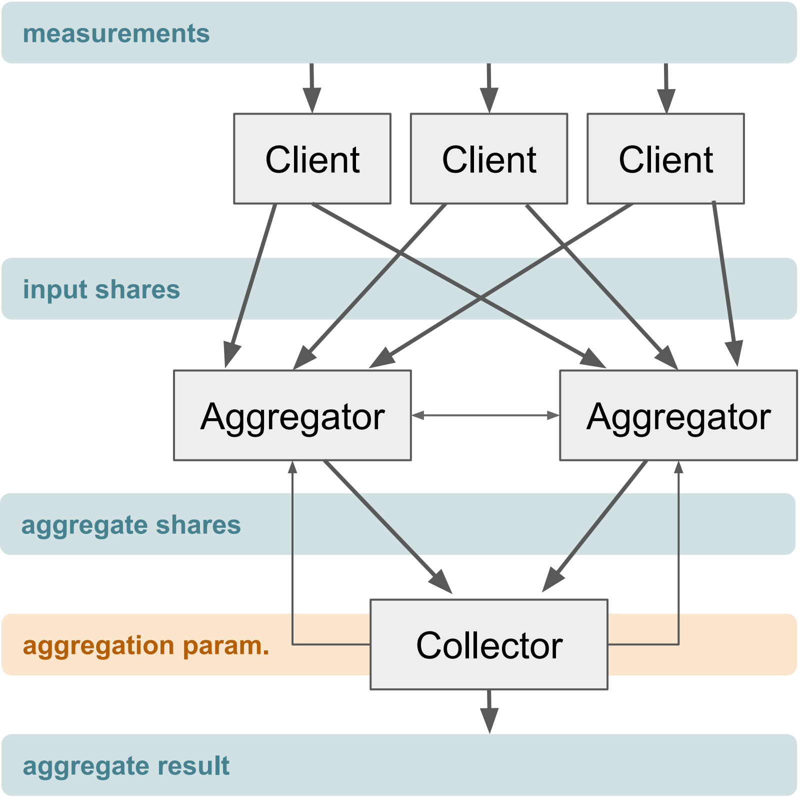
measurements) ⇒

aggregate result



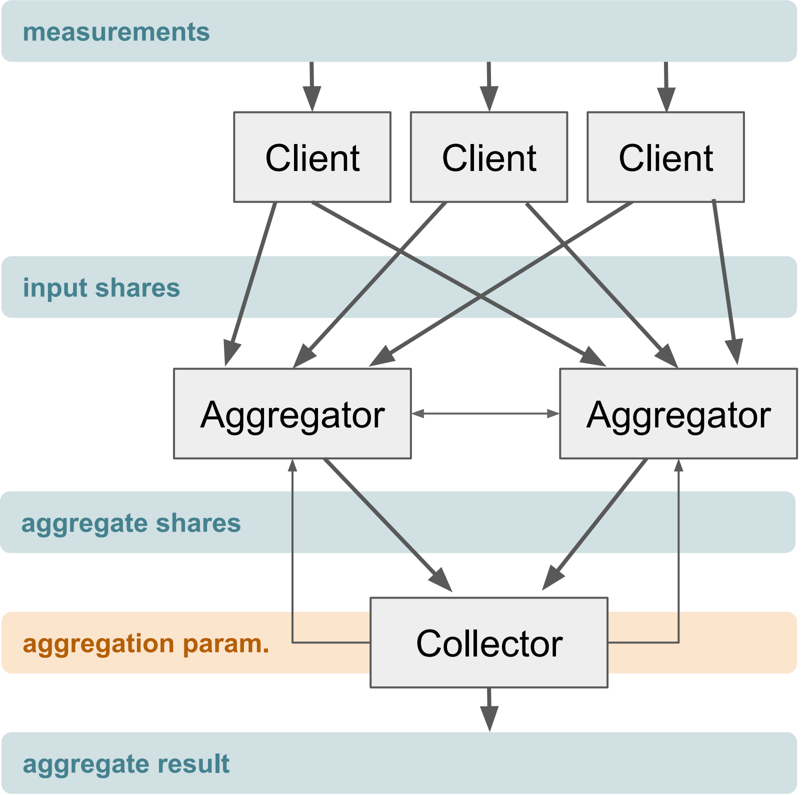
Other candidates

- Prio+ [AGJ+21]
  - Boolean-to-arithmetic conversion via OT extension
    - Compared to Prio, this significantly reduces Client computation for certain measurement types.



