

# PGP Key Servers

# PGP Key Servers

Admin path: **Encryption > PGP Key Servers** (`view_pgp_key_servers.cfm`, `inc/publish_pgp_keyring.cfm`).

This page maintains the **HKP keyserver publish list** — the set of public OpenPGP keyservers Hermes will push (`gpg --send-keys`) recipient public keys to when an admin clicks **Publish** on a keyring row in **Encryption > External Recipients**. Each row is a hostname only (no scheme, no port, no path) stored in the `pgp_keyservers` table.

“ **Important: publish, not lookup.** Despite the page name, the keyserver list is currently **outbound-only**. Hermes does NOT auto-query these servers to fetch a recipient's PGP key at send time — recipient keys must be imported manually (paste-in or file upload) on **Encryption > External Recipients > PGP Keyrings**. The keyservers configured here are used solely by the **Publish** action in `inc/publish_pgp_keyring.cfm`, which pushes a key the operator already holds (typically the local CipherMail server's public key or a recipient's key that was imported and now needs broader distribution).

## What the page does

The page is a thin CRUD over a 3-column table:

<code>pgp_keyservers</code> column	Purpose
<code>id</code>	PK
<code>keyserver</code>	Hostname only, e.g. <code>keys.openpgp.org</code>
<code>note</code>	Free-text label, e.g. "Primary keyserver"

Three actions:

Action	Form value	Effect
--------	------------	--------

Add	<code>action=add</code>	Validates hostname via <code>IsValid("email", "bob@" &amp; ks)</code> (rejects URLs and <code>host:port</code> ), checks for duplicate <code>keyserver</code> , INSERTs the row
Single delete	<code>action=delete</code> with <code>delete_id</code>	DELETE one row by id
Bulk delete	<code>action=bulk_delete</code> with <code>selected_ids</code> (CSV)	DELETE every selected id in a loop

The existing-servers card is a DataTable with select-all + per-row checkboxes + a **Delete Selected** button. There is no per-row enable flag, no protocol/port column, no priority ordering — every row in the table is offered as a publish target in the modal on the keyring page, indexed by `id`.

## What "publish" actually runs

When the operator clicks **Publish** on a keyring row at **External Recipients > PGP Keyrings**, the `publish_gpg_keyring.cfm` include does the following for each selected keyserver:

```
/usr/bin/gpg --homedir /opt/hermes/.gnupg/ \
  --keyserver <hostname-from-gpg_keyservers> \
  --send-keys <recipient-PGP-key-id>
```

The temp script is written to `/opt/hermes/tmp/<token>_publish_gpg_key.sh`, `chmod'd`, executed, and deleted. The standard Hermes temp-script pattern. The keyserver hostname is substituted via `REReplace` of the `THE-KEY-SERVER` placeholder in `/opt/hermes/scripts/publish_gpg_key.sh`.

GPG itself picks the protocol — `gpg` defaults to `hkps://` (HKP over TLS on tcp/443) for a bare hostname when the local `dirmngr` is configured for it; otherwise it falls back to `hkp://` (tcp/11371). Hermes does not pass an explicit scheme.

Failure modes the include recognizes (sets `session.m` and redirects):

PGP stderr fragment	Meaning	<code>session.m</code>
Server indicated a failure	Keyserver rejected the upload (rate limit, policy, malformed key)	22
No name	Local GPG keyring has no user-id matching the requested key id	23
Not found	Local GPG keyring does not hold the requested key id	24
Not a key ID	The key id parameter was malformed	25

A successful publish returns no recognized fragment and falls through to the success branch.

# Recommended seed list

The default install seeds one row:

Hostname	Note
<code>keyserver.ubuntu.com</code>	Ubuntu SKS OpenPGP Public Key Server

Practical 2026 replacements / additions the operator should consider:

Hostname	Network	Caveats
<code>keys.openpgp.org</code>	Identity-verified standalone (Hagrid)	Strips third-party signatures (no web-of-trust); requires email verification before a key becomes searchable by email address; <b>does not distribute revocation certificates the way SKS did</b>
<code>keyserver.ubuntu.com</code>	SKS-style federated	Was the last reliable SKS-network bridge; survives but is no longer broadly federated
<code>pgp.mit.edu</code>	Legacy SKS	Largely defunct in 2026 — uploads may not propagate; leave off unless legacy compatibility is required
<code>&lt;your-org-keyserver&gt;</code>	Internal HKP daemon (e.g. Hagrid)	Useful if the operator runs an authoritative keyserver for their own domain — same publish path

The page does NOT validate keyserver reachability at add time; an unreachable host simply produces a publish failure when the operator clicks Publish later.

