One ciphertext to rule them all

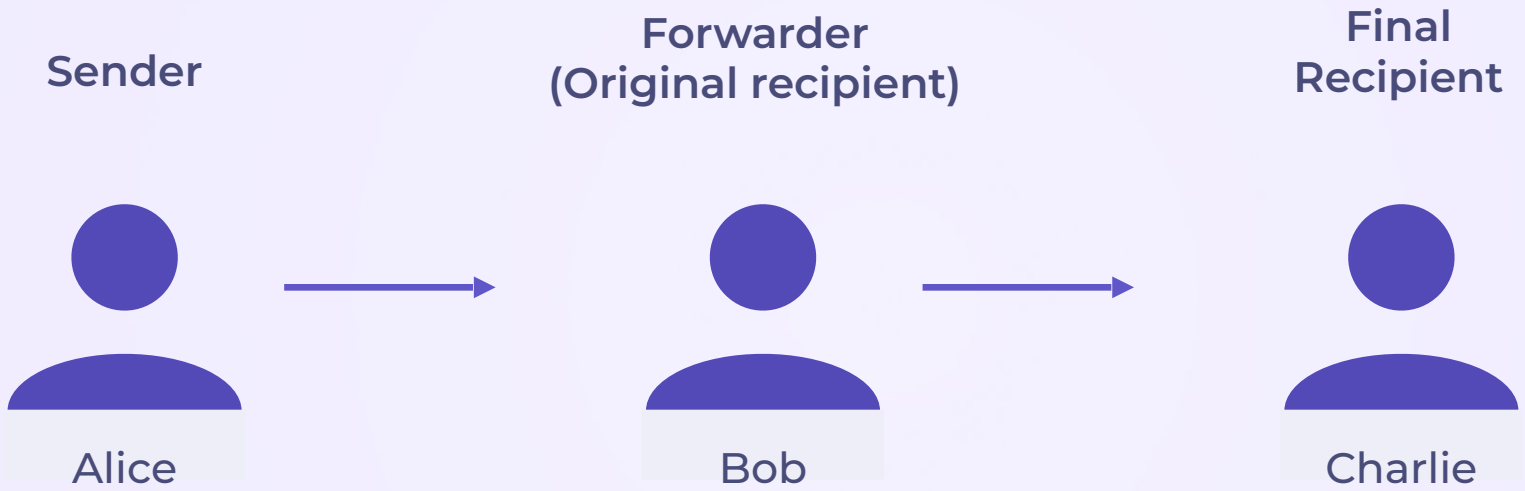
Deploying interoperable E2EE Email Forwarding and Mailing Lists

