**NeXt generation Techno-social Legal Encryption Access and Privacy** nextleap.eu

Grant No. 688722 Project Started 2016-01-01. Duration 36 months

# DELIVERABLE D3.1

# Initial Decentralization Case-studies

Francesca Musiani (CNRS), Ksenia Ershomina (CNRIS)

| | |
|---|---|
| **Beneficiaries:** | **CNRS** (lead) |
| **Workpackage:** | **D3.1 Initial Decentralization Case-studies** |
| **Description:** | This list will be an initial list of ethnographic case-studies of communities focussed on decentralized internet messaging protocols. |

| | |
|---|---|
| **Version:** | 1.0 |
| **Nature:** | Report (R) |
| **Dissemination level:** | Public (PU) |
| **Pages:** | 87 |
| **Date:** | 2016-06-30 |

Project co-funded by the European Commission within the Horizon 2020 Programme

# Table of Contents

# 1. Introduction

This deliverable aims at giving an overview and mapping of 30 projects focused on decentralized Internet messaging protocols. This broad set lists potential case studies, a selection of which will be investigated deeper in Months 6 to 24 of the project with social sciences qualitative methods, including ethnography. This initial set of cases refers to protocols related to those that will be developed within the NEXTLEAP project. The selection of 30 cases was made upon several criteria ranging from their use of open standards, their degree of decentralization, the data collected, the user base etc.; these criteria will be detailed further below.

Although some of these projects may, eventually and ideally, consider upgrading to the protocols designed with the help of NEXTLEAP, assessing the extent to which they may be willing and are available to do so is not the primary purpose of this deliverable. This issue is more closely addressed by D5.1, which is focused on the organization of developer communities in the field of decentralized and encrypted messaging and their possible integration in the NEXTLEAP project. The present deliverable and WP2 as a whole are focused on observing, in a detailed and critical manner, the development and early-user appropriation of decentralized messaging systems, as well as the history and background of the developers and some central pioneer users, while refraining from directly engaging with their adoption of NEXTLEAP protocols.

This deliverable presents each project in a brief form (one page per project), providing a map that will help the NEXTLEAP team to navigate within the rapidly developing universe of existing projects exploring encrypted and/or decentralized messaging. This cartography will also provide a ‘lower layer’ for the choice of the three case-studies to delve into in depth in D3.2 and D3.3. The results of 3.2 will feed into T2.2 in WP2 (the design of the decentralized secure messaging protocols), while the finalization of the surveys, fieldwork, and interviews with the community under study will happen at the end of Year 2.

The two main kinds of projects that we selected for this overview are related to chat clients (messengers) and email clients. First of all, we present several basic protocols used for chat and email-oriented services, and then move on to the criteria for studying and comparing our set of cases.

## 1.1 Protocols and standards for messaging

### 1.1.1 Protocols for asynchronous messaging (e-mail)

The **SMTP** is the protocol originally used for transferring email and as such is one of the oldest standards for asynchronous messaging, first defined in 1982 by the IETF[1] and by default not having provision for content confidentiality. PGP was created to add end-to-end encryption capabilities to e-

1 https://tools.ietf.org/html/rfc821

mail in 1991 by Phil Zimmerman. In part due to pressure from the U.S. government, and in part due to patent claims by RSA Corporation, Phil Zimmerman pushed PGP to become an IETF standard. The OpenPGP set of standards was finally defined in 1997[2], to allow the open implementation of PGP. OpenPGP is implemented in mobile apps, such as the IPGMail for iOS and the Openkeychain key management system for Android and F-Droid. **GnuPG** (GPG) is a free software implementation of the OpenPGP standards developed by Free Software Foundation in 1999 and compliant with the OpenPGP standard specifications. There's a number of e-mail client applications that rely on GPG such as the Enigmail plugin for Thunderbird, Kontact, Evolution. The problem with GPG and alternative implementations is that they have user-friendliness issues, especially in terms of usage and key management, as well as security issues; if an adversary compromises a user's private key, this allows all encrypted messages to be read. Finally, the IETF has recently opened up the OpenPGP Working Group in order to allow the fundamental algorithms to be upgraded and to use more modern cryptographic primitives, such as larger keys. S/MIME is another IETF standard addressing the need for encrypting multi-part emails, i.e. emails with attachments.

### 1.1.2 Protocols for synchronous messaging (chat)

**XMPP open protocol** became an IETF (Internet Engineering Task Force) standard in 2004. XMPP is a federated standard that "provides a technology for the asynchronous, end-to-end exchange of structured data by means of direct, persistent XML streams among a distributed network of globally addressable, presence-aware clients and servers"[3]. There are many implementations of the XMPP specifications. The XMPP foundation gives a list of 70 clients and 25 servers actually using the XMPP protocol[4]. Jabber.org is the original instant messaging service based on XMPP, and it is now one of the biggest nodes on the XMPP network. XMPP traffic or content are not encrypted by default. Network level encryption security using SASL and TLS has been built into the core. In addition, as claimed by the XMPP foundation, a team of developers is working on an upgrade of the standard to support end-to-end encryption[5].

**The OTR protocol** is, in essence, an extension to XMPP released in 2004, providing deniable authentication for users; unlike PGP output which can be later "used as a verifiable record of the communication event and the identities of the participants" (Borisov, Goldberg, Brewer et al. 2004). Thus, OTR is a security upgrade over PGP. The original paper that defines OTR is called ""Off-the-Record Communication, or, Why Not To Use PGP" (Borisov, Goldberg, Brewer et al. 2004). The first OTR implementation was a popular Linux IM client, GAIM. At the present moment it is said to be used by 14 instant messaging clients[6], including earlier versions of Cryptocat (in-browser Javascript client), Jitsi, ChatSecure (XMPP client for Android and iOS). However, OTR is designed for synchronous messaging between two people, and so does not work out-of-the-box for group messaging or asynchronous messaging. There has long been a move away from OTR; the IM+ app for Android, even though having good user ranking and between 5 mln and 10 mln downloads, is

2 https://tools.ietf.org/html/rfc4880

3 https://tools.ietf.org/html/rfc6120#page-13
4 http://xmpp.org/software/clients.html
5 http://xmpp.org/about/technology-overview.html
6 https://otr.cypherpunks.ca/software.php

reported by users on Google Play Market as “abandoned” (last update in 2014). Climm is abandoned since 2010. Recently, a number of variations and alternatives to XMPP have been developed:

**Matrix.org**, released in December 2014, is designed as an “open specification for decentralized communication” using JSON rather than XML like XMPP, is an application-layer communications protocol for federated real-time communication. It is unencrypted by default. However, using the Olm library (Axolotl ratcheting) encryption on a room-to-room basis is optionally achieved. Among the innovative features of Matrix.org compared to other standards is its interoperability, as underlined in several articles: the main goal of the project being to “create an architecture that tackles the interoperability problems that were not addressed by previous approaches” (Prokop, 2015). Others underline its attractiveness for users that results from this interoperability: “where IRC has a high barrier to entry, requiring you to know exactly what server you're connecting to and configure accordingly, Matrix would let you associate with as many public identities as you're willing to share (phone number, email address, Facebook, Google, and so on), as long as they support the Matrix standard. Otherwise requires no setup -- it's just like if you were using any consumer messaging service” (Weinberger, 2014). However, Matrix.org seems to have few users since none of the mainstream IM clients relies on it yet. The “Matrix Console” messaging app for Android has between 1000 and 5000 downloads. The website lists 17 clients and 6 servers using Matrix.org[7].

**The Signal Protocol**, the non-federated protocol developed in 2013 by **Open Whisper Systems**, provides end-to-end encryption for instant messaging. It combines the “Axolotl” DH and KDF ratchets, initial key exchange and the ongoing renewal and maintenance of short-lived session keys. It could be a standard, but is not yet recognized as such; Moxie Marlinspike, the co-author of Axolotl, was inspired with some OTR features, such as the idea of ephemeral key exchange (Marlinspike, 2013), but also added additional security features such as “backward secrecy”, support for asynchronous messaging and group messaging, going above and beyond OTR by allowing also clients to be offline. The Signal protocol is said to be widely used in mobile messaging applications such as Signal (formerly TextSecure and RedPhone), WhatsApp, Secure Chat (by GData). Silent Circle uses a version of the Signal Protocol since 2015 in its Silent Phone. The first step towards “standardization” of Axolotl is the creation of OMEMO.

**OMEMO** is a new *encrypted* extension of XMPP protocol developed in Summer 2015. It uses Axolotl to establish multiple secure sessions with the clients of your contacts. It has been presented to the XMPP Standards Foundation but not yet approved in any official manner[8]. The initial ambition of OMEMO is to build upon the work of the Signal Protocol, OTR and OpenPGP; OTR is said to have “inter-client mobility problems” and can only work when all conversation participants are online, while OpenPGP “does not provide any kind of forward secrecy and is vulnerable to replay attacks” (Straub, 2015). The software implementations of OMEMO are quite few for the moment. Among them, we must cite **Conversations** (XMPP, OMEMO, TLS), an open-source application for Android that counts over 5000 installs via Google Play Market, and an unknown number of installs via F-Droid.

7 https://matrix.org/docs/projects/try-matrix-now.html
8 https://xmpp.org/extensions/inbox/omemo.html

## 1.2 Network-level encryption and decentralization

While this paper (and subsequent work of the NEXTLEAP project) is mostly focused on the application level, it seems important to mention the network-level initiatives, such as P2P routing services or anonymous remailers. There seems to be no functional standards on this level; however, some solutions, such as Tor or I2P, tend to serve as references for different projects.

The **Tor** hidden service protocol offers a platform to develop decentralized and encrypted instant messenger servers. It is used by projects such as the Tor Messenger, Pond and Ricochet. Another example is the decentralized and end-to-end encrypted mobile messenger Briar that relies on the Tor network when available, but could also work over Bluetooth in case of emergency off-the-grid situations. Among further examples, the well-known Tor (and Orbot for Android) aims at providing encryption and anonymity on top of TCP/IP. Anonymous high-latency remailers, as well, are proposing solutions to the problem of transport meta-data leaks in federated messaging, falling under three types: Cypherpunk Anonymous Remailer, Mixmaster, Mixminion. The latter is not currently active, according to the statement on the official website[9]. The statistics on the website show there are currently 18 Mixminion nodes running - compared to almost 1.2K of Tor routers.

There has been a number of new tools developed on the network level. Zero Tier One is an end-to-end encrypted, peer-to-peer virtual network; it is orthogonal to DNS and does not depend on it -- it provides static network addresses which remain stable even if the user changes physical WIFI/networks. CJDNS implements a virtual IPv6 network in which all packets are encrypted to the final recipient, using public key cryptography for network address allocation and a distributed hash table for routing (as described by the CJDNS White Paper on GitHub[10]).

## 1.3 Activity stream standards

Besides private communication, the public-facing posting, messaging and status updates, also called "Activity Streams" that exemplify popular applications such as Twitter and Facebook. are being reconsidered in terms of privacy and decentralization. This work comes mostly from a desire to decentralize Facebook and Twitter, but has not typically taken privacy or security concerns unlike chat and e-mail standards.

"Activity Streams"  is a data format for encoding and transferring activity/event meta-data. The first version of the specification for Activity Streams in XML was published in 2011 by the independent

9 http://mixminion.net/
10 https://github.com/cjdelisle/cjdns/blob/master/doc/Whitepaper.md

Activity Streams Working Group[11]. Since then, the W3C Social Web Working Group is working on new standards for activity streams (status updates)[12]. The current (**ActivityStreams 2.0**) version of the spec is JSON-based. There is also “IndieWeb” versions for decentralized federation like **Micropub** as well as a **ActivityPub** for federating ActivityStreams 2.0.  For this developing space, see http://www.w3.org/Social/WG.

Other open standards are proposed on top of Activity Streams implementations. **Salmon Protocol** aims to define a standard protocol for comments and annotations (open, decentralized, abuse resistant, and user centric). **OStatus** is an open-standard for microblogging, that is built on top of existing open protocols including Atom, **Activity Streams**, **PubSubHubbub, Salmon** and **Webfinger**. The **OStatus** standard was implemented in **GNU social** (formerly StatusNet, a federated microblogging service) and **Quitter** (federated specific installation of GNU Social)**.**

11 http://activitystrea.ms/

12 https://www.w3.org/wiki/Activity_Streams#Implementations

# 2 Towards a set of criteria for categorization of (decentralized) messaging projects

While some projects are products of wide and well-known communities (such as Open Whisper Systems and Tor), new services either re-use the protocols or infrastructure independently by smaller groups and non-institutionalized developing teams. When standards are not available or not satisfying, there is a tendency to (re)use not yet officially standardized protocols and tools as standards such as Signal's Axolotl ratchet. That is why, taken in consideration this moving nebula of standards and non-standardized projects, it is important to start with a mapping based on a defined range of criteria.

All of the 30 project selected are either encrypted and not-decentralized (such as Signal) or encrypted and decentralized (such as Tor). We pay attention to open source projects, however, business closed-source solutions are also of interest. We take into consideration the kinds of data collected by the applications, as well as the purpose of this collect. For instance, some applications (e.g. Wickr) collect user statistics: anonymous information about basic usage statistics, such as the number of messages sent by all users daily, what types of messages our users tend to send (e.g., voice messages more often than text), and so forth. The number of users, their geo-location and the targeted user-groups must as well be defined (whether the app is optimized for anarchists, journalists, human right defenders, power-users or developers, enterprises, government…).

The basic criteria for case descriptions can be presented in the following list:

- Is it open-source? If so, where (link)?
- Is it centralized or decentralized?
- If decentralized, is it federated or P2P?
- If decentralized, does it use a standard?
- What operating systems?
- What standards?
- What data is collected, for what purpose?
- How many users?
- What countries?
- What kinds of users (file-sharing, human right defenders, anarchists, enterprises, government, only power-users, only developers, "ordinary" people)

- Business models? (if any)

An important caveat concerning terminology must be acknowledged here. As regards (de-)centralization and federation, for the time being, we are referring to technology and algorithms. Thus, we do not use the words in the same way as D5.1, that discusses "social federation", i.e. who controls instances of servers.[13] More generally, we debated on whether to include a criterion on (de-)centralization of governance/power structures; we ended up not including it, as it is complicated to get this kind of information without in-depth investigation, which we do not undertake for all the 30 cases. However, it is an aspect that will be thoroughly examined in the three in-depth case studies, and we open it up for further investigation in the conclusions here.

Template for project presentation

The thirty project presentations included in the remainder of this deliverable will each have the following structure:

Name of application

Author(s) (organisation and/or names)

Screenshot if available

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | |
| (De)Centralization | |
| (if decentralized) Federated/P2P? | |
| (if decentralized) Standardized? | |
| Standard(s) | |
| Operating System(s) | |
| Data Collection (/Purpose) | |

13 From this standpoint, Bitcoin is mostly technically decentralized but socially centralized: focused on the core group creating and delivering the software, while users effectively run the same software.

| | |
|---|---|
| Number of users | |
| Type(s) of users | |
| Countries | |
| End-to-end group chat? | |
| End-to-end voice calls? | |

Discursive description and additional information

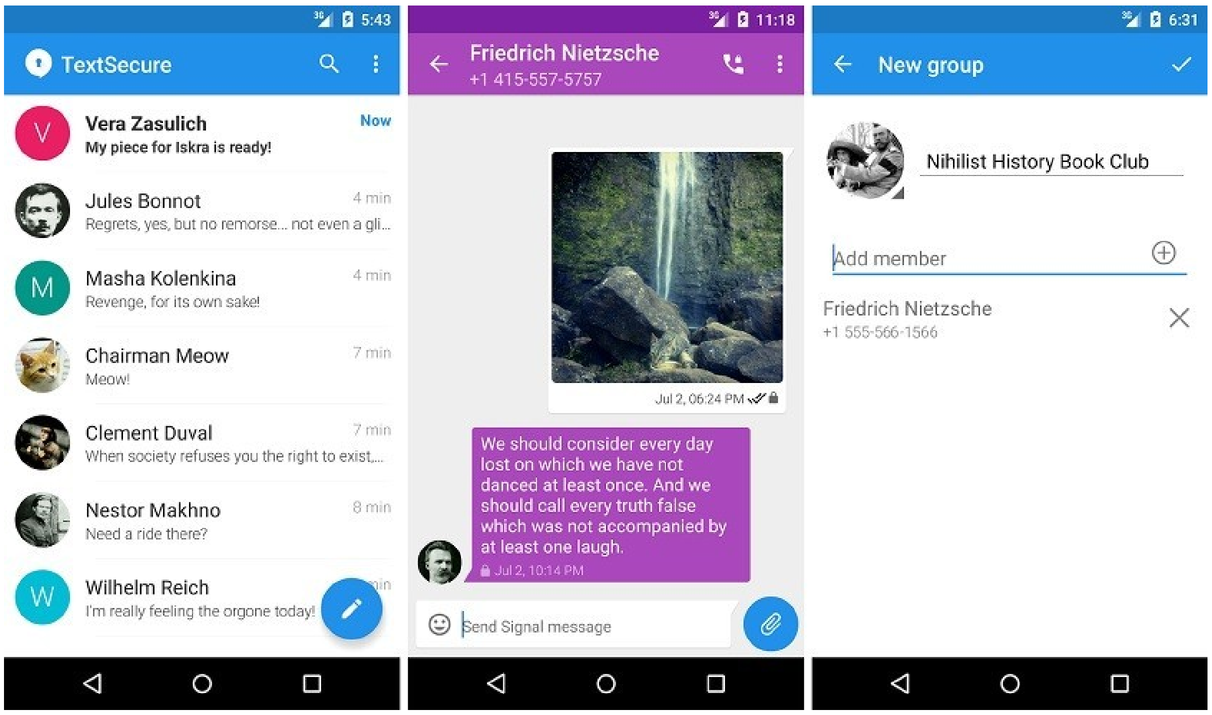
# 3 Case studies

## 3.1 Signal (formerly TextSecure)

Authors: Open Whisper Systems: Moxie Marlinspike, Tyler Reinhard, Trevor Perrin, Lilia Kai

https://whispersystems.org/

https://twitter.com/whispersystems



Vera Zasulich Now
My piece for Iskra is ready!

Jules Bonnot 4 min
Regrets, yes, but no remorse... not even a gli...

Masha Kolenkina 4 min
Revenge, for its own sake!

Chairman Meow 7 min
Meow!

Clement Duval 7 min
When society refuses you the right to exist,...

Nestor Makhno 8 min
Need a ride there?

Wilhelm Reich
I'm really feeling the orgone today!

Friedrich Nietzsche
+1 415-557-5757

Jul 2, 06:24 PM

We should consider every day lost on which we have not danced at least once. And we should call every truth false which was not accompanied by at least one laugh.
Jul 2, 10:14 PM

Send Signal message

Nihilist History Book Club

Friedrich Nietzsche
+1 555-566-1566

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Opensource:<br>**https://github.com/whispersystems** |
| (De)Centralization | Centralized |
| Standard(s) | Axolotl protocol (not a standard but tends to be used as such) |
| Operating System(s) | Android, iOS, Desktop version in beta (Chrome |

| | |
|---|---|
| | add-on) |
| Data Collection (/Purpose) | Desktop version asks for the access to Google contacts, but data not actively collected it appears |
| Number of users | About 1 million |
| Type(s) of users | Mostly activist-oriented, also used by journalists |
| Countries | Used in Russia, France (Nuit Debout movement), USA… |
| End-to-end group chat? | Yes (text messaging, photo and video) |
| End-to-end voice calls? | Yes |

**Discursive description and additional information**

Originally created by Moxie Marlinspike and Trevor Perrin’s Whisper Systems, the firm was sold to Twitter in 2011, at which point things looked uncertain. In 2013, however, TextSecure re-emerged as an open source project under the auspices of a new company, Open Whisper Systems since when it and has gained endorsements from figures such as Bruce Schneier and Edward Snowden. Open Whisper Systems is both a large community of volunteer Open Source contributors, as well as a small team of dedicated grant-funded developers. Their mission: “to advance the state of the art for secure communication, while simultaneously making it easy for everyone to use”.

Signal was designed as an independent end-to-end platform that transports messages across its own data infrastructure. Signal is presenting itself as an app for activists. See for instance their screenshots using names such as Nestor Makhno (a famous Ukrainian anarchist leader) or Vera Zassulitch (an anarchist and feminist). Signal is based on users’ real names and address books. The development team is supported by community donations and grants. There are no advertisements, and it doesn't cost anything to use.

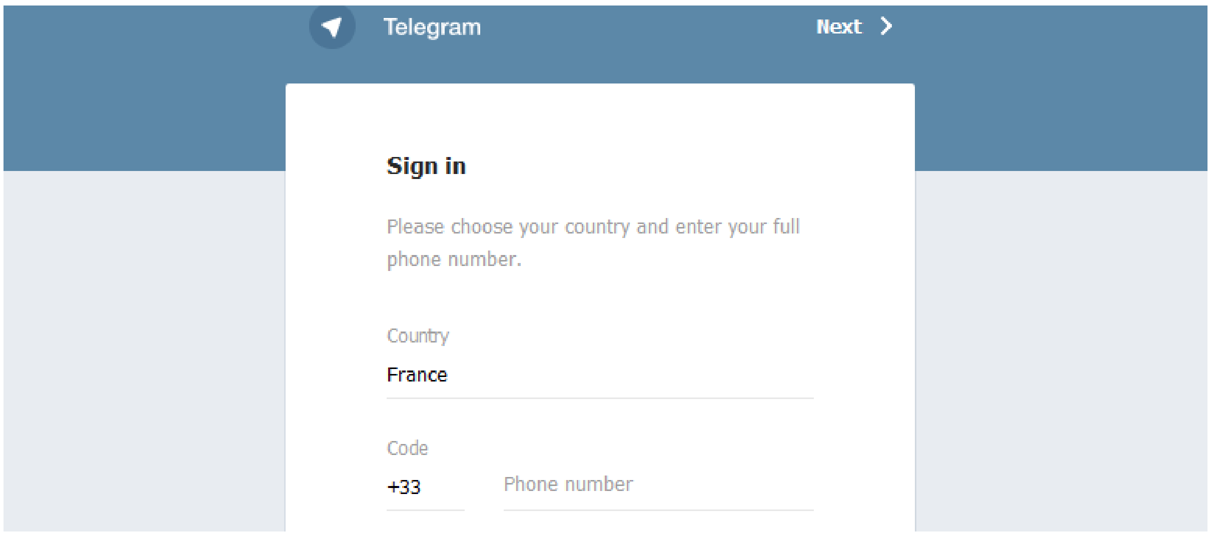
However, Moxie announced on 18 May 2016, that the Signal protocol will be used by Google: “We’re excited to partner with Google on the private communication features of their new smart messaging app, Allo”[14]. On 5 of April 2016 the Signal protocol was implemented in WhatsApp.

14 https://whispersystems.org/blog/allo/

## 3.2 Telegram

Author(s): Pavel and Nikolai Durov

https://telegram.org



Sign in

Please choose your country and enter your full phone number.

Country
France

Code
+33
Phone number

(Screenshot: Web interface).

