Package: dicomtk.fl (via r-universe)

November 14, 2026

Title Federated Learning, Trusted Third-Party Custody, and Differential Privacy for 'DICOM'

Version 0.1.0

Description Federated learning and trusted third-party ('TTP') custody satellite for 'dicomtk'. Provides 'Mainzelliste' REST client integration with offline fallback, polymorphic encryption and pseudonymisation ('PEP') for cross-project unlinkability, hardened Bloom-filter privacy-preserving record linkage ('PPRL') with randomized response bit-flipping and balanced padding, (epsilon, delta)-differential privacy ('DP') federated statistics with a formal privacy budget accountant and minimum-cell k-anonymity suppression, site-package exporters for 'Flower', 'NVIDIA FLARE', and 'OpenFL', and cryptographic 'site_admin' outbound approval gating.

License GPL (>= 2)

URL https://lauyeehow1986-hub.github.io/dicomtk.fl, https://github.com/lauyeehow1986-hub/dicomtk.fl

BugReports https://github.com/lauyeehow1986-hub/dicomtk.fl/issues

Depends R (>= 4.1.0)

Imports R6 (>= 2.5.0), rlang (>= 1.1.0), jsonlite (>= 1.8.0), openssl (>= 2.1.0), httr2 (>= 1.0.0), tibble (>= 3.0.0), stats, utils

Suggests dicomtk (>= 0.6.0), dicomtk.server (>= 1.0.0), testthat (>= 3.0.0), withr, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libssl-dev

Repository https://lauyeehow1986-hub.r-universe.dev

Date/Publication 2026-11-14 15:26:21 UTC

RemoteUrl https://github.com/lauyeehow1986-hub/dicomtk.fl

RemoteRef main

RemoteSha a74941d84682c284fe853d10d3959a16aa2349da

Contents

dp_aggregate_sites . . . 2
dp_budget_accountant . . . 3
dp_federated_summary . . . 4
dp_gaussian_mechanism . . . 5
dp_laplace_mechanism . . . 6
dp_suppress_small_cells . . . 6
fl_capabilities . . . 7
fl_export_site_package . . . 7
fl_site_approval_gate . . . 8
pep_derive_site_token . . . 10
pep_rekey_pseudonym . . . 11
pprl_dice_similarity . . . 11
pprl_encode_bloom . . . 12
pprl_link_cohorts . . . 13
ttp_mainzelliste_client . . . 14

Index **16**

`dp_aggregate_sites` *Aggregate Differentially Private Site Summaries Across a Federation*

Description

Aggregate Differentially Private Site Summaries Across a Federation

Usage

```
dp_aggregate_sites(site_summaries)
```

Arguments

`site_summaries` List of `"dicomtk_fl_site_summary"` objects.

Value

A list with `n_sites`, `total_count`, `weighted_mean`, and `site_ids`.

dp_budget_accountant *Differential Privacy Budget Accountant*

Description

Tracks cumulative `(epsilon, delta)` privacy budget expenditure across federated queries and aborts with `dicomtk_fl_dp_budget_exceeded_error` whenever a query would exceed the configured privacy ceiling.

Public fields

`max_epsilon` Maximum allowed cumulative epsilon.

`max_delta` Maximum allowed cumulative delta.

`spent_epsilon` Cumulative epsilon spent so far.

`spent_delta` Cumulative delta spent so far.

Methods

Public methods:

- `dp_budget_accountant$new()`
- `dp_budget_accountant$spend()`
- `dp_budget_accountant$remaining()`
- `dp_budget_accountant$history()`
- `dp_budget_accountant$clone()`

Method `new()`**:** Initialize a privacy budget accountant.

Usage:

```
dp_budget_accountant$new(max_epsilon = 3, max_delta = 1e-05)
```

Arguments:

`max_epsilon` Positive numeric ceiling for epsilon.

`max_delta` Non-negative numeric ceiling for delta.

Method `spend()`**:** Spend `(epsilon, delta)` privacy budget for a named query.

Usage:

```
dp_budget_accountant$spend(epsilon, delta = 0, query_name = "dp_query")
```

Arguments:

`epsilon` Positive numeric epsilon cost.

`delta` Non-negative numeric delta cost.

`query_name` Descriptive label for the audit ledger.

Returns: Invisibly returns `self`.

Method `remaining()`**:** Return remaining `(epsilon, delta)` privacy budget.

Usage:

```
dp_budget_accountant$remaining()
```

Returns: Named list with `epsilon` and `delta`.

Method `history()`**:** Return the privacy expenditure ledger as a `tibble`.

