End-to-End Encrypted Messaging Protocols: An Overview

Ksenia Ermoshina, Francesca Musiani, Harry Halpin

End-to-end Encrypted Messaging Protocols: An Overview

Ksenia Ermoshina[1], Francesca Musiani[2], and Harry Halpin[3]

[1] i3-CSI, MINES ParisTech, France
ksenia.ermoshina@mines-paristech.fr
[2] ISCC, CNRS/Paris-Sorbonne/UPMC, France
francesca.musiani@cnrs.fr
[3]INRIA, France
harry.halpin@inria.fr

Abstract. This paper aims at giving an overview of the different core protocols used for decentralized chat and email-oriented services. This work is part of a survey of 30 projects focused on decentralized and/or end-to-end encrypted internet messaging, currently conducted in the early stages of the H2020 CAPS project NEXTLEAP.

Keywords: Decentralization; end-to-end encryption; messaging; protocols; NEXTLEAP

Exploratory/Survey Paper

1. Introduction

This exploratory paper first gives an overview of the different core protocols subtending the development of end-to-end encrypted internet messaging (chat and email-oriented) services. In its second part, the paper outlines initial findings of a survey of thirty decentralized and/or end-to-end encrypted projects. The paper also presents the methodological opportunities and challenges of studying such systems with social sciences tools.

Currently, end-to-end encrypted messaging has risen to prominence, with the adoption of end-to-end encrypted messaging by large proprietary applications such as WhatsApp or Facebook Messenger, and the interest in securing communication privacy provoked by the Snowden revelations. In "end-to-end" encrypted messaging, the server that hosts messages for a user or any third-party adversary that intercepts data as the message is *en route* cannot read the message content due to the use of encryption. The "end" in "end-to-end" encryption refers to the "endpoint," which in the case of messaging is the client device of the user rather than the server.

However, open standards for encrypted e-mail and chat are still not seeing widespread use, and a new generation of end-to-end encrypted messaging protocols offer-

ing better security properties are rapidly gaining traction, although most are not yet standardized or decentralized. The academic cryptographic community has renewed impetus post-Snowden to rigorously engage with the "secure messaging problem in the untrusted-server model," a problem that until recently "feels almost intentionally pushed aside" although the problem is perhaps "the most fundamental privacy problem in cryptography: how can parties communicate in such a way that nobody knows who said what" [1]. While the security research community has already began to overview the technical details of these protocols [2], what is missing from the technical work currently in progress is the needs and expectations of users when using encrypted end-to-end messaging applications.

An ongoing study on the use of encryption and decentralized communication tools is being conducted via the H2020 CAPS (Collective Awareness Project) project NEXt-generation Techno-social Legal Encryption Access and Privacy (NEXTLEAP).[1] NEXTLEAP seeks to address, in an interdisciplinary manner, the recent erosion of public trust in the Internet as a secure means of communication that has been prompted by the Snowden revelations. The core objective of NEXTLEAP is to improve, create, validate, and deploy communication protocols that can serve as pillars for a secure, trust-worthy and privacy-respecting Internet able to ensure citizens' fundamental rights. For this purpose, NEXTLEAP seeks to develop an interdisciplinary internet science of decentralization as the basis on which these protocols can not only be built, but become fully (and meaningfully) embedded in society. In this regard, the social aspect of end-to-end encryption must be included in the overall analysis of trust and decentralization at the heart of Internet Science.

The two main kinds of projects that we seek to examine are related to email and chat clients (also called "instant messaging" clients), both of which are considered to be particular cases of messaging. Historically, e-mail is considered asynchronous messaging, where a user does not have to be online to receive the message, while chat is considered asynchronous messaging, where a user has to be online to receive the message. However, in general these distinctions are blurring as any popular chat protocols now support asynchronous messaging. The remainder of this paper presents a genealogy of these fundamental protocols used for email and chat-oriented services, and then moves on to present preliminary findings and open questions.

2. E-mail Protocols

Email is based on standardized and open protocols descended insofar as the fundamental protocols allow interoperability between different email servers, so that a Microsoft server can send email to a Google server. Classically, as revealed by the PRISM program of the NSA, e-mail is sent unencrypted and so the server has full access to the content of e-mail. Thus, there has been a long-standing program to send email "end-to-end" encryption.