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes, for some components.<br>client-side code: open source.<br>server-side code: closed source and proprietary<br>https://core.telegram.org/mtproto<br>https://core.telegram.org/api/end-to-end<br>https://telegram.org/apps#source-code |
| (De)Centralization | Centralized |
| Standard(s) | 256-bit symmetric AES encryption, RSA 2048 encryption |
| Operating System(s) | Android, iOS, Windows Phone, OS X, Ubuntu Touch, MS Windows, Linux |
| Data Collection (/Purpose) | Telegram “stores the data it needs to function properly”[15]: cloud chats, media in secret chats |

15 https://telegram.org/privacy

| | |
|---|---|
| | (encrypted), phone number, password recovery email. |
| Number of users | As of February 2016: 100 million monthly active users, claiming 350,000 new users signing up every day[16] |
| Type(s) of users | Varied.<br>Has been at core of controversy re: use of encrypted messaging by terrorists.<br>Podemos has announced they are using Telegram. |
| Countries | Varied.<br>The Durov brothers were born in Russia, as were some of the key developers, but Telegram is not connected to Russia either legally or physically. Telegram's HQ is in Berlin. |
| End-to-end group chat? | “Secret chats” use end-to-end encryption. |
| End-to-end voice calls? | No |

**Discursive description and additional information**

Telegram is a cloud messaging application that uses both server-client and client-client encryption (for secret chats), initially released August 2013. It is available for a range of operating systems.

Telegram has gained media attention when its broadcasting capabilities have been reportedly being used by jihadists for propaganda activities[17]. However, its active user base is, according to the service, large and varied.

In addition to regular private and group chats, Telegram supports “secret chats”, that use end-to-end encryption, are not part of the Telegram cloud, and can only be accessed on their devices of origin.

Telegram supports two layers of encryption:

- Server-client encryption for Cloud Chats (private and group chats)
- Client-client encryption for Secret Chats.

All data, regardless of type (text, media or files), is encrypted in the same way.

The service holds a ‘Telegram cracking contest’[18]. Articles on Telegram[19].

16 https://telegram.org/blog/100-million
17 http://www.reuters.com/article/us-france-shooting-telegram-shutdown-idUSKCN0T734R20151119

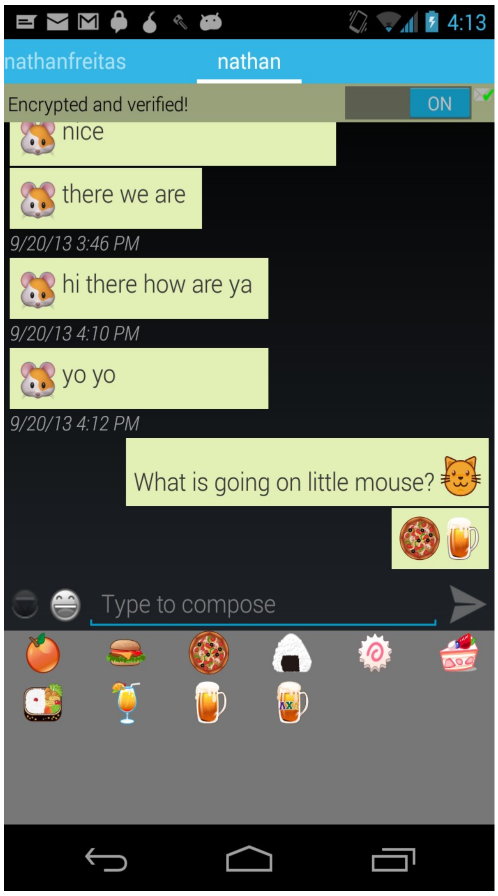
18 https://core.telegram.org/contest300K

19 https://telegram.org/press

## 3.3 ChatSecure

https://chatsecure.org

Authors: Nathan Freitas (Guardian Project), Chris Ballinger



Encrypted and verified!

nice

there we are

9/20/13 3:46 PM

hi there how are ya

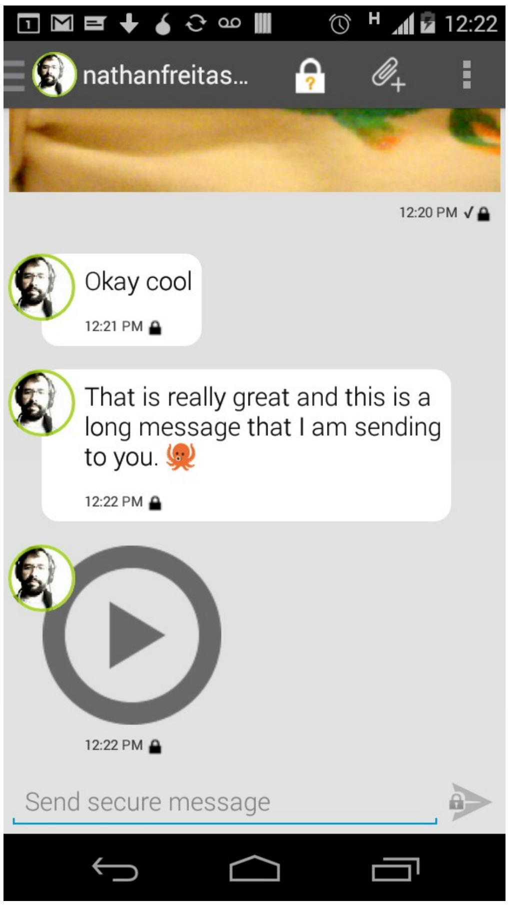
9/20/13 4:10 PM

yo yo

9/20/13 4:12 PM

What is going on little mouse?

Type to compose



12:20 PM

Okay cool

12:21 PM

That is really great and this is a long message that I am sending to you.

12:22 PM

12:22 PM

Send secure message

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes<br>**https://github.com/guardianproject/ChatSecureAndroid** |
| (De)Centralization | Centralized |
| Standard(s) | OTR encryption over XMPP with TLS certificate pinning.<br>Uses Tor as well |
| Operating System(s) | iOS, Android (via The Guardian Project's similar app, formerly named Gibberbot), |
| Data Collection (/Purpose) | "At this time, the only other data that can be submitted to a 3rd party is our opt-in, anonymized crash report data, provided by the open source crash reporting SDK from |

| | |
|---|---|
| | HockeyApp”<br><br>Though, as ChatSecure can be used over other XMPP services (for example, Google Talk), users should be attentive to the privacy policy of these services.<br><br>ChatSecure may collect statistics about the behavior of visitors to its websites. ChatSecure may display this information publicly or provide it to others. However, ChatSecure does not disclose personally-identifying information other than as described below.<br><br>ChatSecure uses cookies to help ChatSecure identify and track visitors, their usage of ChatSecure website, and their website access preferences.<br><br>If ChatSecure, or substantially all of its assets, were acquired, or in the unlikely event that ChatSecure goes out of business or enters bankruptcy, user information would be one of the assets that is transferred or acquired by a third party. Any acquirer of ChatSecure may continue to use your personal information as set forth in this policy”. |
| Number of users | Between 500 000 and 1 000 000 downloads on Google Play |
| Type(s) of users | Activists, journalists, general public |
| Countries | Unknown |
| End-to-end group chat? | yes |
| End-to-end voice calls? | No |

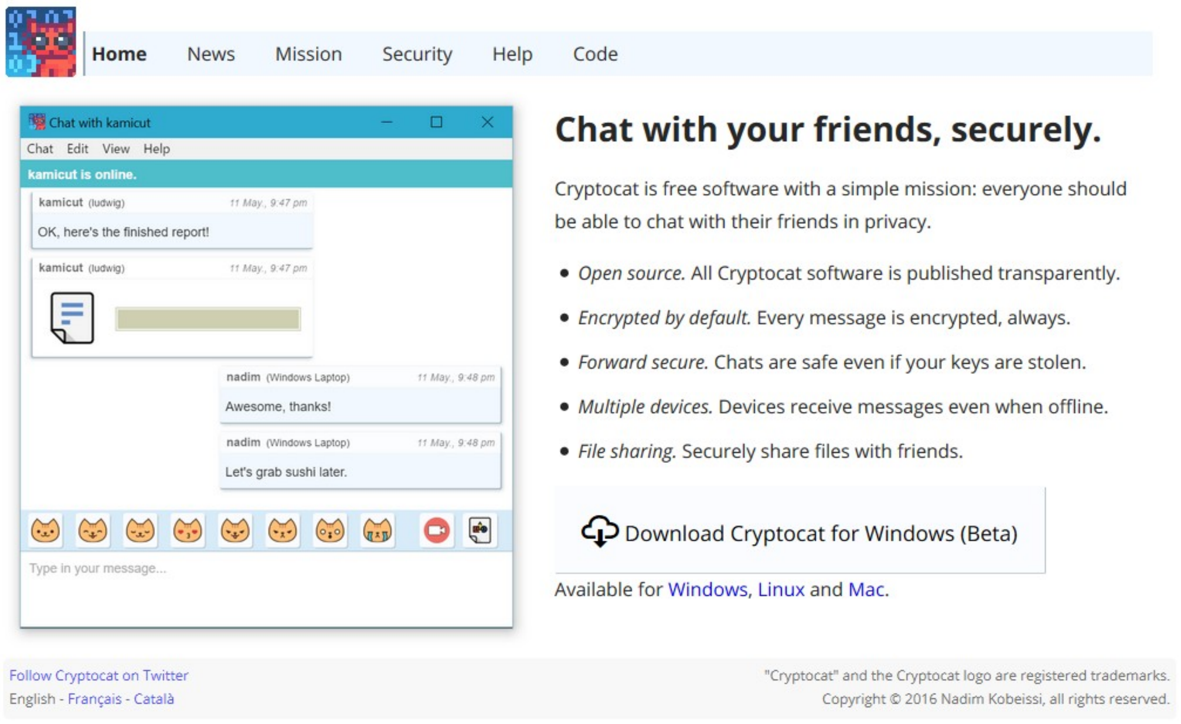
**Discursive description and additional information**

ChatSecure is fully interoperable with other clients that support OTR and XMPP. Promoted by Guardian Project, collaborate with Signal.

## 3.4 Cryptocat

https://crypto.cat/

Author: Nadim Kobeissi and open source contributors



kamicut (ludwig) 11 May, 9:47 pm
OK, here's the finished report!

kamicut (ludwig) 11 May, 9:47 pm

nadim (Windows Laptop) 11 May, 9:48 pm
Awesome, thanks!

nadim (Windows Laptop) 11 May, 9:48 pm
Let's grab sushi later.

Chat with your friends, securely.

Cryptocat is free software with a simple mission: everyone should be able to chat with their friends in privacy.

- *Open source.* All Cryptocat software is published transparently.
- *Encrypted by default.* Every message is encrypted, always.
- *Forward secure.* Chats are safe even if your keys are stolen.
- *Multiple devices.* Devices receive messages even when offline.
- *File sharing.* Securely share files with friends.

Available for Windows, Linux and Mac.

Follow Cryptocat on Twitter
English - Français - Català

"Cryptocat" and the Cryptocat logo are registered trademarks.
Copyright © 2016 Nadim Kobeissi, all rights reserved.

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://github.com/cryptocat/cryptocat<br>Currently in beta release |
| (De)Centralization | Centralized |
| Standard(s) | OMEMO,<br>XMPP |
| Operating System(s) | Windows, OS X, Linux<br>Completely rewritten as desktop software instead of the original web application software in March 2016 |
| Data Collection (/Purpose) | Information regarding “buddy list” and linked devices |
| Number of users | Circa 25,000 (source: N. Kobeissi) |
| Type(s) of users | All types. The idea behind the project is indeed |

| | |
|---|---|
| | to make privacy more accessible for a large public.<br>However, CryptoCat has received an award from Elevate Festival (Austria)[20] as a project for activists. |
| Countries | Available in English, Catalan, and French |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | No |

**Discursive description and additional information**

Cryptocat is an open source, cross-platform desktop application intended to allow encrypted online chatting between users. Cryptocat was first launched on 19 May 2011.

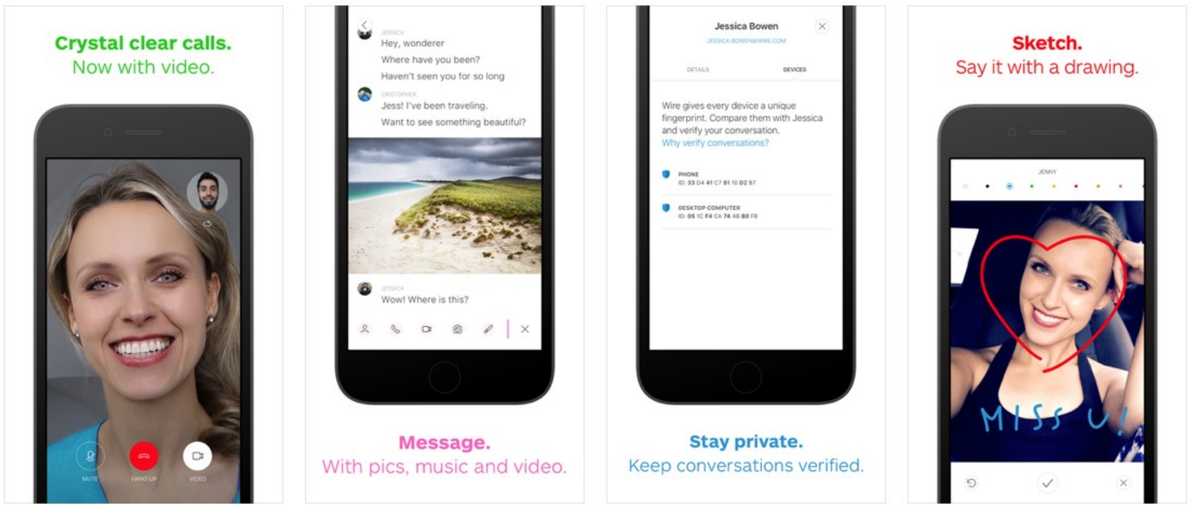
All messages, files and audio/video recordings sent over Cryptocat are end-to-end encrypted. Cryptocat users link their devices to their Cryptocat account upon connection, and can identify each others' devices via the client's device manager.

20 https://elevate.at/websites/2014/en/festival/about/newsmagazine/detail/news/the-2014-elevate-awards/index.html

## 3.5 Wire

Authors: Alan Duric (CEO), Janus Friss (Executive, co-founder Skype), Benny Neugebauer, Jens Kasemets, Panayiotis Lipiridis aka “Lilis” (developer expert at Google)

https://wire.com



Hey, wonderer
Where have you been?
Haven't seen you for so long
Jess! I've been traveling.
Want to see something beautiful?
Wow! Where is this?

Jessica Bowen
Wire gives every device a unique fingerprint. Compare them with Jessica and verify your conversation.
Why verify conversations?

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes, but not entirely: "We've released components of the app that take care of encryption and data handling as Open Source."<br>https://github.com/wireapp/wire<br>https://github.com/wireapp |
| (De)Centralization | No |
| Standard(s) | OTR, Axolotl ratchet and pre-keys optimized for mobile messaging. Voice calls use the WebRTC standard. DTLS is used for key negotiation and authentication. SRTP is used for encrypted media transport. |
| Operating System(s) | iOS, Android<br>Windows, OS X, Web-browser extensions |
| Data Collection (/Purpose) | **1.Address books** (stored on the backend servers to match users on Wire, i.e. to suggest new contacts and to automatically create connections between users)<br><br>**2.Metadata:** |

| | |
|---|---|
| | - Creator: The user who created the conversation.<br>- Timestamp: The UTC timestamp when the conversation was created.<br>- Participants list: The list of users who are participants of that conversation and their devices. This information is used by clients to display participants of the group and to perform end-to-end encryption between clients..<br>- Conversation name: Every user can name or rename a group conversation. Conversation names are not encrypted.<br><br>**3. Usage data:**<br>- Crash reports<br>- Viewed screens data<br>- Aggregated usage statistics<br>- App events data<br><br>“Wire client applications collect usage data with the aim of improving future versions of Wire. Users can disable usage data collection at any time”. Initially data is stored on users’ clients but periodically synchronised with third party services (based in EU only). Crash reports are stored on HockeyApp. All other types of usage data are stored on the Localytics service. |
| Number of users | Between 500 000 and 1 000 000 on Google Play. According to venturebeat.com (March 2016) there are about 150,000 to 250,000 sign-ups each month. However,<br>“the firm has remained tight-lipped about how many active users it has”. |
| Type(s) of users | Seems like being addressed to the general public (no specifically “activist-oriented” discourse). |
| Countries | Germany, Switzerland |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes |

**Discursive description and additional information**

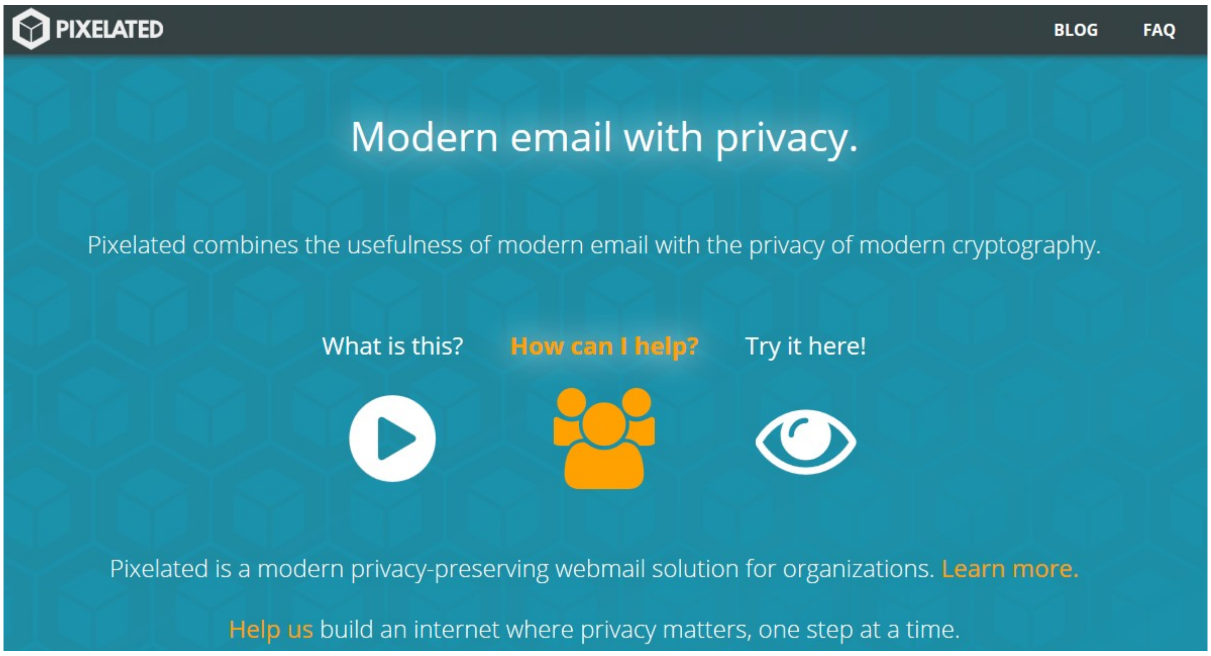
Based in Switzerland with a development center in Berlin, Germany. Wire is not financed by ads. Wire provides multi-device support for end-to-end encryption, end-to-end encrypted group chats, end-to-end encrypted 1:1 audio and video calls, voice only End-to-end encrypted group calls, has a desktop version. Has pages on Instagram, Twitter, Facebook (11450 likes). Supports SoundCloud and YouTube to share links as embeds in the chat.

## 3.6 Pixelated/LEAP

Authors: The Pixelated Team (8-10 people) and LEAP Encryption Access Project (8-10 people).

https://pixelated-project.org/

https://leap.se/



Modern email with privacy.

Pixelated combines the usefulness of modern email with the privacy of modern cryptography.

Pixelated is a modern privacy-preserving webmail solution for organizations. Learn more.

Help us build an internet where privacy matters, one step at a time.

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>Source code available on Github under AGPL license<br>https://github.com/pixelated/<br>https://leap.se/en/docs |
| (De)Centralization | Currently federated (based on email), wants to move to decentralized. |
| Standard(s) | Aims at “maximum interoperability with existing solutions” for secure email, i.e. use of GPG/PGP, SMTP/TLS, HKS |
| Operating System(s) | Bitmask: currently Linux, Mac OS in the works<br>Pixelated: browser based - cross platform<br>LEAP platform: Debian Linux |

| | |
|---|---|
| Data Collection (/Purpose) | Private keys are encrypted with the users' password and then stored on the server. |
| Number of users | Pixelated is hosted for approximately 100 initial users. It also ships with the latest version of bitmask - the LEAP client.<br>LEAP has been deployed by at least 4 providers. It's still in test use and the total number of users is unknown. |
| Type(s) of users | With respect to LEAP, Pixelated wants to focus on "ease-of-use". Pixelated is explicitly not especially aimed at activists.[21]<br><br>As the authors claim on their website[22], "tech-savvy anti-surveillance-minded individuals" are the initial market for Pixelated. However, they are aiming at a broader target group. |
| Countries | ThoughtWorks is funding a team to work on Pixelated full-time, currently based in Hamburg (Germany) and São Paulo (Brazil).<br>LEAP is developed by a distributed team located in the Americas and Europe. |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional information**

LEAP provides a toolbox for setting up email providers with a focus on security and encryption. It also develops the bitmask client application to access these services. Pixelated adds a webmail interface to the LEAP platform to lower the barrier of entry. LEAP focusses on enabling email providers to host a more secure service that eases encryption for the client.

Pixelated is presented as a software application (not a service in itself). It aims at giving organizations the means to host an email solution that provides a web interface as well as privacy through PGP encryption. Its discourse focuses a lot on ease-of-use. Pixelated runs the encryption and decryption routines on the Pixelated server, thus hosts the private keys on the server. It is still under development and requires a LEAP-enabled backend server.

LEAP and Pixelated are both developed independently and then deployed by providers such as Riseup.net, calyx.net, codigosur and colnodo.

