EUROPEAN UNION AGENCY
FOR CYBERSECURITY

# SECURE GROUP COMMUNICATIONS

for incident response and operational communities

JULY 2019

# ABOUT ENISA

The European Union Agency for Cybersecurity (ENISA) has been working to make Europe cyber secure since 2004. ENISA works with the EU, its member states, the private sector and Europe's citizens to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. Since 2019, it has been drawing up cybersecurity certification schemes. More information about ENISA and its work can be found at www.enisa.europa.eu.

## CONTACT

For contacting the authors please use CSIRT-Relations@enisa.europa.eu
For media enquiries about this paper, please use press@enisa.europa.eu.

## LEGAL NOTICE

Notice must be taken that this publication represents the views and interpretations of ENISA, following the consultation of experts in this area. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013.
This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

## COPYRIGHT NOTICE

Catalogue number: TP-03-19-540-EN-N, ISBN: 978-92-9204-303-2, DOI: 10.2824/062571

# CONTENTS

# EXECUTIVE SUMMARY

As of July 2019 there are more than 414 incident response teams in Europe[1]. These teams work together to respond to cyber-attacks and need to use secure and reliable communication channels to share threat and incident information while protecting European citizens and businesses. With a number of cyber security incidents and an attack surface that increase every day, spanning from large infrastructures to the end users, there is the need to improve operational cooperation, preparedness and information exchange by promoting the use of complete and scalable secure communication solutions.

This document provides an overview of available solutions best known at the time of writing. Intentionally, it does not provide a conclusion that can be directly applied to all communities. However, it can serve as a starting point for a tailored evaluation.

The process works by first identifying available solutions best known at the time of writing of this report. The initial requirements were defined based on the use case of a group of incident response teams forming a decentralised community. The document provides and prioritizes criteria for the selection of available solutions for this scenario. The first round of chosen criteria were: secure group communication, archive, attachments, open specification, license, availability for all operating systems and maturity. It is important to underline that this report does not address the quality of encryption. After listing products along these criteria in a first round, available solutions scoring in all criteria are descripted. The secondary selection criteria used were: hosting, add-on compatibility, compatibility with existing infrastructure, backward compatibility and forward secrecy.

During the project, it has been found that various solutions best known at the time of writing did not fit all criteria either because, at the time of writing, they do not provide the required end-to-end encryption, they did not provide mature multi-platform support, they were not offering a secure group chat or they were missing an open specification. On closer examination of the various solutions and their evaluation with encrypted mailing lists, it became clear that having a message archive as well as forward secrecy is a challenge. Due to forward secrecy, in many end-to-end encrypted solutions, only existing members can access the full history, but newly added members can only access from the point when they have been added.

Based on this, a parallel setup of a chat solution and an encrypted mailing list seems the most fitting to cater both synchronous and asynchronous communication needs for the scenario of group of incident response teams forming a decentralised community considered in this document.

[1] ENISA CSIRTs by Country - Interactive Map https://www.enisa.europa.eu/csirts-map

# 1. INTRODUCTION

Computer Security Incident Response Teams (CSIRTs) around the world deal with security events, such as malware outbreaks or vulnerability discoveries, etc. Incident response teams are often organized in communities such as CSIRTs Network[2], TF-CSIRT[3], FIRST[4] and other regional, sub regional or sectorial communities. Typical information exchanged among teams include threat intelligence, indicators of compromise (IoCs), malware samples and details about relevant incidents.

To facilitate information exchange among teams and improve reaction time to security incidents, tailored communication solutions are required. These teams are often organized in groups forming a decentral community that needs to cooperate and have secure and reliable communication channels to share information.

**This document serves as a starting point for other incident response communities to conduct their own evaluation and see how these tools can fit their sizes and needs.**

The type of work and decentralized organisational structure of these communities impose tough requirements on the chosen communication solutions. First and foremost, solutions must implement end-to-end encryption protocols for group messaging because highly sensitive information is exchanged. Thus, to reduce the amount of required trust in providers, solutions must implement end-to-end encryption with verifiable keys defined in an open specification. To allow archive of previous incidents lesson learnt, these solutions must provide a way to archive conversations and storage of attachments. Finally, on premise hosting and the selection of free software allows independent operation and extensibility by the managing member of the community.

Recent cryptographic reports such as the ECRYPT 2018" Algorithms, Key Size and Protocols Report" [5] focuses on fundamental algorithms and protocols. Previous ENISA work on the topic was 2014 "Algorithms, key sizes and parameters report"[6] and "Study on cryptographic protocols"[7] Other projects, such as "Applied Crypto Hardening: bettercrypto.org"[8] provide good-practice configurations for server administrators. In contrast, this study focuses on real-world communication solutions for CSIRTs and it does not address the quality of encryption.