Other candidates

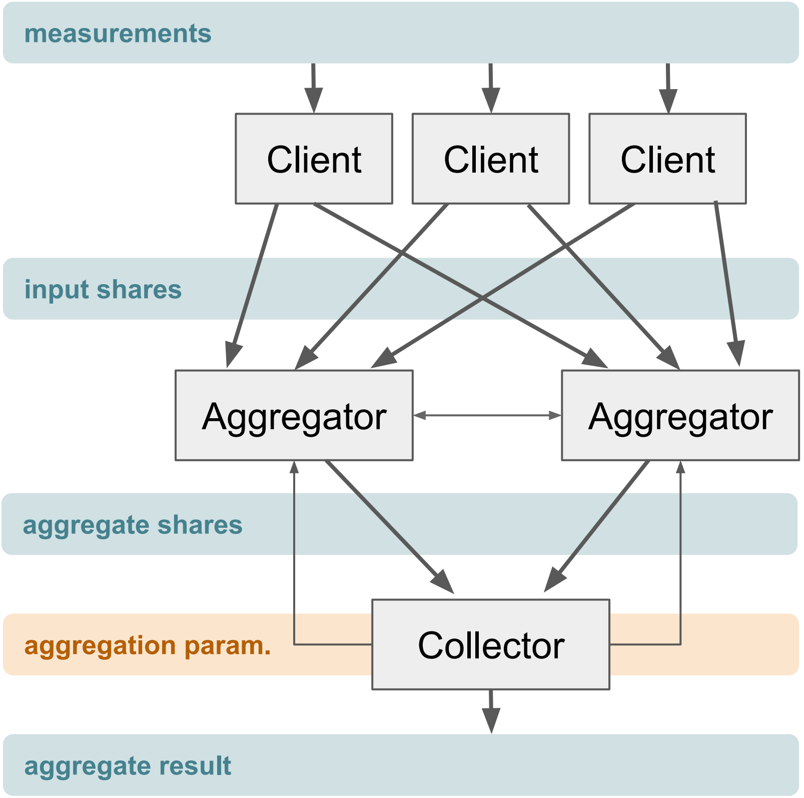
- Prio+ [AGJ+21]
  - Boolean-to-arithmetic conversion via OT extension
    - Compared to Prio, this significantly reduces Client computation for certain measurement types.
- Masked LARk [PCG+21]
  - Compute gradient descent over plaintext features and "masked" labels
    - **Challenge** – Private, verifiable *gradient descent computation* that fits this architecture



One scheme to rule them all? *Nope.*

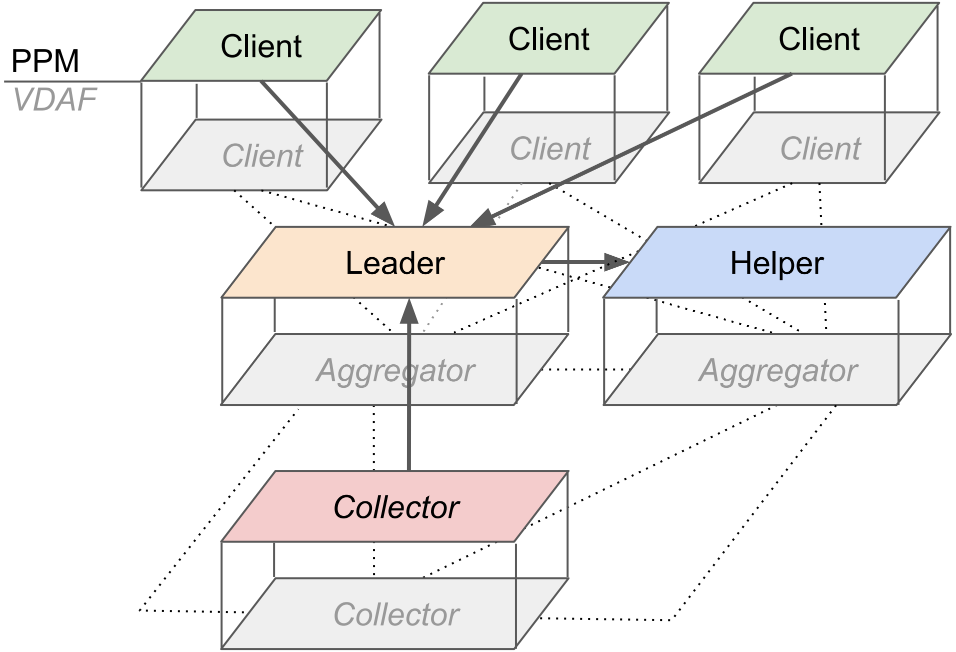
Verifiable Distributed Aggregation Functions (VDAFs)

