OpenPGP Key Replacement

draft-ietf-openpgp-replacementkey

Daphne Shaw, Andrew Gallagher

OpenPGP @ IETF121 November 2024

Basic Design — Recap

draft-ietf-openpgp-replacementkey

- New signature subpacket for use in revocations and self-sigs
- Contains a class (flags) octet, and zero or more target records of the form:
  - Record length in octets
  - Version and Fingerprint of the target key
  - Imprint of the target key (similar to the fingerprint, but using the sig's hash algo)
- Can be used in normal or "inverse" directions, distinguished by a flag bit in the class
  - Paired normal and inverse subpackets form a key equivalence binding
- Provides a hint for how to find a new key, and a preferred ordering of equivalent keys

Recent Changes

draft-ietf-openpgp-replacementkey

- No subpacket version field
- Target records have a length field (for robust parsing)
- Explicit guidance for encryption subkey selection
- Updates to draft name and terminology — no more please :-)

Major Outstanding Questions

draft-ietf-openpgp-replacementkey

- Target record format #15
- Preferred Key Server subpacket #22
- UX guidance #17

Target Record Format — Recap (1/8)

draft-ietf-openpgp-replacementkey

- Key equivalence requires similar cryptographic strength to a subkey bind
  - v4 fingerprints use a deprecated hash algorithm
    - But exploit requires second-preimage attack
  - Imprint field means that we only rely on the signature’s hash algorithm
    - Similar in principle to Attested Certifications subpacket design
  - In many cases, the imprint field will be the same as the fingerprint field
    - Should we deduplicate?

Target Record Format (2/8)

draft-ietf-openpgp-replacementkey

- Remember that:
  - Imprints only exist to cover a corner case
  - Fingerprint required for keyserver lookup (for foreseeable future)
  - Cryptographic strength equals that of imprint, not fingerprint
  - When v4 keys die, SHA1 dies
- Goal: find balance of easy lookup, cryptographic strength, and elegance

Target Record Format — Option 1 (3/8)

draft-ietf-openpgp-replacementkey

- No deduplication (current state):
  - Fingerprint is required to look up target
  - Imprint is required to verify target
- 32 octets is negligible compared to PQC key size
- Keeps the parser simple

Target Record Format — Option 2 (4/8)

draft-ietf-openpgp-replacementkey

- Opportunistic deduplication:
  - Fingerprint is required to look up target
  - Imprint is only required to verify target IFF it differs from fingerprint
- Absence of duplicate field inferred from record length
- Lossless

Target Record Format — Option 3 (5/8)

draft-ietf-openpgp-replacementkey

- Lossy deduplication:
  - Imprint is required to verify target (and locate it?)
  - Fingerprint is optional
- Deprecates fingerprints without providing alternative lookup mechanism:
  - We must fall back on non-universal lookups (WKD, autocrypt)
  - …or else we must develop new keyserver lookup (when? who?)

Target Record Format — Option 0 (6/8)

draft-ietf-openpgp-replacementkey

- Still on the table:
  - Remove imprint
  - Back to original target record format
  - Relies on fingerprint digest (e.g. SHA-1) for cryptographic hardness
    - Exploitability very low

Target Record Format — Detour (7/8)

draft-ietf-openpgp-replacementkey

- <columbo> Just One More Thing </columbo>

Preferred Key Server subpacket (1/2)

draft-ietf-openpgp-replacementkey

- Draft currently says:

  A Preferred Key Server subpacket MAY be included in the revocation or direct key signature to recommend a location and method to fetch the replacement certificate

- Indicates the PKS of both the current key and *all* target key(s)
- Ideally, PKS could be set independently for current key *and* each target
- Alternatively, don’t provide PKS hints for targets

Preferred Key Server subpacket (2/2)

draft-ietf-openpgp-replacementkey

- If we want to make the PKS apply only to the target key, either:
- Option 1: remove references to PKS of target key(s)
  - Receiving implementations are on their own
- Option 2: put PKS info in target records instead
  - Current key, each target can have unique (or no) PKS, no constraints

Target Record Format — Decision (8/8)

draft-ietf-openpgp-replacementkey

- AG: Please don’t get rid of fingerprints
  - Withholding useful information is not a virtue
  - SHA1 will die a natural death, don’t panic
  - 32 octets isn’t worth it
- Should we add an optional PKS field to target records?
  - …in which case we need fingerprint to perform the lookup
- Decision required (on both fingerprints/imprints and PKS)

UX Guidance — Recap (1/2)

draft-ietf-openpgp-replacementkey

- How do we expect users to create the bidirectional signatures?
- Do users have to have both/all primary keys online simultaneously?
- Or do we add the subpacket to one key first, and then prompt the user to consent to the binding when the equivalent(s) next comes online?

UX Guidance — Proposal (2/2)

draft-ietf-openpgp-replacementkey

1. On (creation | revocation) of a key, or on triggering an expert option:
   - Ask whether this is a (replacement | original) for another key(s)
   - Add a replacement key subpacket as necessary
   - If the other private key is available, add/update the inverse subpacket
2. On receiving a cert, if a new unpaired RKS refers to our private key:
   - Ask for confirmation, then add/update the inverse subpacket

Minor Outstanding Questions

draft-ietf-openpgp-replacementkey

- Is the current guidance re selection of encryption subkeys sufficient? #14
- Two-octet record length field? (e.g. PKS field may overflow 255 octets) #20
- Is the draft terminology clear?

Timeline

draft-ietf-openpgp-replacementkey

- Nov 19 (2 weeks): new draft with outstanding questions addressed
- Dec/Jan: cleanup, finalisation
- Jan/Feb: implementations?

Further Information

draft-ietf-openpgp-replacementkey

- Draft: https://datatracker.ietf.org/doc/html/draft-ietf-openpgp-replacementkey
- Repo: https://andrewgdotcom.gitlab.io/openpgp-replacementkey