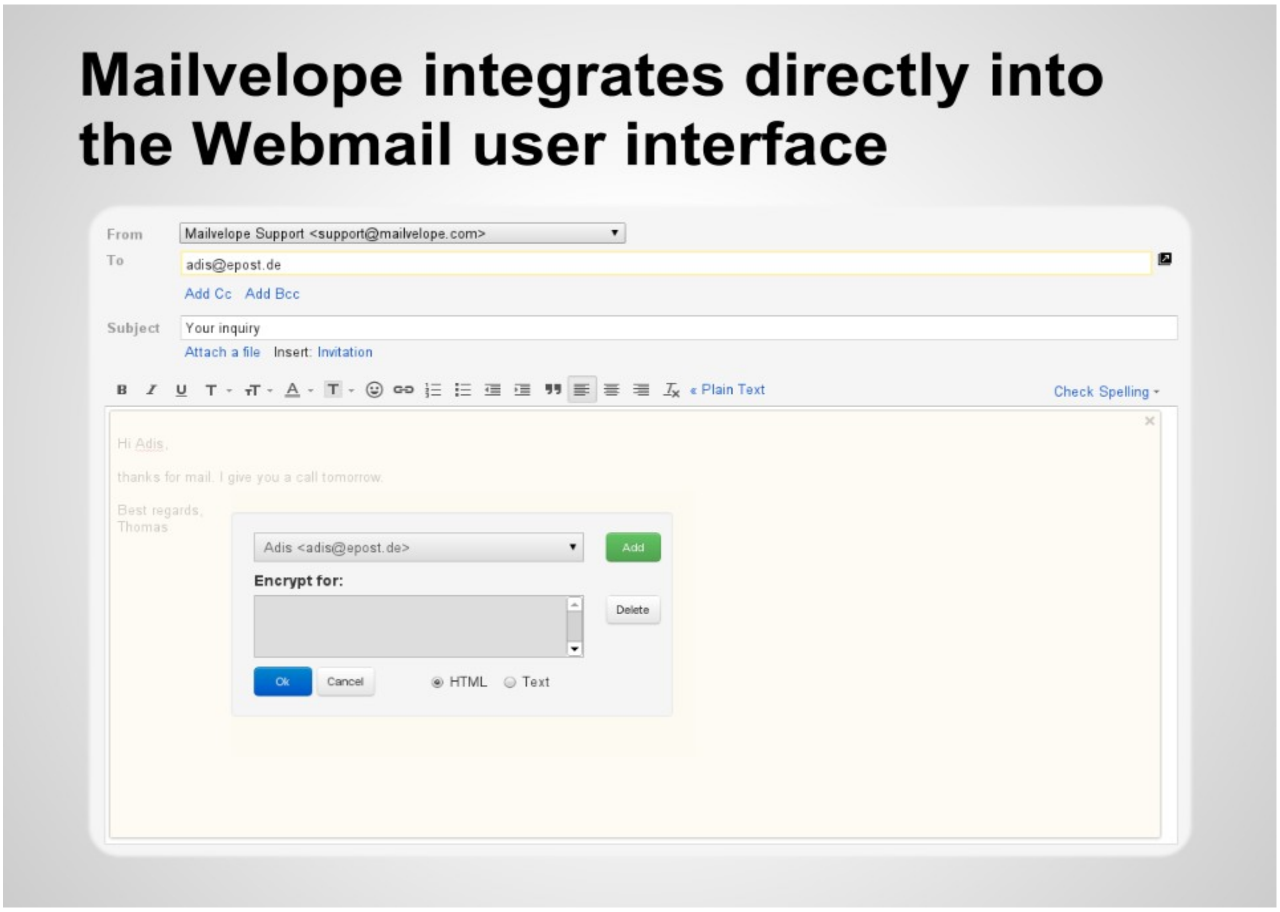
21 https://pixelated-project.org/faq/#how-can-i-keep-myself-safe

22 https://pixelated-project.org/faq/

## 3.7 Mailvelope

Authors: Thomas Oberndörfer, Tankred Hase, Ilian Sapundshiev

https://www.mailvelope.com/


Mailvelope integrates directly into the Webmail user interface

From: Mailvelope Support <support@mailvelope.com>
To: adis@epost.de
Subject: Your inquiry

Hi Adis,

thanks for mail. I give you a call tomorrow.

Best regards,
Thomas

Adis <adis@epost.de>
Encrypt for:

HTML

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes https://github.com/mailvelope/mailvelope |
| (De)Centralization | Decentralized trust model |
| (if decentralized) Federated/P2P? | |
| (if decentralized) Standardized? | Yes |
| Standard(s) | OpenPGP standard<br><br>SMTP standard |
| Operating System(s) | Web-browser plugin (Chrome, Firefox) |

| | |
|---|---|
| | Can be configured to work with arbitrary Webmail provider |
| Data Collection (/Purpose) | Mailvelope stores the keys in the local storage of the browser. The user must verify the data collection parameters of the browser. |
| Number of users | 211 721 users on Chrome |
| Type(s) of users | Activists, journalists (supported by the Knight International Journalism Fellowship), used by companies as well (e.g. Tine20, GMX and WEB.DE) |
| Countries | available in Chinese (China), French, German, Norwegian, Polish, Russian, Spanish<br>Tested in Uganda |
| End-to-end group chat? | no |
| End-to-end voice calls? | no |

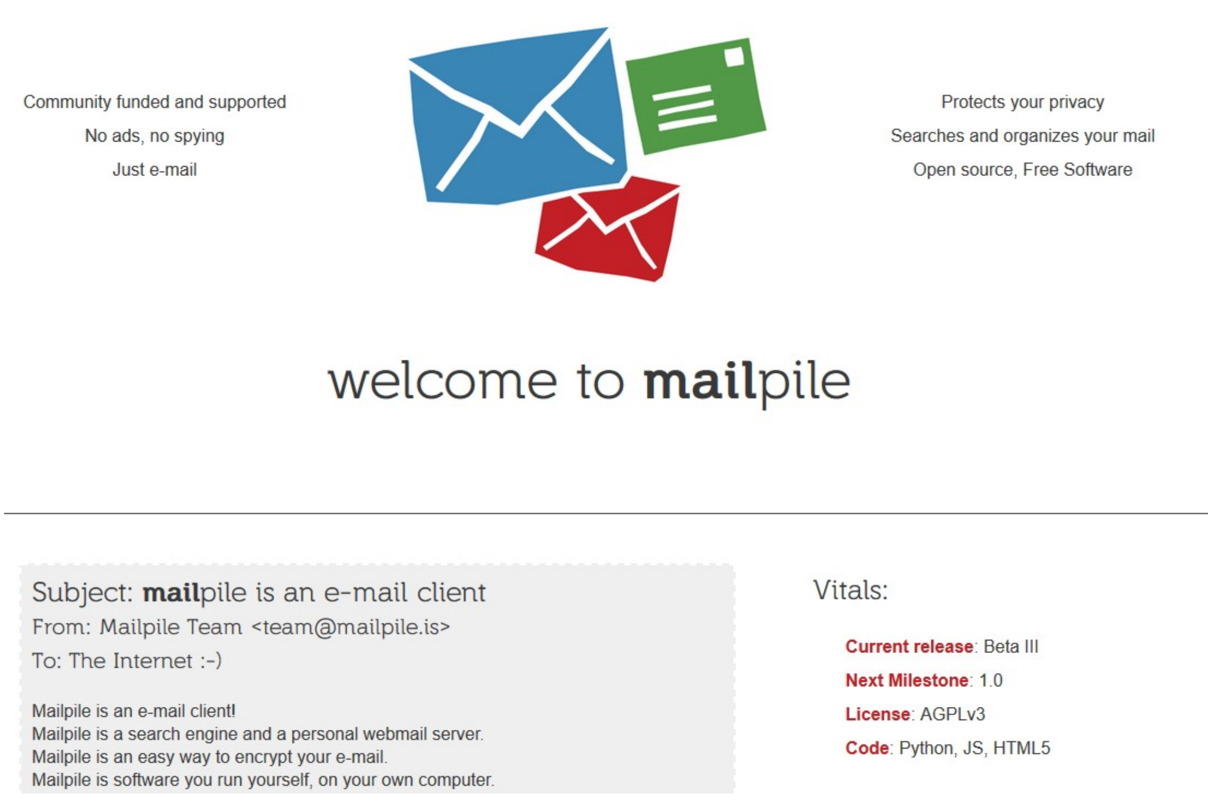
**Discursive description and additional information**

A well-known ancestor of this project is FireGPG developed by Maximilien Cuony and others starting around 2007 and discontinued in 2010. Mailvelope is a new project without dependencies to GPG. It's based on OpenPGP.js which is an open source project that was initially developed by recurity-labs.com. Mailvelope's development started in March 2012 and has been available in the Chrome Web Store since August 2012. The recently published version 1.4 of Mailvelope comes with new features for encrypting files. Since September 2013 it got funding from Open Technology Fund, for its first security audits. Mailvelope is based on the work of the following open source projects: OpenPGP.js, email.js, DOMPurify, Bootstrap, jQuery, Oxygen Icons.

## 3.8 Mailpile

Authors: Bjarni Rúnar Einarsson, Brennan Novak, Smari McCarthy and the Mailpile open source developer team

https://www.mailpile.is/


Community funded and supported
No ads, no spying
Just e-mail
Protects your privacy
Searches and organizes your mail
Open source, Free Software
welcome to mailpile

Subject: **mail**pile is an e-mail client
From: Mailpile Team <team@mailpile.is>
To: The Internet :-)

Mailpile is an e-mail client!
Mailpile is a search engine and a personal webmail server.
Mailpile is an easy way to encrypt your e-mail.
Mailpile is software you run yourself, on your own computer.

Vitals:

**Current release**: Beta III
**Next Milestone**: 1.0
**License**: AGPLv3
**Code**: Python, JS, HTML5

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source, software only<br>https://github.com/mailpile/Mailpile |
| (De)Centralization | Email client is installed and runs on users' computers |
| (if decentralized) Federated/P2P? | Federated |
| (if decentralized) Standardized? | Yes |
| Standard(s) | Plug-in/SMTP standard client<br>Thunderbird PGP replacement, still under development. |

| | |
|---|---|
| Operating System(s) | Linux, OS X, Windows |
| Data Collection (/Purpose) | Mailpile is self-hosted |
| Number of users | N/A |
| Type(s) of users | N/A |
| Countries | Available in 14+ languages |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional info**

Mailpile is an email client (i.e. does not operate email servers or stores private keys) and it has a heavy focus on encryption and privacy features by default.

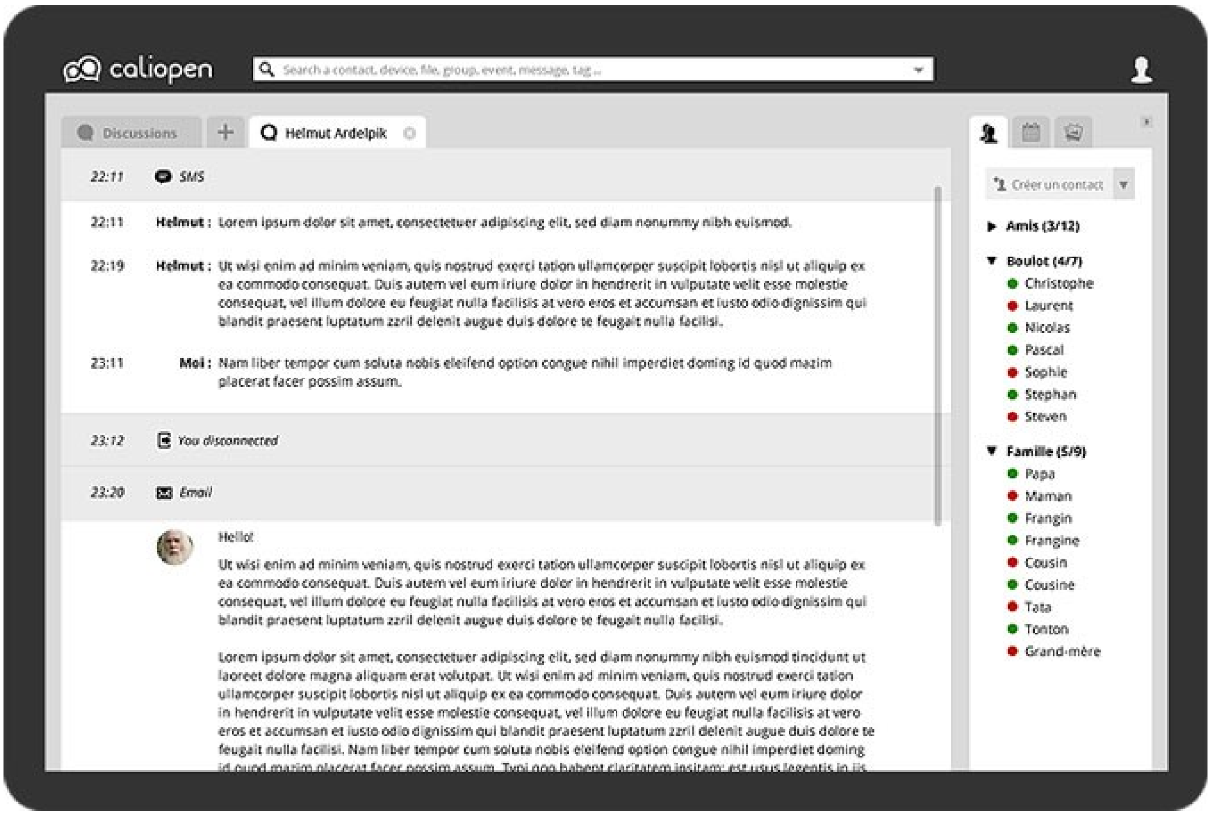
Mailpile natively supports PGP encryption and stores all locally generated files in encrypted form on-disk; thus, it explicitly positions itself against encrypted messaging services that ‘store your e-mail data on their servers’[23]. They also emphasize ease-of-use, as a remedy to what they describe as the ‘most fundamental problem of encryption’: being unable to send encrypted email to someone who does not have an encryption key him/herself. It is still under development.

23 https://www.mailpile.is/faq/#enc-12

## 3.9 Caliopen

Authors: Laurent Chemla (Gandi, Quadrature du Net), Alexis Mineaud, Guillaume Gauvrit, Tim Pillard

https://caliopen.org/


caliopen
Search a contact, device, file, group, event, message, tag ...
Discussions
Helmut Ardelpik
Créer un contact

22:11 SMS

22:11 **Helmut :** Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod.

22:19 **Helmut :** Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

23:11 **Moi :** Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

*23:12 You disconnected*

*23:20 Email*

Hello!

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis

- ▶ **Amis (3/12)**
- ▼ **Boulot (4/7)**
  - 🟢 Christophe
  - 🔴 Laurent
  - 🟢 Nicolas
  - 🟢 Pascal
  - 🔴 Sophie
  - 🟢 Stephan
  - 🔴 Steven
- ▼ **Famille (5/9)**
  - 🟢 Papa
  - 🔴 Maman
  - 🟢 Frangin
  - 🟢 Frangine
  - 🔴 Cousin
  - 🟢 Cousine
  - 🔴 Tata
  - 🟢 Tonton
  - 🔴 Grand-mère

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | OpenSource (GPLv3)<br>https://caliopen.github.io/<br>https://github.com/CaliOpen |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | Is expected to be federated |
| (if decentralized) Standardized? | Yes |
| Standard(s) | SMTP |
| Operating System(s) | Both web and mobile interfaces are expected to be developed |
| Data Collection (/Purpose) | N/A |
| Number of users | N/A as not yet open to beta |

| | |
|---|---|
| Type(s) of users | Project is aimed mostly at tech experts able to build services on top of it. |
| Countries | France |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional information**

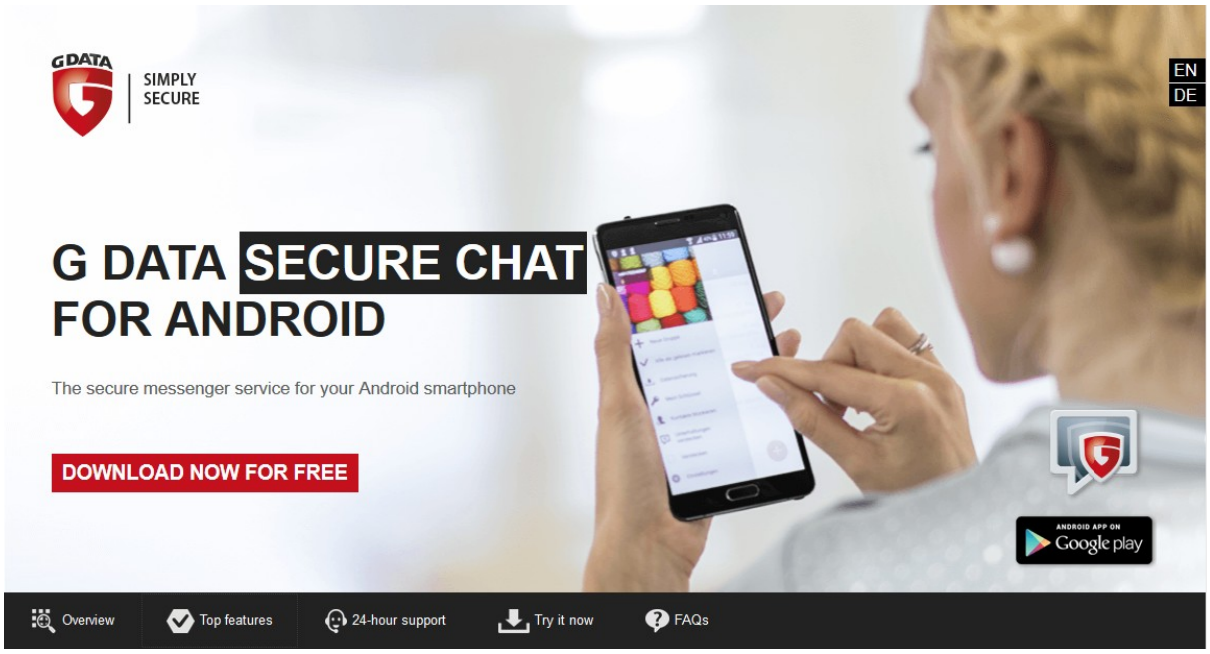
Project launched in 2014 after Snowden revelations and the PRISM case. Supported by Gandi and Quadrature du Net. Presented as a software distribution or “software suite” (not a service in itself). It lets users (structures or institutions) build secure mail clients on top of it. Mission: to make an easy to use service that will help to build and manage secure mailing. A “Caliopen label” is planned in order to deliver labels to the projects that run on Caliopen.

For now, the project is not under development. However, it is discussed in french-speaking Linux community and has participated in OpenPGP summit. The page on GitHub is actively updated.

## 3.10 G Data Secure Chat

Author(s): Germany-based software company G Data

https://www.securechat.com/en


G DATA SECURE CHAT FOR ANDROID
The secure messenger service for your Android smartphone
DOWNLOAD NOW FOR FREE
SIMPLY SECURE
EN
DE
ANDROID APP ON Google play
Overview
Top features
24-hour support
Try it now
FAQs

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Core is open source<br>https://github.com/GDATASoftwareAG<br>In Github, direct fork from TextSecure (Open Whisper Systems) code<br>App as a whole is not open source |
| (De)Centralization | Centralized<br>Germany-based servers |
| Standard(s) | Based on Axolotl/OMEMO |
| Operating System(s) | Android-only |
| Data Collection (/Purpose) | Keys exist only on devices and firm says it has ‘no access at all to your data’. |
| Number of users | N/A, although G Data Software has a facebook page with about 65,000 likes |
| Type(s) of users | Look and content of the website seem to indicated that the service is aimed at a general |

| | |
|---|---|
| | public |
| Countries | Germany-based |
| End-to-end group chat? | Yes (+ encrypted SMS and self-destructing messages) |
| End-to-end voice calls? | No |

**Discursive description and additional information**

The app bears several similarities to Signal, and proposes to import and encrypt existing messaging data for safe keeping. Users (including in groups) are enrolled by performing phone number verification for each account. Ease-of-use is emphasized (by reviewers as well[24]) and users can switch between secure chat (free messaging across secure infrastructure), secure SMS (across carrier infrastructure at the user's cost) and insecure SMS.

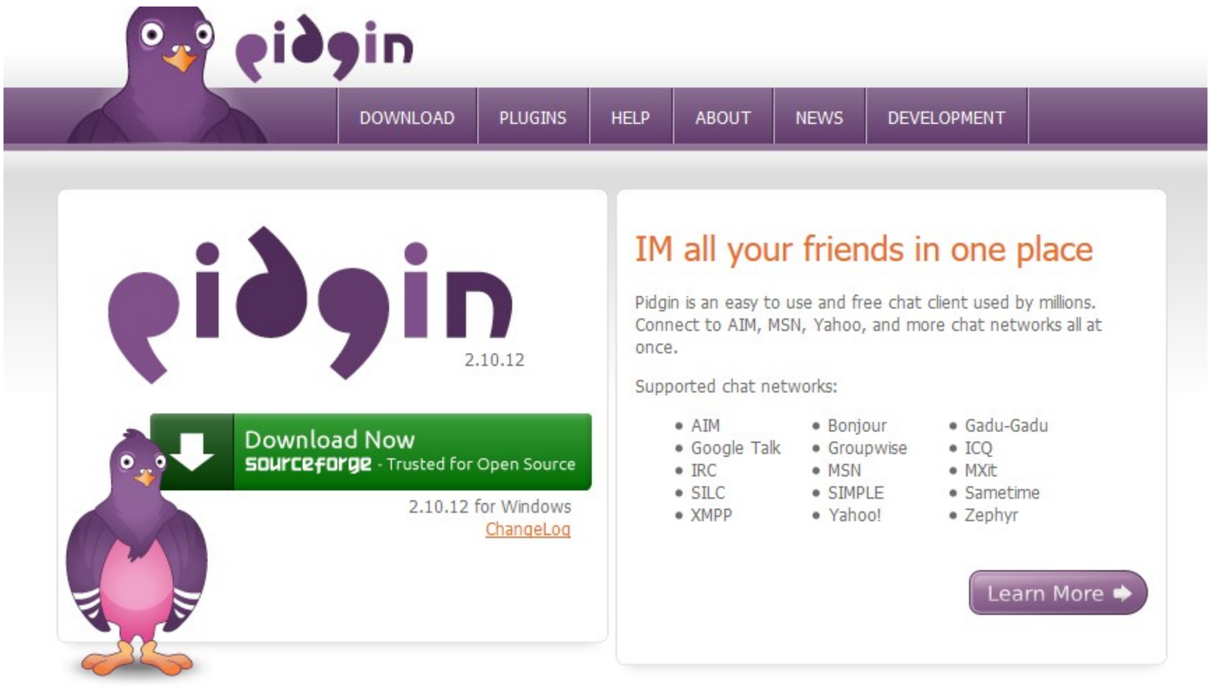
The product seems aimed at a general public (with promo texts like : "A snapshot from holiday, quickly sent via the messenger service – a nice greeting directly on the smartphone instead of a postcard. But does the image really reach only the recipient? Yes! G DATA SECURE CHAT makes sure of it. With its reliable encryption, nobody but you can access your data"). Unlike Signal, that uses in its promo texts activist and anarchist references (i.e. screenshots with Vera Zasulich, Nestor Makhno etc), G Data seems to address a larger, not activist or advocate-only public (possibly a post-Snowden effect?)

24 http://www.pcmag.com/article2/0,2817,2491997,00.asp

## 3.11 Pidgin

Authors: Mark Spencer and contributors

https://pidgin.im/



IM all your friends in one place

Pidgin is an easy to use and free chat client used by millions. Connect to AIM, MSN, Yahoo, and more chat networks all at once.

Supported chat networks:

- AIM
- Google Talk
- IRC
- SILC
- XMPP
- Bonjour
- Groupwise
- MSN
- SIMPLE
- Yahoo!
- Gadu-Gadu
- ICQ
- MXit
- Sametime
- Zephyr

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://www.pidgin.im/download/ |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | Federated |
| (if decentralized) Standardized? | Yes |
| Standard(s) | Based on **libpurple**, a library that has support for many instant messaging applications<br>XMPP+OTR |
| Operating System(s) | Linux, Windows (Adium for OS X) |
| Data Collection (/Purpose) | N/A |

| | |
|---|---|
| Number of users | Estimated as over 3 million |
| Type(s) of users | All types |
| Countries | Translated into 60+ languages[25] |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes, using Farstream, then through XMPP |

**Discursive description and additional info**

Pidgin is widely used for its Off-The-Record Messaging (OTR) plugin, which provides the end-to-end encryption.