This project on secure communication solutions has been conducted with a specific community and scenario in mind. This community could be a group of incident response teams forming a decentral community or an operational community grouped in an information sharing and analysis centre (ISAC). This model community already have in place chat, encrypted email and a shared secure space on the web, where to share information, like many existent communities. The idea is to move from a set of tools and systems, created over time, to a more scalable and integrated set of tools. On this baseline, the document follows a methodology for evaluating best-known solutions at the time of writing. It explicitly does not provide results that can be automatically re-used for different communities or use case. However, it serves as a starting

[2] CSIRTs Network http://csirtsnetwork.eu/
[3] TF-CSIRT https://tf-csirt.org/
[4] FIRST - Forum of Incident Response and Security Teams https://www.first.org/
[5] Algorithms, Key Size and Protocols Report (2018) www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf
[6] ENISA, "Algorithms, key size and parameters report", 2014 https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014
[7] ENISA, "Study on cryptographic protocols", 2014, https://www.enisa.europa.eu/publications/study-on-cryptographic-protocols
[8] Wolfgang Breyha, David Durvaux, Tobias Dussa, L. Aaron Kaplan, Florian Mendel, Christian Mock, Manuel Koschuch, Adi Kriegisch, Ulrich Pöschl, Ramin Sabet, Berg San, Ralf Schlatterbeck, Thomas Schreck, Alexander Würstlein, Aaron Zauner, Pepi Zawodsky "Applied Crypto Hardening: bettercrypto.org", version 1.x, 2018-12-21, https://bettercrypto.org/

point for other operational communities to conduct their own evaluation and see how these tools could fit their sizes and needs.

## 1.1 PREVIOUS ENISA WORK ON THE TOPIC

Since 2005, ENISA has been supporting Member States and CSIRT communities in EU to build and advance their incident response capabilities with handbooks, online & onsite trainings and dedicated projects. ENISA's portfolio of work is related to setting up, running or developing capabilities of Computer Security Incident Response Teams (CSIRTs). The goal is to define minimum common baseline practices across the EU to improve operational cooperation, preparedness and information exchange for the next generation of cyber-attacks. More info can be found at https://www.enisa.europa.eu/csirt-services

Relevant ENISA efforts are:

- Reference Security Incident Taxonomy Working Group[9]
- Exploring the opportunities and limitations of current Threat Intelligence Platforms[10]
- Actionable Information for Security Incident Response[11]
- Detect Share Protect - Solutions for Improving Threat Data Exchange[12]
- Proactive Detection of Network Security Incidents – Honeypots[13]
- Proactive Detection of Network Security Incidents – Data feeds – internal and external[14]

Moreover, the following relevant trainings are also available for free download:

- Proactive incident detection: handbook and VM[15]
- Automation in incident handling: handbook and VM[16]
- Honeypots: handbook and VM[17]
- Presenting, correlating and filtering various feeds: handbook and 2 VMs[18]

## 1.2 METHODOLOGY

This document presents an example of how to select suitable candidates from a large number of best known software solutions and see how they can fit the needs of the members of a specific incident response community. The key is to find many potential solutions and limit them quickly step by step.

One task is to find out about type of users to aim for. This means asking a number of questions, such as:

[9] Reference Security Incident Taxonomy Working Group  - RSIT- WG https://github.com/enisaeu/Reference-Security-Incident-Taxonomy-Task-Force

[10] ENISA, "Exploring the opportunities and limitations of current Threat Intelligence Platforms", 2018, https://www.enisa.europa.eu/publications/exploring-the-opportunities-and-limitations-of-current-threat-intelligence-platforms

[11] ENISA, "Actionable Information for Security Incident Response", 2015, https://www.enisa.europa.eu/publications/actionable-information-for-security

[12] ENISA, "Detect Share Protect - Solutions for Improving Threat Data Exchange", 2013, https://www.enisa.europa.eu/publications/detect-share-protect-solutions-for-improving-threat-data-exchange-among-certs

[13] ENISA, "Proactive Detection of Network Security Incidents – Honeypots", 2012, https://www.enisa.europa.eu/publications/proactive-detection-of-security-incidents-II-honeypots

[14] ENISA, "Proactive Detection of Network Security Incidents – Data feeds", 2011, https://www.enisa.europa.eu/publications/proactive-detection-report

[15] ENISA, "Proactive incident detection training", https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational#proactive-incident-detection

[16] ENISA, "Automation in incident handling training", https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational#automation_incident

[17] ENISA, "Honeypots training", https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational#honeypots

[18] ENISA, "Presenting, correlating and filtering various feeds training",https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/technical-operational#presenting--correlating-and-filtering-various-feeds

- What communication workflows are already well established?
- What solutions are in use? What experience background do people have with them?
- What type of support the different users provide? For example hosting, budget or training?
- How important are the different aspects of the solutions? For example in communication speed or attachment size?

The second task is to find and weight potential solutions. For this, a variant of David A. Wheeler's IRCA approach[19] was used where IRCA stands for

- **I**dentify candidates
- **R**ead existing reviews
- **C**ompare the leading programs' basic attributes to your needs
- **A**nalyze the top candidates in more depth.

In the initial discovery phase, overview articles, search engines and Wikipedia entries were used to find candidates[20]. In the review phase, credible third party descriptions or assessments from experts in the area were used to get an overview of the claimed properties of a solution. The first list of findings was reviewed in the light of the particular use case and refined.

**The main goal of this exercise is to find solutions capable of allowing secure communications among incident response and operational communities.**

## 1.3 CRITERIA

In this section, the evaluation criteria are presented. As discussed in the introduction, it is important to note that this study’s focus is on incident response teams forming a decentralised community. Other scenarios may require different criteria and will most probably lead to a different prioritization and outcome.

