Privacy Preserving Measurement
Internet-Draft
Intended status: Standards Track
Expires: 5 January 2026

M. Thomson
Mozilla

4 July 2025

Distributed Aggregation Protocol (DAP) Report Binding Extensions
draft-thomson-ppm-dap-dp-ext-02

Abstract

Report extensions to the Distributed Aggregation Protocol (DAP) are defined to support new modes of operation for the protocol. This includes a batched submission mode where an intermediary can collect reports prior to settling on an exact task configuration. It also includes more flexibility for modes where a differential privacy mechanism is applied as part of the aggregation process. Report extensions bind the content of reports to the chosen options to ensure that the guarantees that both DAP and differential privacy provide are maintained.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://martinthomson.github.io/dap-dp-ext/draft-thomson-ppm-dap-dp-ext.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-thomson-ppm-dap-dp-ext/.

Discussion of this document takes place on the Privacy Preserving Measurement Working Group mailing list (mailto:ppm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ppm/. Subscribe at https://www.ietf.org/mailman/listinfo/ppm/.

Source for this draft and an issue tracker can be found at https://github.com/martinthomson/dap-dp-ext.

Status of This Memo

This Internet-Draft will expire on 5 January 2026.

Table of Contents

1. Introduction 2
2. Conventions and Definitions 4
3. Late Task Binding 4
4. Privacy Budget Consumption 5
4.1. Privacy Budget Report Extension Format 6
4.2. Privacy Budget Usage 7
5. Scoping Extensions 7
5.1. Requester (Website) Identity 8
5.2. Report Partition 8
6. Task Configuration Extensions 9
6.1. Privacy Budget Task Extension 9
6.2. Single Requester Task Extension 10
7. Security Considerations 10
8. IANA Considerations 10
9. References 11
9.1. Normative References 11
9.2. Informative References 11
Acknowledgments 12
Author's Address 12

1. Introduction

The core Distributed Aggregation Protocol (DAP) [DAP] is suited to a deployment model where reports are submitted directly to the leader as they are generated.

DAP report extensions can be used to enable alternative deployment models and approaches. This document defines a set of extensions to enable two key changes to how DAP is used:

* A operating mode where an intermediary is responsible for gathering reports and performing batch submissions for aggregation.

* A operating mode where DAP is part of a differentially-private system where the differential privacy mechanism is applied as part of the aggregation process.

No changes are needed to enable batched report submission in DAP. The challenge is that DAP assumes that clients know all the details of the aggregation task when generating reports. The late_binding report extension (Section 3) loosens this requirement, allowing the intermediary to make some choices about task configuration without coordinating with clients.

One consequence of late binding is the scope of anti-replay protections needs to much broader. For this, scoping report extensions (Section 5) are defined to help constrain the scope of reports over which replay protection operates.

For differential privacy, its effective implementation depends on being able to limit contributions from participants, or set bounds on sensitivity. The basic mechanism that DAP uses to cap contributions is record anti-replay. Aggregators are responsible for ensuring that the same report cannot be aggregated more than once. An honest participant will contribute a limited number of reports and can rely on at least one aggregator preventing each report from being used multiple times.

The default configuration of DAP also depends on having differential privacy configuration known to clients when reports are generated. Clients need to know the parameters of the differential privacy mechanism that is configured so that they can properly account for any privacy loss. This can limit the applicability of the protocol outside of certain narrow patterns.

This document defines extensions to DAP that allow reports to be bound to a specific amount of differential privacy budget expenditure (Section 4). For cases where the DAP aggregators are responsible for applying the differential privacy mechanism -- namely the addition of noise -- this ensures that clients can depend on the correct amount of noise being applied.

For a differentially-private system, scoping extensions (Section 5) can be essential. Scoping constraints can be used to ensure that contributions from different contexts are not combined in a way that would cause sensitivity bounds to be exceeded.

This document also defines a task provisioning extension (see Section 3.3 of [TASKPROV]) that lists the report extensions that need to included in every report. These task extensions also set constraints on the value of those report extensions.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Late Task Binding

DAP presently requires that a client be aware of the task that it is contributing to. The identity of the task is bound to each report through the inclusion of the task ID in the call to the sharding function of the Verifiable Distributed Aggregation Function (VDAF); see Section 5.1 of [VDAF].