Usage:

```
dp_budget_accountant$history()
```

Returns: A `tibble`.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
dp_budget_accountant$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`dp_federated_summary` *Compute (epsilon, delta)-DP Federated Site Summary*

Description

Enforces minimum cohort size `k_min`, clips values to `[lower, upper]` to bound global sensitivity, and releases noisy count, mean, and variance.

Usage

```
dp_federated_summary(
  x,
  lower,
  upper,
  epsilon = 1,
  delta = 1e-06,
  k_min = 10L,
  accountant = NULL,
  site_id = "SITE_1"
)
```

Arguments

`x`	Numeric vector of local site measurements.
`lower`	Lower clipping bound.
`upper`	Upper clipping bound (`upper` > `lower`).
`epsilon`	Total epsilon budget for this site summary.
`delta`	Total delta budget for Gaussian mean/variance release.
`k_min`	Minimum cohort size required before release (default `10L`).
`accountant`	Optional `dp_budget_accountant` instance.
`site_id`	Site identifier string.

Value

A list of class "dicomtk_fl_site_summary".

---

dp_gaussian_mechanism *Gaussian (epsilon, delta)-Differential Privacy Mechanism*

Description

Gaussian (epsilon, delta)-Differential Privacy Mechanism

Usage

```
dp_gaussian_mechanism(
  value,
  sensitivity,
  epsilon,
  delta,
  accountant = NULL,
  query_name = "gaussian_query"
)
```

Arguments

value	Numeric scalar or vector true statistic.
sensitivity	L2 global sensitivity of value.
epsilon	Positive privacy parameter.
delta	Failure probability in (0, 1).
accountant	Optional dp_budget_accountant instance.
query_name	Audit label for accountant.

Value

Perturbed numeric vector.

dp_laplace_mechanism *Laplace Differential Privacy Mechanism*

Description

Laplace Differential Privacy Mechanism

Usage

```
dp_laplace_mechanism(
  value,
  sensitivity,
  epsilon,
  accountant = NULL,
  query_name = "laplace_query"
)
```

Arguments

value	Numeric scalar or vector true statistic.
sensitivity	L1 global sensitivity of value.
epsilon	Positive privacy parameter.
accountant	Optional dp_budget_accountant instance.
query_name	Audit label for accountant.

Value

Perturbed numeric vector.

dp_suppress_small_cells *Suppress Small Contingency Cells for k-Anonymity*

Description

Suppress Small Contingency Cells for k-Anonymity

Usage

```
dp_suppress_small_cells(counts, k_min = 10L)
```

Arguments

counts	Numeric or integer vector of cell counts.
k_min	Minimum non-zero cell count allowed without suppression (default 10L).

Value

Integer vector with `0 < count < k_min` replaced by `NA_integer_`.

---

fl_capabilities *Package-level federated audit & configuration helper*

Description

Package-level federated audit & configuration helper

Usage

```
fl_capabilities()
```

Value

A named list describing `dicomtk.fl` capabilities and defaults.

Examples

```
fl_capabilities()
```

---

fl_export_site_package *Export a Governed Federated Learning Site Bundle (Flower, NVFlare, or OpenFL)*

Description

Validates a signed `site_admin` approval receipt against `cohort_manifest`, then writes a framework-specific site configuration bundle (`"flower"`, `"nvflare"`, or `"openfl"`) to `out_dir`.

Usage

```
fl_export_site_package(
  cohort_manifest,
  out_dir,
  framework = c("flower", "nvflare", "openfl"),
  approval_gate,
  approval_receipt,
  dp_config = list(clip_norm = 1, noise_multiplier = 1.1)
)
```

Arguments

`cohort_manifest`	Named list describing the local site cohort and feature schema.
`out_dir`	Output directory path where the site bundle is written.
`framework`	Target FL framework: `"flower"`, `"nvflare"`, or `"openfl"`.
`approval_gate`	A `fl_site_approval_gate` R6 instance.
`approval_receipt`	Signed `dicomtk_fl_approval_receipt` issued by `site_admin`.
`dp_config`	Optional differential privacy hyperparameters (`clip_norm`, `noise_multiplier`).

Value

Invisibly returns a list summarizing the exported bundle and written files.

`fl_site_approval_gate` *Cryptographic Site-Admin Outbound Approval Gate*

Description

Enforces that no federated learning site bundle or summary table leaves an institution without an explicit cryptographic sign-off from an authenticated `site_admin` principal, verifying minimum-cell k-anonymity and manifest SHA-256 integrity.

Public fields

`k_min` Minimum cell count required for outbound approval.

Methods

Public methods:

- `fl_site_approval_gate$new()`
- `fl_site_approval_gate$manifest_digest()`
- `fl_site_approval_gate$approve_export()`
- `fl_site_approval_gate$verify_receipt()`
- `fl_site_approval_gate$revoke_receipt()`
- `fl_site_approval_gate$clone()`

Method `new()`**:** Initialize an outbound approval gate.