In 2015, Pidgin scored seven out of seven points on the Electronic Frontier Foundation's secure messaging scorecard. The points were awarded for:

- having communications encrypted in transit
- having communications encrypted with keys the providers don't have access to,
- enabling users to independently verify their correspondent's identities,
- having past communications secure if the keys are stolen,
- code open to independent review
- having their security designs well-documented
- having recent independent security audits

However, Wired pointed out that the underlying libpurple codebase that Pidgin is based on is "known for its bountiful security bugs even if OTR itself is not[26]".

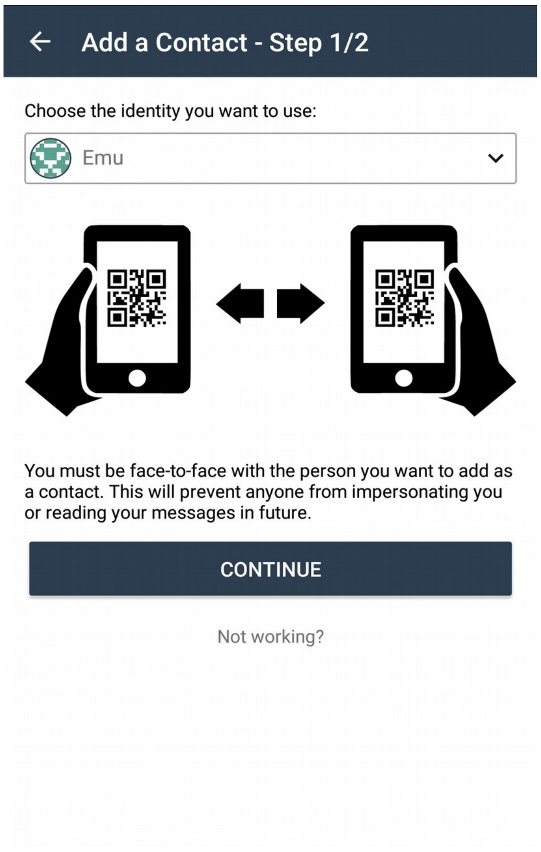
25 https://pidgin.im/about/

26 https://www.wired.com/2015/10/Tor-just-launched-the-easiest-app-yet-for-anonymous-encrypted-im/

## 3.12 Briar

Authors: Michael Rogers (Freenet, Limewire), Eleanor Saitta, Bernard Tyers (Open Rights Group, Tor), Ximin Luo (Tor, Debian, Freenet), Torsten Grote (Free Software Foundation)

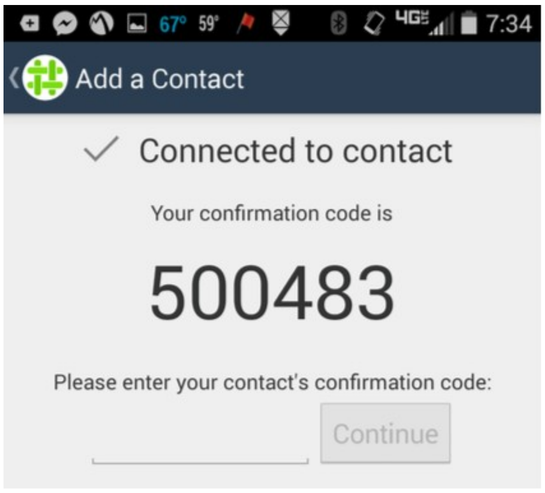
https://briarproject.org


Add a Contact - Step 1/2

Choose the identity you want to use:

Emu

You must be face-to-face with the person you want to add as a contact. This will prevent anyone from impersonating you or reading your messages in future.


Add a Contact

Connected to contact

Your confirmation code is

500483

Please enter your contact's confirmation code:

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | OpenSource<br>https://code.briarproject.org/akwizgran/briar/tree/master |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | P2P |
| (if decentralized) Standardized? | not |
| Standard(s) | Built on top of Tor Hidden Services and on Bramble |

| | |
|---|---|
| Operating System(s) | Android version currently in beta |
| Data Collection (/Purpose) | N/A |
| Number of users | N/A |
| Type(s) of users | “Designed for activists, journalists and everyone else who needs a safe, easy and robust way to communicate” |
| Countries | N/A |
| End-to-end group chat? | Provides encrypted forums |
| End-to-end voice calls? | N/A |

**Discursive description and additional information**

Briar is a messaging app designed for activists, journalists, and anyone else who needs a safe, easy and robust way to communicate. Briar doesn't rely on a central server - messages are synchronized directly between the users' devices. If the Internet's down, Briar can sync via Bluetooth or Wi-Fi, keeping the information flowing in a crisis. If the Internet's up, Briar can sync via the Tor network. Briar has received funding from Small Media, the Open Internet Tools Project, Access and the Open Technology Fund. However, the app is not yet accessible through the App stores. Runs over Bramble.[27]

“The original use cases for Briar have always been as a communications tool for people at risk, whether they're journalists, political activists, folks trying to work together to organize a union, or sex workers or domestic violence victims who need to check in with friends to stay safe.”[28]

The vision of the project as exposed by one of its founders, Eleanor Saitta, is not about building fixed solution but more about providing kits for communities in order to give them a possibility to create their own tools according to their needs and ideas. It is about decentralizing communities: “When we decentralize not just a technical system but a social process at the core of managing collective infrastructure, we spread agency out through the community, give people a sense of ownership, and change their social relationship with the system.”[29]

27 https://dymaxion.org/essays/briarvision.html

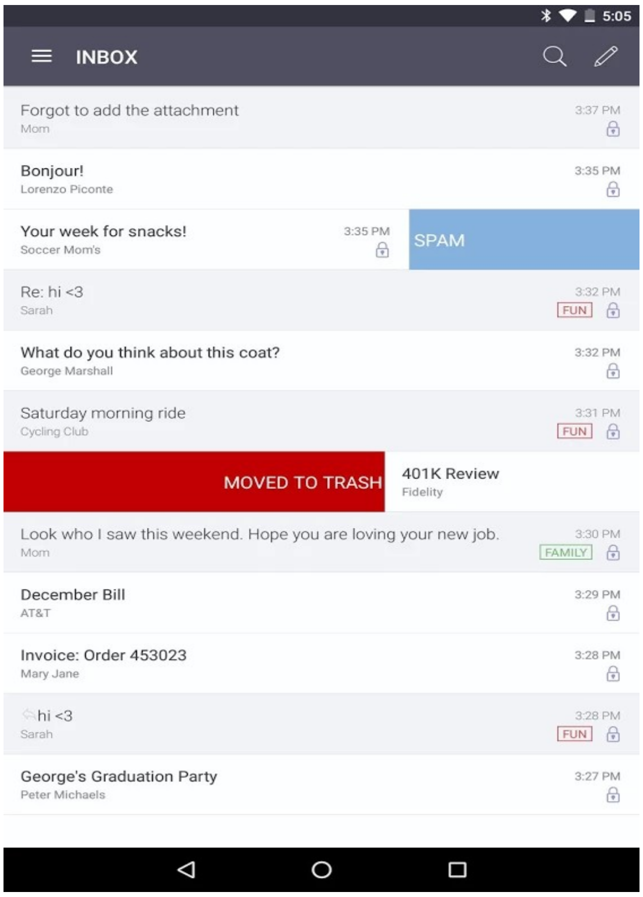
28 Ibid.

29 Ibid.

## 3.13 Protonmail

Authors: Jason Stockman, Andy Yen and Wei Sun

https://protonmail.com


5:05

INBOX

Forgot to add the attachment
Mom
3:37 PM

Bonjour!
Lorenzo Piconte
3:35 PM

Your week for snacks!
Soccer Mom's
3:35 PM
SPAM

Re: hi <3
Sarah
3:32 PM
FUN

What do you think about this coat?
George Marshall
3:32 PM

Saturday morning ride
Cycling Club
3:31 PM
FUN

MOVED TO TRASH
401K Review
Fidelity

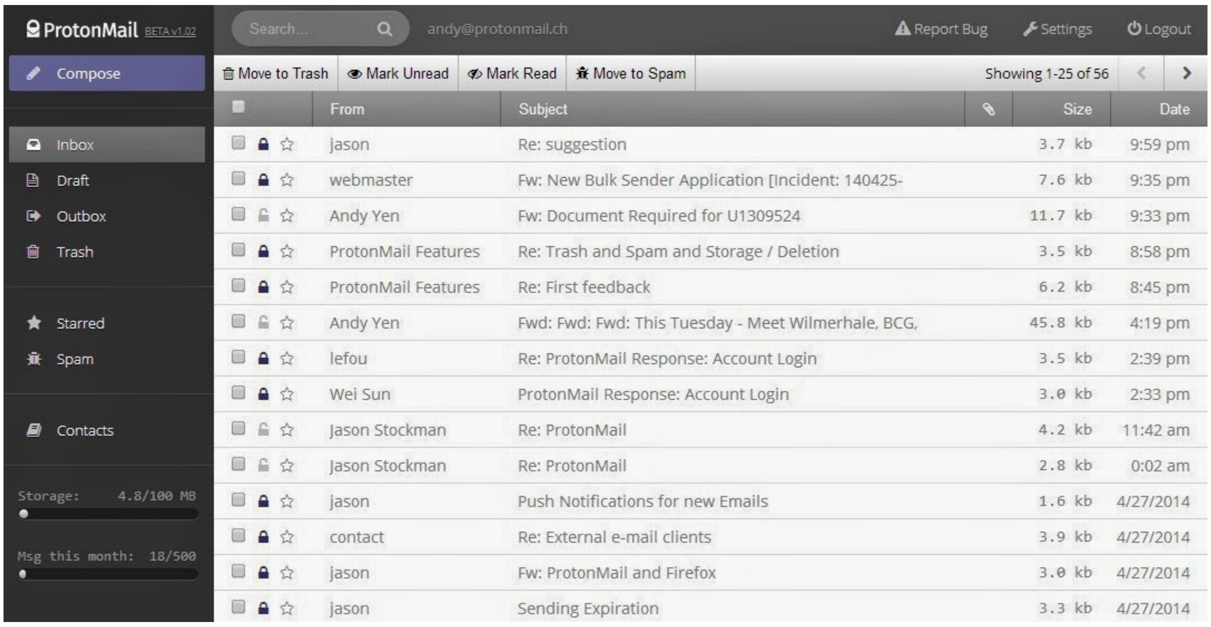
Look who I saw this weekend. Hope you are loving your new job.
Mom
3:30 PM
FAMILY

December Bill
AT&T
3:29 PM

Invoice: Order 453023
Mary Jane
3:28 PM

hi <3
Sarah
3:28 PM
FUN

George's Graduation Party
Peter Michaels
3:27 PM



| From | Subject | Size | Date |
|---|---|---|---|
| jason | Re: suggestion | 3.7 kb | 9:59 pm |
| webmaster | Fw: New Bulk Sender Application [Incident: 140425- | 7.6 kb | 9:35 pm |
| Andy Yen | Fw: Document Required for U1309524 | 11.7 kb | 9:33 pm |
| ProtonMail Features | Re: Trash and Spam and Storage / Deletion | 3.5 kb | 8:58 pm |
| ProtonMail Features | Re: First feedback | 6.2 kb | 8:45 pm |
| Andy Yen | Fwd: Fwd: Fwd: This Tuesday - Meet Wilmerhale, BCG, | 45.8 kb | 4:19 pm |
| lefou | Re: ProtonMail Response: Account Login | 3.5 kb | 2:39 pm |
| Wei Sun | ProtonMail Response: Account Login | 3.0 kb | 2:33 pm |
| Jason Stockman | Re: ProtonMail | 4.2 kb | 11:42 am |
| Jason Stockman | Re: ProtonMail | 2.8 kb | 0:02 am |
| jason | Push Notifications for new Emails | 1.6 kb | 4/27/2014 |
| contact | Re: External e-mail clients | 3.9 kb | 4/27/2014 |
| jason | Fw: ProtonMail and Firefox | 3.0 kb | 4/27/2014 |
| jason | Sending Expiration | 3.3 kb | 4/27/2014 |

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source clients, but closed source server code. |
| (De)Centralization | Centralized |
| Standard(s) | ProtonMail has its own custom encryption format, but claims to be working on  support for the OpenPGP standard. |
| Operating System(s) | Android, iOS, Windows, OS X |
| Data Collection (/Purpose) | Protonmail does not save any tracking information. By default, it does not record metadata such as the IP addresses used to log into accounts. To protect user privacy, ProtonMail does not require any personally identifiable information to register. |
| Number of users | Between 100 000 and 500 000 on Google Play<br>Had about 1 000 000 users[30] by December 2015. |
| Type(s) of users | General public, activists |
| Countries | Switzerland, France (Nuit Debout), Russia… |

30 According to Steier, Henning (16 December 2015). "Protonmail Ende Januar offen für alle". *Neue Zürcher Zeitung* (in German).

| | |
|---|---|
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description/any additional info**

This project provides encrypted email on the server, based on the model of Lavabit, Snowden's original email provider. However, this approach has been criticized by Moxie Marlinspike as effectively pointless and not providing any security, since the server still has the keys.[31] However, Protonmail claims that being in a Swiss jurisdiction provides more protection. More recently, they claim also to be moving towards end-to-end encryption where the server doesn't have the keys (based on PGP), but this is still under development.

One interesting feature is that a user can set an optional expiration time on ProtonMail's encrypted emails, so they will be automatically deleted from the recipient's inbox once they have expired. This technology works for both emails sent to other ProtonMail users, and encrypted emails sent to non-ProtonMail email addresses."

31 https://moxie.org/blog/lavabit-critique/

## 3.14 Jitsi

Author(s): Emil Ivov and contributors[32]

https://jitsi.org/

jitsi.org
JITSI
DOWNLOAD
DEVELOPMENT
DOCUMENTATION
SUPPORT
SECURE COMMUNICATIONS
Secure audio and video calls. All your calls through Jitsi are encrypted and completely secure. Enable the encrypted chat to start a private conversation with your friends.
DOWNLOAD JITSI
Start using it now
JITSI MEET
WEB CONFERENCES
JITSI
SECURE VIDEO CALLS AND CHAT
JITSI VIDEOBRIDGE
VIDEO RELAYING FOR EFFICIENT CONFERENCES
LIBJITSI
THE JITSI MEDIA LIBRARY

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Free and open source<br>https://github.com/jitsi/jitsi<br>Released under the Apache licence |
| (De)Centralization | Yes |
| (if decentralized) Federated/P2P? | Yes |
| (if decentralized) Standardized? | Yes |
| Standard(s) | MSNP (Microsoft messaging service), OSCAR (AIM/ICQ/MobileMe), SIP/SIMPLE, XMPP, YMSG, ZRTP (Zimmerman's standard) |

32 https://jitsi.org/Development/TeamAndContributors

| | |
|---|---|
| | Could support P2P as it handles IPv6 |
| Operating System(s) | Linux, Mac OS X, Windows. Beta packages built for Android available. |
| Data Collection (/Purpose) | Jitsi emphasizes that content of the conversation is private, but metadata are still visible to any intercepting party. |
| Number of users | N/A |
| Type(s) of users | N/A, although Jitsi remarks that "Throughout the years our community has grown to include members and contributors from Brazil, Bulgaria, Cameroon, China, Estonia, France, Germany, India, Japan, Romania, Spain, Switzerland, UK, USA, and others." |
| Countries | Available in Austrian, English, French, German, Bulgarian, Japanese, Spanish, Italian, Romanian, Greek and 25 more |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes |

**Discursive description and additional information**

Jitsi is an audio/video Internet phone and instant messenger written in Java. It supports some of the most popular instant messaging and telephony protocols such as SIP, Jabber/XMPP (and hence Facebook and Google Talk), AIM, ICQ, MSN, Yahoo! Messenger.

The development of Jitsi was started by Emil Ivov at the time he was a student at the University of Strasbourg, France. Originally the project was known as SIP Communicator.

Jitsi presents its end-to-end encryption as a feature that "prevents eavesdropping by automatically encrypting your conversation for the full duration of its trip across the Internet", and emphasizes how this means that that even your voice calling service (Google, Facebook, Jit.si, etc.) cannot listen to the contents of your call. Jitsi follows up by presenting straight away the limits of their approach, such as:

- Metadata: The content of the conversation is private, but information about the call (such as the accounts and IP addresses involved, and the length and time of the call) are still visible to any intercepting party, and often stored by your voice calling service.
- Lapses in verification: If users do not verbally confirm the codes at the beginning of the conversation, it is possible for an eavesdropper to go unnoticed. Also it is possible that the

person on the other end is not who you think they are. This is particularly true if you do not know the person well enough to recognize by voice.

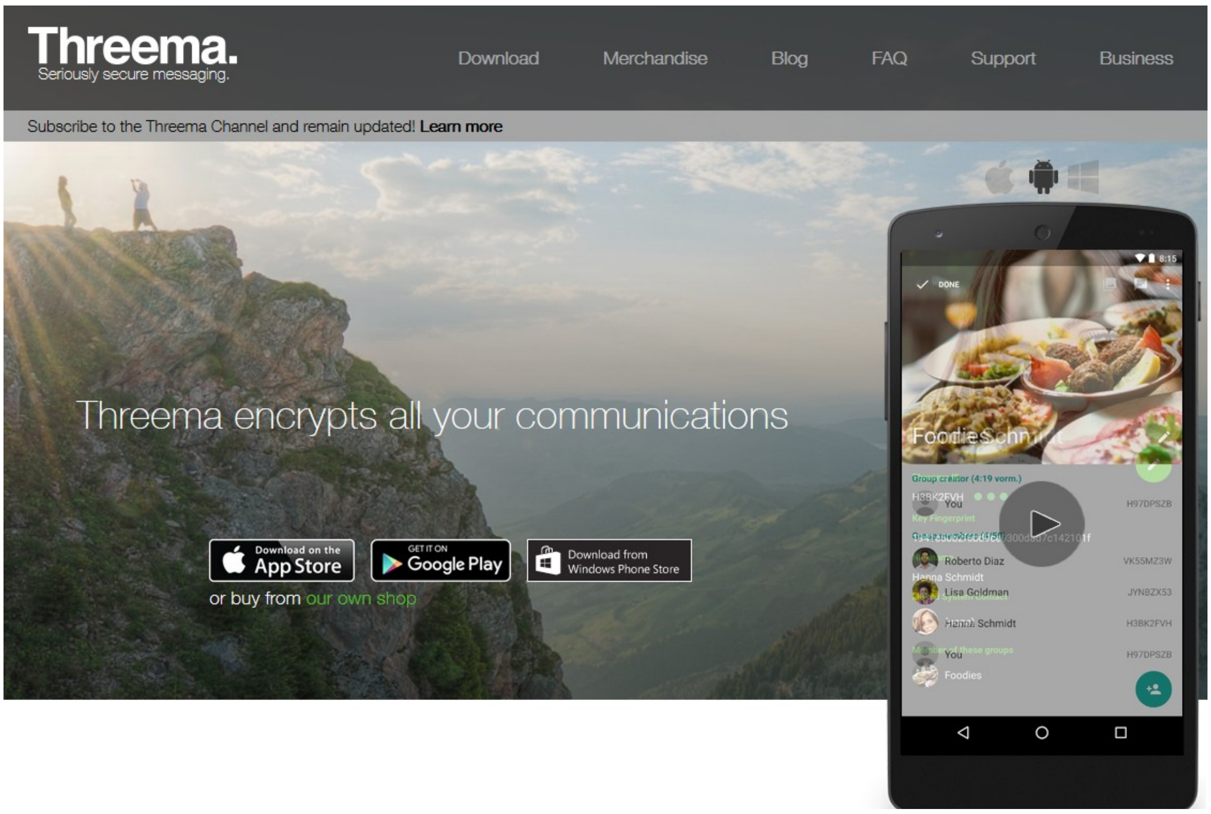
- Vulnerabilities on either end: End-to-end encryption protects data in transit, but it cannot ensure the security of the device, computer, or person on either end of the call.

In 2014, Jitsi scored 6 out of 7 points on the EFF's secure messaging scorecard.

## 3.15 Threema

Author(s): Manuel Kasper, Martin Blatter, Silvan Engeler, Threema GmbH

https://threema.ch/en


Threema.
Seriously secure messaging.
Download
Merchandise
Blog
FAQ
Support
Business
Subscribe to the Threema Channel and remain updated! Learn more
Threema encrypts all your communications
Download on the App Store
GET IT ON Google Play
Download from Windows Phone Store
or buy from our own shop

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Closed source, proprietary |
| (De)Centralization | Centralized, Threema servers<br>(however, key pairs generated in decentralized way on user device) |
| Standard(s) | Based on the NaCl library (open source) Uses asymmetric 256-bit ECC-based encryption. |
| Operating System(s) | iOS, Android, Windows Phone |
| Data Collection (/Purpose) | The company says that "Using Threema ought to generate as little data on servers as possible [...] For that reason, data like e.g. contacts or group chats are stored in a decentralized way on user devices, instead of on a Threema server. |

| | |
|---|---|
| | Threema servers assume the role of a switch; messages and data get forwarded, but not permanently stored. The following types of data get temporarily stored to enable asynchronous communication: messages and group chats, anonymized addresses and phone numbers from address book.<br><br>Key pairs are generated in a decentralized way on user device.<br><br>Threema doesn't log which Threema IDs are communicating.<br><br>Offers "Validation Logging" feature to confirm that messages are end-to-end encrypted. |
| Number of users | 3.5 million users as of 2015 |
| Type(s) of users | Varied, most of them from German-speaking countries |
| Countries | Germany and Switzerland mainly |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes, voice messages. |

**Discursive description and additional information**

Swiss-based end-to-end encrypted messenger that supports file exchange, photos, videos, voice messages, QR codes, polls and most types of files.

As of November 2015, Threema has a score of 6 out of 7 points on the EFF’s secure messaging scorecard, only losing a point because of the closed source code. The code has, however, recently sustained an independent external audit.

The company points out that Threema's encryption code is open to independent review as it is based on NaCI library, and provides the Validation Logging feature.[33] It adds: “Making the app's code open to the public is out of question for Threema. By disclosing the source code, Threema would divulge its asset. [...] In the end, a certain amount of trust in the provider of a secure communication solution is inevitable – just as one also has to trust the developer of the platform on which Threema is installed. This of course goes against best practice in security research and so Threema’s decision to not let the research community scrutinize its code is ridiculed as “security by obscurity.”

33 https://threema.ch/validation

## 3.16 Pond

Author: Adam Langley (Senior Staff Software Engineer, Google)

https://www.imperialviolet.org/ (dedicated blog section, https://pond.imperialviolet.org/, is no longer accessible from a web browser, but archives available on GitHub (https://github.com/agl/pond)

Screenshot N/A

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://github.com/agl/pond<br>In stasis (although not shut down) |
| (De)Centralization | Federated<br><br>Group signature scheme that allows your pondserver (analogous to mailserver) to verify that the person ponding you is a member of your allowed group, but cannot verify who exactly in the group it is.<br><br>A default server is still active. |
| Standard(s) | N/A; message transport provided over Tor<br>OTR |
| Operating System(s) | Darwin (base for OS X), Linux, FreeBSD |
| Data Collection (/Purpose) | A server can learn when a user is online.<br><br>A server can learn how many messages a user receives, and when they receive them.<br><br>A server can ensure that only authorised users can communicate with a Pond user to prevent spam.<br><br>A server knows a group public key for each user that it accepts mail for.<br><br>The network can know that a given host is running Pond.<br><br>“We save all state in a single file that is encrypted with a random session key and overwritten for every update”.<br>All messages are deleted a week after the |

| | |
|---|---|
| | reception |
| Number of users | N/A |
| Type(s) of users | N/A though in the current state of the project we can suppose that it aims at mostly tech-savvy users |
| Countries | Mostly United States |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional information**

Pond is a "one-man-show" application developed by Adam Langley, a Senior Staff Software Engineer at Google.