The late_binding report extension (codepoint 0xTBD) signals to aggregators that a report was not bound to a specific task when it was created.

Late task binding might be useful when reports are collected by an intermediary. The client that generates the report in this case might be unaware of how the report will ultimately be aggregated. This allows the intermediary to defer the creation of a task until it has determined the necessary parameters for the task.

When sharding and protecting reports, the task ID is replaced with a the fixed, 32-byte sequence of b13e8440f1cdb4da51eed3967e0a2652d27f5005bc35f751daf188b4b746708b (in hex). Specifically, this includes the task_id that is passed to the Vdaf.shard function and is encoded in the InputShareAad struct; see Section 4.5.2 of [DAP].

> This is the output from SHA-256 [SHA2] when passed an ASCII-encoded [ASCII] input of 'no task_id'.

Removing the binding of reports to tasks means that a report might be aggregated across any task that permits the use of this extension.

Binding reports to tasks is defense against attacks that misdirect reports to unintended reports, including those configured with weaker security margins. The best defense against this is to ensure that all tasks that use this are configured with roughly equivalent parameters. Spoiling of results through misdirection is safeguarded by the verification that is performed at the preparation phase of the VDAF.

Misdirecting reports this way can still lead to limited spoiling of tasks, within the bounds permitted for a single task. However, any entity that can perform such redirection is potentially able to effect the same outcome more effectively through entirely falsified reports.

Enforcing anti-replay for tasks that are configured to accept reports with this extension cannot be applied on a per-task scope. A single anti-replay state MUST be used for all reports that include the late_binding report extension.

This could increase the cost of meeting anti-replay requirements. The intent with this extension is that additional constraints, such as one or more of the scoping extensions (see Section 5), will be used to make it more feasible for an aggregator to comply with anti-replay requirements.

4. Privacy Budget Consumption

The gathering of reports can be modeled as the expenditure of privacy budget by a client. That is, clients treat the creation of a report from private information as a limited release of information.

Total privacy loss in this case is determined by the combination of two factors:

* How the report is aggregated.

* How many reports are produced.

If aggregation includes the application of an appropriate differential privacy mechanism (that is, added noise; see [DWORK], [DAP-DP], and Section 8.5 of [DAP]), the client might rely on an understanding of that mechanism to model privacy loss. However, without finer controls, clients need to attribute a fixed privacy loss to each report. Consequently, the client needs to limit the number of reports it generates.

A budget ensures that the total privacy loss can be bounded while providing more flexibility in how reports are constructed. Privacy loss associated with any report (or information release) can be adjusted. Importantly, the amount of noise added to aggregates is based on the expended budget. In general, spending more privacy budget means that less noise is needed to maintain the same level of privacy; conversely, spending less budget means more noise.

A budget might be specified in terms of a metric (like the epsilon parameter in (竜, 隆)-differential privacy) that is expended with each information release.

For example, for an overall budget of 竜=10 might be split four ways: (0.5, 1.5, 2, 6). Noise might then be added, drawing from a distribution with a width inversely proportional to the budget spent; that is, a distribution with a standard deviation proportional to 2, 2/3, 1/2, and 1/6 respectively.

This extension gives clients the ability to manage privacy budget expenditure. By binding the amount of budget spent to each report, the client can transfer responsibility for applying differential privacy to the aggregation service. The addition of noise by the aggregation service can ensure a better trade-off between the amount of added noise and privacy parameters.

4.1. Privacy Budget Report Extension Format

The privacy_budget report extension (codepoint 0xTBD) encodes the amount of privacy budget that the client considers to be expended as a result of producing a report.

The value of the codepoint is an encoding of the number of micro-epsilons of budget that are expended, using as many bytes as needed to encode the value in network byte order. Each unit is a one-millionth of an epsilon (竜) as used in (竜, 隆)-differential privacy.

The micro-epsilon value is encoded as a 32-bit integer in network byte order. This permits the expenditure of up to 竜=4294.967295.

> Note: A separate report extension could be defined to change the scale of this value or switch to a different unit, as necessary.

