Thunderbird

Thunderbird Email Security

Plans and Challenges

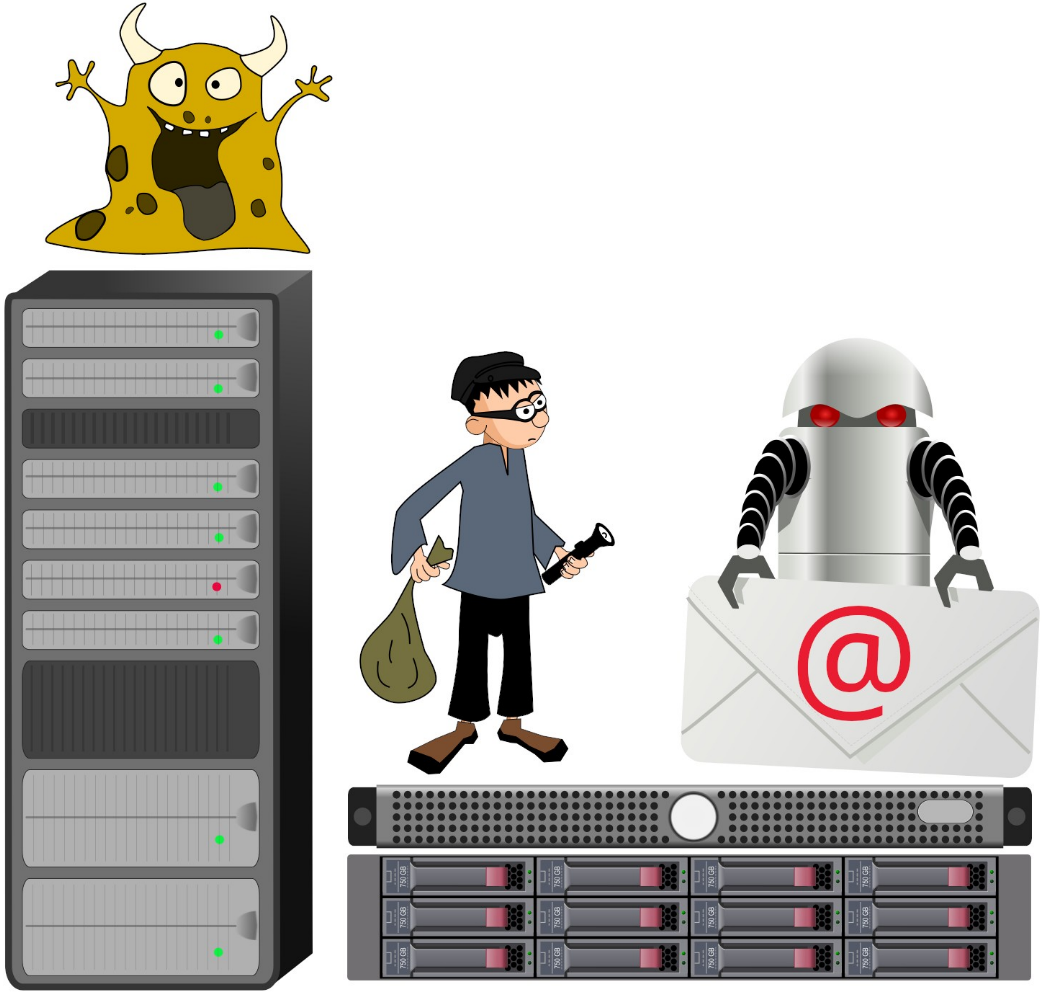
04.02.2024 Kai Engert — Sr. Security Software Developer

Slides v3 from 2024-01-31

Who can access your emails ?

(view or manipulate)

- **Robots living on email servers**
- **Mass-Surveillance Monsters**
- **Cybercriminals**



Images from openclipart.org

No protection while emails are stored on servers.

We need more than TLS Transport security

We need end-to-end (E2E) security
- encryption to achieve confidentiality
- digital signatures, to be certain who sent an email