The following seven criteria have been identified as important:

- **Secure Group Communication:** The main goal of this evaluation is to find solutions capable of allowing secure communications among teams dealing with security incidents, such as CSIRTs. Thus, secure group communication is the main requirement for all discussed solutions. One specific requirement in this study is that the intended solution provides communication groups with a certain level of security, authenticated members and a key management where users are associated to cryptographic identity keys. Furthermore, there must be a way to verify relationships between identity keys and users via out-of-band verification schemes. Conclusively, a simple TLS-based in-transit-encryption is not sufficient for the intended use case of CSIRT communication.
- **Archive:** An archive must be provided to look up previous discussions. Furthermore, it must be possible to add new members to an existing group discussion and allow them access to the message history. This is a difficult criteria for secure communication groups. In many encryption solutions, only existing members can access the full history, but newly added members can only access from the point when they have been added. Furthermore, for CSIRTs, it should be possible to have functional (non-personalized) mailboxes/accounts.
- **Attachments:** To exchange files, such as documents, source code, or IOCs, the solution should provide the ability to exchange attachments of any file type.

[19] David A. Wheeler, How to Evaluate Open Source Software / Free Software (OSS/FS) Programs, Revision August 5, 2011, https://dwheeler.com/oss_fs_eval.html

[20] For example:
https://en.wikipedia.org/wiki/Comparison_of_instant_messaging_clients
https://docs.google.com/spreadsheets/d/1-UlA4-tslROBDS9IqHalWVztqZo7uxlCeKPQ-8uoFOU
https://www.eff.org/node/82654
https://systemausfall.org/wikis/howto/CryptoMailingLists

- **Open Specification: The communication protocol/solution should be specified in an open way to allow auditing of security features and reliability requirements.** This requirement is explicitly not about end-user documentation, but **documentation about the architecture**.
- **License:** To allow auditing the code by independent reviewers, the source code of the intended communication solution should be licensed under a Free Software license approved by the Free Software Foundation (FSF)[21] or the Open Source Initiative (OSI)[22] (CIRCL, 2018). Popular licenses include Apache v2[23], GNU GPL[24] and Xorg-Style (aka MIT)[25]. Closed source solutions will be annotated as “proprietary”. Because some solutions provide encryption capabilities as add-ons for proprietary products, this requirement is relaxed in these cases, to only cover the add-on. When there are multiple implementations of a solution with different licenses, but Free Software Solutions exist for all operating systems, these solutions are annotated with “Free Software”. This is especially true for many open messaging specifications with multiple clients.
- **All Operating Systems (OS):** The solution should be available on all major desktop operating systems (Windows, Mac OS, GNU/Linux) and mobile operating systems (iOS, Android). This requirement is only applicable for client software.
- **Maturity:** Due to its long-term operation as one important communication medium among CSIRTs, the intended solution should be “future proof”. While this criteria is difficult to assess, the authors consider a finalized protocol specification, clients released as stable versions and a stable business model as indicators for a project’s maturity.

The remaining criteria have been identified as secondary selection criteria. These often require more detailed discussions and may not be answered easily. Furthermore, some make only sense for email-based solutions.

- **Hosting:** It is differentiated between completely *decentralized*, *federated*, *centralized*, and centralized but hosted *on-premise* solutions. While completely decentralized solutions form a communication network using distributed data structures shared by all clients, federated solutions require a number of hosted nodes that process and distribute messages hosted by participating CSIRTs. For centralized solutions, a special attention is given to on-premise solutions that can be hosted by the managing organization. In addition, for centralized solutions, the legal base should be in the EU or EFTA countries.
  In general, more decentralized solutions are preferred for inter-CSIRT communication.
- **Add-on Compatibility:** Email-based solutions should allow an integration into best known email programs at the time of writing, to be adopted by end-users. This criteria is not relevant for standalone clients.
- **Compatibility with Existing Infrastructure:** This shall just describe if there is an easy integration with already existing administration infrastructure, like a directory service.
- **Backward Compatibility:** Older conversations, available in archives, must be still accessible years after their creation. For encrypted communication, this means that older identity keys should be available or messages were re-encrypted.

[21] Free Software Foundation (FSF) https://www.fsf.org
[22] CIRCL Computer Incident Response Center Luxembourg, CSIRT Tooling: Best Practices in Developing, Maintaining and Distributing Open Source Tools, 2018-11-06, https://github.com/CIRCL/compliance/blob/master/csirt-tooling-best-practices/index.md
[23] Apache v2 https://www.apache.org/licenses/LICENSE-2.0
[24] GNU GPL  https://www.gnu.org/licenses/gpl-3.0.en.html
[25] Xorg-Style MIT https://cgit.freedesktop.org/xorg/xserver/tree/COPYING

- **Forward Secrecy:** Due to the popularity of Double-Ratchet-based protocols[26], such as the Signal Protocol[27], OMEMO[28] and others, the property of forward secrecy received a lot of attention in recent years in the Privacy Enhancing Technologies community. An attacker compromising the keys at a specific point in time should not be able to decrypt *previously* recorded encrypted communication from the past. Note that "perfect" forward secrecy cannot be achieved in an asynchronous protocol, because an online connection is required to execute a key agreement for a new ephemeral key that does not depend on a previous one. In addition to forward secrecy, the opposite direction has recently been defined as "post-compromise security" because the previous terms "backward secrecy" or "future secrecy" were confusing. An attacker compromising keys at a specific point in time should not be able to decrypt future communication, i.e., the protocol should be "self-healing".