SMTP (Simple Mail Transfer Protocol) is the protocol originally used for transferring email and as such is one of the oldest standards for asynchronous messaging, first

[1] https://nextleap.eu

defined in 1982 by the IETF[2] and by default not having provision for content confidentiality using end-to-end encryption. PGP (Pretty Good Privacy) was created to add end-to-end encryption capabilities to e-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, to allow the open implementation of PGP. OpenPGP is implemented in open-source software such as Thunderbird with the Enigmail plug-in as well as in mobile apps, such as the IPGMail for iOS and the Openkeychain key management system for Android and F-Droid. GPG (GnuGPG) is a free software implementation of the OpenPGP standards developed by Free Software Foundation in 1999 and compliant with the OpenPGP standard specifications, serving as the free software foundation for most modern PGP-enabled applications. Recently, 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[3] is another IETF standard addressing the need for encrypting e-mail. In contrast to PGP that is based on a decentralized "Web of Trust" between users who accept and sign each others keys (and so "offloads" the complexity of key management to the end-user), S/MIME uses a centralized public key infrastructure to manage keys. Thus, while it has been adopted by some large centralized institutions, it has had much less success among the general public and so is not part of the study.

The problem with implementations of OpenPGP such as GPG is that they are difficult for most users to understand and use, especially in terms of usage and key management [3]. These problems extend to security: if an adversary compromises a user's private key, this allows all encrypted messages to be read. In general, while OpenPGP has had a resurgence of interest since 2013, it has not had as much deployment by ordinary users due to the aforementioned issues around user-friendliness and the fact that OpenPGP expects the users to understand the fundamentals of cryptography, such as public and private keys.

3. Chat Protocols

Unlike e-mail that is started as high-latency and asynchronous messaging, chat protocols began as low-latency synchronous messaging, although recently the line has become more blurred as many chat protocols allow asynchronous message delivery. There has long been an intuition that more and more messaging is moving from e-mail to messaging, although it seems that e-mail is still widely used.

XMPP (eXtensible Message and Presence Protocol) became an IETF (Internet Engineering Task Force) standard in 2004 for chat, and is probably the most widely used standardized chat protocol. 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,

[2] https://tools.ietf.org/html/rfc821

[3] https://tools.ietf.org/html/rfc3851

presence-aware clients and servers."[4] There are many implementations of the XMPP specifications, with the XMPP Foundation giving a list of 70 clients and 25 servers using the XMPP protocol.[5] 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, although 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.[6]

The OTR (Off-the-Record) protocol released in 2004 is an extension to XMPP to provide end-to-end encryption. It also provides deniable authentication for users, unlike PGP messages, which can be later "used as a verifiable record of the communication event and the identities of the participants" [4]. OTR is a security upgrade over PGP at least insofar as it does not have long-term public keys that can be compromised. The original paper that defines OTR is called ""Off-the-Record Communication, or, Why Not To Use PGP" [4]. 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,[7] including earlier versions of Cryptocat (in-browser Javascript client), Jitsi, and ChatSecure (XMPP client for Android and iOS). However, OTR is designed for synchronous messaging between two people, and so does not work for group messaging or asynchronous messaging. There seems to be a move away from OTR; the IM+ app for Android, even though having good user ranking and between 5 and 10 million downloads, is reported by users on Google Play Market as "abandoned" (last update in 2014). A further inquiry will be conducted in order to understand whether the reasons of abandonment are due to usability issues, to cryptographic failures or to other factors such as financial problems, maintenance costs, team conflicts or fusion with bigger projects.

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, it is an application-layer communications protocol for federated real-time communication. It is unencrypted by default. However, using the Olm library (Axolotl ratcheting from the Signal Protocol, described below) encryption can be 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" [5]. 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" [6]. However, Matrix.org seems to have few users since none of the

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

mainstream IM clients relies on it yet. The website lists 17 clients and 6 servers using Matrix.org.[8]

The Signal Protocol, the non-federated protocol developed in 2013 by Open Whisper Systems, provides end-to-end encryption for groups. Moxie Marlinspike, the co-author of Signal, was inspired with some OTR features, such as the idea of ephemeral key exchange [7], but also added additional security features such as future secrecy, support for asynchronous messaging and group messaging, going above and beyond OTR by allowing also clients to be offline. The Signal Protocol uses the "Axolotl" keyratchet for initial key exchange and the ongoing renewal and maintenance of short-lived session keys, so there is not only no long-term key that can be compromised. This provides forward secrecy so that the compromise of a short-term key does not compromise past keys (so that an adversary can decrypt past messages) as well as "backwards secrecy" (also called "future secrecy") so that the compromise of a key does not endanger future messages. It could be a standard, but is not yet recognized as such. The Signal protocol is said to be widely used in mobile messaging applications such as Signal (formerly TextSecure and RedPhone), WhatsApp[9], Secure Chat (by GData). Silent Circle uses a version of the Signal Protocol since 2015 in its Silent Phone. Recently Facebook announced the implementation of Signal Protocol for their Messenger[10]. The first step towards "standardization" of the Signal Protocol so far has been the creation of OMEMO.