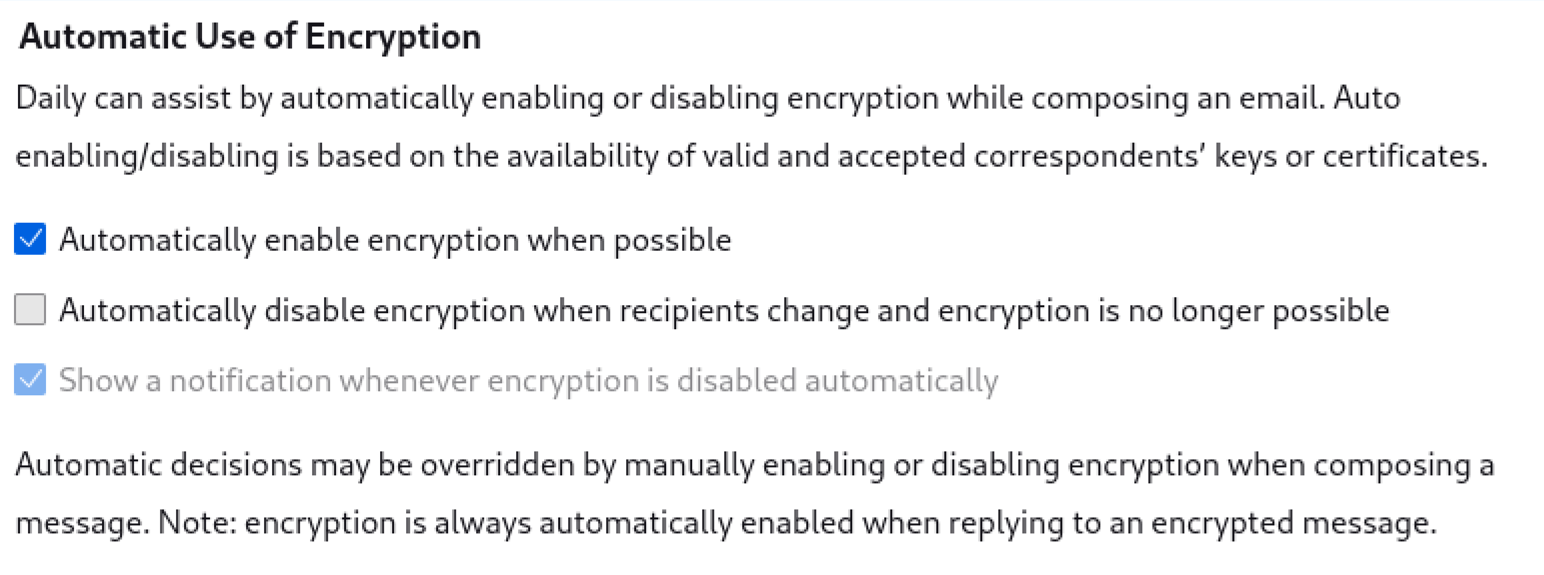
Thunderbird supports two separate E2E technologies:

- S/MIME - (since 2004)
- OpenPGP - (previously Enigmail Add-on, now fully integrated since 2020)

Past Improvements

- Unified status feedback when reading
- Composing: Unified controls to enable/disable encryption
- Composing (OpenPGP): interactive key assistant
- Composing: Reminders if can encrypt

Recently added “Encrypt if possible”



Automatic Use of Encryption

Daily can assist by automatically enabling or disabling encryption while composing an email. Auto enabling/disabling is based on the availability of valid and accepted correspondents’ keys or certificates.

☑ Automatically enable encryption when possible

☐ Automatically disable encryption when recipients change and encryption is no longer possible

☑ Show a notification whenever encryption is disabled automatically

Automatic decisions may be overridden by manually enabling or disabling encryption when composing a message. Note: encryption is always automatically enabled when replying to an encrypted message.

Recent improvements for OpenPGP:

- Secret keys can be protected with their own passphrase, independent of Primary Password. (Still need to a implement a cache.)
- Improved Autocrypt-compatible key distribution headers, including keys of participants in a group conversation (“Gossip”).
- Publishing to keys.openpgp.org

Challenges

- We see emails with mixed technology,
  e.g. OpenPGP message wrapped in
  an outer S/MIME layer (e.g. from G/Suite)
- What to do if digital signature cannot be verified?
  Give feedback about bad status,
  or show no status at all?

Digital signatures with HTML/CSS

- Users want email that looks pretty, not plaintext
- HTML/CSS can manipulate what's shown on screen, when reading and while composing
- Sender and recipient may see different messages, also shown by researchers.
- Show weaker signature status for messages with HTML/CSS ?
- Unresolved problem, looking for suggestions.

Only small portion of emails use S/MIME or OpenPGP.

The technologies aren’t used much, because there are barriers of entry, it’s complicated to manage, and it can have unexpected consequences.

- Difficult to access encrypted email from secondary devices
- Users can lose secret keys and lose access to archive of encrypted email

It’s necessary to involve the user.

- User must be willing to accept the consequences
- User must be willing to take care of the secret key file(s)
  (or agree to lose their archive in the worst case scenario).

Wa want more people to use encryption and signatures

- Full automatism not possible, heterogeneous ecosystem
- We must better assist users.
- Which technology is easier?
- Focus in past years was OpenPGP, is that still a good idea?
- Future of OpenPGP is uncertain,
  because of the problematic LibrePGP fork.
- Conflicting specifications, incompatible implementations and keys, and little hope for a unified specification.
- PGP might become less interoperable
  and more complicated to use.