[26] Key management algorithm that was developed by Trevor Perrin and Moxie Marlinspike in 2013. It can be used as part of a cryptographic protocol to provide end-to-end encryption for instant messaging. After an initial key exchange it manages the ongoing renewal and maintenance of short-lived session keys. It combines a cryptographic ratchet based on the Diffie–Hellman key exchange (DH) and a ratchet based on a key derivation function (KDF) like e.g. a hash function and is therefore called a double ratchet. https://en.wikipedia.org/wiki/Double_Ratchet_Algorithm

[27] Signal Protocol https://signal.org/docs/

[28] OMEMO https://conversations.im/omemo/

# 2. OVERVIEW OF SOLUTIONS

First, a list of communication solutions best known at the time of writing is created to provide a coarse-grained overview. This is done by evaluating solutions best known at the time of writing with respect to the seven criteria identified as important selection criteria. After the first step, individual solutions that fulfil with all important criteria are filtered into a second step. These are discussed in detail, also with respect to secondary criteria.

## 2.1 SOLUTIONS NOT COVERED

While it is acceptable that non-email solutions require the installation of new clients, email-based solutions should integrate with the teams existing email infrastructure. For the purpose of this study email-based solutions that are hosted in the cloud and require the users to create new email accounts were not covered. These do not integrate with existing email accounts and are thus not applicable to the scenario envisioned in this study.

## 2.2 INDICATORS AND ABBREVIATIONS

If not otherwise specified inside the tables, the following indicators and abbreviations are used throughout this assessment:

| | |
|---|---|
| ● | full |
| ○ | no support |
| ○/● | partial support |
| ? | unclear |
| N/A | not applicable |

## 2.3 FIRST STEP

For a better overview, solutions best known at the time of writing are categorized into coarse classes: Open Messaging Specifications, Central Messaging, Messengers, Encrypted Email Mailing lists, Email Encryption Gateways.

Obviously, there is some overlap among all these categories. The most prominent example is email communication, which is based on Internet Engineering Task Force - IETF[29] standards, such as Simple Mail Transfer Protocol (SMTP)[30] and Internet Message Access Protocol (IMAP)[31], i.e. it is an Open Messaging Specification. Still, email can be used in a large number of different setups and software configurations. Thus, there is a differentiation between different email setups in their own categories.

### 2.3.1 Open Messaging Specifications

Solutions best known at the time of writing based on openly specified standards. These typically have a wide range of different clients, as any developer is allowed to implement these specifications.

[29] Internet Engineering Task Force https://www.ietf.org/
[30] Simple Mail Transfer Protocol (SMTP) https://tools.ietf.org/html/rfc5321
[31] Internet Message Access Protocol (IMAP) https://tools.ietf.org/html/rfc3501

**Table 1:** Overview of open messaging specifications

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| IRC | https://ircv3.net | ○/● (opt. OTR[32]) | ● (via bouncer) | ● not encrypted | ● | Free Software | ● | ○ (no encrypt., v3 in dev) |
| Kontalk (based on XMPP) | https://kontalk.org | ● (OpenPGP [33]) | ○ | ● | ● | GPLv3 | ● | ○ (no desktop clients) |
| Matrix | https://matrix.org | ● | ● | ● | ● | Apache v2 (Riot) | ● | ● |
| PSYC1 | https://psyc.eu | ○/● (OTR) | ● | ● not encrypted | ● | Free Software | ○ | ○ (PSYC2 in dev) |
| Ricochet | https://ricochet.im | ● | ○ | ○ | ● | BSD[34] | ○ | ○ |
| Tox | https://tox.chat | ● | ○ | ● | ● | Free Software | ● | ○ |
| XMPP | https://xmpp.org | ● (OTR / OMEMO) | ○/● (XEP-313[35]) | ● (XEP-363[36]) | ● | Free Software | ● | ● |

## 2.3.2 Central Messaging

Solutions best known at the time of writing that typically provided as on-premise or SaaS solutions, these messaging systems provide unified access and easy on-boarding.

**Table 2:** Overview of central messaging solutions

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| Discord | https://discordapp.com | ○ | ● | ● | ○ | proprietary | ● | ● |
| Flock | https://flock.com | ○ | ● | ● | ○ | proprietary | ● | ● |
| Gitter | https://gitter.im | ○ | ● | ● | ● | MIT | ● | ● |
| Keeperchat | https://keeperchat.com | ● | ○ | ● | ○ | proprietary | ● | ● |
| Keybase | https://keybase.io | ● | ● | ● | ● | Client: BSD | ● | ● |
| Mattermost | https://mattermost.com | ○ | ● | ● | ● | MIT/propr | ● | ● |
| NextCloud Talk | https://nextcloud.com/talk | ○/● | ● | ● | ○ | AGPL[37] | ● | ○ |
| Rocket | https://rocket.chat | ○/● (OTR) | ○/● | ● not encrypted | ● | MIT | ● | ○ (Better E2E encr. in dev) |

[32] Off-the-Record Messaging Protocol version 3 https://otr.cypherpunks.ca/Protocol-v3-4.1.1.html
[33] OpenPGP https://www.openpgp.org/about/standard/
[34] BSD https://en.wikipedia.org/wiki/BSD_licenses
[35] XEP-0313 https://xmpp.org/extensions/xep-0313.html
[36] XEP-363 https://xmpp.org/extensions/xep-0363.html
[37] AGPL https://en.wikipedia.org/wiki/Affero_General_Public_License

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Slack | https://slack.com | ○ | ● | ● | ○ | proprietary | ● | ● |
| Spectrum | https://spectrum.chat | ○ | ● | ● | ○ | proprietary | ○ | ○ |
| Zulip | https://zulipchat.com | ○ | ● | ● | ○ | Apache | ● | ● |

### 2.3.3 Messenger

Solutions best known at the time of writing that, coming from a user experience perspective, focus on mobile first, nowadays messengers dominate the market for private users.