Lara Bruseghini, Aron Wussler

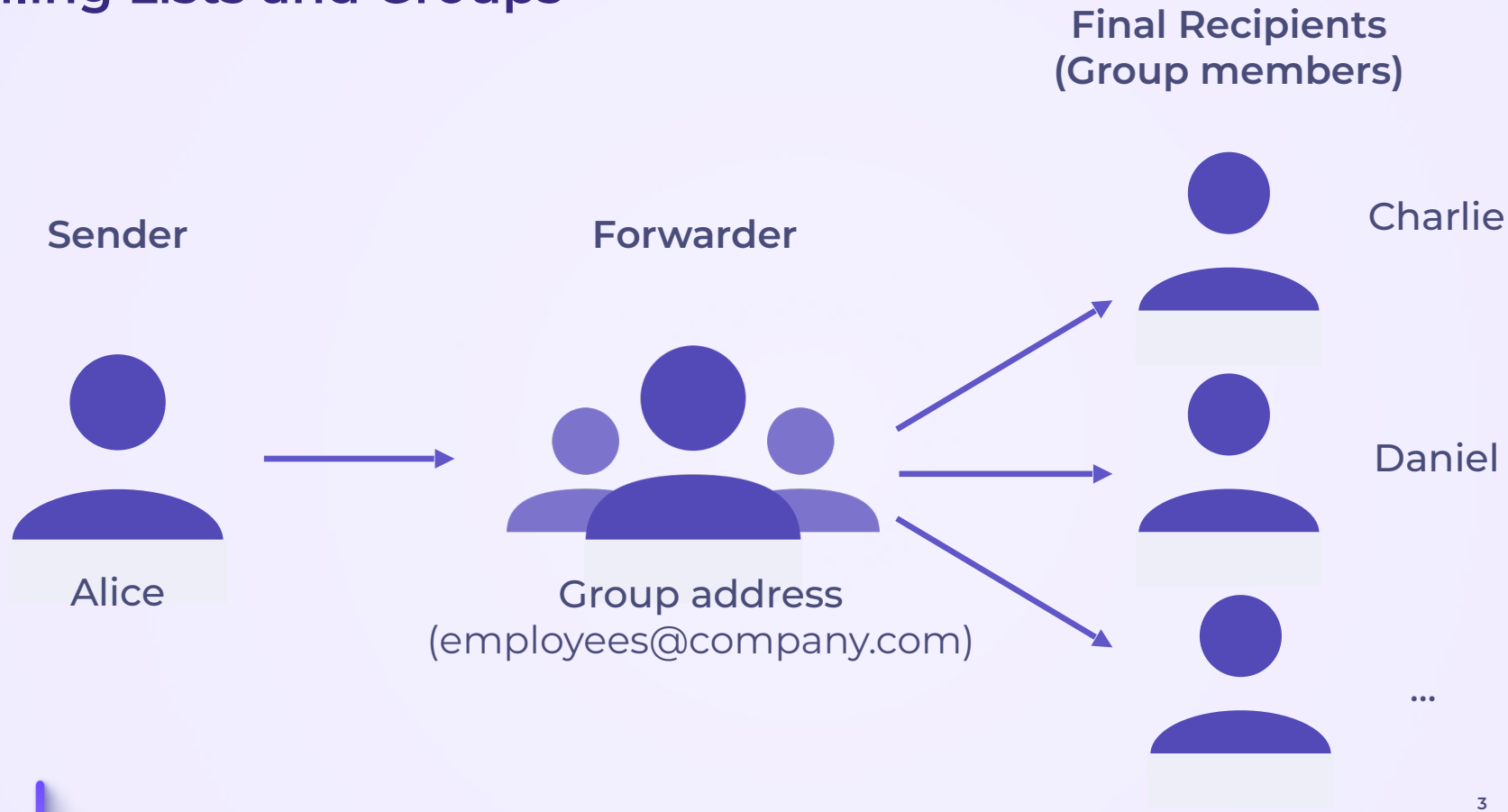
May 10th, 2026

Proton | Privacy by default

Automatic email forwarding



Mailing Lists and Groups



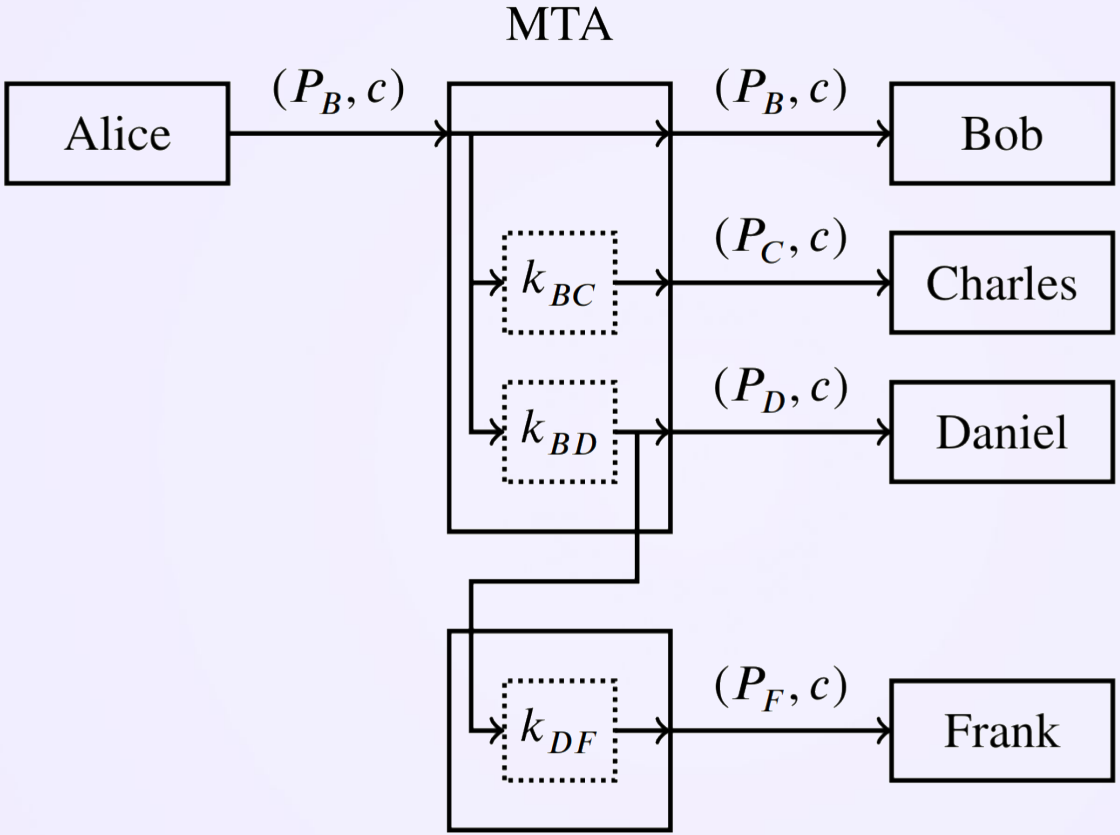
Use Case and Requirements

- Temporary trust delegation (e.g. Out-of-office)
  - Final recipient trusted to access email contents, but not the forwarder’s secret keys
- Groups / private mailing lists preserving E2EE, hiding from the sender the set of recipients
  - Single encryption key (also for sender’s performance)
  - Recipients may be added or removed without updating the distributed public key
- Interoperability across clients and backwards compatibility
  - The sender must not have to update their software to be able to send an email
  - OpenPGP 🫶

Threat model and security guarantees

- Forwarding server (cryptographic proxy) should not be able to read messages
- Final recipient should only be able to read the forwarded messages
  - Server trusted with respecting filters / time interval(*) (e.g. forward only emails received during time off)