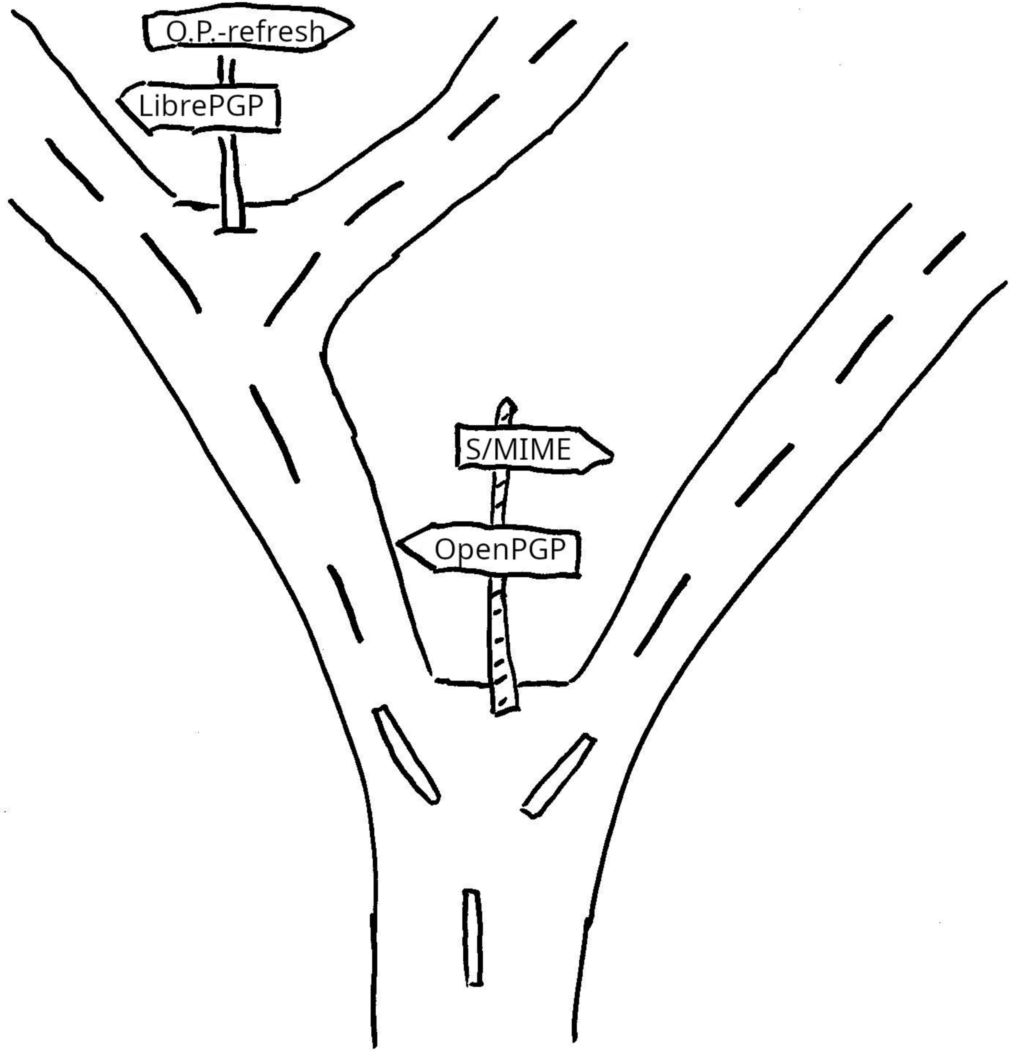
What should we do?

Continue to support both.

Suggestion to change focus:

Make S/MIME easier to use, eliminate entry barriers, declare as preferred technology for users with limited threat model ?

Declare OpenPGP is for users with a broader threat model, who must accept higher complexity ?



S/MIME

- More widely available in email applications.
- If you trust Certificate Authorities (CA), then S/MIME is easier to use than OpenPGP (no manual checking of keys)
- Appropriate for limited threat model, protects against passive reading.
- Remaining risk of falsely issued certificates, e.g. by CAs who get compelled or hacked (see DigiNotar)
- CAs are regularly audited, don’t want to lose their reputation
- The risk of falsely issued certificates might be acceptable for many, but still, the risk remains.

Remove S/MIME barrier of entry?

- Allow everyone to get a certificate for free ?
- Support obtaining (and renewing) a personal email certificate from within the email client.
- Certificate Transparency, using redacted certificates, that contain a hash instead of the email address ?
- Implement certificate directories (like keyservers), using the information from the transparency logs ?

OpenPGP

- Users, who don’t want to accept the risk of falsely issued S/MIME certificates (or OpenPGP keys with false user ID), which means they prefer stronger security over simplicity, can use OpenPGP with **manual** key ownership verification, at the cost of having to learn a more complex technology.
- Making OpenPGP easier to use might become a lower priority.
- OpenPGP related development in Thunderbird might prefer changes that improve security and interoperability.

Thank you!

Slides:
https://kuix.de/fosdem2024