Table 3: Overview of modern messengers

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| Babelnet | https://www.babelnet.com | ● | ○ | ● | ○/● (XMPP[38] based) | proprietary | ○ | ○ |
| Black Berry Messenger | http://bbm.com/en/ | ○ | ○ | ● | ○ | proprietary | ○ | ● |
| Briar | https://briarproject.org | ● | ○ | ○ | ○ | GPLv3 | ○ | ○ |
| DeltaChat (based on email) | https://delta.chat | ● | ○/● | ● | ● | Free Software | ○ | ○ |
| Facebook Messenger | https://www.messenger.com | ● | ○ | ● | ○ | proprietary | ● | ● |
| Gadu-Gadu | https://www.gadu-gadu.pl | ○ | ○ | ● | ○ | proprietary | ● | ○ |
| ICQ | https://icq.com | ○ | ○ | ● | ○ | proprietary | ● | ● |
| iMessage | https://support.apple.com/explore/messages | ● | ○ | ○ | ○ | proprietary | ○ | ● |
| Jami | https://jami.net/ | ● | ○ | ● | ○ | GPLv3 | ● | ○ |
| KakaoTalk | https://www.kakaocorp.com | ○ | ○ | ● | ○ | proprietary | ● | ○ |
| Line | https://line.me | ● | ○ | ● | ○ | proprietary | ○ | ● |
| Signal | https://www.signal.org | ● | ○ | ● | ○ | GPLv3 | ● | ● |
| Skype | https://www.skype.com | ○ | ○ | ○ | ○ | proprietary | ● | ● |
| Surespot | https://www.surespot.me | ○ | ○ | ● | ○ | GPLv3 | ○ | ○ |
| Telegram | https://telegram.org | ○ | ● (super-groups) | ● | ● | GPLv2 | ● | ● |
| Tungsten | https://tungsten-labs.com | ● | ○ | ● | ○ | proprietary | ● | ○ |
| Threema | https://threema.ch | ● | ○ | ● | ○ | proprietary | ● | ● |
| Viber | https://www.viber.com | ● | ○ | ● | ○ | proprietary | ● | ● |

[38] XMPP https://xmpp.org/

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Whatsapp | https://www.whatsapp.com | ● | ○ | ● | ○ | proprietary | ● | ● |
| Wickr | https://wickr.com | ● | ○ | ● | ○ | proprietary | ● | ● |
| Wire | https://wire.com | ● | ○ | ● | ○ | (A)GPL | ● | ● |

### 2.3.4 Encrypted Email Mailing lists

Solutions best known at the time of writing based on the email standards and OpenPGP or S/MIME (Secure/Multipurpose Internet Mail Extensions)[39]. These solutions support an encrypted mailing list among all participants.

**Table 4:** Overview of encrypted email mailing lists: OpenPGP and S/MIME

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| Mailman PGP | https://gitlab.com/J08nY/mailman-pgp | re-encrypt | ○/● | ● | ● | GPLv3 | ● | ○ (Unmain-tained) |
| Proposed OpenPGP extension for Mailing lists | https://gnupg.org/ftp/people/neal/openpgp-mailing-lists.pdf | ● | ○/● | ● | ● | GPLv3 | ● | ○ (2016 proposal) |
| Office 365 Message Encryption (OME) | - | re-encrypt | ○/● | ● | ● | proprietary | ? | ● |
| Petidomo | http://petidomo.sourceforge.net/#x1-300005.2 | re-encrypt | ○/● | ● | ● | GPLv3 | ● | ○ (No commits since 2017-01) |
| RedIRIS's PGP scripts | https://www.rediris.es/pgp/app/pgplist/index.html.en | ● | ○/● | ● | ● | ? | ● | ○ (last update 2008) |
| Schleuder | https://schleuder.org | re-encrypt | ○/● | ● | ● | GPLv3 | ● | ● |
| Sympa S/MIME | http://www.sympa.org/documentation/sympa-smime/ | re-encrypt | ○/● | ● | ● | GPLv2 | ● | ● |

[39] S/MIME https://tools.ietf.org/html/rfc3851

### 2.3.5 Email Encryption Gateways

Solutions best known at the time of writing based on email standards: encryption gateways can transparently handle encryption and decryption for organizations. Most of the selected criteria are not directly applicable to email encryption gateways. Some solutions offer a TLS secured web-interface. The typical scenario is that emails are sent in plaintext to the gateway and then encrypted for the transmission over the Internet to the intended recipient. Thus, "encrypted groups" can be set up as usual. Some gateways may provide tools and policies to ease this, but in the scope of this study, it was not possible to evaluate each gateway in-depth.

**Table 5:** Overview of encryption email gateways

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| CipherMail | https://www.ciphermail.com | N/A | N/A | N/A | N/A | AGPLv3 (+ proprietary versions) | N/A | ● |
| NoSpamProxy Encryption | https://www.nospamproxy.de/de/produkt/nospamproxy-encryption/ | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| Proofpoint Email Encryption | https://www.proofpoint.com | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| Symantec Email Encryption | https://www.symantec.com/products/gateway-email-encryption | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| Trend Micro | https://www.trendmicro.com | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| Virtru | https://www.virtru.com | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| Voltage SecureMail | https://voltage.com | N/A | N/A | N/A | N/A | proprietary | N/A | ○ |
| Zertificon Email Encryption Gateway | https://www.zertificon.com/en/solutions/email-encryption-gateway | N/A | N/A | N/A | N/A | proprietary | N/A | ● |
| ZixMail | https://www.zixcorp.com | N/A | N/A | N/A | N/A | proprietary | N/A | ● |