- draft-patton-cfrg-vdaf-01
  - Defines syntax and (informal) security goals for VDAFs
  - Specifies two constructions: Prio and Poplar



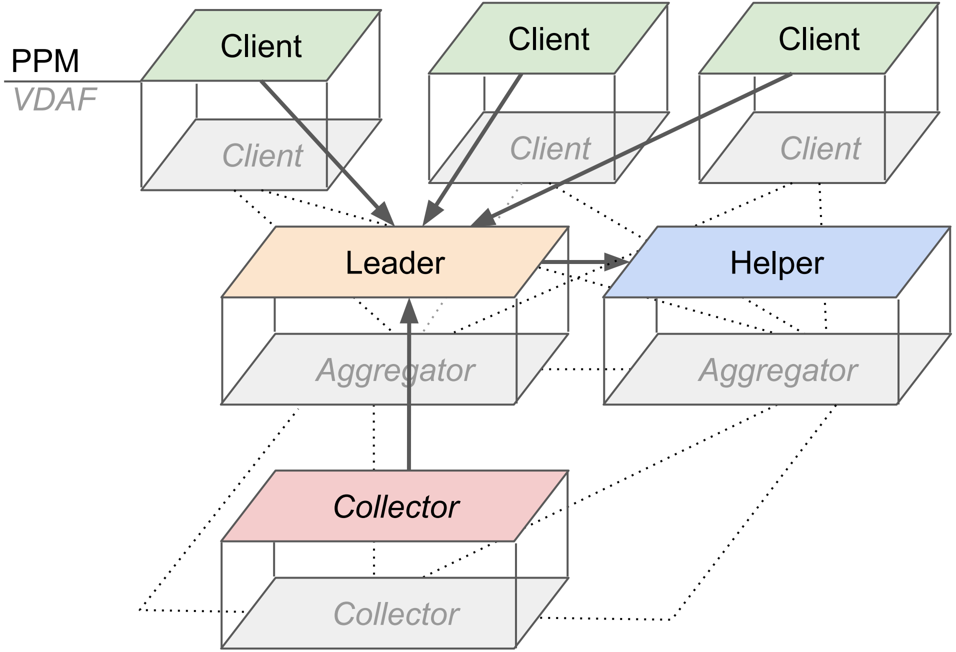
4. The Privacy-Preserving Measurement (PPM) protocol

- draft-gpew-priv-ppm-01 – A protocol for evaluating a VDAF over HTTPS



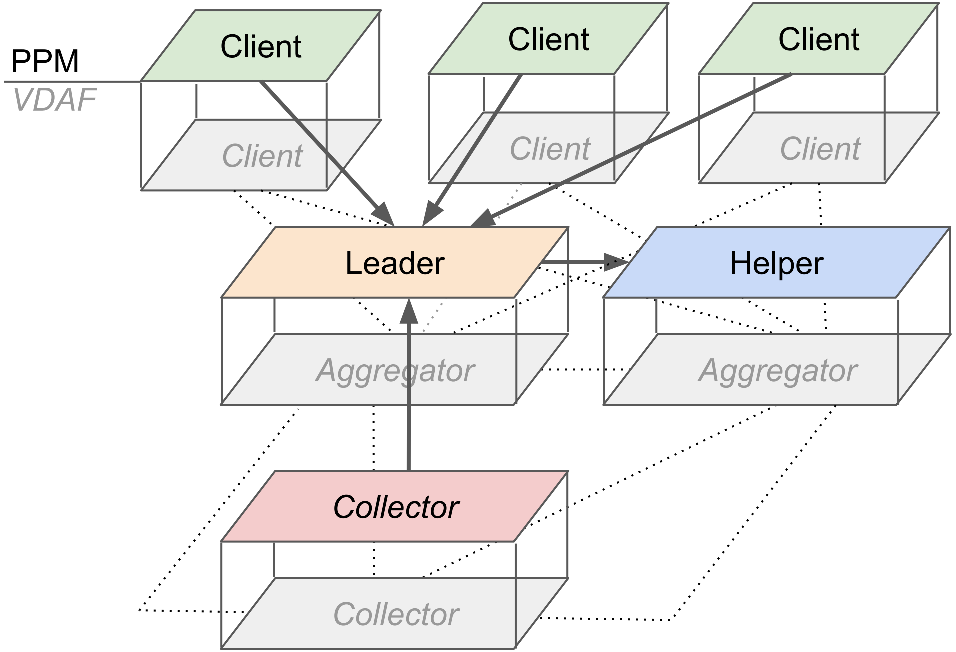
4. The Privacy-Preserving Measurement (PPM) protocol