Usage:

```
fl_site_approval_gate$new(gate_secret, k_min = 10L)
```

Arguments:

`gate_secret` Secret HMAC-SHA256 signing key held by the site governance enclave.

`k_min` Minimum cell count threshold (default `10L`).

Method `manifest_digest()`**:** Compute canonical SHA-256 digest of an outbound manifest.

Usage:

```
fl_site_approval_gate$manifest_digest(manifest)
```

Arguments:

`manifest` Named list describing the outbound cohort/model manifest.

Returns: Hexadecimal SHA-256 digest string.

Method `approve_export()`**:** Issue a signed `site_admin` approval receipt for an outbound manifest.

Usage:

```
fl_site_approval_gate$approve_export(
  principal,
  manifest,
  dp_epsilon = NULL,
  dp_delta = 0,
  min_cell_count
)
```

Arguments:

`principal` List or RBAC principal containing `user_id` and `role`.

`manifest` Named list describing the outbound cohort/bundle.

`dp_epsilon` Optional declared DP epsilon budget.

`dp_delta` Optional declared DP delta budget.

`min_cell_count` Minimum cell count in the outbound dataset/summary.

Returns: An object of class `"dicomtk_fl_approval_receipt"`.

Method `verify_receipt()`**:** Verify a `site_admin` approval receipt against an outbound manifest.

Usage:

```
fl_site_approval_gate$verify_receipt(receipt, manifest)
```

Arguments:

`receipt` A `"dicomtk_fl_approval_receipt"` object.

`manifest` Named list matching the approved manifest.

Returns: `TRUE` if valid; otherwise aborts with `dicomtk_fl_unauthorized_export_error`.

Method `revoke_receipt()`**:** Revoke a previously issued approval receipt.

Usage:

```
fl_site_approval_gate$revoke_receipt(receipt_id)
```

Arguments:

`receipt_id` Character scalar receipt ID.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
fl_site_approval_gate$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`pep_derive_site_token` *Derive a Project- and Site-Scoped Polymorphic Pseudonym (PEP)*

Description

Computes a cryptographically unlinkable project- and site-specific pseudonym from a root TTP `master_pseudonym` using keyed HMAC-SHA256 domain separation, and seals an AES-256-GCM envelope so a TTP can re-key the pseudonym for another project without exposing the root identity.

Usage

```
pep_derive_site_token(
  master_pseudonym,
  project_id,
  site_id,
  pep_key,
  context_salt = "v1"
)
```

Arguments

`master_pseudonym`	Root TTP pseudonym (character scalar).
`project_id`	Target research project identifier.
`site_id`	Participating hospital/site identifier.
`pep_key`	Secret PEP master key.
`context_salt`	Optional protocol version/context salt (default `"v1"`).

Value

An object of class `"dicomtk_fl_pep_token"`.

Examples

```
tok <- pep_derive_site_token("TTP-1234", "PROJ_A", "SITE_1", "secret-key")
tok$pseudonym
```

pep_rekey_pseudonym *Re-Key a PEP Token for a Different Project or Site*

Description

Unwraps the sealed AES-256-GCM PEP envelope at the TTP and derives a new project- and site-scoped pseudonym without returning the root identity.

Usage

```
pep_rekey_pseudonym(
  token,
  target_project_id,
  target_site_id,
  pep_key,
  context_salt = "v1"
)
```

Arguments

token	A "dicomtk_fl_pep_token" object.
target_project_id	Destination project identifier.
target_site_id	Destination site identifier.
pep_key	Secret PEP master key.
context_salt	Optional protocol context salt (default "v1").

Value

A new "dicomtk_fl_pep_token" scoped to target_project_id and target_site_id.

pprl_dice_similarity *Compute Sorensen-Dice Similarity Between Two Bloom Filters*

Description

Compute Sorensen-Dice Similarity Between Two Bloom Filters

Usage

```
pprl_dice_similarity(bf1, bf2)
```

Arguments

bf1	A "dicomtk_fl_bloom" object.
bf2	A "dicomtk_fl_bloom" object of identical length.

Value

Numeric scalar similarity in `[0, 1]`.

`pprl_encode_bloom` *Encode Quasi-Identifiers into a Hardened PPRL Bloom Filter*

Description

Encodes patient quasi-identifiers into character q-grams hashed via keyed HMAC-SHA256 double-hashing into an `m`-bit Bloom filter, with optional randomized response bit-flipping (`flip_prob`) and constant-weight balanced padding (`balanced = TRUE`) to resist frequency cryptanalysis.