  - They cannot impersonate the forwarder (limit their key capabilities to decryption, not sign/ encrypt/certify)
- Both are partially trusted entities (distributed trust):
  - By colluding they can reconstruct Bob's encryption key (e.g. access to other emails)
- Email sender should be unaware of forwarding (preserve forwarder's privacy)
  - They might also have pinned the forwarder's public key

E2EE forwarding via proxied ECDH



Deployment at scale

(Paid-only feature)

Released in 2023

39K E2EE forwarding setups

Released in 2024

13K groups

44K members

User experience

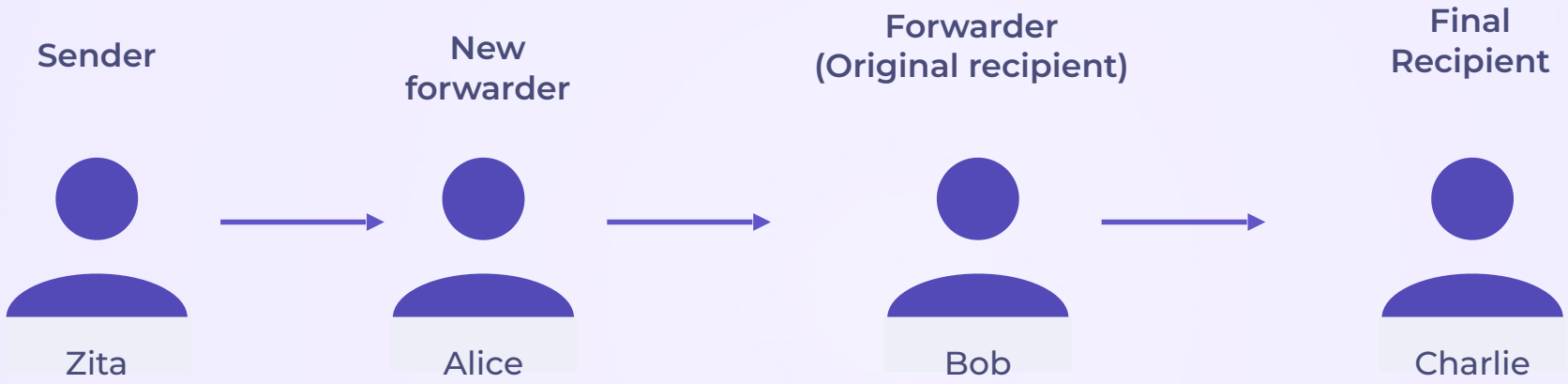
The protocol was designed with user experience in mind, keeping it **as close to “normal” unencrypted forwarding and mailing lists**:

- Original sender not aware that the forwarding is taking place, or who are the final recipients
- External E2EE (OpenPGP) messages must also be forward-able
- Seamless message decryption and signature verification for all involved parties
- Take advantage of & don’t disrupt existing trusted keys
    - Between Alice and Bob, as well as Bob and the final recipients
- Don’t require additional setup steps

