Opportunistically encrypted mailing lists with Mailman and Autocrypt

Contents

1 Project details / administrativia

Name and e-mail address

holger krekel holger@merlinux.eu[1]

Project name

Opportunistically encrypted mailing lists with Mailman and Autocrypt

Requested funding amount in USD

39000

Project term in whole months e.g. 3, 6, 12, or 24

9 months

Endorser's name

[1] mailto:holger@merlinux.eu

Daniel Kahn Gillmor

Address

merlinux gmbH Reichsgrafenstr. 20 79102 Freiburg Germany

2 Project Description

This project is to implement, showcase and guide new, opportunistic approaches for Mailing List encryption, furthering Internet privacy and decentralization. Opportunistic encryption means that mail apps of subscribers which support encryption can send and receive encrypted e-mail without causing unreadable mail for those without encryption support. This is fundamentally different from past approaches which aimed for an all-or-nothing approach.

Mailing lists are typically used in the free/libre and open source software (F/LOSS) community for public discussion. However, they're also used by private groups including small business teams, conference organizers, activists, and others for whom the need for communication privacy outweighs the potential advantages of public communications.

Many of these groups also need threaded, subject-based communication rather than the contact-oriented chat-style messaging which e.g. Signal or WhatsApp offer. Furthermore, privacy sensitive groups often want control over their core communication infrastructure rather than being locked into reliance on centralized services. Finally, opportunistically encrypted mailing lists reduce the information exposed to bulk data collection and help lay the ground for end-to-end encrypted email group communication.

E-mail and the Open Web are at the heart of distributed Internet architectures which give participants agency and privacy regarding their communication, application and content infrastructure. Despite continous doom-saying, e-mail remains the largest open federated identity and messaging ecosystem, anchoring the web and mobile phones. However, encrypted e-mail has failed to see wide adoption outside of specialist communities.

Even within groups where encrypted e-mail has been successfully adopted by most individuals, communication over mailing lists either remains in cleartext (i.e., unencrypted), or is encrypted, but subject to difficult and cumbersome key management and confusing user experience. If we want encrypted e-mail to be something that people can adopt without too much pain, mailing lists need to be included in the discussion. What is needed is not to try harder with the old approaches but to try something different. We indeed work from a new perspective which is well described in RFC7435 ("Opportunistic Security: Some Protection Most of the Time", https://tools.ietf.org/html/rfc7435.html#section-1.2).

Our project will integrate Mailman, the popular mailing list software, with Autocrypt, a fresh approach to opportunistic e-mail encryption. Autocrypt provides both mechanisms and guidance for e-mail application implementers that are interested in offering easy-to-use, privacy-focused encrypted e-mail. Autocrypt dramatically simplifies the user experience of e-mail encryption compared to traditional models, and aims at widespread adoption based on interoperable specifications, a simplified mental model for users, and encouraging a comprehensible and minimal user interface. Among other mail apps, Enigmail and K-9 Mail are currently implementing full Autocrypt Level 1 support. Some providers (e. g. privacy-focused "posteo.de") have added support for securing Autocrypt headers.

We are heading for a design and an implementation of opportunistically-encrypting mailing lists with little or no configuration needed. We want to avoid requiring extra administrative efforts from both operators and users of mailing lists. This is strongly related to Autocrypt's design goal of "Don't ask users about keys, ever!". Concretely, if a subscriber's e-mail client supports Autocrypt, e-mail messages from the client to the mailman service (and vice versa) can be encrypted in some cases, rendering the content unreadable for a subscribers's e-mail provider. In such cases, the encrypted messages will be better protected from message interception and bulk data collection.

User experience, simple and clear mental models, and incentives that enable private communication are a major component of this work. For example, each mailing list message sent out by Mailman could indicate the "encryption status" of the list and of the message itself in the footer and/or in headers of each message. This could include the overall encryption/cleartext recipient ratio and whether the message was properly encrypted by the original sender. The precise semantics will be determined through community discussions and in particular during a multi-day hackathon. We plan to invite and involve interested parties from Mailman, Autocrypt and related communities to discuss and produce guidance on Autocrypt-enabled opportunistic mailing list modes.
We will also operate a best-practice-driven mailman server with new opportunistic encryption modes. Through such real-world usage we gather insights and feedback which then flow back into refining our developments.