## What is NOT on this page

Several things an operator might reasonably expect from a "PGP Key Servers" page that are intentionally elsewhere or absent:

Expectation	Where it actually lives
Per-server enable/disable toggle	Not implemented — every row is a publish target
Search-order priority	Not applicable — publish iterates the explicit selection from the modal, not the full list

Expectation	Where it actually lives
Inbound recipient-key auto-lookup at send time ( <code>gpg --search-keys / recv-keys</code> )	Not implemented anywhere in Hermes; recipient keys must be imported manually on <b>External Recipients &gt; PGP Keyrings</b>
Automatic refresh of imported keys (re-fetch + merge updates)	Not implemented; operators must re-import a key if a recipient rotates
DANE <code>OPENPGPKEY</code> DNS lookup	Not currently surfaced in the Hermes admin or CipherMail engine config
WKD (Web Key Directory) discovery at <code>https://&lt;domain&gt;/.well-known/openpgpkey/...</code>	Not currently surfaced in the Hermes admin or CipherMail engine config
HKP port override	Not on this page; GPG picks the port
Encryption policy decisions ("fail closed vs send plaintext if no key")	<a href="#">Encryption Settings</a> , not here

The page is deliberately scoped to one job: **a list of HKP endpoints the publish flow can push to.**

# When the operator should populate this list

Two practical scenarios:

1. **The organization wants its own gateway PGP key to be publicly discoverable.** Add the operator's preferred public keyserver(s), then publish the local CipherMail key from **External Recipients > PGP Keyrings**. External counterparties running `gpg --recv-keys` against the same keyserver can then pull it for encrypting mail back to Hermes-served users.
2. **A specific recipient has asked for their key (which the operator already holds locally) to be pushed somewhere centralized.** Less common — usually recipients self-publish — but the workflow supports it.

If the deployment never publishes keys outward (typical Community deployments that use S/MIME exclusively, or PGP deployments that exchange keys out-of-band via attachment), this page can remain empty with no functional impact.

# Container and database touch-points

Component	Location	Role
Page	<code>config/hermes/var/www/html/admin/2/view_pgp_key_servers.cfm</code> ( <code>hermes_commandbox</code> )	CRUD UI
Publish include	<code>config/hermes/var/www/html/admin/2/include/publish_pgp_keyring.cfm</code> ( <code>hermes_commandbox</code> )	Builds + runs the temp <code>gpg --send-keys</code> script
Template script	<code>/opt/hermes/scripts/publish_pgp_key.sh</code>	Single line: <code>/usr/bin/gpg --homedir /opt/hermes/.gnupg/ --keyserver THE-KEY-SERVER --send-keys THE_KEY_ID 2&gt;&amp;1</code>
GPG home	<code>/opt/hermes/.gnupg/</code> (bind-mounted into <code>hermes_commandbox</code> )	Local GPG keyring holding the keys eligible for publish
Storage	<code>pgp_keyservers</code> in <code>hermes</code> DB ( <code>hermes_db_server</code> )	The list itself
Engine	<code>hermes_ciphermail</code> (separate from <code>publish</code> — handles actual signing/encryption at send time)	NOT touched by this page; this page only manages the GPG outbound-publish list

The publish flow runs **gpg on** `hermes_commandbox` (which has the `/opt/hermes/.gnupg/` keyring bind-mounted) — not inside `hermes_ciphermail`. CipherMail keeps its own per-recipient PGP store in the `djigzo` DB for actual encryption/decryption operations.

## Related

- [External Recipients](#) — per-counterparty key store; the **Publish** action that consumes this list lives on the keyring sub-page there
- [Encryption Settings](#) — outbound encryption policy that decides whether absence of a recipient PGP key blocks the message or falls through to plaintext
- [Internal CA](#) — sibling page for the S/MIME side of recipient key issuance and trust
- **Advanced Settings** (sidebar link to `/ciphermail/`) — CipherMail's own admin UI for the deep PGP keyring operations the Hermes admin does not surface

---

Revision #48

Created 2026-05-31 12:52:38 UTC by Dino Edwards

Updated 2026-06-20 13:33:21 UTC by Dino Edwards