OMEMO is a new encrypted extension of XMPP protocol developed in 2015 that effectively copies the Signal Protocol and adopts it to XMPP. It has been presented to the XMPP Standards Foundation but not yet approved in any official manner.[11]10 OMEMO builds upon the work of the Signal Protocol as 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" [8]. The software implementations of OMEMO are growing such as conversations, an open-source application for Android that counts over 5000 downloads via Google Play Market, and an unknown number of installs via F-Droid.

4. Network-level Anonymity

While this work is mostly focused on the application level, it seems important to mention the network-level initiatives, such as P2P routing services or anonymous remailers that can add supplementary privacy properties to end-to-end encrypted messaging. For example, end-to-end encryption does not usually allow a user to be anonymous to the server or third-party without additional network-level encryption. 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.

[8] And the "Matrix Console" messaging app for Android reportedly has "between 1000 and 5000 downloads".

[9] WhatsApp turns to end-to-end encryption by default in April 2016.

[10] https://whispersystems.org/blog/facebook-messenger/

[11] https://xmpp.org/extensions/inbox/omemo.html

The Tor hidden service protocol offers a platform to develop decentralized and encrypted instant messenger servers. It is used by default 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.

Tor provides only anonymity for network addresses, but not metadata such as the sender, recipient, and time of message such as are kept in the email header in the time of email or can be deduced by the server. There has also historically been work on anonymous high-latency remailers to fix these 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.[12] 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 experimental tools developed on the network level that, while not guaranteeing anonymity, provide some level of encryption. Zero Tier One is an end-to-end encrypted, peer-to-peer virtual network that 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.[13]

5. Towards a set of criteria for categorization of 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, we have proceeded with a mapping based on a defined range of criteria. We do not include the case-by-case mapping details in this paper for lack of space[14] but we briefly introduce the criteria that guided the analysis, and discuss some of its preliminary findings.

All of the 30 projects that are included in the mapping[15] are either centralized, with encrypted messages stored on (but not readable by) a central trusted authority, or decentralized, and so not having no central trusted authority for even storing messag-

[12] http://mixminion.net/

[13] https://github.com/cjdelisle/cjdns/blob/master/doc/Whitepaper.md

[14] The full 30 case studies can be downloaded from https://nextleap.eu

[15] Briar, Caliopen, ChatSecure, CoverMe, CryptoCat, Equalit.ie, GData, i2P, Jitsi, Mailpile, Mailvelope, ParanoiaWorks, Patchwork, Pidgin, Pixelated, Pond, Protonmail, qTOX, Ricochet, Scramble, Signal, SilentCircle, SureSpot, Teem/SwellRT, Telegram, Threema, TorMessenger, Vuvuzela, Wickr, Wire

es. Decentralized systems are either federated (allowing multiple servers, including users setting up their own servers), or peer-to-peer (allowing direct communication between client devices). For the purpose of subsequent investigations with social science methods of ethnography, in-depth interviews and documental research, there are a number of features we seek to identify. We pay particular 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 collection. 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 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…).

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. We should thus distinguish it from the "social federation", i.e. the question about who controls, at a socio-political level, the instances of servers. For instance, from this standpoint, Bitcoin is mostly technically decentralized but socially centralized: there is a single core group creating and delivering the software, while users effectively run the same software that calculates transactions in a decentralized way. In order to analyze the (de)centralization of governance/power structures in messaging, we have to conduct an in-depth investigation. The further ethnographic and sociological analysis will aim at a deeper understanding of different models of socio-economical federation these protocols and tools produce. 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.

6. Preliminary findings and methodological concerns

This diversity poses a methodological challenge of representation and accuracy, which will be further delved into as the research progresses; however, for the time being, this research opens the way to a number of preliminary socio-technical observations of the end-to-end encrypted 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 reflection on the future of privacy-enhancing technologies.

As we had the occasion to observe in previous mapping research on P2P services [9], 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 our mapping covers only 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 the in-depth ethnographic analysis, as a precious source of 'lessons learned' in terms of user recruitment and governance models.

A social perspective is necessary for the design and refinement of technical protocols, with a focus on whether or not users understand and value the various security properties of the protocols. For example, do users understand what a "key" is and forward secrecy? Often protocol designers make assumptions about whether or not ordinary users can understand the security and privacy properties of their protocols. For example, almost all protocols from PGP to Signal use methods such as "out-of-band fingerprint verification" to determine whether or not the recipient of their message really is who they think they are. It is unclear if users actually use these techniques to verify the identity of their contacts. Another example that has been debated in the technical community is deniable authentication. While a protocol may be technically deniable, would this cryptographic deniability hold up socially, much less in court? Answering these kinds of questions influences the kinds of protocols that can be designed by the research community. Lastly, why do only some protocols enable decentralization via open standards? It is unclear if users prefer (or can even tell the difference between) peer-to-peer solutions and centralized services. Between these two extremes, there is the question of how users make trust decisions in open and federated environments such as PGP and XMPP where users could run their own software or delegate this to a trusted group. Answering these questions is vitally important to ground the design of new decentralized protocols and refine existing ones to become decentralized.

