“Standardising by running code”: the Signal protocol and de facto standardisation in end-to-end encrypted messaging

Ksenia Ermoshina, Francesca Musiani

"Standardizing by running code": The Signal protocol and *de facto* standardization in end-to-end encrypted messaging

Ksenia Ermoshina

Francesca Musiani

Center for Internet and Society, CNRS, Paris, France

Centre Internet et Société (CIS-CNRS), 59-61 rue Pouchet, 75849 Paris cedex 17
Corresponding author email: francesca.musiani@cnrs.fr

Ksenia Ermoshina (PhD, MINES ParisTech) is a postdoctoral researcher at the Center for Internet and Society (CIS) of the French National Centre for Scientific Research (CNRS), and is an Associate Researcher at the Citizen Lab, Munk School of Global Affairs, University of Toronto. Her research focuses on information operations within the Russian-Ukrainian armed conflict, including digital threats to journalists and civil society organizations, Internet censorship, and surveillance. Her previous work as a postdoctoral fellow with the NEXTLEAP research project studied the usage and development of end-to-end encrypted messaging and email protocols and clients. Ksenia will join the ranks of CNRS as an assistant research professor in October 2019.

Francesca Musiani (PhD, MINES ParisTech) is Associate Research Professor (*chargée de recherche*) at the French National Centre for Scientific Research (CNRS), and Deputy Director of its Center for Internet and Society (CIS). She is also an associate researcher with the Centre for the Sociology of Innovation (i3/MINES ParisTech), and academic editor for the *Internet Policy Review*. Her research explores Internet governance. She was one of the Principal Investigators for the European H2020 NEXTLEAP (Next-Generation Techno-Social and Legal Encryption, Access, and Privacy) project (2016–2018).

"Standardizing by running code": The Signal protocol and *de facto* standardization in end-to-end encrypted messaging

After Edward Snowden's revelations, encryption of online communications at a large scale and in a usable manner has become a matter of public concern. The most advanced and popular among recently-developed encryption protocols is currently the Signal protocol. While the Signal protocol is widely adopted and considered as an improvement over previous ones, it remains officially unstandardized, even though there is an informal draft elaborated towards that goal. The analysis of how this protocol was introduced and swiftly adopted by various applications, and of subsequent transformations of the encrypted messaging ecosystem, sheds light on how a particular period in the history of secure messaging has been marked by a "*de facto* standardization". What can we learn about existing modes of governance of encryption and the histories of traditional standardization bodies, when analyzing the approach of "standardization by running code" adopted by Signal? And finally, how does the Signal protocol challenge a "linear", evolution-based vision of messaging history? Drawing from a three-year qualitative investigation of end-to-end encrypted messaging, from a perspective informed by science and technology studies (STS), we seek to unveil the ensemble of processes that make the Signal protocol a quasi-standard.

Keywords: Encryption; secure messaging; standards; standardization; Signal; protocols

Introduction

As it can hardly be disputed anymore, Edward Snowden's 2013 revelations have been a landmark event in the development of the secure communication field. Encryption of communications at a large scale and in a usable manner became a matter of public concern, with a new cryptographic imaginary taking hold, one which sees encryption as a necessary precondition for the formation of networked publics (Myers

West, 2018). Alongside the turning of encryption into a full-fledged political issue, the Snowden revelations catalyzed long-standing debates within the field of secure messaging protocols. The cryptography community (in particular, academic and free software collectives) renewed their efforts to create next-generation secure messaging protocols in order to overcome the limits of existing protocols, such as PGP (Pretty Good Privacy) and OTR (Off-the-Record Messaging). As next-generation encryption is shaping the ways in which we can securely communicate, exchange, store content on the Internet, it is important to unveil the recent and less-recent history of these protocols and their key applications, to understand how the opportunities and constraints they provide to Internet users came about, and how both developer communities and institutions are working towards making them available for the largest numbers.

One of the leading motivations behind this effort consisted in facilitating key exchange and key verification processes[1], previously identified as the main obstacles to mass adoption of encryption (Whitten & Tygar, 1999). The most advanced and popular of these next generation protocols is currently the Signal protocol (formerly called Axolotl[2]), firstly introduced by the messaging application Signal, and adopted or forked by other instant messaging applications, ranging from WhatsApp and Wire to Matrix and Conversations. While the Signal protocol is widely adopted and considered as an improvement over both OTR and PGP, it remains officially unstandardized, even

1 In public key cryptography, key exchange is the method by which cryptographic keys are exchanged between two parties; key verification is any way that lets you match a key to a person, making sure that it is indeed that person who uses the key (see e.g. https://ssd.eff.org/en/glossary/key-verification)

2 From the name of an aquatic salamander with great self-healing capabilities, as the protocol can "heal itself" by preventing, in some cases, an attacker from accessing encrypted communications even after a session key is compromised.

though there is an informal draft elaborated towards that goal by the protocol's creators, Trevor Perrin and Moxie Marlinspike.