- draft-gpew-priv-ppm-01 – A protocol for evaluating a VDAF over HTTPS
    - Addresses a variety of operational issues (establishing secure channels, data recovery, picking a VDAF to run, etc.)



4. The Privacy-Preserving Measurement (PPM) protocol

- draft-gpew-priv-ppm-01 – A protocol for evaluating a VDAF over HTTPS
  - Addresses a variety of operational issues (establishing secure channels, data recovery, picking a VDAF to run, etc.)
  - Additional security considerations:
    - Optional defenses against Sybil attacks
    - Support for differential privacy



How to contribute

- Join the PPM mailing list: ppm@ietf.org
- Provide feedback on:
  - draft-patton-cfrg-vdaf-01 (VDAF)
  - draft-gpew-priv-ppm-01 (PPM)
- Got an interesting paper, or a use case you're wondering about? Bring it to the list!

References

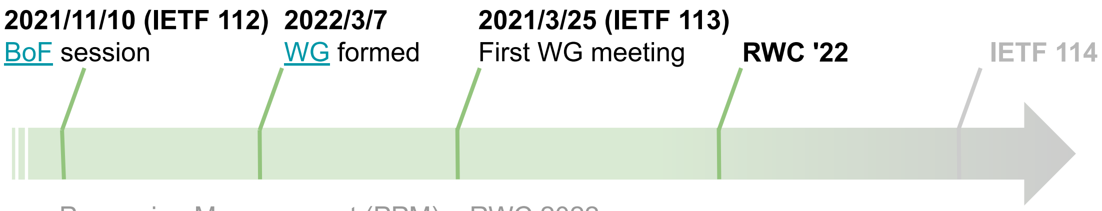
[AGJ+21] Addanki et al. "Prio+: Privacy Preserving Aggregate Statistics via Boolean Shares." ePrint #2021/576.

[CGB17] Corrigan-Gibbs-Boneh. "Prio: Private, Robust, and Scalable Computation of Aggregate Statistics." NSDI 2017.

[BBCG+19] Boneh et al. "Zero-Knowledge Proofs on Secret-Shared Data via Fully Linear PCPs." CRYPTO 2019.

[BBCG+21] Boneh et al. "Lightweight Techniques for Private Heavy Hitters". IEEE S&P 2021.

[PCG+21] Pfeiffer et al. "Masked LARk: Masked Learning, Aggregation and Reporting worKflow." arXiv:2110.14794.