Pond focused a lot on anonymity and privacy properties. Pond deliberately has no kind of identity system, with the communication between two parties being entirely anonymous. Each pairing between two Pond users starts with a "shared secret": because Pond has no accounts or identities, the name of the contact is chosen solely by the sender.

However, its difficulty of use was debated.

Pond was designed to be a "secure by default" system. For example, it has no non-secure version to fall back to and message transport is provided over the Tor network, masking sender and recipient identities and locations.

Pond deals with the 'first contact' issue in the following way: communication can only occur between two people who've already agreed to communicate. That two-way communication is established in the first place by mutual agreement of a passphrase, which can be communicated through IM, voice chat, in person, or however else two parties want to share it (a method called "PANDA"). Both parties post a (time-limited) message derived from the passphrase to a server and use this to share keys with each other. Once this key sharing has taken place, the passphrase isn't used, and even if an attacker discovers it, it's no longer useful[34].

The dedicated GitHub repository reads as of May 2016:

34 This process is not described by Pond direct sources but by an Ars Technica commentary: http://arstechnica.com/security/2014/10/the-secure-smartphone-that-wont-get-you-beaten-with-rubber-hoses/

“*Pond is in stasis*, and has been for several years. I hope that some of the ideas prove useful in the future, but people should use something better polished and reviewed[35]. I've no plans to shutdown down the default server, but **new users should look elsewhere**.”

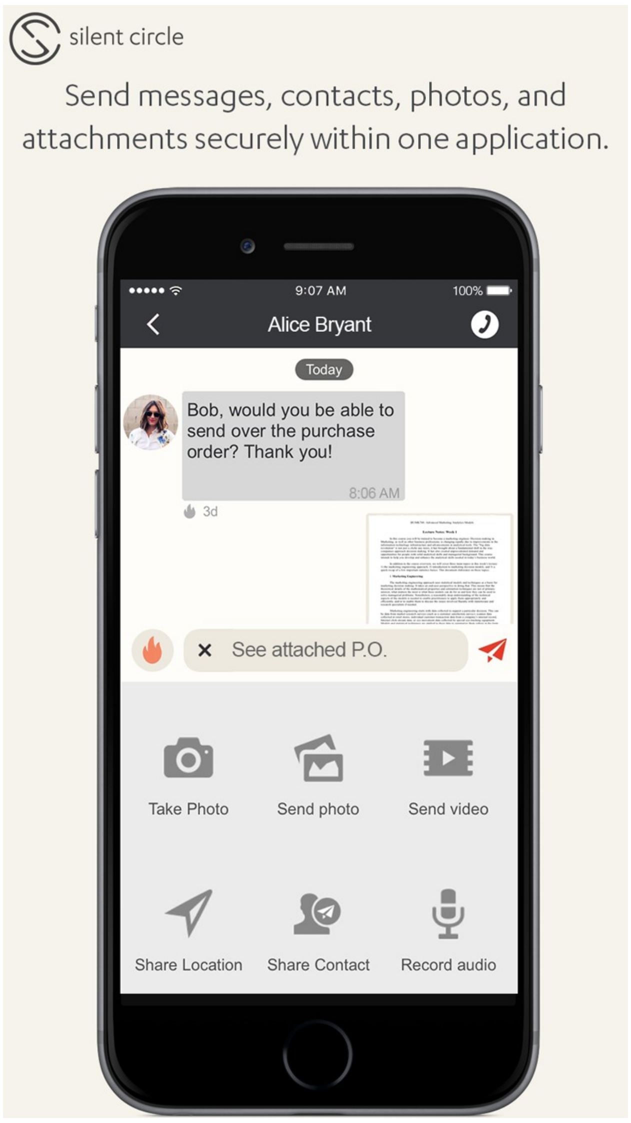
Although Pond is itself in stasis, it has provided the foundations for several experimentations with anonymity and privacy features in instant messaging.

35 Here, Pond directs to Open Whisper Systems.

## 3.17 Silent Circle / Silent Phone

Authors: Phil Zimmerman (PGP), Jon Callas (Apple, Tesla, Kroll-O'Gara, Counterpane, and Entrust), Mikre Janke, Matt Neiderman

https://www.silentcircle.com


silent circle
Send messages, contacts, photos, and attachments securely within one application.
9:07 AM
100%
Alice Bryant
Today
Bob, would you be able to send over the purchase order? Thank you!
8:06 AM
3d
See attached P.O.
Take Photo
Send photo
Send video
Share Location
Share Contact
Record audio

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Partly https://github.com/SilentCircle |
| (De)Centralization | No |
| (if decentralized) Federated/P2P? | No |
| (if decentralized) Standardized? | |

| | |
|---|---|
| Standard(s) | Axolotl (Signal variant) for chat.<br>ZRTP protocol[36] for voice. |
| Operating System(s) | iOS, Android, SilentOS |
| Data Collection (/Purpose) | Track users behavior on their website<br>Collect information using online form submissions |
| Number of users | N/A |
| Type(s) of users | Mostly companies-oriented and government-oriented service.<br><br>However, in their case-study[37] they also speak about a possibility to deploy Silent Circle solutions for “global media companies” and reporters working in “hostile environments” |
| Countries | Switzerland, USA |
| End-to-end group chat? | yes |
| End-to-end voice calls? | Yes (up to 6 calling)<br>Video calls possible |

**Discursive description/any additional info**

Silent Circle provides both software (Silent Phone App) and hardware (Blackphone) solutions for secure communication. In spring 2016 they released the Blackphone 2, a mobile phone that provides device encryption by default through harnessing the company’s operating system (Silent OS) and Google Android technology. The Blackphone 2 currently retails at $799. Software packages are from $9.99 to $39.99 per month[38].

At the heart of Silent OS, Blackphone 2 has a built-in Security Center that enables users to easily manage their privacy and security settings in one place. Users can control and fine-tune the individual app permissions and the data to which the apps have access.

36 ZRTP - Zimmermann Real-time Transport Protocol was developed by Silent Circle's Phil Zimmermann. ZRTP is a cryptographic key-agreement protocol to negotiate the keys for encryption between two end points in a Voice over Internet Protocol (VoIP) phone telephony call based on the Real-time Transport Protocol. It uses Diffie–Hellman key exchange and the Secure Real-time Transport Protocol (SRTP) for encryption.

37 https://www.silentcircle.com/uploads/misc/SilentCircle_Case-Studies.pdf

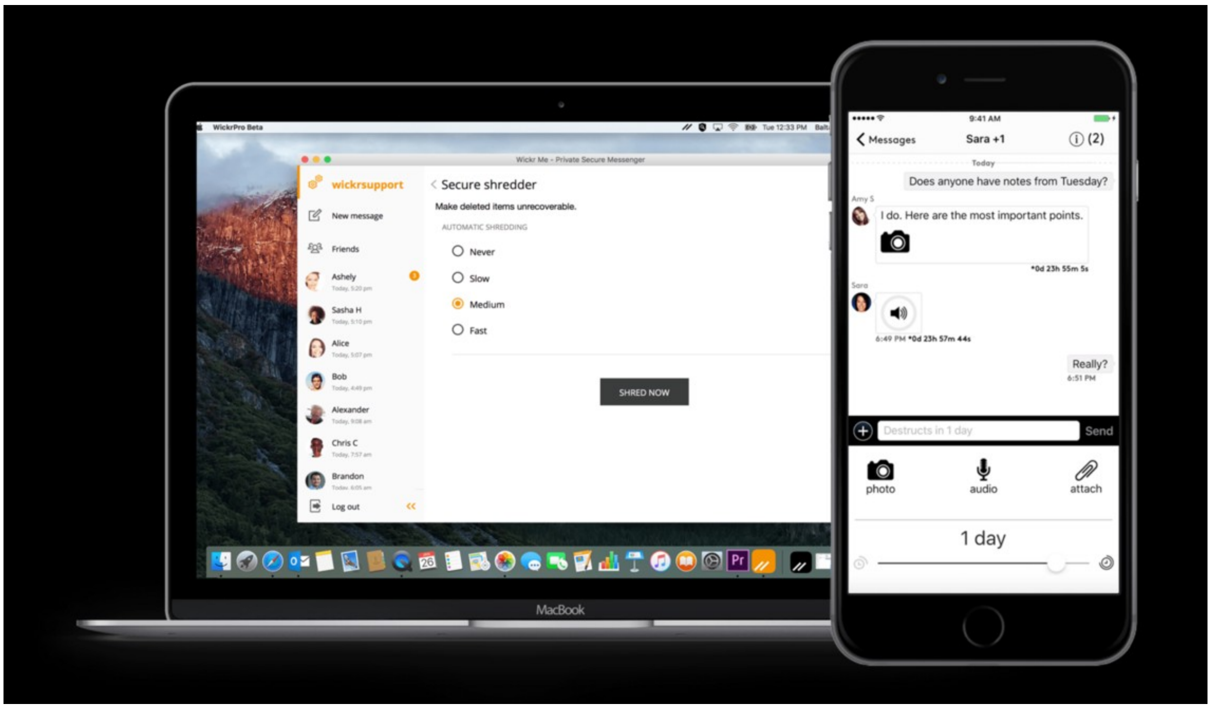
38 According to http://www.gbnews.ch/english/silent-circle-a-specialist-in-encrypted-mobile-services

Silent App can send and receive large files up to 100MB per message. It also proposes  “scheduled burn functionality” that lets choose when messages get deleted on both ends. Proposes packages for entreprises (protecting message exchange, voice calls …).

## 3.18 Wickr

Authors: Nico Sell, Mark Fields, Fitz John Flynn, Chris Howell, Braden Seely, Jennifer Detrani, York Sell, Darlene Miranda (Wickr Labs)

https://wickr.com



Secure shredder

Make deleted items unrecoverable.

AUTOMATIC SHREDDING

Never
Slow
[✓ Medium]
Fast

Today

Does anyone have notes from Tuesday?

Amy S: I do. Here are the most important points.
*0d 23h 55m 5s

Sara
6:49 PM *0d 23h 57m 44s

Really?
6:51 PM

Destructs in 1 day

1 day

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Closed<br>Has three patents:<br>U.S. Patent No. 8,625,805 – Digital Security Bubble<br>U.S. Patent No. 8,707,454 – Multi Party Messaging<br>U.S. Patent No. 8,954,726 – Digital Security Bubble |

| | |
|---|---|
| (De)Centralization | No. |
| (if decentralized) Federated/P2P? | Said to be P2P but...[39] |
| (if decentralized) Standardized? | |
| Standard(s) | No. |
| Operating System(s) | iOS, Android, OS X, Windows, Linux |
| Data Collection (/Purpose) | ● Date an account was created<br>● Type of device(s) on which such account was used<br>● Date of last use<br>● Total number of sent/received messages<br>● A ciphertext copy of any undelivered encrypted messages which have not been deleted yet from our server, which we are not able to decrypt<br>● Privacy list mode setting (allow/block list)<br>● Number of users on the privacy list<br>● Number of external ID's (email addresses and phone numbers) connected to the account, but not the plaintext external IDs themselves<br>● Limited records of recent account activity (does not include data and message content or routing and delivery information)<br>● Wickr version number<br><br>Aggregate usage data: anonymous information about basic usage statistics, such as the number of messages sent by all Wickr users daily, what types of messages our users tend to send (e.g., voice messages more often than text), and so forth[40]. |
| Number of users | Android app: between 500 000 and 1 000 000 downloads<br>N/A for other platforms |
| Type(s) of users | General public and recently (since Feb 2016) enterprises and governments |
| Countries | USA (Chicago, NJ) |

39 https://wickr.com/uploads/files/7008696031631791 65-wickr-whitepaper-final.pdf

40 https://wickr.com/privacy-policy#whatwecollect

| | |
|---|---|
| End-to-end group chat? | Yes, up to 10 users |
| End-to-end voice calls? | Yes |

**Discursive description and additional information**

End2end encrypted messenger, using “multiple-layers” encryption and decentralized key distribution. Since they are not open source, you have to rely on their description, which speaks of AES-256 and ECDH521 to ensure a “double encryption” of the message as well as the keys to read it. All this is put in a TLS-standard “box.” Given that the codebase uses a ‘proprietary algorithm’ and is not open source, they are not providing any real security guarantees.

Provides two kinds of solutions: a professional (“Wickr professional”, since February 2016) and a basic level (“Wickr messenger” - encrypting audio, video, voice, and text messages).

Mobile app notifies if someone takes a screenshot of a conversation on his or her iOS device (function not possible on the desktop version). “Our mission is to help organizations and individuals around the world ensure the privacy of their intellectual property and communications, regardless of the data format.  To this end, we build state-of the-art technology and tools to safeguard the integrity and security of personal and business data—in transit and at rest—with greater efficacy than any other solution in the market”[41].

As of January 5, 2015, Wickr has a score of 5 out of 7 points on the Electronic Frontier Foundation's secure messaging scorecard. It has received points for having communications encrypted in transit, having communications encrypted with keys the provider doesn't have access to (end-to-end encryption), making it possible for users to independently verify their correspondent's identities, having past communications secure if the keys are stolen (forward secrecy), and having completed a recent independent security audit. It is missing points because its code is not open to independent review (open source), and because its security design is not well-documented[42].

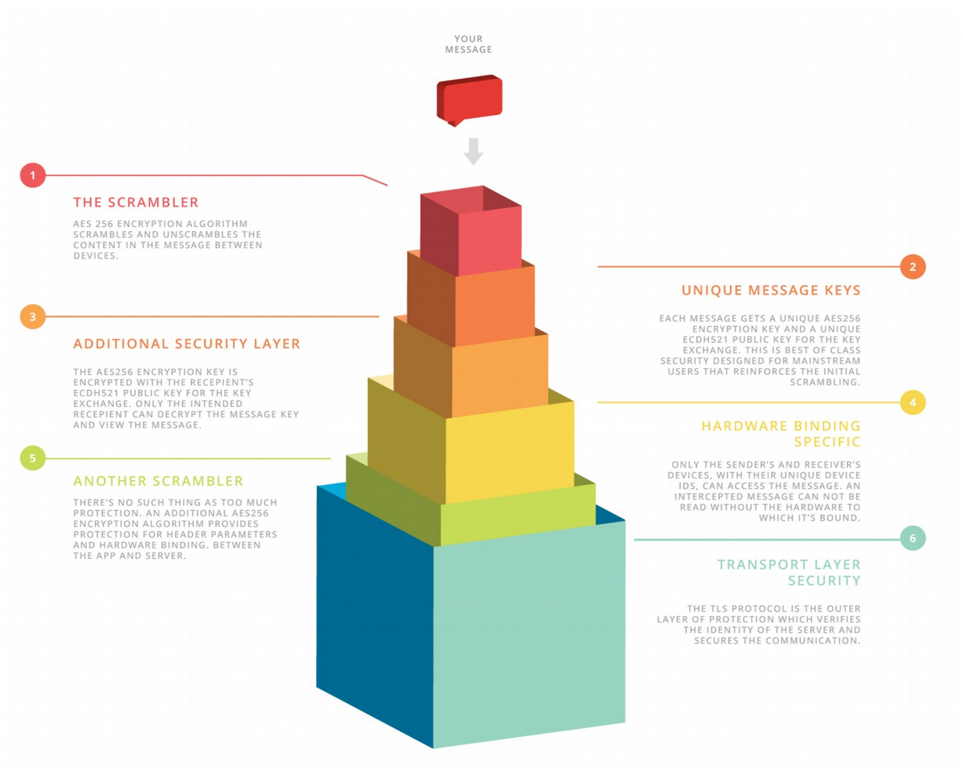
Wickr has reacted publicly to the case Apple vs FBI, publishing a statement that states Wickr is ready to reveal certain data if a proper request has been addressed by the government: “At a time of emerging terrorist threats, we strongly support law enforcement’s mission to stop extremists through all means available within the bounds of the law. The Wickr team has developed comprehensive Law Enforcement Guidelines to assist federal and local agencies in understanding how our platform works and what information can potentially be provided in the course of a lawful investigation”[43]. “Wickr accepts service of court orders, search warrants, and subpoenas for information by email from law enforcement and government agencies, provided that these legal requests are sent from an official

41 https://wickr.com/about-us

42 https://www.eff.org/secure-messaging-scorecard

43 https://wickr.com/about-us/blog/2016/01/04/wickr-transparency-report

government email address of the requesting agent. Law enforcement and/or government agencies should submit legal requests directly from their official government email address.”[44]



THE SCRAMBLER

AES 256 ENCRYPTION ALGORITHM SCRAMBLES AND UNSCRAMBLES THE CONTENT IN THE MESSAGE BETWEEN DEVICES.

UNIQUE MESSAGE KEYS

EACH MESSAGE GETS A UNIQUE AES256 ENCRYPTION KEY AND A UNIQUE ECDH521 PUBLIC KEY FOR THE KEY EXCHANGE. THIS IS BEST OF CLASS SECURITY DESIGNED FOR MAINSTREAM USERS THAT REINFORCES THE INITIAL SCRAMBLING.

ADDITIONAL SECURITY LAYER

THE AES256 ENCRYPTION KEY IS ENCRYPTED WITH THE RECEPIENT'S ECDH521 PUBLIC KEY FOR THE KEY EXCHANGE. ONLY THE INTENDED RECEPIENT CAN DECRYPT THE MESSAGE KEY AND VIEW THE MESSAGE.

HARDWARE BINDING SPECIFIC

ONLY THE SENDER'S AND RECEIVER'S DEVICES, WITH THEIR UNIQUE DEVICE IDS, CAN ACCESS THE MESSAGE. AN INTERCEPTED MESSAGE CAN NOT BE READ WITHOUT THE HARDWARE TO WHICH IT'S BOUND.

ANOTHER SCRAMBLER

THERE'S NO SUCH THING AS TOO MUCH PROTECTION. AN ADDITIONAL AES256 ENCRYPTION ALGORITHM PROVIDES PROTECTION FOR HEADER PARAMETERS AND HARDWARE BINDING. BETWEEN THE APP AND SERVER.

TRANSPORT LAYER SECURITY

THE TLS PROTOCOL IS THE OUTER LAYER OF PROTECTION WHICH VERIFIES THE IDENTITY OF THE SERVER AND SECURES THE COMMUNICATION.

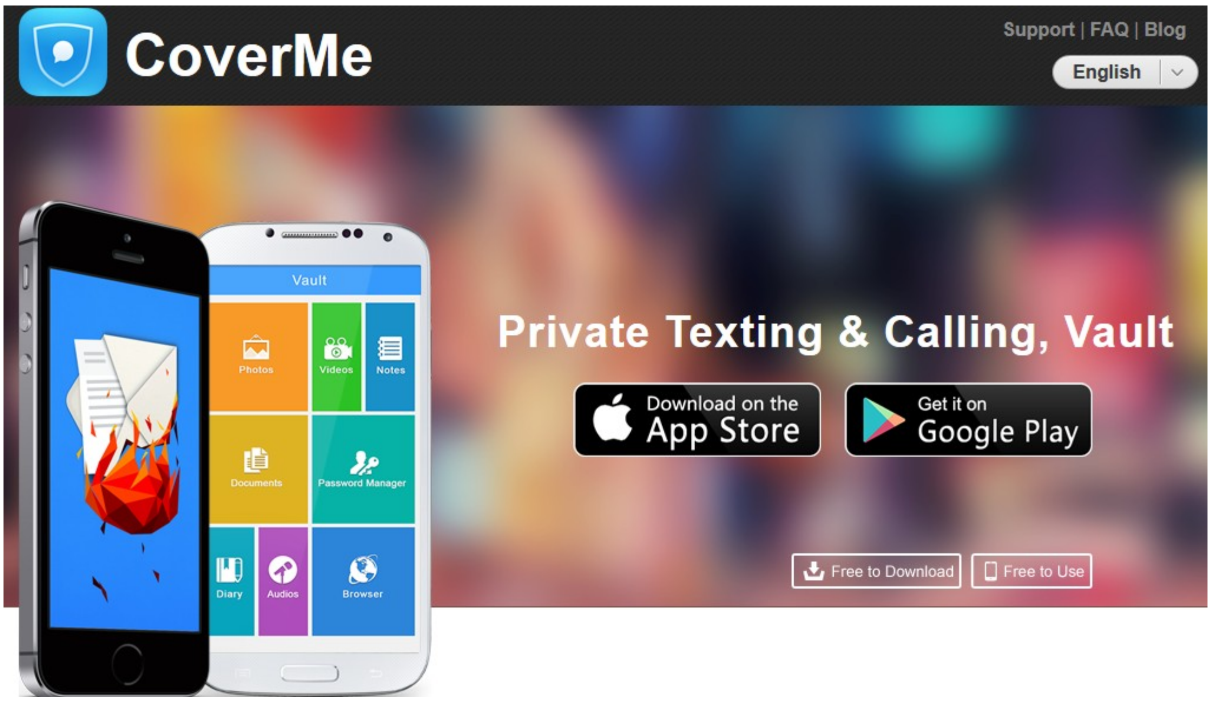
Wickr’s multilayer encryption

44 https://wickr.com/legal-process-guidelines

## 3.19 CoverMe

Author(s): CoverMe, inc. (based in San Francisco, CA)

http://coverme.ws/en/



| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Closed source, proprietary |
| (De)Centralization | Centralized |
| Standard(s) | For encryption: AES-128, AES-256 and RSA |
| Operating System(s) | iOS, Android (mobile devices only) |
| Data Collection (/Purpose) | Secret contacts, photos and videos are stored locally on user device. The user is the only person who can access the information by entering the correct password.<br><br>The company states that when a user shares a message, a secret contact, a photo or a video to a friend via CoverMe, an encrypted copy is cached on CoverMe servers “for performance reasons”. However, no one, including employees of CoverMe, Inc. could decrypt the cached copy as the encryption is unbreakable |

| | |
|---|---|
| | without the password - which only the user knows. The cached copy will be cleared after a short time and is not archived. |
| Number of users | Launched in November 2013 “after a beta period in which over 500,000 users signed up to use the service and exchanged over 500 million messages.”<br><br>Available on the Apple App Store and Google Play for respective operating systems (Between 100 000 and 500 000 downloads on Google Play). |
| Type(s) of users | Varied. |
| Countries | N/A |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes |

**Discursive description and additional information**

CoverMe is a mobile application for iPhone and Android. It claims to offer “military-grade encryption” protection for calls, messaging and all personal and private information such as photos, videos, call logs and contacts.

CoverMe proposes an encrypted “Personal Vault” that can only be accessed with a private password and is aimed at “completely protect[ing] the information in the phone from unauthorized access”, described as a “safe deposit box” where particularly sensitive contacts, photos/videos, messages, logs etc. are kept hidden.