Technical deployment challenges

- Key setup tied to forwarding acceptance
  - users are already used to accept forwarding or group membership requests
  - we “hide” a key exchange in this step
  - and prompt for partial force-upgrade (old clients can’t read forwarding keys)
- Nested subgroups and forwarding chains (coming up)
- Key Transparency integration (almost out of the box)
  - Message signatures are preserved
  - Forwarding decryption keys (Charlie’s) don’t need inclusion in KT
- Standardization effort (coming up)

Forwarding chains



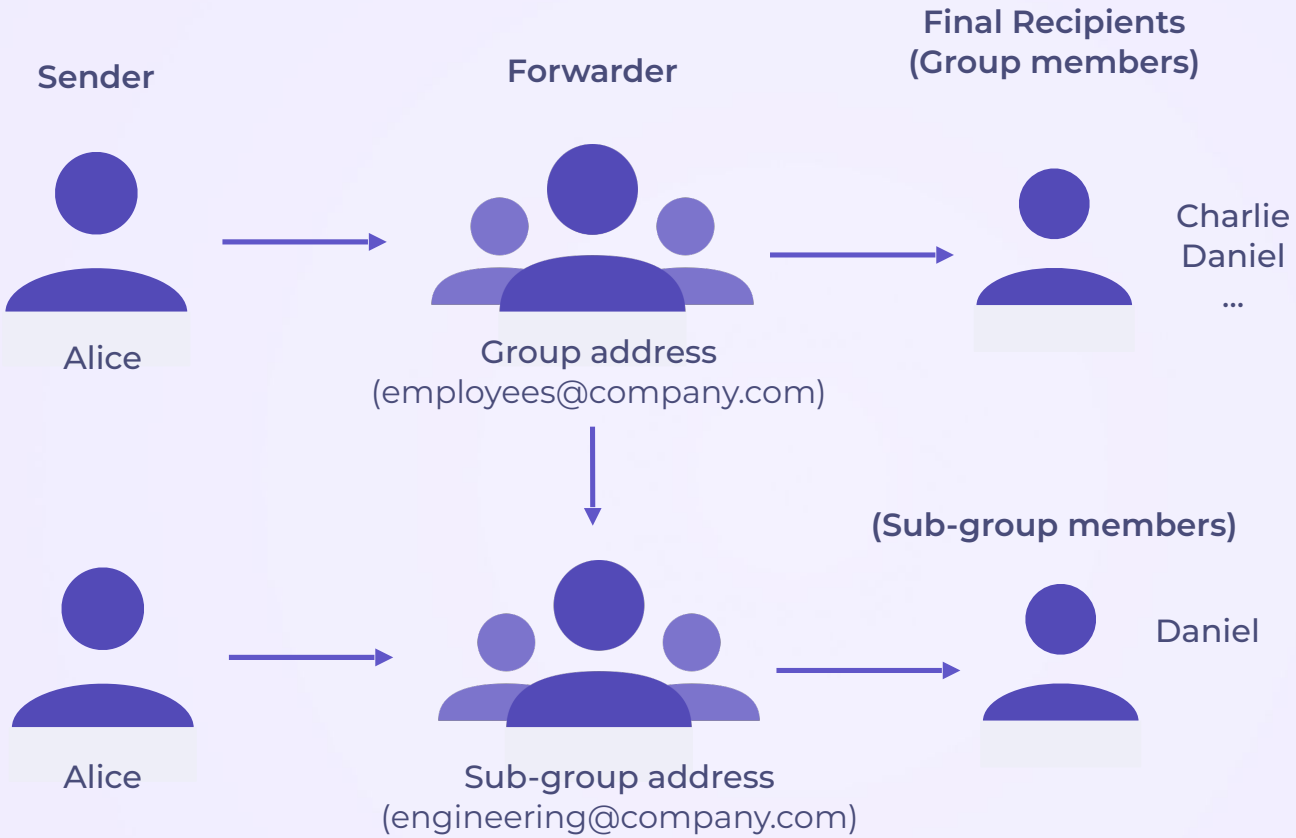
Forwarding chains

We have artificially limited the forwarding chain to 1:

- We're not confident that this will hold with the PQC forwarding scheme
    - Important to avoid feature regression
- Avoid complexity in key distribution
- No loops guarantee

... Or disable E2EE

Nested groups



Nested groups

- Mathematically, the system allows for unlimited levels of recursion
  - Unbounded forwarding chain and nested groups
- Each level of recursion requires a new "decryption subkey" due to the public key binding with the ciphertext KDF
- Groups require dynamic key changes when nesting
- Propagating those changes securely is complex
- We're working only now on nested groups, years later