Usage

```
pprl_encode_bloom(
  record,
  secret_key,
  m = 1024L,
  k = 15L,
  q = 2L,
  flip_prob = 0,
  balanced = TRUE
)
```

Arguments

`record`	Named list (or 1-row data.frame) of quasi-identifiers.
`secret_key`	Shared consortium HMAC-SHA256 linkage key.
`m`	Base Bloom filter length in bits (default `1024L`).
`k`	Number of hash functions per q-gram (default `15L`).
`q`	Character q-gram width (default `2L`).
`flip_prob`	Randomized response bit-flipping probability in `[0, 0.5)` (default `0.0`).
`balanced`	Logical; if `TRUE`, concatenates `bits` and `1L - bits` so Hamming weight is strictly `m` (default `TRUE`).

Value

An object of class `"dicomtk_fl_bloom"`.

pprl_link_cohorts *Privacy-Preserving Record Linkage Across Two Site Cohorts*

Description

Privacy-Preserving Record Linkage Across Two Site Cohorts

Usage

```
pprl_link_cohorts(
  cohort_a,
  cohort_b,
  secret_key,
  id_col = "record_id",
  threshold = 0.85,
  m = 1024L,
  k = 15L,
  flip_prob = 0,
  balanced = TRUE
)
```

Arguments

cohort_a	Data frame with record_id and quasi-identifier columns.
cohort_b	Data frame with record_id and quasi-identifier columns.
secret_key	Shared consortium HMAC-SHA256 linkage key.
id_col	Name of the identifier column (default "record_id").
threshold	Minimum Dice similarity for a match (default 0.85).
m	Bloom filter base length (default 1024L).
k	Number of hash functions (default 15L).
flip_prob	Randomized response bit-flipping probability (default 0.0).
balanced	Use balanced constant-weight Bloom filters (default TRUE).

Value

A tibble with id_a, id_b, and dice_similarity.

`ttp_mainzelliste_client` *Trusted Third-Party (TTP) Mainzelliste Client*

Description

R6 client for interacting with a Mainzelliste Trusted Third-Party (TTP) pseudonymisation service, with an offline deterministic mode for air-gapped testing and simulation.

Public fields

`base_url` Base URL of the Mainzelliste REST service.

`mode` Either `"offline"` or `"mainzelliste"`.

Methods

Public methods:

- `ttp_mainzelliste_client$new()`
- `ttp_mainzelliste_client$create_session()`
- `ttp_mainzelliste_client$request_pseudonym()`
- `ttp_mainzelliste_client$resolve_temp_id()`
- `ttp_mainzelliste_client$clone()`

Method `new()`**:** Initialize a Mainzelliste TTP client.

Usage:

```
ttp_mainzelliste_client$new(
  base_url = NULL,
  api_key = NULL,
  mode = c("offline", "mainzelliste"),
  ttp_secret = "dicomtk-fl-default-ttp-secret"
)
```

Arguments:

`base_url` Optional Mainzelliste REST URL.

`api_key` Optional API key for Mainzelliste REST calls.

`mode` `"offline"` (default) or `"mainzelliste"`.

`ttp_secret` Secret key used for deterministic offline TTP IDs.

Method `create_session()`**:** Create a TTP session.

Usage:

```
ttp_mainzelliste_client$create_session()
```

Returns: A list with `session_id`, `mode`, and `created_at`.

Method `request_pseudonym()`**:** Request a root TTP pseudonym for patient identifying attributes.

Usage:

```
ttp_mainzelliste_client$request_pseudonym(session_id, idat)
```

Arguments:

`session_id` Active session identifier from `create_session()`.

`idat` Named list of identifying attributes (e.g., `vorname`, `nachname`, `geburtsdatum`).

Returns: A list containing `pid`, `temp_id`, and `mode`.

Method `resolve_temp_id()`**:** Resolve an offline temporary token back to its root TTP pseudonym.

Usage:

```
ttp_mainzelliste_client$resolve_temp_id(temp_id)
```

Arguments:

`temp_id` Temporary ID returned by `request_pseudonym()`.

Returns: Character scalar root TTP pseudonym.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
ttp_mainzelliste_client$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Index

dp_aggregate_sites, 2
dp_budget_accountant, 3
dp_federated_summary, 4
dp_gaussian_mechanism, 5
dp_laplace_mechanism, 6
dp_suppress_small_cells, 6

fl_capabilities, 7
fl_export_site_package, 7
fl_site_approval_gate, 8

pep_derive_site_token, 10
pep_rekey_pseudonym, 11
pprl_dice_similarity, 11
pprl_encode_bloom, 12
pprl_link_cohorts, 13

ttp_mainzelliste_client, 14