The interdisciplinary character of NEXTLEAP project provides us access to several important communities working on improving messaging protocols and encryption, such as the LEAP/Pixelated team, Cryptocat, Open Whisper Systems, Briar, CJDNS, Tor and others. We plan a set of interviews with the teams of three selected projects, as well as observations during important cryptography, decentralization and privacy-related events. We are focusing on both developers and users. Thanks to previous research conducted in the field of activist-targeted technologies, we have connections within several activist user communities in different countries (France, Germany, UK, Austria, Greece, Russia, Mexico, Tunisia, and Lebanon). We will focus on the patterns of adoption/rejection of different messengers/mailing clients, on users' "careers" (e.g. studying usages of encryption and privacy enhancing technologies in dynamic relations to the activist careers and life trajectories), with a specific interest

in the so-called “digital migration problem” (shifting from a non-encrypted tool to using end-to-end encryption).

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 [10], the number of users is explicitly made visible on the websites of the projects. It becomes an important tool for building user communities and empowering the impact of such activist projects. Whereas our analysis of the 30 projects shows that very few projects openly give the number of their active users (possibly due to privacy issues). A further exploration of the three selected cases will investigate these specific politics. In this context, bringing methods of social science to the topic of secure messaging protocols may be useful to elucidate the underlying processes of building user communities.

The analyzed projects propose several solutions to the problem of data storage. Indeed, despite the guarantees of “no personal data collection”, some projects still 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 developing communities are aware of the problem and are seeking for alternatives with minimal data storage, and opt for stronger decentralization. The analysis shows that it is the question of metadata that appears to be an area of active research, stimulating experiments with standards and architectures (e.g. Vuvuzela’s usage of “noise” to obfuscate metadata discussed in Ref. [11]).

A look at visual aspects, such as the design of interfaces 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 the developers perceive them. 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 (e.g. ‘Nestor Makhno’ or ‘Vera Zassulitch’), 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 “encryption is used by jihadists”-type allegations [12]. 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 [13], some of them particularly related to encryption (or the break-

ing 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 needs to take into account, and indeed is already taking into account: architecture is politics, but it is not a substitute for politics [14].

7. Conclusions

The overview of the protocols presented in this short paper is focused on stabilizing a list of potential case studies among decentralized internet messaging projects. A further selection of these will be investigated in depth in the future with qualitative methods, including ethnography and in-depth interviews. This is deemed necessary as the proliferation of projects addressing encryption, decentralization, or both, in the field of messaging has not led so far to massive adoption outside of a few large centralized companies such as WhatsApp, for a number of factors that go beyond technology to include difficulty of use, economic sustainability, and unclear socio-legal status of encrypted communication. Thus, the development of a related Internet science requires insight from both social science and ICTs to understand the successes and failures in the design of end-to decentralized protocols.

Considering the lively and constantly-evolving ecosystem of standardized and non-standardized projects in the field of decentralized and encrypted 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 short paper 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-requisite to an in-depth, case study-based inquiry. However, this social science research is ultimately meant to feed back into the 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.

References

1. Rogaway, P. (2015). “The moral character of cryptographic work”. IACR Distinguished Lecture at *Asiacrypt 2015* http://web.cs.ucdavis.edu/~rogaway/papers/moral.pdf
2. Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., & Smith, M. (2015, May). SoK: Secure Messaging. In *2015 IEEE Symposium on Security and Privacy* (pp. 232-249). IEEE.
3. Whitten, A., Tygar.J. D. (1999). Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8* (SSYM'99), Vol. 8. USENIX Association, Berkeley, CA, USA, 14-14.
4. 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
5. Prokop, A. (2015) "Solving the WebRTC Interoperability Problem", in *NoJitter*, http://www.nojitter.com/post/240169575/solving-the-webrtc-interoperability-problem
6. 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
7. Marlinspike, M. (2013). "Advanced cryptographic ratcheting", published in OpenWhisperSystems blog on 26 of November 2013, https://whispersystems.org/blog/advanced-ratcheting/
8. 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
9. 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.
10. 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
11. Van den Hooff, Lazar, et al. (2015) "Vuvuzela: Scalable Private Messaging Resistant to Traffic Analysis", in *Proceedings of SOSP'15*, http://dx.doi.org/10.1145/2815400.2815417
12. 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
13. Musiani, F., Cogburn, D. L., DeNardis, L., Levinson, N. S. (dir.) (2016), "The Turn to Infrastructure in Internet Governance", Palgrave Macmillan.
14. Agre, P. E. (2003). P2P and the promise of internet equality. *Communications of the ACM*, *46*(2), 39-42.