## 2.4 SECOND STEP

For the second round of assessments, email encryption gateways were omitted completely due to the following reasons: for a decentralized setup, like the one considered for the scenario of this report, gateways would need to be deployed at each of the teams, which have heterogeneous infrastructures. Thus integrating these with existing email gateways is difficult. Also, one of the advertised advantages of email encryption gateways is that they work with range of different encryption techniques (OpenPGP/MIME, S/MIME, password-based-encryption) with external recipients. Because a community of CSIRTs is a closed group of users, external recipients are a rare use case. Furthermore, if email decryption is handled at the

Internet gateway and then stored unencrypted on IMAP or transmitted to the endpoint, additional encryption must be employed to re-transmit this for consumption on mobile devices.

Table 6: overview of the solutions best known at the time of writing that score all first step criteria

| Tool | Website | Encrypt Groups | Archive | Attach-ments | Specifi-cation | License | All OS | Maturity |
|---|---|---|---|---|---|---|---|---|
| Matrix | https://matrix.org | ● | ● | ● | ● | Apache v2 (Riot) | ● | ● |
| Schleuder | https://schleuder.org | re-encrypt | ○/● | ● | ● | GPLv3 | ● | ● |
| XMPP | https://xmpp.org | ● (OTR / OMEMO) | ○/● (XEP-313) | ● (XEP-363) | ● | Free Software | ● | ● |

In the following paragraphs, these solutions are discussed in-depth. First, the primary criteria presented in Table 6 are discussed. This includes an assessment whether the design of the encrypted group communication fits into the scenario of having decentralized teams organized in a community. Furthermore, whether the provided archive and attachment capabilities are suitable and the availability of an open specification. For these solutions the remaining secondary criteria presented in Section 2 are discussed, namely Hosting (*decentralized*, *federated*, *centralized*, *on-premise), A*dd-on Compatibility, Compatibility w/ Existing Infrastructure, Backward Compatibility and Forward Secrecy.

### 2.4.1 Mailing Lists

In general, mailing lists have the advantage of an organized threaded archive that can also be accessed at a later point. Importantly, users that have been added later can also access this archive. This allows reconstructing a discussion even for someone who was not a member of the lists at that point.

Mailing lists can be hosted on-premise and easily integrate with the existing infrastructure. They provide no additional forward secrecy due to their reliance on OpenPGP that typically have key rollovers of over 1 year. On the client side at the time of writing, OpenPGP add-ons exist for all systems.

Schleuder[40] is a well maintained candidate of a re-encrypting mailing list based on OpenPGP. While Sympa[41] is well maintained but it only supports S/MIME, not OpenPGP that users, like the one considered for the scenario of this report,  are more accustomed to in compliance with RFC 2350[42] .

#### 2.4.1.1 Schleuder

For Schleuder mailing lists, a mailing list key pair is generated by the list provider. A sender encrypts his/her email on the client side using this public key and transmits the email to the mailing list provider. On the mailing list provider side, the email is decrypted and re-encrypted using the individual recipients' public keys.

The threat model of Schleuder is a "wanted Man-in-the-Middle": The emails are protected against wiretapping by the *email provider*, but can be read during re-encryption by the *list provider*. However, if the list provider is compromised at a specific point in time, previous

[40] Schleuder https://schleuder.org/
[41] Sympa https://www.sympa.org/
[42] RFC 2350 "Expectations for Computer Security Incident Response" https://www.rfc-editor.org/rfc/rfc2350.txt

discussions cannot be decrypted by the adversary, only future communication is compromised. Conclusively, one can view the list provider to be "just another" subscriber to the mailing list that has no way of accessing old conversations.

To provide better resilience when deploying Schleuder, it should be considered to run fallback Schleuder instances at the local organization to switch over to in case the main Schleuder instance cannot be reached. A few simple feature additions to Schleuder, such as adding an import function for a fixed subscriber list would ease administration of this deployment.

**Mailing lists have the advantage of an organized threaded archive that can also be accessed at a later point.**

#### 2.4.1.2 OpenPGP/MIME vs. OpenPGP "Inline"

Simply encrypting/signing the payload of an email is called OpenPGP "Inline". This is not standardized properly and requires a separate encryption for each attachment. A more comfortable attachment and encoding handling with OpenPGP according to the standard is only provided by with clients implementing OpenPGP/MIME. Thus, in case OpenPGP encrypted mailing lists are chosen, it is recommended to use modern email client add-ons that implement OpenPGP/MIME.

### 2.4.2 Matrix

In Matrix, group communication takes place in *rooms*. The protocol uses a one-to-one Double Ratchet (as in the Signal Protocol) between each other in a full mesh ("Olm ratchet")[43]. A simpler hash ratchet is used to encrypt sequences of messages from each device to other devices in a room (Megolm ratchet)[44].
The state of each device's megolm ratchet ("Megolm key") is sent to all the other devices in the room over the secure one-to-one Olm channel, such that they can decrypt the messages and message history as long as they have the necessary Megolm session keys. The sessions are regularly re-established to avoid reusing the same key throughout the lifetime of the room (especially as users join/part the room). This scheme allows partial forward secrecy, but no post-compromise security (called "backward secrecy" in Matrix' specification) as long as the session is continued and not re-established. This design allows newly added members to decrypt previous communication by retrieving Megolm session keys from other participants.