The project is implemented by professional F/LOSS community developers who have founded and maintained several successful projects (for example pytest, tox and PyPy, all in use by Mozilla internally) and is supported by key community members of the Mailman and Autocrypt efforts. Project team members work together for years and have successfully delivered on many non-profit and commercial proposals.

A Mozilla grant for our project would significantly help to pave the way for privacy-sensitive organizations to achieve better protected group communications. It would also support small independent providers to offer best-practise mailing list servers to their users. Last but not least, it would help to grow the movement around Autocrypt as a new promising way for more pervasive e-mail encryption on the Internet.

3 Outcomes and how they further Mozilla's mission

The Opportunistic Mailing list Encryption project will run for nine months and produce three outcomes:

1. Released F/LOSS code for Mailman3 / Autocrypt integration, facilitating a new experimental opportunistic encryption mode which encrypts mails between the mailman service and e-mail clients, and which shows group and message encryption status/capability in the footer of each message. The released code consists of the "muacrypt" package, and a plugin and/or patches for mailman3 (to be determined during project).

Estimated costs: $24,000.

2. Published documentation on how to setup and operate a best-practice mailman3 server on top of Debian 9, Postfix, letsencrypt and nginx. The documentation will be an enhanced version of the existing template for the already operative "lists.codespeak.net" service. For beta-test and gathering user-feedback, we will offer select groups use of the codespeak.net service with the experimental mailing list encryption mode.

Estimated costs: $8,000.

3. Organizing a multi-day gathering for developers and projects interested in encrypted mailing list communications, to result in a report with guidance on opportunistic encryption for re-mailers as well as the mailman implementation itself.

Estimated costs (travel, location, writing): $7,000.

These outcomes and our project's activities relate to many of the 10 Mozilla Manifesto principles with particular focus on strengthening

- decentralization, privacy and security of core Internet communication infrastructure (principles 01, 04, 06)
- transparent community-based practices to promote participation with and trust (principles 07, 08) into open source, e-mail Internet communication infrastructure.

As to technical details of our main code outcome, new opportunistic modes through Mailman and Autocrypt integration, we consider the following noteworthy for reviewers:

- Autocrypt specifies how to generate an Autocrypt key, transfer and parse public keys and settings through headers of regular e-mails. Sending encrypted mails by CCing group members is made easier through the Autocrypt-Gossip header but requires a sender to know all recipient e-mail addresses and Autocrypt keys. However, Ccing everyone does not fit the project goal of allowing group membership to change over time and still have threaded discussion (users should not need to "reply all" to start a new thread).

- For the opportunistic mailing list encryption mode, a Mailman plugin (and/or mailman core-integrated code) will create a per-list Autocrypt account managed through enhancing the "muacrypt" project. Muacrypt keeps track of a subscriber's encryption settings by parsing incoming mail messages and determining the "encryption status" for each outgoing message. No special interface is needed on the mailing list web page and no extra duties need to be performed by the mailing list operator or users.
- A crucial design challenge occurs when a mailing list has a mixed set of subscribers (some with and some without Autocrypt support). In this case, Mailman could send out encrypted and unencrypted mails respectively, or it could send encrypted mails only to those with Autocrypt support. Morover, including an "encryption status" information in each message's footer can help interested subscribers to understand the situation and can provide incentives to upgrade their e-mail program. Encryption status information can also be added and specified in message headers such that an e-mail client can programmatically understand and display the encryption status.
- Some mailing lists are set to be publicly archived directly by Mailman. This configuration choice may not be compatible with the goal of protecting the message contents from an external eavesdropper. This project will explore the interaction between these settings and propose reasonable and opinionated guidance.
- With the current Autocrypt specification, a Mailman list will need to have the DMARC-mitigation "replace-from" action enabled to a) allow the list's public key to be added to outgoing mails in an Autocrypt-compliant way, and b) send properly signed and encrypted mails to those subscribers receiving encrypted mails.
- Schleuder (https://schleuder.nadir.org/) is an existing solution for encrypted mailing lists that works with key servers and requires expertise and maintenance both from list administrators and users to maintain key consistency. We are in good contact with Schleuder developers and will invite them (Outcome 3) to share and discuss on how to introduce Autocrypt to re-mailer software.
- Our approach does not remove the need to trust the mailing list service operator but Outcome 2 aims at spreading the knowledge to operate and maintain mailing list services. We will include support for OnionMX (https://github.com/ehloonion/onionmx), a new practice for allowing e-mail servers to relay messages to each other by routing them through the

Tor network.