2021/11/10 (IETF 112)
BoF session

2022/3/7
WG formed

2021/3/25 (IETF 113)
First WG meeting

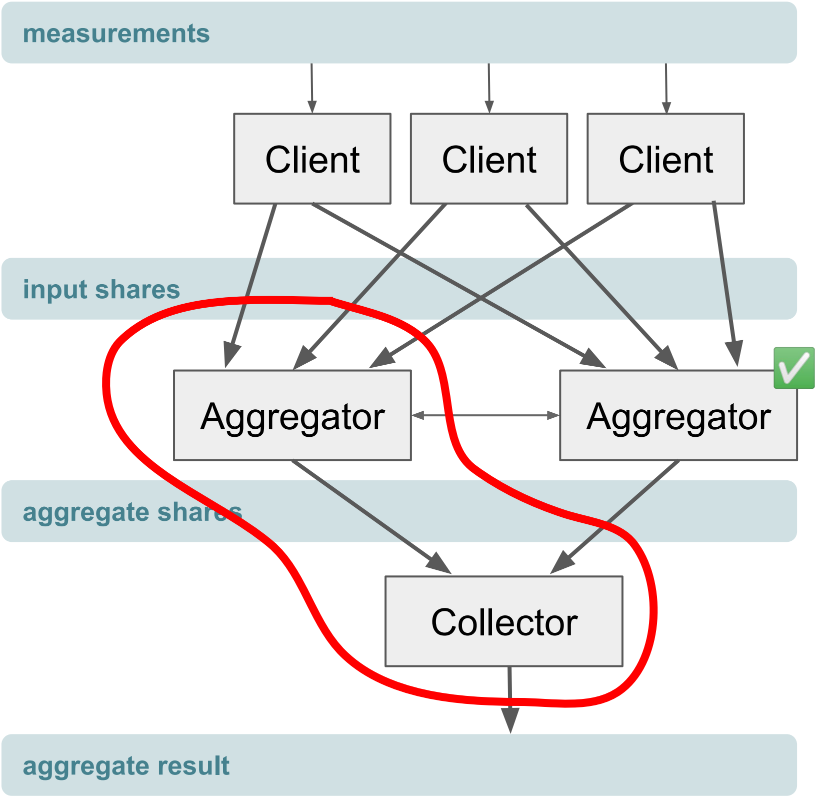
RWC '22

IETF 114

Backup slides

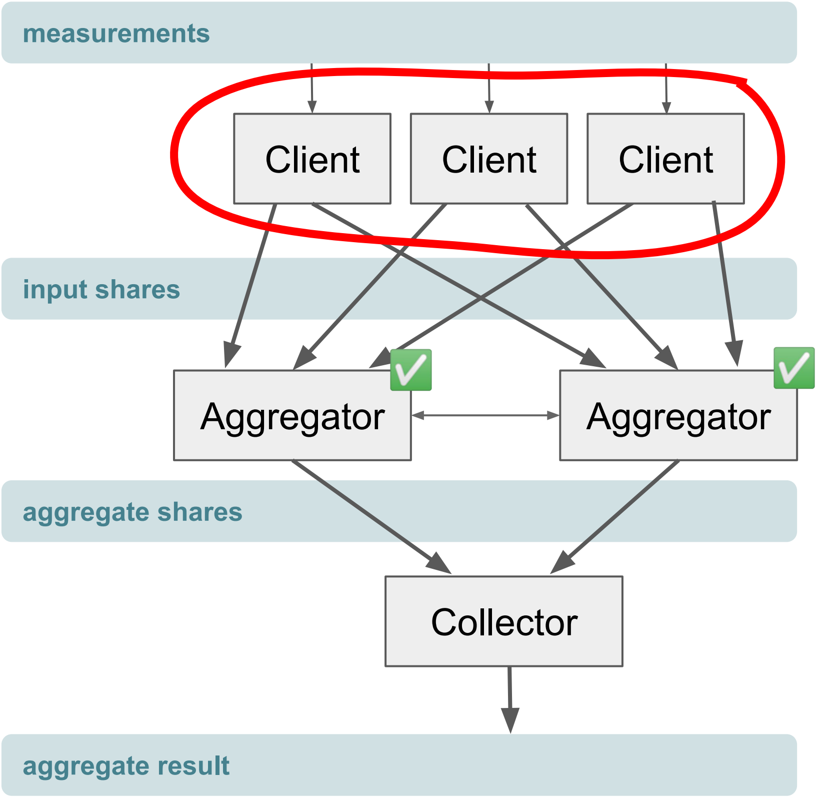
Security Requirements

- **Privacy** – If *at least one Aggregator is honest*, then no server learns anything beyond the aggregate result.



Security Requirements

- **Privacy** – If *at least one Aggregator is honest*, then no server learns anything beyond the aggregate result.
- **Correctness** – If *all Aggregators implement the protocol correctly*, then the Collector correctly computes the aggregate result over measurements *uploaded by honest clients*.



Contributors to VDAF/PPM drafts

Richard Barnes
David Cook
Armando Faz-Hernández
Tim Geoghegan
Charlie Harrison
Anthony Miyaguchi
Brandon Pitman
Christopher Patton
Eric Rescorla
Phillipp Schoppmann
Christopher Wood