The delta (隆) parameter is not directly bound to reports. This parameter is rarely used in privacy budgeting. A maximum value might be fixed as part of the system, with the final value being chosen based on the total report volume and -- where applicable -- the total number of tasks that each client might contribute to.

> | Note: Where the delta (隆) value is non-zero, and a client might
> | generate many reports, clients might also need to limit the
> | number of reports to prevent the overall delta value from
> | growing large.

4.2. Privacy Budget Usage

An aggregator that is configured to apply a differential privacy mechanism can operate in one of two modes: either one where the privacy budget value is validated and reports that contain a small value are rejected; or, where the minimum privacy budget value is used to determine the parameters for the differential privacy mechanism.

In the first mode, aggregators each validate this parameter as part of validating each report. The value in the report is compared with the value configured for the task. A report that contains a value that is lower than the value configured for the task is the result of a client that expects that the aggregators will add more noise than the task configuration presently allows. Aggregators MUST reject reports with a privacy budget value that is smaller than their configured privacy budget.

Alternatively, aggregators could adjust the parameters of the differential privacy mechanism they use to match the smallest privacy budget that was included in reports. For long-running tasks that produce multiple outputs over time, it is only necessary to ensure that each output contain noise that is based on the minimum budget expenditure of the reports that are included in that aggregate.

This report extension can be used to protect reports that are conveyed from client by untrusted entities, especially where those entities might be able to choose any task, as enabled by the late_binding report extension (Section 3). This parameter ensures that the entity cannot direct reports to a task that has an inadequate differential privacy mechanism.

5. Scoping Extensions

The DAP report extensions in this section might be used to either constrain the use of reports for tasks that are configured with matching values or group reports for the purposes of detecting duplicates.

Including additional scoping information can also ensure that reports do not get reused outside of their intended scope.

This section defines report extensions that carry requester identity (Section 5.1) and report partition (Section 5.2).

5.1. Requester (Website) Identity

Reports might be requested by an entity that operates at lower trust level than the entity that assembles the report. The entity at the lower trust level might not have access to the information necessary to generate the report.

The requester_identity report extension (codepoint 0xTBD) contains an encoding of the entity that requested the report be created.

For example, an application could ask the operating system to generate a report using information that would normally be withheld from it. Similarly, a website could ask a web browser to generate a report based on otherwise secret information. In either case, the release of information for report is conditional on it only being used by a specific aggregation service under terms that have been previously established with the aggregators. Binding the report to the identity of the requester ensures that any use of the system can be accounted for as coming from that requester.

The specific encoding used in this extension will depend on the application. This extension therefore contains a sequence of application-defined bytes. However, the use of a globally-unique identifier, such as an origin ([ORIGIN]) or serialized site ([SITE]), reduces the likelihood of name collisions. A name collision might either allow two requesters that share an aggregator to share and reuse each others reports (or perhaps to marginally increase the odds of having reports spuriously detected duplicates).

5.2. Report Partition

This extension allows a client to bind a report to an application-defined label. This allows applications to partition reports and have each partition managed separately.

The report_partition report extension (codepoint 0xTBD) contains an application-defined sequence of bytes.

The use of this report extension allows aggregators to partition their state for tracking reports. Duplicate reports only need to be tracked across a matching partition, for detecting duplicates within a task or for detecting duplicates across tasks.

The selection of partition values might need to be coordinated with aggregators. If partitions are used by aggregators, the amount of state the aggregator tracks is increased by the number of partitions. This represents an increase in total storage, in exchange for reducing the scope over which that storage needs to be consistent.

An aggregator could constrain the values that are accepted for this extension, rejecting reports that lack the extension or have disallowed values.

6. Task Configuration Extensions

DAP tasks that are created with the DAP task configuration extension [TASKPROV] can set constraints on the reports that are accepted for a task.

```
enum {
  task_budget (0xTBD),
  single_requester (0xTBD),
  (2^16-1)
} ExtensionType;

uint16 ReportExtensions<2..2^16-2>;
```

Task provisioning extensions are defined that govern the use of the privacy_budget (Section 4) and requester_identity (Section 5.1) report extensions. These task provisioning extensions ensure that reports submitted to the task include the corresponding report extensions and that reports with invalid values are rejected.