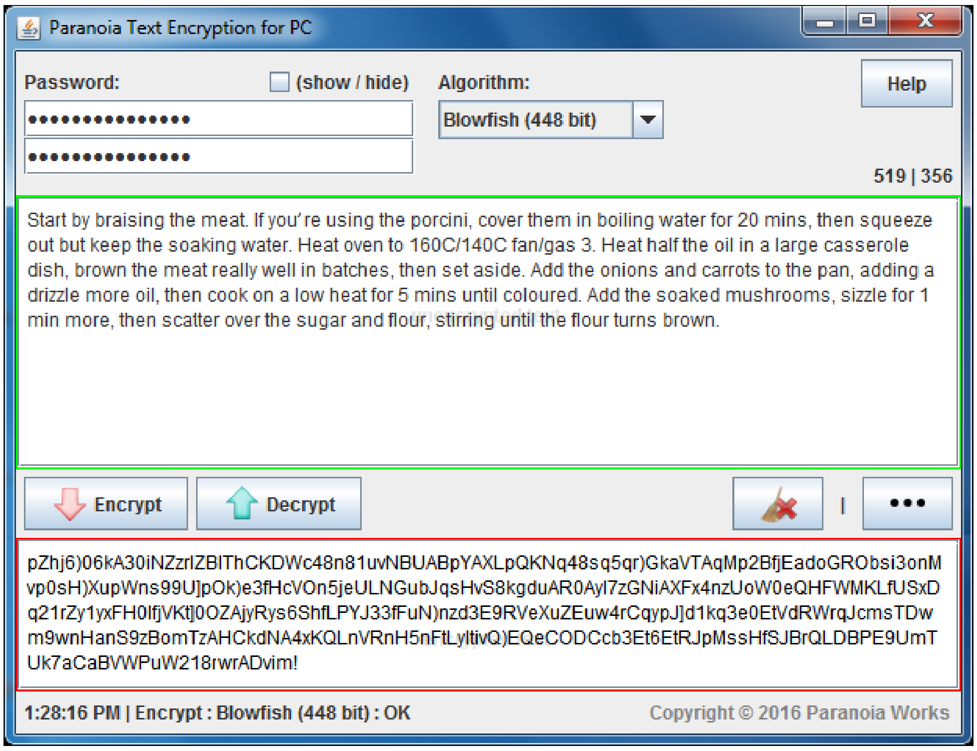
CoverMe also proposes the “Decoy Password” functionality, i.e. a separate password allowing to maintain a visible set of contacts and messages less strongly secured. When the Decoy is used to login, neither the Personal Vault nor its contents are visible.

Heavy ‘military’ narrative throughout the FAQ, e.g.: “To put it in perspective even the lowest level (AES-128) of any of these encryption mechanisms generates a key that is so variable, there are more atoms in the known universe than there are combinations! (Roughly 2128 Vs. 272 atoms!) All the computers in the world combined could not currently crack such a combination in under a thousand years. This is why the military use it in all their communications equipment!”

## 3.20 Paranoia Works

Authors: “P.W.” (Paranoia Works)

http://paranoiaworks.mobi



Password: •••••••••••••••
•••••••••••••••

Algorithm: Blowfish (448 bit)

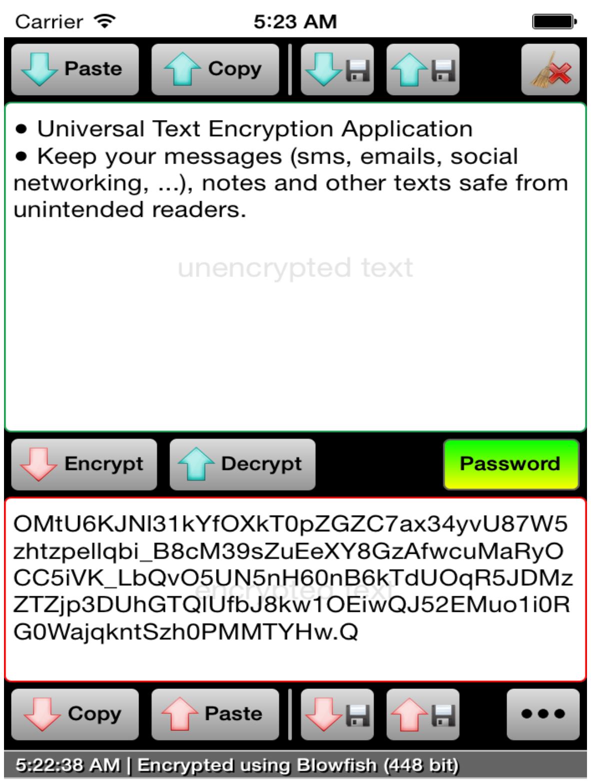
519 | 356

Start by braising the meat. If you’re using the porcini, cover them in boiling water for 20 mins, then squeeze out but keep the soaking water. Heat oven to 160C/140C fan/gas 3. Heat half the oil in a large casserole dish, brown the meat really well in batches, then set aside. Add the onions and carrots to the pan, adding a drizzle more oil, then cook on a low heat for 5 mins until coloured. Add the soaked mushrooms, sizzle for 1 min more, then scatter over the sugar and flour, stirring until the flour turns brown.

```
pZhj6)06kA30iNZzrlZBlThCKDWc48n81uvNBUABpYAXLpQKNq48sq5qr)GkaVTAqMp2BfjEadoGRObsi3onM
vp0sH)XupWns99U]pOk)e3fHcVOn5jeULNGubJqsHvS8kgduAR0Ayl7zGNiAXFx4nzUoW0eQHFWMKLfUSxD
q21rZy1yxFH0lfjVKt]0OZAjyRys6ShfLPYJ33fFuN)nzd3E9RVeXuZEuw4rCqypJ]d1kq3e0EtVdRWrqJcmsTDw
m9wnHanS9zBomTzAHCkdNA4xKQLnVRnH5nFtLyltivQ)EQeCODCcb3Et6EtRJpMssHfSJBrQLDBPE9UmT
Uk7aCaBVWPuW218rwrADvim!
```

1:28:16 PM | Encrypt : Blowfish (448 bit) : OK

Copyright © 2016 Paranoia Works



• Universal Text Encryption Application
• Keep your messages (sms, emails, social networking, ...), notes and other texts safe from unintended readers.

OMtU6KJNI31kYfOXkT0pZGZC7ax34yvU87W5zhtzpellqbi_B8cM39sZuEeXY8GzAfwcuMaRyOCC5iVK_LbQvO5UN5nH60nB6kTdUOqR5JDMzZTZjp3DUhGTQlUfbJ8kw1OEiwQJ52EMuo1i0RG0WajqkntSzh0PMMTYHw.Q

5:22:38 AM | Encrypted using Blowfish (448 bit)

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes (Android : http://www.paranoiaworks.mobi/download/) |
| (De)Centralization | No |
| Standard(s) | The app provides multiple levels of encryption: Blowfish 448bit, AES 256bit, RC6 256bit, Serpent 256bit, Twofish 256bit, GOST 256bit + (Threefish 1024bit and SHACAL-2 512bit for Pro Version) ciphers are available.The encryption engine is based on the Bouncy Castle Crypto Library (Java) and Crypto++ (C++). |
| Operating System(s) | Android, iOS + cross-platform versions (Windows, Linux, OS X) + Online version[45] |
| Data Collection (/Purpose) | N/A |
| Number of users | N/A<br>Google Play between 100 000 and 500 000 users |
| Type(s) of users | N/A |

45 https://pteo.paranoiaworks.mobi/

| | |
|---|---|
| Countries | Czech Republic etc. |
| End-to-end group chat? | no |
| End-to-end voice calls? | no |

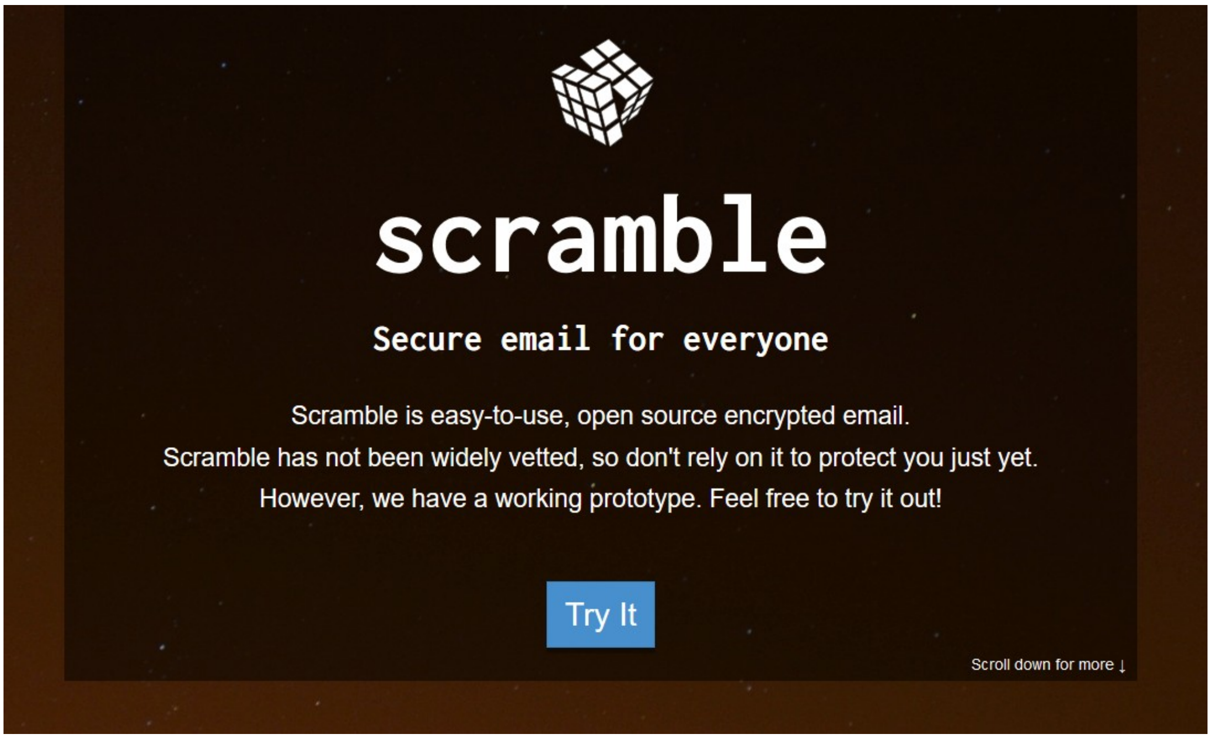
**Discursive description and additional information**

The app lets user encrypt his/her text, and then copy it to a standard messenger and send the encrypted text as a normal email. The receiver has to have the app installed on his/her phone/laptop. When he/she gets the email with the encrypted message, he/she only needs to copy the text and insert it into the app along with the password to get the message decrypted. Encrypts SMS, emails, social networking, files, notes and other texts. Texts are encrypted using encryption algorithms: AES (Rijndael) 256bit, RC6 256bit, Serpent 256bit, Blowfish 256bit/448bit, Twofish 256bit and GOST 256bit ciphers are available. Steganographic function is available.

## 3.21 Scramble

Authors: Jaekwon and DC {dcposch, jaekwon}@scramble.io

https://dcposch.github.io/scramble/



Scramble is easy-to-use, open source encrypted email.
Scramble has not been widely vetted, so don't rely on it to protect you just yet.
However, we have a working prototype. Feel free to try it out!

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://dcposch.github.io/scramble/<br>Prototype-stage |
| (De)Centralization | Partly decentralized |
| (if decentralized) Federated/P2P? | Federated, inspired by Moxie Marlinspike's Convergence.io |
| (if decentralized) Standardized? | Yes |
| Standard(s) | OpenPGP<br>HTML5 crypto.getRandomValues() strong random number generator for encryption |

| | |
|---|---|
| Operating System(s) | N/A |
| Data Collection (/Purpose) | Encryption is always performed end-to-end on the clients. Scramble servers have no knowledge of the message contents.<br>Private keys are stored on the server in encrypted form. |
| Number of users | For prototype release, Scramble is running three initial notaries[46], all located in California. They are looking for volunteers--especially outside the US--who can run additional notaries. |
| Type(s) of users | N/A |
| Countries | United States for the time being |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional information**

Scramble claims only servers store only ciphertext so mail content is protected from the server. Mail sent from one Scramble address to another is OpenPGP encrypted and signed at the user's browser—so the server never sees the plaintext.

End-to-end encryption in Scramble is thus described:

"When a user first registers on a Scramble server, an OpenPGP keypair is generated on the client. The public key of the recipient is used for encrypting email, and the key of the sender is used for authentication. The private key of the recipient decrypts the message. Users don't have to manage and back up their keys---or even know what a key is. Scramble does it for you."

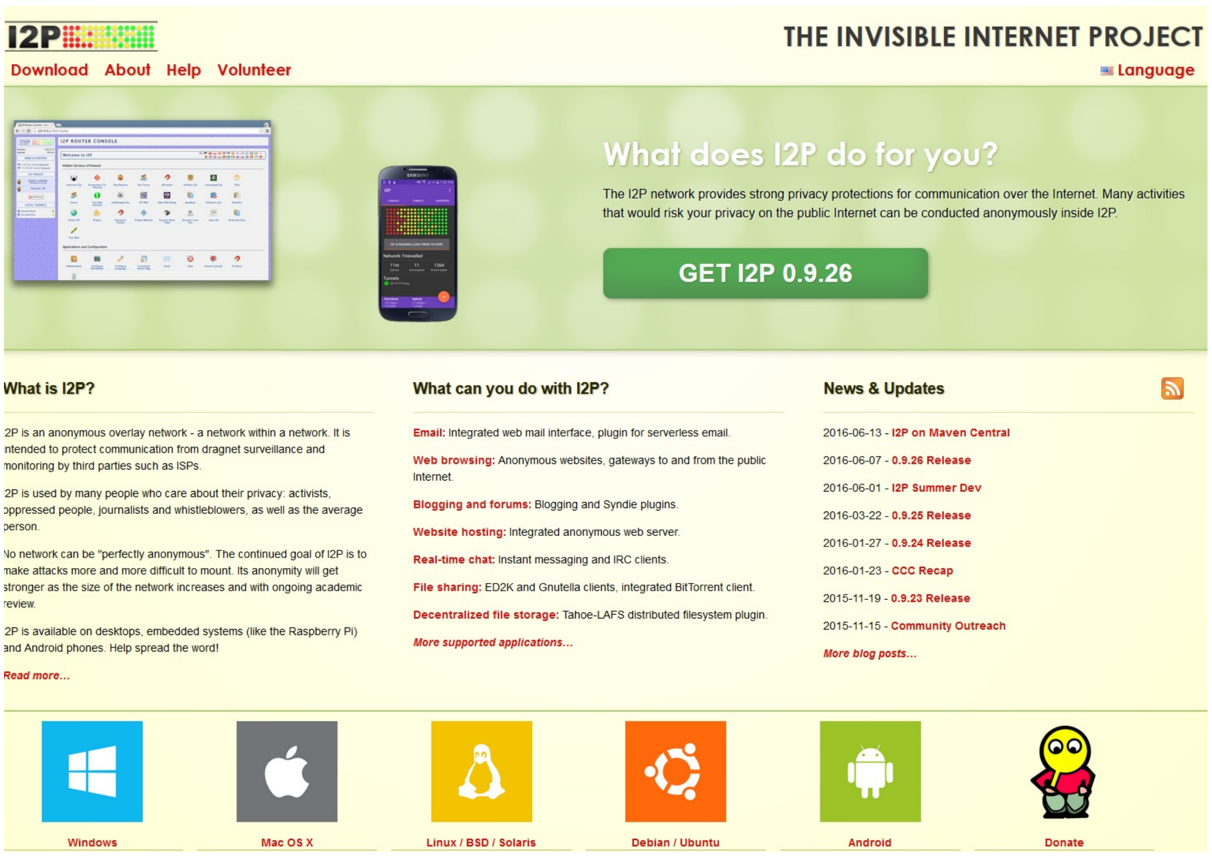
Scramble emphasizes ease-of-use ("includes a user-friendly webmail interface"; "Scramble aims to provide strong security guarantees without compromising on usability. Webmail is the easiest way to reach a wide audience, but it comes with its own set of challenges.").

46 "To look up a public key, the client asks a set of notaries---ideally ten or more, run by independent organizations and in different countries. If enough of them answer and they all agree, then the response is trusted. An attacker can't pretend to be a notary: the client comes with a public key for each trusted notary, and the notaries sign their responses. This allows the client to verify."

## 3.22 I2P

Authors: The Invisible Internet Project Team[47]

https://geti2p.net/en/



What does I2P do for you?

The I2P network provides strong privacy protections for communication over the Internet. Many activities that would risk your privacy on the public Internet can be conducted anonymously inside I2P.

**What is I2P?**

2P is an anonymous overlay network - a network within a network. It is ntended to protect communication from dragnet surveillance and monitoring by third parties such as ISPs.

2P is used by many people who care about their privacy: activists, oppressed people, journalists and whistleblowers, as well as the average person.

No network can be "perfectly anonymous". The continued goal of I2P is to make attacks more and more difficult to mount. Its anonymity will get stronger as the size of the network increases and with ongoing academic review.

2P is available on desktops, embedded systems (like the Raspberry Pi) and Android phones. Help spread the word!

*Read more…*

**What can you do with I2P?**

**Email:** Integrated web mail interface, plugin for serverless email.

**Web browsing:** Anonymous websites, gateways to and from the public Internet.

**Blogging and forums:** Blogging and Syndie plugins.

**Website hosting:** Integrated anonymous web server.

**Real-time chat:** Instant messaging and IRC clients.

**File sharing:** ED2K and Gnutella clients, integrated BitTorrent client.

**Decentralized file storage:** Tahoe-LAFS distributed filesystem plugin.

*More supported applications…*

**News & Updates**

2016-06-13 - **I2P on Maven Central**

2016-06-07 - **0.9.26 Release**

2016-06-01 - **I2P Summer Dev**

2016-03-22 - **0.9.25 Release**

2016-01-27 - **0.9.24 Release**

2016-01-23 - **CCC Recap**

2015-11-19 - **0.9.23 Release**

2015-11-15 - **Community Outreach**

*More blog posts…*

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://github.com/i2p/i2p.i2p |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | P2P |
| (if decentralized) Standardized? | No |
| Operating System(s) | Cross-platform (Windows, Mac OS X, Debian/Ubuntu, Linux, Android) |
| Data Collection (/Purpose) | In principle, none. I2P notes that ‘even the end points (destinations) are cryptographic identifiers’ |

47 https://geti2p.net/en/about/team

| | |
|---|---|
| Number of users | N/A but community appears large. |
| Type(s) of users | The I2P site notes that the network is ‘used by many people who care about their privacy: activists, oppressed people, journalists and whistleblowers, as well as the average person’ |
| Countries | English, Spanish. Incomplete translations exist in Russian, French,Romanian, German, Swedish, Italian, Portuguese, Chinese, Dutch, Polish, Hungarian, Arabic, Japanese, Estonian |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | No |

**Discursive description and additional information**

I2P is an anonymous P2P overlay network. It provides a layer that applications can use to send messages to each other in an anonymous and secure fashion. The network itself is message based, but there is a library available to allow communication on top of it. All communication is end to end encrypted (in total there are four layers of encryption used when sending a message). The end-points ("destinations") are cryptographic identifiers as well.

An important focus is anonymity. To anonymize the messages sent, each client application has their I2P router build a “tunnel”, i.e. a sequence of peers that pass messages in one direction (to and from the client, respectively). In turn, when a client wants to send a message to another client, the client passes that message out one of their outbound tunnels targeting one of the other client's inbound tunnels, eventually reaching the destination. Every participant in the network chooses the length of these tunnels, and in doing so, makes a tradeoff between anonymity, latency, and throughput according to their own needs. This is meant to optimize the number of peers needed to complete a satisfactorily anonymous and secure communication for both parties. I2P has been in beta since 2003 and is essentially a Tor competitor.

## 3.23 Teem/SwellRT (formerly Kune)

Authors: Comunes Collective (comunes.org)

Teem: https://play.google.com/store/apps/details?id=eu.p2pvalue.pear2pear

SwellRT: http://swellrt.org/

Formerly Kune

http://kune.ourproject.org/

Frequently, we see the scope of action of **our group/collective/organization** on Internet **limited by** several aspects such as distance, **economic** means **or technical knowledge**, which make us heavily dependent on others. What would happen if we were able to overcome these problems ourselves without any technical and economic effort?

Kune allows the collaborative management of a collective with just an Internet connection.

What would you be able to do with Kune?

You could create an online group space for the creation of collaborative documents, build your group website, holding and planning meetings in real time, interact or share your contents with other users of your group or others in your social network.

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source (both code and libraries)<br>https://github.com/P2Pvalue/swellrt<br><br>Released under GNU AGPLv3 license. |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | Federated |
| (if decentralized) Standardized? | Yes |

| | |
|---|---|
| Standard(s) | Apache (ex-Google) Wave Federation Protocol (an extension for the XMPP protocol) |
| Operating System(s) | Cross-platform |
| Data Collection (/Purpose) | Kune states that "*by itself* [it] does not store information about what users write using Kune. All this information is stored by the ones responsible for the server and this is their responsibility to inform users about what is done with their data." |
| Number of users | N/A, although Kune has active support and is used by several institutions and organizations besides Comunes. |
| Type(s) of users | Kune has a special focus on free culture and social movement needs, although it aims to appeal at individuals and organizations in need of a collaborative tool they can own in a distributed manner. |
| Countries | Varied, available in 10+ languages. |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | NO |

**Discursive description and additional information**

Team/SwellRT is a free/open source distributed social network with an important focus on collaboration in addition to communication. It focuses on online real-time collaborative editing, decentralized social networking, and web publishing, and places an emphasis on workgroups. Teem is the Android App, while the desktop version is SwellRT. They are both re-branded versions of Kune, which is in turn a re-branded version of Apache Wave with various improvements made. Kune is built on top of the XML-based Apache Wave software framework, originally developed by Google as Google Wave (Jabber/XMPP) and donated to the Apache Foundation after Google abandoned the product.

Teem/SwellRT are not encrypted or secure messaging based applications. They do have https (SSL) enabled by default while all the user interaction happens (not only in the login). Kune has plans to provide a layer of encryption by default for all content, but currently the software does not encrypt user data, so users needs to have trust in the server.

## 3.24 (n+1)sec

Authors: The eQualit.ie team (https://equalit.ie/#team)

https://equalit.ie

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://github.com/equalitie/np1sec<br>Implemented as a FLOSS library in C++ |
| (De)Centralization | Yes |
| (if decentralized) Federated/P2P? | Federated |
| (if decentralized) Standardized? | Yes |
| Standard(s) | Variation of OTR |
| Operating System(s) | Cross-platform |
| Data Collection (/Purpose) | N/A |
| Number of users | In development - N/A |
| Type(s) of users | N/A |

| | |
|---|---|
| Countries | N/A |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | No |

**Discursive description and additional information**

The software is developed by the team/collective eQualit.ie, whose activities include several projects and a consultancy activity. eQualit.ie is partners with organizations such as Human Rights Watch and Civil Rights Defenders, and boasts a privacy-oriented Manifesto[48]. eQualit.ie is also engaged into popularizing encryption and privacy for advocacy (organization of trainings, developing online manuals).