Standardization effort

- The mechanism is meant to allow forwarding across email providers
  - Third party MTAs could act as forwarding proxies too
  - Mail OpenPGP clients need to add support for the forwarding decryption key format
- After several internal iterations we published an IETF personal draft
  - Main decisions: KDF v2 format, 0x40 key flag
- Working with the community to discuss the modernization approaches

Future work

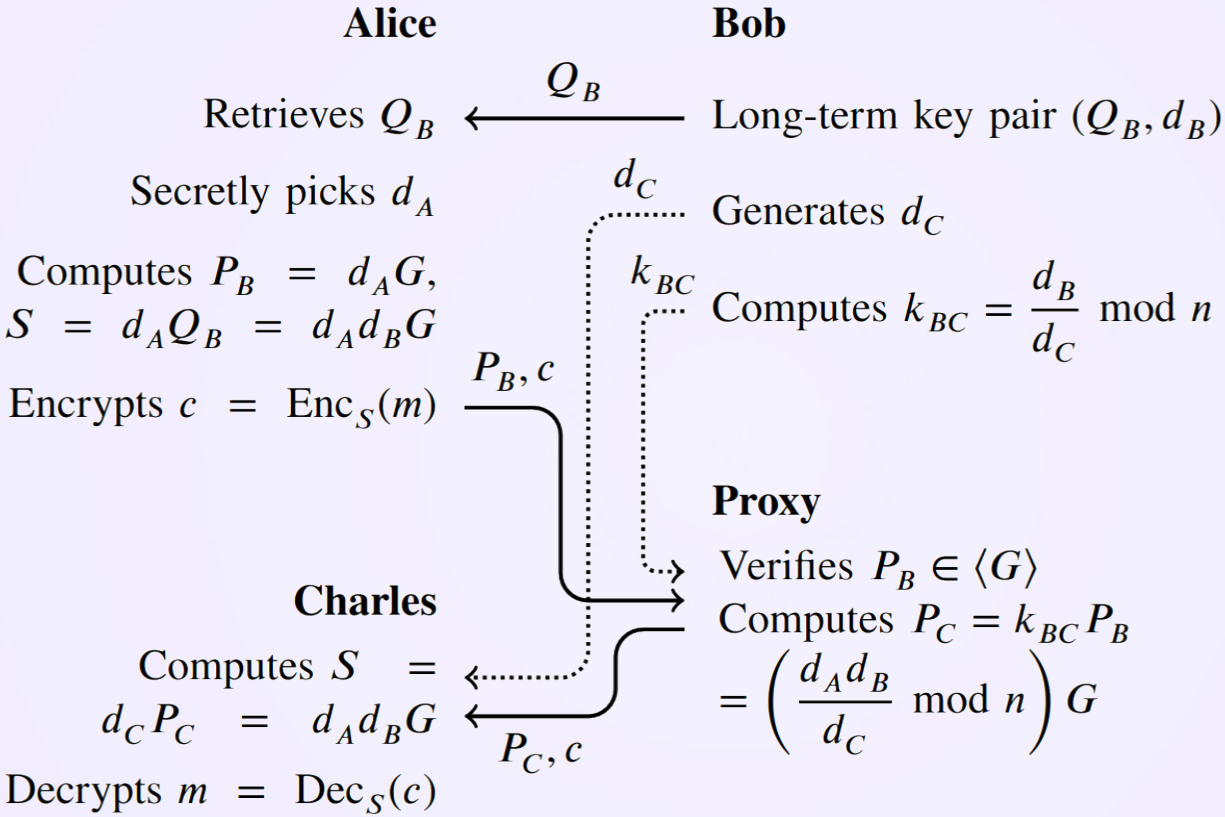
We currently only support forwarding for "Legacy ECDH" OpenPGP keys

- Adaptation for the newly standardized OpenPGP V6 X25519 is fairly straightforward
  - the key format definition will require some work
- PQC integration seems possible [1], but investigation is still at the academic/theoretical level
- Improve support for key rotation mechanism and short-lived keys in OpenPGP
  - Bob's inbox is less exposed if their forwarding key is reconstructed

[1] “*Bounded CCA2 Secure Proxy Re-encryption Based on Kyber*”, *S. Sato, J. Shikata*

Q&A

Proxy re-encryption scheme



Preprint: https://arxiv.org/abs/2211.06992