6.1. Privacy Budget Task Extension

The task_budget task provisioning extension contains the minimum privacy budget that can be expended on the task. The value is encoded identically to the privacy_budget report extension; see Section 4.

The task_budget task provisioning extension both establishes a requirement for reports to include the privacy_budget report extension and sets an upper bound on the amount of privacy budget that can be expended. Reports that contain no privacy_budget report extension or those that contain a privacy_budget report extension value smaller than the value in the task_budget extension MUST be rejected.

The value of this extension can be used to bound the noise that is applied by a centrally managed differential privacy mechanism. Aggregators use the minimum value across all submitted reports to determine how much noise is added; setting a minimum budget allows that noise to be bounded.

6.2. Single Requester Task Extension

This single_requester task provisioning extension contains the identity of a requester. The format of this is identical to the requester_identity report extension; see Section 5.1.

Use of this extension indicates that all reports submitted to the task MUST include a requester_identity report extension with the specified value. All reports that omit that extension or contain a different value MUST be rejected.

7. Security Considerations

Security factors specific to each extension is enumerated in the respective sections: Section 3, Section 5.1, Section 5.2, and Section 4.

Use of DAP is subject to the security considerations of DAP (Section 8 of [DAP]) and the VDAF that is in use (Section 9 of [VDAF].

8. IANA Considerations

This document registers report extensions in the "Report Extension Identifiers" registry established in Section 9.2.2 of [DAP].

New report extension registrations are tabulated in Table 1.

Value	Name	Reference
TBD	late_binding	Section 3
TBD	privacy_budget	Section 4
TBD	requester_identity	Section 5.1
TBD	partition	Section 5.2

Table 1: DAP Extensions

This document registers task provisioning extensions in the "Taskbind Extensions" registry established in Section 7.2 of [TASKPROV].

New task provisioning extensions are tabulated in Table 2.

Value	Name	Reference
TBD	task_budget	Section 6.1
TBD	single_requester	Section 6.2

Table 2: Task Provisioning Extensions

9. References

9.1. Normative References

[DAP] Geoghegan, T., Patton, C., Pitman, B., Rescorla, E., and C. A. Wood, "Distributed Aggregation Protocol for Privacy Preserving Measurement", Work in Progress, Internet-Draft, draft-ietf-ppm-dap-15, 29 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ppm-dap-15>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[TASKPROV] Wang, S. and C. Patton, "Task Binding and In-Band Provisioning for DAP", Work in Progress, Internet-Draft, draft-ietf-ppm-dap-taskprov-02, 5 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ppm-dap-taskprov-02>.

9.2. Informative References

[ASCII] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/rfc/rfc20>.

[DAP-DP] Chen, J., McMillan, A., Patton, C., Talwar, K., and S. Wang, "Differential Privacy Mechanisms for DAP", Work in Progress, Internet-Draft, draft-wang-ppm-differential-privacy-00, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-wang-ppm-differential-privacy-00>.

[DWORK] Dwork, C. and A. Roth, "The Algorithmic Foundations of Differential Privacy", Now Publishers, Foundations and Trends速 in Theoretical Computer Science vol. 9, no. 3-4, pp. 211-407, DOI 10.1561/0400000042, 2013, <https://doi.org/10.1561/0400000042>.

[ORIGIN] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <https://www.rfc-editor.org/rfc/rfc6454>.

[SHA2] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/rfc/rfc6234>.

[SITE] WHATWG, "HTML - Living Standard", 26 January 2021, <https://html.spec.whatwg.org/#site>.

[VDAF] Barnes, R., Cook, D., Patton, C., and P. Schoppmann, "Verifiable Distributed Aggregation Functions", Work in Progress, Internet-Draft, draft-irtf-cfrg-vdaf-15, 17 June 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-vdaf-15>.

Acknowledgments

Roxana Geambesu noted that a binding to requester identity (Section 5.1) was an important component of a robust differential privacy system design. David Cook provided useful feedback about the design and document.

Author’s Address

Martin Thomson
Mozilla
Email: mt@lowentropy.net