(n+1)sec is a secure group messaging protocol (authored with support from the Open Technology Fund), and a FOSS library. The main idea behind this project is to allow secure instantaneous communications between any number of people. The protocol wishes to provide end-to-end security of synchronous communications by building on recent advancements in cryptographic research.

According to its website, security properties of (n+1)sec include:

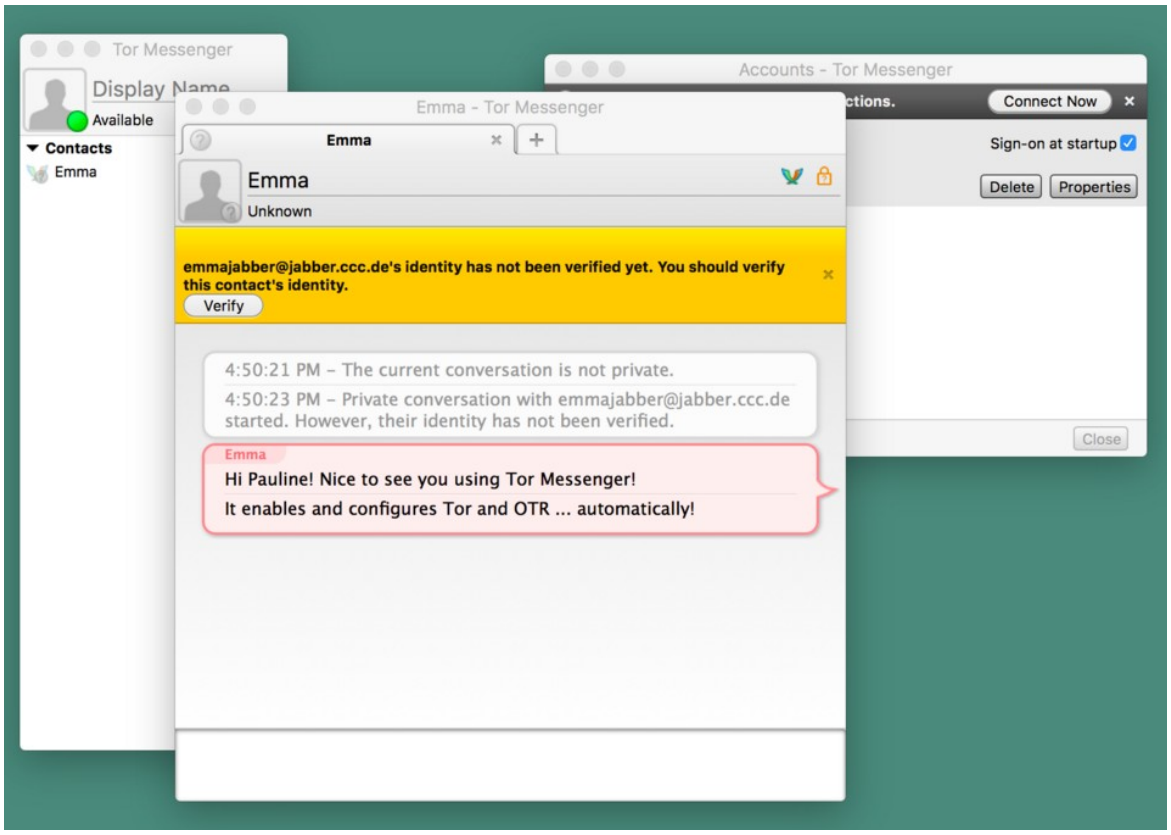
- **Confidentiality:** the conversation is not readable to an outsider
- **Forward secrecy:** conversation history remains unreadable to an outsider even if participants' encryption keys are compromised
- **Deniable authentication:** Nobody can prove your participation in a chat
- **Authorship:** A message recipient can be assured of the sender's authenticity even if other participants in the room try to impersonate the sender
- **Room consistency:** Group chat participants are confident that they are in the same room
- **Transcript consistency:** Group chat participants are confident that they are seeing the same sequence of messages

48 https://equalit.ie/equalit-ie-manifesto/

## 3.25 Tor Messenger

Authors: Jacob Appelbaum, Subkhir Singh (Code), Roger Dingledine, Isabella Bagueros

https://trac.torproject.org/projects/tor/wiki/doc/TorMessenger#Downloads



emmajabber@jabber.ccc.de's identity has not been verified yet. You should verify this contact's identity.

4:50:21 PM – The current conversation is not private.

4:50:23 PM – Private conversation with emmajabber@jabber.ccc.de started. However, their identity has not been verified.

Emma

Hi Pauline! Nice to see you using Tor Messenger!

It enables and configures Tor and OTR ... automatically!

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Yes<br>https://gitweb.torproject.org/tor-messenger-build.git |
| (De)Centralization | Yes |
| (if decentralized) Federated/P2P? | no |
| (if decentralized) Standardized? | yes |
| Standard(s) | Supports XMPP<br>Based on Tor |
| Operating System(s) | Cross-platform<br>Linux (32-bit and 64-bit), Windows and OS X versions |

| | |
|---|---|
| Data Collection (/Purpose) | N/A |
| Number of users | N/A |
| Type(s) of users | Activists, tech savvy |
| Countries | N/A |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | no |

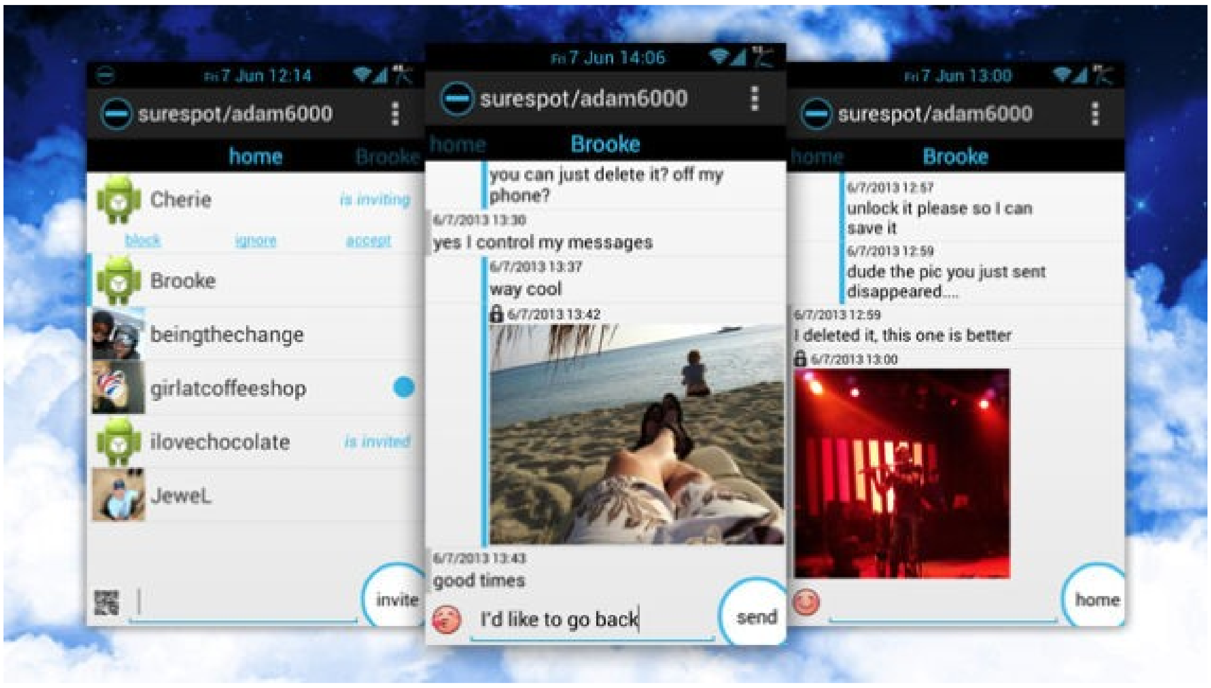
**Discursive description and additional information**

Tor Messenger is a cross-platform chat program that aims to be secure by default and sends all of its traffic over Tor. It is based on Instabird and supports a wide variety of transport networks, including Jabber (XMPP), IRC, Google Talk, Facebook Chat, Twitter, Yahoo, and others; enables Off-the-Record (OTR) Messaging automatically; and has an easy-to-use graphical user interface localized into multiple languages. Beta-version released in March 2016. Tor Messenger now supports OTR conversations over Twitter DMs (direct messages).

## 3.26 SureSpot

Authors: SureSpot LLC

https://surespot.me/



Cherie is inviting
block ignore accept
Brooke
beingthechange
girlatcoffeeshop
ilovechocolate is invited
JeweL

you can just delete it? off my phone?
6/7/2013 13:30
yes I control my messages
6/7/2013 13:37
way cool
6/7/2013 13:42
6/7/2013 13:43
good times
I'd like to go back

6/7/2013 12:57
unlock it please so I can save it
6/7/2013 12:59
dude the pic you just sent disappeared....
6/7/2013 12:59
I deleted it, this one is better
6/7/2013 13:00

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Clients are open source<br>https://github.com/surespot |
| (De)Centralization | Centralized |
| Standard(s) | |
| Operating System(s) | iOS, Android |
| Data Collection (/Purpose) | surespot stores the following data on its servers:<br><br>● Usernames.<br>● Friend relationships (who is friends with who, blocked who, ignored who, deleted who).<br>● Conversation relationships (how many friends currently you have a "conversation" with - meaning have a |

| | |
|---|---|
| | sent or received a message with).<br>● Messages in the amount of MAX_MESSAGES_PER_USER (currently 1000) which each have a server timestamp, to username, from username, and encrypted content, or link to encrypted content (image or file).<br>● Encrypted message file data (image or other - anything but encrypted message content) is stored (encrypted in the same way text messages are) on rackspace cloud files.<br>● Total messages sent per user.<br>● Total images sent per user.<br>● Current message count per user. (How many messages they have stored in the database currently, will always be <= MAX_MESSAGES_PER_USER (currently 1000)).<br>● Signing (DSA) public keys and versions.<br>● Encryption (DH) public keys and versions.<br>● Encrypted "friend images" or avatars and friend aliases that are assigned to certain usernames. These are encrypted with a key generated from ECDH key derivation of assigning identity's private/public keypair.<br>● Google GCM id (used for push messaging) which is related to the username in the surespot database.<br>● Apple APN token (used for push messaging) which is related to the username in the surespot database.<br>● If voice messaging is purchased, a purchase token given to us by Google or Apple which is related to the username in the surespot database.<br>● Server logs may contain any of the above information and are in a 20 log - 5MB per log rotation. |
| Number of users | Between 100 000 and 500 000 on Google Play |
| Type(s) of users | General public |
| Countries | USA (Colorado) |
| End-to-end group chat? | Not yet |

| | |
|---|---|
| End-to-end voice calls? | no |

**Discursive description/any additional info**

SureSpot is based on the end-to-end encryption with symmetric-key encryption 256 bit AES GCM using keys created with 521 bit ECDH shared secret derivation. SureSpot also enables users to verify the public key fingerprint of friends offline which adds another layer of protection by revealing any MITM attacks. Uses Google Cloud Messaging aka push messaging.

SureSpot states that users can delete messages not only from their phone but also from the receiver's phone and SureSpot server. The server limits message storage to 1000 messages, after which it will automatically delete the oldest message.

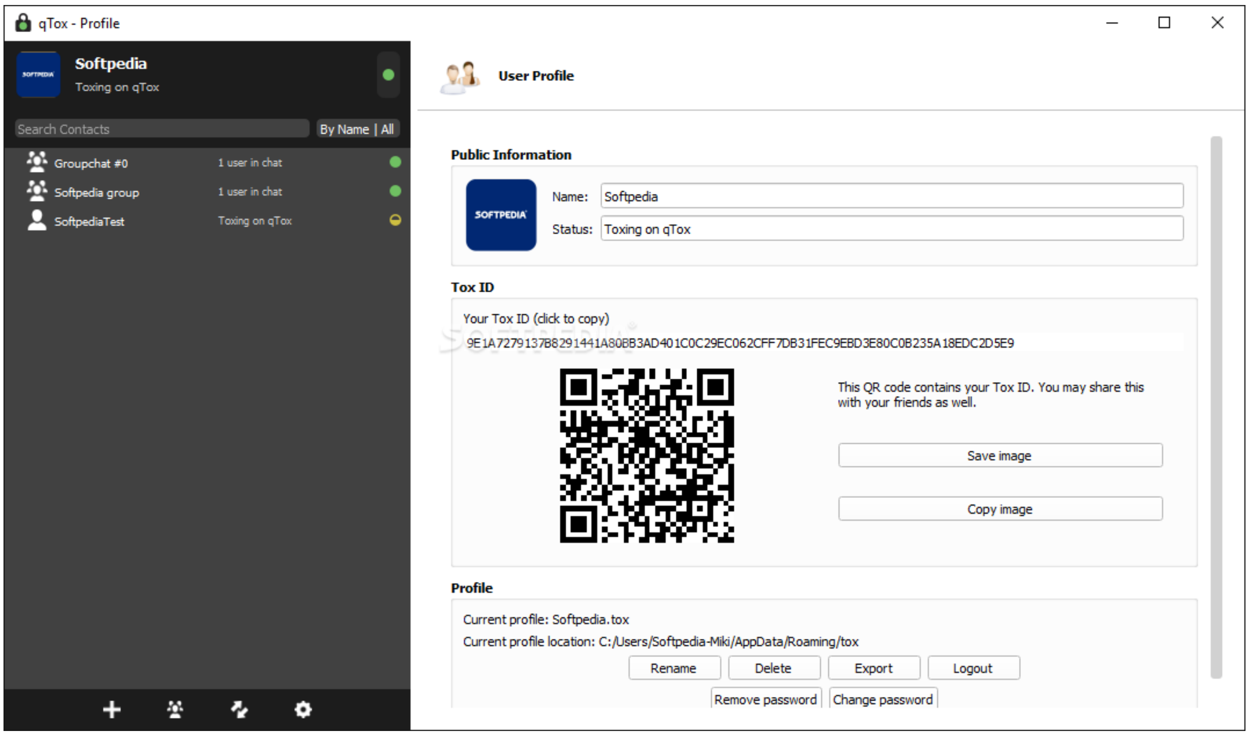
SureSpot also had a "story" engaging ISIS members said to have been using SureSpot app. SureSpot published a statement on this incident:

*"If any confusion exists as to the role surespot played in these events, consider a simple analogy: if someone were to mail a box containing illegal goods to an unsuspecting recipient and that recipient was arrested by law enforcement, would that be the fault of the mail carrier? Was USPS hacked? Or, did they safely deliver the package? Mail carriers are not responsible for advising individuals on who they should trust or accept mail from. Due to the sensitive nature of encryption services in an era of increased terror threat, companies like surespot will always be under attack by media and those working to thwart such threats. We should all remember that media is largely influenced and often entirely controlled by governments and special interest groups. It is in the interest of authorities to project the idea that surespot was hacked in order to discourage more people from using the app, because it is so secure. Don't Believe the Hype*[49]*."*

49 http://surespotencryptedmessenger.blogspot.fr/2015/09/dont-believe-hype.html

## 3.27 TOX

Authors: Developers remain anonymous but use the following handles Jfreegman, stal, Impyy, nurupo, mannol, irungentoo



Public Information

Name: Softpedia

Status: Toxing on qTox

Tox ID

Your Tox ID (click to copy)

9E1A7279137B8291441A80BB3AD401C0C29EC062CFF7DB31FEC9EBD3E80C0B235A18EDC2D5E9

This QR code contains your Tox ID. You may share this with your friends as well.

Profile

Current profile: Softpedia.tox

Current profile location: C:/Users/Softpedia-Miki/AppData/Roaming/tox

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | OpenSource<br>https://github.com/tux3/qTox<br>Core: https://github.com/irungentoo/toxcore |
| (De)Centralization | Decentralized |
| (if decentralized) Federated/P2P? | P2P |
| (if decentralized) Standardized? | Yes |
| Standard(s) | Based on TOX protocol (Tox Client Standard[50]) that is not an IETF standard |

50 https://www.gitbook.com/book/tox/tox-client-standard/details

| | |
|---|---|
| Operating System(s) | Linux, Windows, OS X<br>Alternatives for Android: “Antox”<br>Alternatives for iOS: “Antidote” |
| Data Collection (/Purpose) | All the data storage parameters are regulated by the Tox standard[51] |
| Number of users | N/A<br>There are between 3250 and 2800 nodes per minute (statistics from March to June 2016)[52] |
| Type(s) of users | Activists, p2p advocates, tech<br>General public |
| Countries | According to the map of nodes (crawler[53]) leading countries: Russia, US, Germany, France... |
| End-to-end group chat? | Yes |
| End-to-end voice calls? | Yes (Group voice calls also) |

**Discursive description and additional information**

qTox is peer-to-peer and encrypted build on top of the home-baked Tox protocol, runs over Linux or Windows, supports OTR, file transfer, video calls, screen sharing functions. Uses onion routing for friends discovery. The underlying Tox protocol encrypts all traffic using NaCl library. Specifically, Tox employs Curve25519 for its key exchanges, xsalsa20 for symmetric encryption, and Poly1305 for MACs. Tox was accepted into the Google Summer of Code as a Mentoring Organization in 2014 and 2015.

“Tox pushes [secure communication] forward in that there’s not really a central server…but as it’s currently designed, it allows a direct IP-to-IP connection [that can be tracked]. That’s the problem with this whole anonymous space. Nine out of ten people who are trying to do it don’t really know what the problem is. The problem is metadata.” (critics by the founders of Ricochet)[54].

51 https://tox.gitbooks.io/tox-client-standard/content/data_storage/data_storage.html

52 https://toxstats.com/

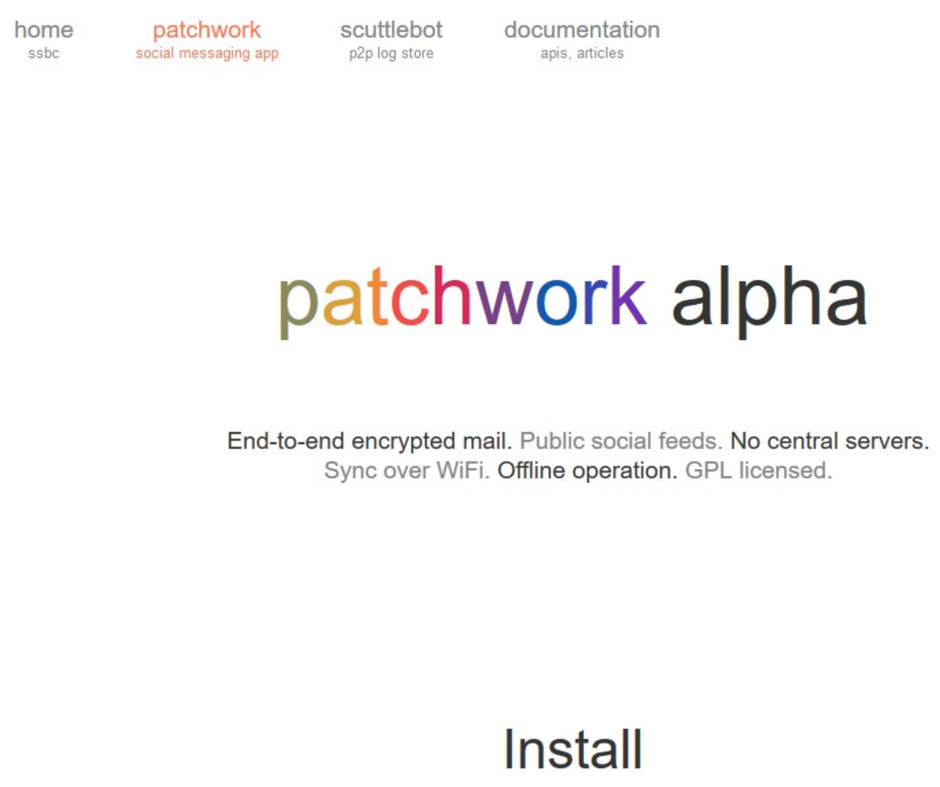
53 http://tox.viktorstanchev.com/

54 https://www.wired.com/2014/09/new-encrypted-chat-program-thwarts-nsa-eliminating-metadata/

## 3.28 Patchwork

Authors: Secure Scuttlebutt Consortium

https://ssbc.github.io/patchwork/



home
ssbc

patchwork
social messaging app

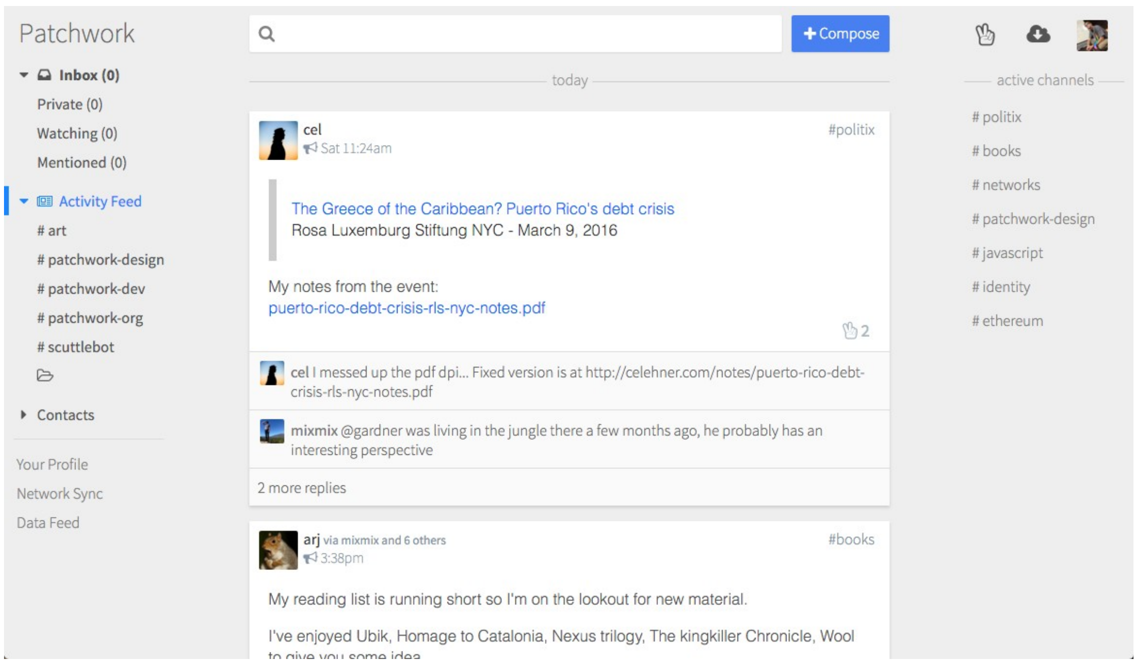
scuttlebot
p2p log store

documentation
apis, articles

patchwork alpha

End-to-end encrypted mail. Public social feeds. No central servers.
Sync over WiFi. Offline operation. GPL licensed.