In their reference implementation, the exchange of Megolm session keys can be configured per room. The possible settings are: "Anyone", "Members only (since the point in time of selecting this option)", "Members only (since they were invited)" and "Members only (since they joined)".

The solution Matrix provides is fully decentralized and federated. In practice, each user is registered at a homeserver and all messages in a communication room are replicated over all homeservers of the room's participants. Thus, every homeserver connected to a room, through a user in this room, keeps a content of the room's history. If a homeserver goes down, the conversation can still go on as the remaining homeservers are still exchanging messages. If a homeserver is back online it can re-synchronise messages, i.e., receiving old ones from other homeservers and inserting its own into the timeline of others.

Matrix is openly specified and implemented. On one hand, its reference implementation "Riot" has already several features and allows end-to-end encryption. On the other hand, Riot on desktop systems is implemented using web technology, and native clients, such as nheko-reborn, still lack in certain areas, such as encryption reliability. While there seem to be a

[43] Olm: A Cryptographic Ratchet, https://matrix.org/docs/spec/olm.html
[44] Megolm group ratchet, https://matrix.org/docs/spec/megolm.html

development team of several full-time persons, at time of writing, there is no indication that this team has found a stable business model yet.

In 2018, the French Ministry of Digital reached out to the Matrix developers and a collaborative project called DINSIC started[45]. Since January 2019, Matrix solutions are getting rolled out to the ministries[46]. This effort also led to increased work on finalizing the Matrix Version 1.0 Specification. In parallel, ANSSI and EY[47] are working on an audit. Interesting features on the implementation roadmap in regards to end-to-end encrypted are end-to-end capable search and turning on end-to-end encryption by default for rooms with private history.

### 2.4.3 XMPP

XMPP (Extensible Messaging and Presence Protocol) is an extensible standard for real-time communication. It is openly specified since 1999. The difficulty of evaluating XMPP lies in its history as an evolving standard. There exists a number of different cryptographic protocol extensions XEPs (XMPP Extension Protocol)[48]. The currently valid ones on the Standard track are XEP-0373: OpenPGP for XMPP[49], XEP-0384: OMEMO[50], XEP-0396: Jingle Encrypted Transports – OMEMO[51]. In addition, OTRv3[52]/OTRv4[53] is often used as an encryption protocol on-top of XMPP. A sensible selection of XEPs, that are required to provide a secure and modern group communication, has been defined by the Conversations team in their XMPP Compliance Tester[54]. While this tester is focused on server providers, the selected XEPs must also be implemented on the client side to provide the required features. This makes it especially difficult to select the correct clients. If XMPP is chosen, a list of preferred clients should be distributed among participating organizations. Again, a difficult trade-off lies in the “archive” and “forward secrecy” requirement. While OMEMO and OTR provide forward secrecy, there is no way to provide an archive of messages[55]. The newer OpenPGP extension, on the other hand, allows archiving, but no forward secrecy. A lot of comparisons between Matrix and XMPP can be found on the Internet. Here the focus is on three main points:

1) If one Matrix homeserver fails, Matrix can still re-synchronize missing messages between the remaining homeservers participating in a group discussion. Due to its real-time communication focus, this is not part of a normal XMPP server. Thus, **Matrix is in principle more resilient to network failures and disruptions.**
2) Matrix’ Megolm protocol is unique in its properties and has been especially designed for group chats, choosing a promising set of tradeoffs. However, only the primary client Riot supports all features.
3) XMPP is a mature federated protocol with a long history. In contrast to Matrix, it has been proven in real world deployments. Matrix protocol is still in development, at the time of writing, and has a lot of room for improvement and stability issues.

[45] Matrix and Riot confirmed as the basis for France’s Secure Instant Messenger app, https://matrix.org/blog/2018/04/26/matrix-and-riot-confirmed-as-the-basis-for-frances-secure-instant-messenger-app/
[46] Matthew Hodgson, Matrix in the French State, FOSDEM, Feb 2019, https://matrix.org/blog/wp-content/uploads/2019/02/2019-02-01-FOSDEM-Matrix-1.0.pdf
[47] Matrix in the French State https://fosdem.org/2019/schedule/event/matrix_french_state/
[48] XEPs (XMPP Extension Protocol  https://xmpp.org/extensions/
[49] XEP-0373: OpenPGP for XMPP https://xmpp.org/extensions/xep-0373.html
[50] XEP-0384: OMEMO Encryption https://xmpp.org/extensions/xep-0384.html
[51] XEP-0396: Jingle Encrypted Transports – OMEMO https://xmpp.org/extensions/xep-0396.html
[52] Off-the-Record Messaging Protocol version 3 https://otr.cypherpunks.ca/Protocol-v3-4.1.1.html
[53] Off-the-Record Messaging protocol. OTR version 4 https://github.com/otrv4/otrv4/blob/master/otrv4.md
[54] XMPP compliance tester, https://compliance.conversations.im/
[55] OMEMO Multi-End Message and Object Encryption, https://conversations.im/omemo/

# 3. RELEVANT RECENT IDEAS

## 3.1 PRACTICAL ENCRYPTED MAILING LISTS

A new proposal has been published as a whitepaper named "Practical Encrypted Mailing Lists"[56] (Walfield, 2018). It does not require plaintext access on the list provider. For this, the encryption public keys of all subscribers are bound as subkeys to a new OpenPGP key. A new key flag is defined for special handling of these subkeys. To send a new email to this list, the sender selects this OpenPGP key, the email is encrypted to all subkeys and then relayed by the list provider to all recipients. While a prototype implementation is available[57], it requires changes to the OpenPGP standard and it is not available in a stable GnuPG release or other OpenPGP implementations, such as OpenPGP.js[58] or OpenKeychain[59]. Thus, while the proposal is sound, it is not considered as a stable candidate, at the time of writing. However, it serves as an example that innovations may become available as implementations of new cryptographic approaches are evolving.