- The outcomes of our project will likely help discussions around evolving the Autocrypt specification to support mailing list modes which do not replace the From: header but such guidance itself is not part of our project outcomes.

4 endorsement (Daniel Kahn Gillmor)

Encrypted e-mail has been available for decades, but never widely used, due to a combination of usability and maintenance problems. The Autocrypt project is making great headway in providing guidance for encrypted e-mail that is actually usable and maintainable, though it doesn't have the same security properties we've traditionally expected from encrypted mail. The Autocrypt approach is both opportunistic (see https://tools.ietf.org/html/rfc7435) and opinionated – making it straightforward for non-technical users to do the most reasonable thing in a mixed environment where communications confidentiality isn't always possible.
While the Autocrypt project has a functional handle on end-to-end encryption of directly-sent e-mail messages, it doesn't yet have a good story for messages sent via mailing lists or other re-mailing engines. The current proposal aims to bridge that gap and provide not only guidance but a thoughtful, user-centric implementation that will integrate Autocrypt's vision of simplicity and usability with one of the most widely-used mailing list platforms we have available today. While not every mailing list specifically needs this functionality (e.g. some lists are publicly archived and indexed already), it ought to be readily available for those lists which *do* need it. The mixture of F/LOSS implementation and guidance are important contributions to the Internet as a whole, moving this critical part of our shared infrastructure in a healthier direction. I have read the text of this application and I believe the applicants are technically capable of achieving the work outlined. I hope they get the chance to do this work.

5 how the project is managed / core-team

The following developers are planned to be part-time contracted though Merlinux, bringing their knowledge and experience to the effort:

- Holger Krekel, founder of Merlinux, co-founder of the Autocrypt effort and core author of several Open Source projects (including pytest, tox, PyPy all used by Mozilla internally).
- Florian Schulze, who led the initial prototyping of the community-run mail server set-up at https://lists.codespeak.net with initial documentation at https://github.com/codespeaknet/sysadmin/blob/master/docs/notes.rst

and also is the core maintainer of https://devpi.net, an open source system for managing Python Packaging workflows,

- Azul, co-founder of the Autocrypt effort, co-authoring Muacrypt and involved in the LEAP Encryption Access Project (which received MOSS funding in 2016).

All three have been working together for years and have already established good practises of collaborating on joint projects. These practises include a review-based work flow, regular IRC and A/V meetings as well as meeting in person several times per year. In addition to the people mentioned here we also plan to invite and involve key developers from re-mailer projects, paying for their travel and accomodation through this grant. It's also possible that we hire experienced developers from the communities we are integrating with.

merlinux is a 14-year old limited liability company based in Freiburg, Germany, which has successfully delivered on Open Source group development proposals for many non-commercial and commercial entities.

6 community

We are substantially involved in the following existing open-source projects and communities:

- https://autocrypt.org – the community website for the Autocrypt specification, also listing current implementation efforts of the Thunderbird-extension Enigmail, K-9 Mail for Android and Delta.Chat, a new telegram-style messenger which uses e-mail addresses and providers instead of operating own centralized servers.
- https://list.org – the GNU Mailing list manager ("Mailman") in wide use across the Internet. Its founder and long-time maintainer, Barry Warszaw, supports our effort.
- https://muacrypt.readthedocs.io – an ongoing effort by Holger Krekel and azul which implements a command line tool and a Python Interface to achieve Autocrypt support for Mail program setups. The major software release deliverables of the Opportunistic Encrypted Mailman Autocrypt mode will majorly consist in enhancing muacrypt.
- https://lists.codespeak.net – a best-practise setup of Mailman on Debian 9, inaugurated by Florian Schulze through the NEXTLEAP EU research project on decentralized messaging. The live-running instance is documented here: https://github.com/codespeaknet/sysadmin/blob/master/docs/notes.rst

Our three projected outcomes all involve and am to grow the communities behind these projects. We are confident that our efforts will continue and be taken up after the Mozilla grant duration.

7 guidelines / CoC

https://www.python.org/psf/codeofconduct/