Install



Patchwork

Inbox (0)
Private (0)
Watching (0)
Mentioned (0)
Activity Feed
# art
# patchwork-design
# patchwork-dev
# patchwork-org
# scuttlebot
Contacts
Your Profile
Network Sync
Data Feed

+Compose

today

cel #politix
Sat 11:24am

The Greece of the Caribbean? Puerto Rico's debt crisis
Rosa Luxemburg Stiftung NYC - March 9, 2016

My notes from the event:
puerto-rico-debt-crisis-rls-nyc-notes.pdf

2

cel I messed up the pdf dpi... Fixed version is at http://celehner.com/notes/puerto-rico-debt-crisis-rls-nyc-notes.pdf

mixmix @gardner was living in the jungle there a few months ago, he probably has an interesting perspective

2 more replies

arj via mixmix and 6 others #books
3:38pm

My reading list is running short so I'm on the lookout for new material.

I've enjoyed Ubik, Homage to Catalonia, Nexus trilogy, The kingkiller Chronicle, Wool to give you some idea

active channels
# politix
# books
# networks
# patchwork-design
# javascript
# identity
# ethereum

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Open source<br>https://github.com/ssbc/patchwork |
| (De)Centralization | Yes |
| (if decentralized) Federated/P2P? | P2P |
| (if decentralized) Standardized? | No |
| Operating System(s) | Aiming at Cross-platform? |
| Data Collection (/Purpose) | N/A |
| Number of users | Several hundred |
| Type(s) of users | Tech-savvy users (for the time being) |
| Countries | |
| End-to-end group chat? | |
| End-to-end voice calls? | |

**Discursive description and additional information**

Patchwork is a decentralized messaging and sharing application. Patchwork's github page notes a number of features that make it stand out: by-default end-to-end encryption, software running locally, free and open source software, ability to work offline as data is saved to users' disks, federation ("Users are not bound to one server/host (what we call 'pubs') and do not have to trust the servers"), aimed at ease-of-use ("It's very easy to setup and maintain your own pub").

At present, potential users need to contact an SSB team member (in #scuttlebutt on Freenode) to get onto the network. Patchwork notes that "That's our informal barrier to entry right now, since we're not prepared for lots of users yet."

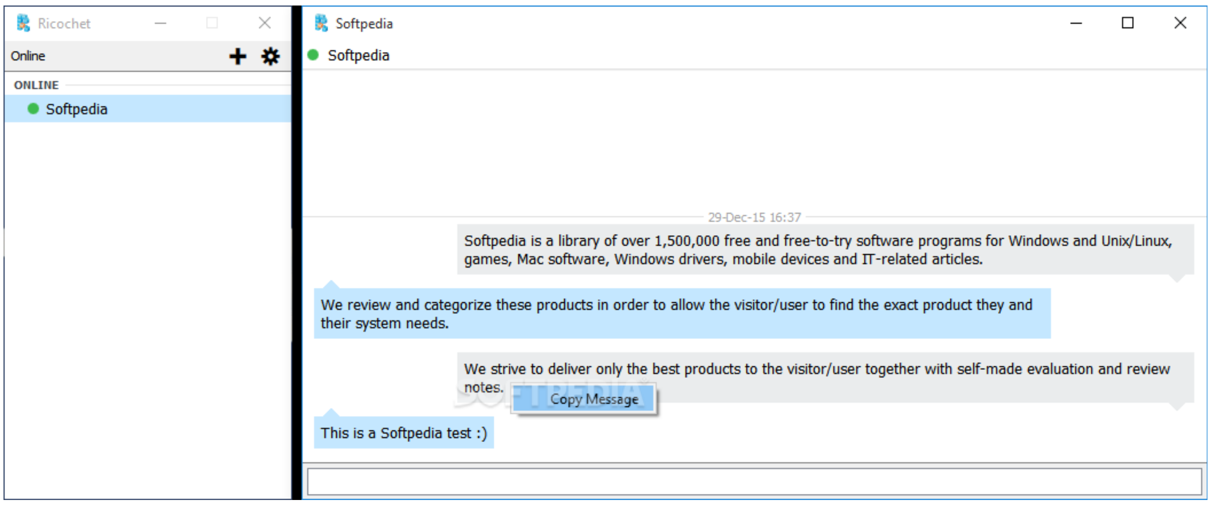
The protocol it's based upon is available on Github[55] but it's not compatible with PGP or SMTP.

55 https://github.com/ssbc/secure-scuttlebutt

## 3.29 Ricochet

Authors: John Brooks, Invisible.im group

https://ricochet.im/



Softpedia is a library of over 1,500,000 free and free-to-try software programs for Windows and Unix/Linux, games, Mac software, Windows drivers, mobile devices and IT-related articles.

We review and categorize these products in order to allow the visitor/user to find the exact product they and their system needs.

We strive to deliver only the best products to the visitor/user together with self-made evaluation and review notes.

This is a Softpedia test :)

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Opensource<br>https://github.com/ricochet-im/ricochet/ |
| (De)Centralization | yes |
| (if decentralized) Federated/P2P? | Uses Tor Hidden servers re federation |
| (if decentralized) Standardized? | no |
| Standard(s) | no |
| Operating System(s) | Windows, OS X (10.7 or later), and a generic Linux binary package |
| Data Collection (/Purpose) | N/A |
| Number of users | N/A |
| Type(s) of users | Journalists (“his primary motivation was to protect the anonymity of sources who contact journalists”), tech, activists |
| Countries | Australia, US... |

| | |
|---|---|
| End-to-end group chat? | yes |
| End-to-end voice calls? | no |

**Discursive description and additional information**

Ricochet is an instant messaging system designed around Tor hidden services. Ricochet is a modern alternative to TorChat, which hasn't been updated in several years, and to Tor Messenger, which is still in beta. A goal of the Invisible.im group is to help people maintain privacy by developing a "metadata free" instant messaging client.

Ricochet is a peer-to-peer instant messaging system built on the Tor Network hidden services. User’s login is a hidden service address, and contacts connect to it (not to an intermediate server) through Tor. The rendezvous system makes it extremely hard for anyone to learn user’s identity from its address.

The mission of the project is defined is to implement a real-time messaging system with these properties:

- Users aren't personally identifiable by contacts or their address
- Communication is authenticated and private
- No person or server can access contact lists, message history, or other metadata
- Resist censorship and monitoring at the local network level
- Resist blacklisting or denial of service against users
- Accessible and understandable for non-technical users
- Reliability and interactivity comparable with traditional IM services[56].

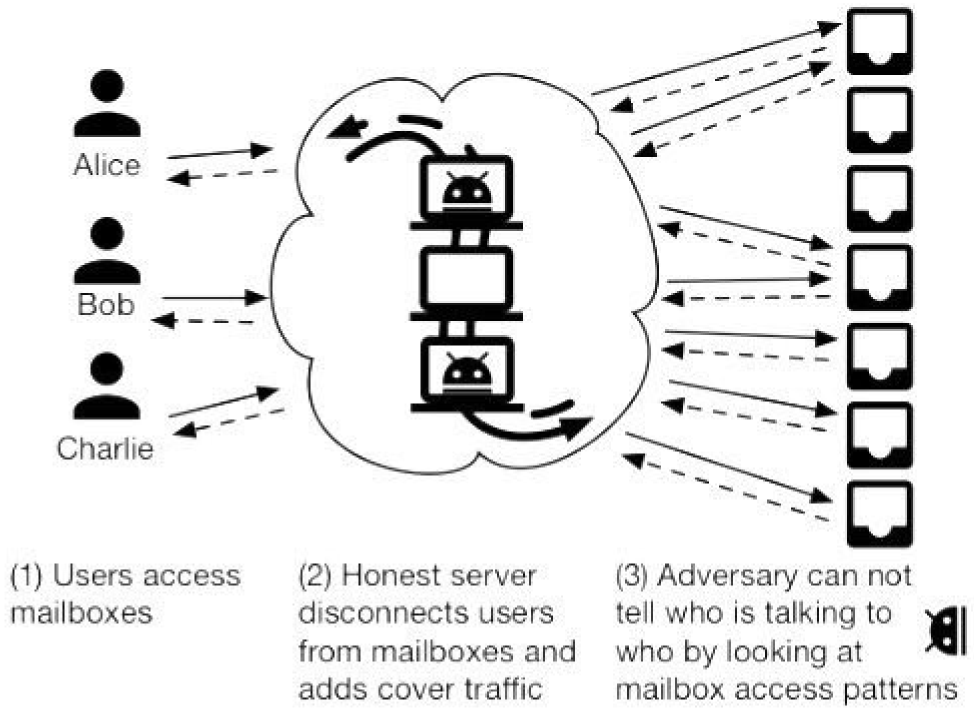
In February 2016, Ricochet's developers made public a security audit that had been sponsored by the Open Technology Fund and carried out by the NCC Group in November 2015. The results of the audit were "reasonably positive". The audit identified "multiple areas of improvement" and one vulnerability that could be used to deanonymize users. According to Brooks, the vulnerability has been fixed in the latest release.

Invisible.im recently got $10,000 from Blueprint for Free Speech, an Australian non-profit, to fund Brooks’ development costs and with that group as a fiscal sponsor now, Invisible.im can also apply for grants as an NGO.

56 https://github.com/ricochet-im/ricochet/blob/master/doc/design.md

## 3.30 Vuvuzela

Authors: Nickolai Zeldovich, David Laza, Matei Zaharia (MIT).


Alice
Bob
Charlie
(1) Users access mailboxes
(2) Honest server disconnects users from mailboxes and adds cover traffic
(3) Adversary can not tell who is talking to who by looking at mailbox access patterns

| | |
|---|---|
| Open/Closed Source (/What level)<br>Link to the source code if available | Opensource<br>https://github.com/davidlazar/vuvuzela |
| (De)Centralization | Centralized |
| Standard(s) | No |
| Operating System(s) | Desktop versions for Windows, OS X<br>Mobile apps not yet un beta |
| Data Collection (/Purpose) | Unable to encrypt two kinds of metadata: the number of idle and active users. The adversary does, however, know that two users whose messages reached the first server within some window of time have been talking. That is why Vuvuzela uses obfuscation and noise to make it difficult to survey. |

| | |
|---|---|
| Number of users | N/A |
| Type(s) of users | Oriented on activists and tech savvy for the moment. |
| Countries | USA |
| End-to-end group chat? | No |
| End-to-end voice calls? | No |

**Discursive description and additional information**

In the description of the project, authors underline the drawbacks and weak points of Tor and position their tool as an alternative. They also insist on the central role of metadata in surveillance quoting NSA former director Michael Hayden's famous phrase "We kill people based on metadata". They also insist on the scalability of their solution, talking about millions of users.

Vuvuzela is a messaging system that protects the privacy of message contents and message metadata. Users communicating through Vuvuzela do not reveal who they are talking to, even in the presence of powerful nation-state adversaries. Vuvuzela uses cryptography and mix networking to hide as much metadata as possible and adds noise to metadata that can't be encrypted efficiently. Vuvuzela is unable to encrypt two kinds of metadata: the number of idle users (connected users without a conversation partner) and the number of active users (users engaged in a conversation). Without noise, a sophisticated adversary could use this metadata to learn who is talking to who. However, the Vuvuzela servers generate noise that perturbs this metadata so that it is difficult to exploit. The Vuvuzela system is different from Tor in that it prevents outside observers from telling the difference between when a person is sending messages, receiving messages or doing neither.

Instead of using a single server, Vuvuzela uses three. Corresponding to the three servers, every message sent through the system is wrapped in three layers of encryption. The first server peels off the first layer of encryption before passing messages on to the second server. But it also randomly permutes their order. The second server peels off the second layer of encryption and permutes the message order yet again. Only the third server sees which messages are bound for which memory addresses. Here's where the noise comes in: When the first server passes on the messages it's received, it also manufactures a slew of dummy messages, with their own encrypted destinations. The second server does the same. So statistically, it's almost impossible for the adversary to determine even whether any of the messages arriving within the same time window ended up at the same destination[57].

57 https://news.mit.edu/2015/untraceable-anonymized-communication-guaranteed-1207

# 4. More decentralized and encrypted messaging software

While we have chosen, for the purpose of this deliverable, to focus on a 30-cases sample that is more likely to inform our subsequent choice of three case-studies for in-depth analysis, it is useful to point out, in this page, some additional resources on the great variety of systems that belong in the decentralized and/or encrypted messaging "galaxy", either because they adopt similar architectures and standards, or because the communities of developers interact/intersect. These may be studied in detail later.

In terms of peer-to-peer systems, three other notable systems are **Tribler** (https://www.tribler.org/)

from the Technical University of Delft and **Gnutella** (http://www.gnutellaforums.com/), that is currently being worked on by INRIA Rennes (Christian Grothoff) and support p2p systems, i.e.. file-sharing applications such as SecuShare (http://secushare.org/). There is also **FreeNet** for decentralized file storage (https://freenetproject.org/). A new product that was just released in for file storage along with a "chat" feature similar to IRC that could be investigated is **Semaphor** (https://spideroak.com/)

We have not delved into blockchain-based systems. The larger critique of such systems is that by preserving messages on the blockchain, they are essentially making the metadata public, which is major security and privacy flaw. In detail, these systems are **Twister** (http://twister.net.co/), based on blockchains for micro-blogging ala Twitter, and **BitMessage** (https://bitmessage.org) for encrypted messaging. Some Bitcoin wallets are also claiming anonymity properties and featuring messaging such as **DarkWallet** (https://github.com/darkwallet/darkwallet), although development seems to have stopped on DarkWallet. In general, blockchain-based solutions may be more useful in preserving transparency on server transactions rather than on a user's messaging inbox.

We have not looked into all proprietary extensions and business products. In particular, the most interesting is Google's **End-to-end e-mail**, a Google Chrome extension that should support PGP (https://github.com/google/end-to-end) but is still under development. There are a large number of non-standardized encrypted e-mail solutions aimed at business/government that do not support PGP and aren't open-source, so they are of limited interest to our case-studies, such as **Armortext** (http://armortext.com), **Virtru** https://www.virtru.com/, and **Ceerus** (https://www.ceerus.com). Furthermore, the **Peer.io** software has been discontinued, due to the failure of the start-up. **Whiteout**, software also created end-to-end PGP-enabled software, but the company failed and some of the developers have gone on to work at Thoughtworks, the developer of Pixelated.

We did not investigate many ActivityStreams-based projects, primarily because they are for the most part using end-to-end encryption or because of their security or privacy claims. Their claims to privacy are based on being self-hosted and decentralized based on standards-based federation. Thus, it will be useful for NEXTLEAP to help them develop security and privacy considerations, and become aware of privacy and security concerns. The main codebase is **ActivityPump** (http://pump.io/) by

Evan Prodromou although there is also **MediaGoblin** (http://www.mediagoblin.org/) by Chris Webber for federated media-sharing. Most interestingly, ActivityStreams 2.0 was also adopted by the CAPS project D-CENT (http://dcentproject.eu/) in their **Objective8** software for decentralized deliberation (https://objective8.dcentproject.eu/). However, given Objective8 has no real privacy considerations and does not use end-to-end encryption, it was not covered.

While other non-federated software being used by D-CENT software seems to have many users such as Betri Reykjavík, it does not appear there was any update of Objective8. Furthermore, the W3C Social Web Working Group did not reach consensus on a single format for federated social networking. Thus, there are systems based on the Semantic Web model such as **SoLiD: Social Linked Data** (http://crosscloud.org) that also use a federated model but based on W3C Semantic Web standards. However, this software also has no security or privacy-claims besides being self-hosted and does not use standardized APIs or formats. Even worse, it uses cross-origin certificates for identity such as **WebID+TLS** (https://www.w3.org/2005/Incubator/webid/spec/) that are in process of being deprecated due to privacy and security concerns.[58] More popular has been the "**IndieWeb**" approach, based on microformats (http://microformats.org/) in HTML, that individual web-masters can use in order to federate their blog posts and other data. Again, there are no security or privacy-claims besides self-hosting. A thorough security and privacy review of these technologies is necessary, as they have little communication with either the security and privacy research community or other security and privacy focussed open source efforts.

58 https://groups.google.com/a/chromium.org/forum/#!topic/security-dev/pX5NbX0Xack

# 5 Conclusions

Considering the lively and constantly-evolving ecosystem of standardized and non-standardized projects in the field of encrypted instant messaging, it is important that a multi-year interdisciplinary effort such as NEXTLEAP starts with a comprehensive mapping of relevant protocols first, relevant projects applying them next, based on a defined range of criteria.

This deliverable presents a first exploration in this regard, especially peculiar from an interdisciplinary standpoint, inasmuch as it is elaborated by social scientists and is meant to serve their needs in the first place, as a pre-requirement to an in-depth, case study-based inquiry. However, this social science research is ultimately meant to feed back into NEXTLEAP's development of technical protocols – protocols that are not only technically sound, but made for users and able to find their way into networked societies that are increasingly concerned about the security and confidentiality of their online communications.

This deliverable shows the great diversity of protocols and applications used by developers seeking to apply decentralization and/or end-to-end encryption to instant messaging systems -- a diversity that is further demonstrated in the appendix. This poses a methodological challenge of representativeness and accuracy that we have tried to embrace at best in the present document. Furthermore, this research opens the way to a number of preliminary observations which will be useful as we select, in the coming two months, the three in-depth case studies -- and beyond that, for an accurate socio-technical portrait of the end-to-end encrypted instant messaging field.

Despite the prevalence of free and open source software projects, proprietary software is not absent in this landscape, revealing both a potentially fruitful 'business-to-business' market for end-to-end encryption and a lack of open-source and standards adoption by mainstream applications. Open source itself is multi-layered and sometimes hybrid, with the code on the client side being open source and the server side being proprietary. Perhaps unsurprisingly, the proprietary features are more important in applications destined to a business-to-business use, while free and open source software is predominant for tools destined to activists and tech-savvy users. This transparency of code and encryption protocols is aimed not only at improving the project, but also at creating an emulation around the project producing communities of peer reviewers, experts, beta-testers and advanced users who participate in a collective reflexion on the future of privacy-enhancing technologies.

As we had the occasion to observe in previous mapping research on P2P services (Méadel and Musiani, 2015), part of the reason why there is such a great diversity and complexity in this field is the relatively short life span of several projects. While this deliverable only maps, among the 30 cases, projects that are currently active (with one exception, Pond, 'in stasis' albeit not deactivated), our preliminary research revealed countless others that, after two or three years of pre-beta phase, and sometimes less, stopped development with no evident explanation. While in more than a few cases, the motives behind this are primarily related to a technical experimentation that did not deliver as hoped or expected, a number of additional factors may also be responsible, including the failure to develop an economic model, the internal governance of FOSS development groups, and the inability to rally a critical mass of users around the app (possibly due to a lack of ease-of-use, as discussed below). These socio-technical factors will be useful to observe in the cases eventually selected for D3.3, as a precious source of 'lessons learned' in terms of user recruitment and governance models.

The target audience of the applications is far from being limited to tech-savvy and activist groups; several projects are aimed at widespread use, and user-friendliness appears to be the main issue that stands between this wish and its realization in practice. Interestingly, in some instances where user feedback is visible on the App Store or Google Play, it shows the 'digital migration'-related issues faced by end-to-end encryption; for example, this model is perceived as problematic because both sender and receiver have to install the app for encryption to take place, which complicates usage.

In the case of civic mobile and web applications studied previously (Ermoshina, 2014), the number of users is explicitly shown on the main page. It becomes an important tool for building user communities and empowering the impact of such activist projects. Whereas the present deliverable shows that very few projects openly give the number of their active users (possibly due to privacy issues). A further exploration planned in D3.3 on the three selected cases may investigate these specific politics.

The projects examined here propose several solutions to the problem of data storage. Indeed, despite the privacy concerns and guarantees explicitly stated by the developers, some projects store important amounts of data on the servers (such as usage statistics, device information, keys, usernames or friend relations). Developers tend to explain it by technical requirements (e.g. proposing better user experience based on the collected usage statistics). However, this preliminary inquiry shows that some communities are seeking for alternatives with minimal data storage, implying choices of stronger decentralization. The metadata level appears to be the most delicate one that divides developer communities (e.g. Vuvuzela vs Tor) and stimulates experiments with standards and architectures.

A look at visual aspects, such as the design of interfaces -- which we have sought to briefly account for by showcasing screenshots -- and the design of diagrams and graphics to explain the functioning of the applications, is also revealing of the different publics targeted by the applications and how they are perceived by the developers. General public-oriented systems use very 'politically neutral' imagery, resorting to the very classical 'Alice and Bob' while stressing that their tools are for 'everyone' (e.g. "sharing photos from holidays"), while tools meant for companies emphasize in both visuals and words the security aspect. Other narratives boast fictional anarchist leaders or real-life activists, which also strongly inform the target audience.

A related issue is the powerful 'double' narrative on end-to-end encryption. If on one hand, the discourse on empowerment and better protection of fundamental civil liberties is very strong, several projects show in parallel a desire/need to defend themselves from the 'you're used by jihadists'-type allegations (Sanger and Perlroth, 2015). This narrative is fueled by previous and current ones about decentralized technologies and peer-to-peer, with their history of allegedly 'empowering-yet-illegal' tools. These issues are taking place in the broader context of discussions about governance by infrastructure and civil liberties (Musiani et al, 2016), some of them particularly related to encryption (or the breaking of it), such as the Apple vs. FBI case and WhatsApp proposing, since April 2016, encryption by default. Thus, the present research hints at something that we will thoroughly address in the in-depth case studies -- something a large majority of the projects mapped in this deliverable needs to take into account, and indeed is already taking into account: architecture is politics, but it is not a substitute for politics (cf. Agre, 2003).[59]

59 As further discussed in D2.1.

# 6 References and resources

Agre, P. E. (2003). P2P and the promise of internet equality. *Communications of the ACM*, *46*(2), 39-42.

Borisov, N., Goldberg, I., Brewer, E (2004) "Off-the-Record Communication, or, Why Not To Use PGP", in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, 10.1145/1029179.1029200 https://otr.cypherpunks.ca/otr-wpes.pdf

Ermoshina, K. (2014). "Democracy as pothole repair: Civic applications and cyber-empowerment in Russia". In *Cyberpsychology: Journal of Psychosocial Research on Cyberspace, 8*(3), article 1. doi: 10.5817/CP2014-3-4

Marlinspike, M. (2013). "Advanced cryptographic ratcheting", published in OpenWhisperSystems blog on 26 of November 2013, https://whispersystems.org/blog/advanced-ratcheting/

Méadel, C., Musiani, F. (coord.) (2015). *Abécédaire des architectures distribuées*, Presses des Mines.

Musiani, F., Cogburn, D. L., DeNardis, L., Levinson, N. S. (dir.) (2016), *The Turn to Infrastructure in Internet Governance*, Palgrave Macmillan.

Prokop, A. (2015) "Solving the WebRTC Interoperability Problem", in *NoJitter*, http://www.nojitter.com/post/240169575/solving-the-webrtc-interoperability-problem

Sanger, D. and Perlroth, N. (2015) Encrypted Messaging Apps Face New Scrutiny Over Possible Role in Paris Attacks, New York Times, http://www.nytimes.com/2015/11/17/world/europe/encrypted-messaging-apps-face-new-scrutiny-over-possible-role-in-paris-attacks.html

Straub, A. (2015) "OMEMO Encryption", *a protoXEP standards track proposed to XMPP foundation* on 25th of October 2015, https://xmpp.org/extensions/inbox/omemo.html#intro-motivation

Weinberger, M. (2014) "Matrix wants to smash the walled gardens of messaging", published on September 16, 2014 in *ITworld*, http://www.itworld.com/article/2694500/unified-communications/matrix-wants-to-smash-the-walled-gardens-of-messaging.html