## 3.2 GROUP WEB KEY DIRECTORY FOR OPENPGP

Deploying end-to-end encrypted communication solutions for groups face the challenge of managing who is part of a group and how to find the public keys associated with a person. Using a central solution in the use case analysed here would make it a single-point of failure.

**Deploying end-to-end encrypted communication solutions for groups raises the challenge of managing who is part of a group and how to find the public keys associated with a person.**

When this is not the case, a new solution could be based on the Web Key Directory (WKD)[60], as proposed and implemented by GnuPG[61] since 2016. This mechanism could be extended for group establishment. WKD works like this: Per email address, a TLS connection is established to the email address' domain to retrieve the public key. Now, the email client can automatically encrypt to this email address. WKD builds upon the wide spread usage of the X.509[62] based certificate system for web servers. It allows for a better user experience for end-to-end encrypted mails.

To extend WKD for groups, a list of email addresses could be transferred. This way each organisation announces which email addresses belong to a group. Email clients would need to be extended to do this extra request and to be able to encrypt to a number of recipients automatically. The idea seems promising for smaller communities.

## 3.3 RESILIENT COMMUNICATION

In case of catastrophic events or an Internet lock-down, alternative means of establishing a peer-to-peer network are required that are not covered by the previously discussed solutions. These forms of communication are often called "Off-Grid Communication". At IETF, the Bundle Protocol (RFC 5050bis[63]) is being standardized for Store-Carry-Forward routing ("Delay-Tolerant Networks"). This allows forwarding of messages hop-by-hop directly over different

[56] Neal H. Walfield, Practical Encrypted Mailing Lists, 2016, ftp://ftp.gnupg.org/people/neal/openpgp-mailing-lists.pdf
[57] Neal's encrypted mailing Lists, https://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git;h=refs/heads/neal/encrypted-mailing-lists
[58] OpenPGP.js https://openpgpjs.org/
[59] OpenKeychain https://www.openkeychain.org/
[60] Web Key Directory, https://wiki.gnupg.org/WKD, https://datatracker.ietf.org/doc/draft-koch-openpgp-webkey-service/
[61] GnuPG https://www.gnupg.org/
[62] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://tools.ietf.org/html/rfc5280
[63] Bundle Protocol Version 7 https://tools.ietf.org/html/draft-ietf-dtn-bpbis-12

underlying convergence layers, such as Bluetooth or Wifi Direct. Similarly, Briar[64] and Wind[65] allow direct communication between devices over Bluetooth. These solutions could, in the future, be considered as fall-back mechanisms for short distance and mesh-networks.

## 3.4 INTERESTING RELATED RECENT CRYPTOGRAPHIC RESEARCH

Besides the difficult trade-off between archive access and forward secrecy/post-compromise security, there are additional security properties that could be evaluated. An example is "transcript consistency"[66] (Unger, 2015; Rösler, 2018) that defines the assurance that all members of a conversation are seeing the same message transcript, rather than messages which are selectively delivered or re-ordered to only some members, or messages which contain different plaintext for different members.

[64] Briar, https://briarproject.org
[65] Wind: Off-Grid Services for Everyday People, https://guardianproject.info/wind
[66] The interested reader is referred to academic papers, such as "SoK: secure messaging" (Unger, 2015), "More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema" (Rösler, 2018), and "ENISA: Study on cryptographic protocols"

# 4. BIBLIOGRAPHY

CIRCL Computer Incident Response Center Luxembourg, *CSIRT Tooling: Best Practices in Developing, Maintaining and Distributing Open Source Tools*, 2018-11-06, https://github.com/CIRCL/compliance/blob/master/csirt-tooling-best-practices/index.md

ECRYPT CSA *Algorithms, Key Size and Protocols Report (2018)* European Network of Excellence in Cryptology (ECRYPT CSA), 2018, www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf

P. Rösler, C. Mainka and J. Schwenk, *More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema*, IEEE European Symposium on Security and Privacy (EuroS&P), 2018

N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, & M. Smith, *SoK: secure messaging*, IEEE Symposium on Security and Privacy (SP), 2015

N. H. Walfield, *Practical Encrypted Mailing Lists*, 2016, ftp://ftp.gnupg.org/people/neal/openpgp-mailing-lists.pdf

# 5. GLOSSARY AND ACRONYMS

Please refer to ENISA glossary https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/bcm-resilience/glossary and ENISA list of acronyms https://www.enisa.europa.eu/media/media-press-kits/enisa-glossary

TP-03-19-540-EN-N

## ABOUT ENISA

The European Union Agency for Cybersecurity (ENISA) has been working to make Europe cyber secure since 2004. ENISA works with the EU, its member states, the private sector and Europe’s citizens to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe’s critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU.  Since 2019, it has been drawing up cybersecurity certification schemes. More information about ENISA and its work can be found at www.enisa.europa.eu.

**ENISA**
European Union Agency for Cybersecurity

**Athens Office**
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

**Heraklion office**
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu

ISBN: 978-92-9204-303-2,
DOI: 10.2824/062571