The analysis of how this protocol was introduced and swiftly adopted by several applications, and of the subsequent transformations of the encrypted messaging ecosystem, sheds light on how a particular period in the history of secure messaging has been marked by a "*de facto* standardization" revolving around Signal. As we will address in more detail later, we call "*de facto* standardization" the process through which a feature or a system that is successfully implemented in a variety of scenarios is identified as "*something that works*", is iterated and redeployed by others beyond its original developers, and becomes a quasi-standard through practice and implementation, therefore, without undergoing any institutionalized standardization procedure. By analyzing how this process has happened in the case of Signal, we will address the following questions. What can we learn about existing modes of governance of encryption and the histories of traditional standardization bodies, when analyzing the approach of "standardization by running code" adopted by Signal? And finally, how does the Signal protocol challenge a "linear", evolution-based vision of messaging history?

We analyze the case of the Signal protocol as it unveils an alternative history of secure messaging, where protocols mutually influence and borrow from each other; we will see how they even try to revert to their predecessors[3] and renew them in the light of new norms and requirements brought forward by Signal, such as forward secrecy[4],

[3] See the "renewal" of OTR with OMEMO, and of OpenPGP with Autocrypt, described further in this paper.

[4] Forward/future secrecy is a feature ensuring that a user's session keys will not be compromised even if the private key of the server is compromised, and, in particular, it is meant to protect past sessions against future compromises of secret keys or passwords.

nowadays accepted as the necessary minimum. Drawing from a three-year ethnographic investigation of end-to-end encrypted messaging, from a perspective informed by science and technology studies (STS), we seek to unveil the ensemble of processes that make the Signal protocol a quasi-standard. In its conclusions, the paper seeks to comment on the governance implications of this quasi-standardization process, both for the history of end-to-end encrypted messaging field and for the future of the main existing Internet governance standardization bodies, such as the IETF.

The structure of the paper is as follows. First, we situate our work within the relevant literature, in particular the multi-disciplinary work on the making of standards, and Internet governance research, in particular its recent STS-informed strand. We then proceed to outline our methodology and data collection protocol before moving on to situating the development of Signal in its socio-political, technical and economic context. This is followed by two empirically-driven sections that describe what we call the "de facto standardization" dynamic around Signal, and the complex and closely intertwined relationship between the culture and ethos of developers, the possible licensing choices, and the elaboration of business models that unfolded in this case. Finally, we move from the recent history to the future of Signal by describing its "feedback loop" effect (how older protocols are being refurbished due to its most recent developments) and offer some conclusions and directions for future research.

Looking at history of encryption through the STS lenses of standardization and Internet governance

This paper seeks to contribute primarily to two fields of analysis in science and technology studies that heavily interact with work done in other disciplines, including Internet history, economics and political sciences to name but three. The first one has a long-standing tradition in STS: the study of standards and their making -- how technical objects and protocols become the norm in their respective domains and sectors of activity, either by official, institutionalized means or treading less linear paths, or a mix of both. The second field has only recently begun to be connected to STS, but is otherwise a widely studied research field for scholars in a large number of disciplines: it is the study of Internet governance. By looking at the dynamics of "informal standardization" of encryption and understanding it, from an STS perspective, as part of Internet governance, we seek to shed light on those aspects of encryption today that have to do with its adoption by the general public, keener attention to usability, and design as a social and political choice.

As the field of modern encryption protocols is rapidly evolving (as Signal lead developer Moxie Marlinspike summarizes in a landmark blogpost, "the ecosystem is moving"; Marlinspike, 2016), we acknowledge that some of the empirical elements obtained through interviews with key actors of this field have already in a way become history, as some of the controversies faced by Signal have been resolved since then, and the messenger, as well as the protocol, have been transformed. However, as the authors continue to be engaged in the field of secure messaging (in particular, one of them is a usability researcher for a messaging application), they are constantly exposed to critiques and debates over Signal's design and governance choices. We have tried to reflect some of these debates in this paper. In this way, conducting an embedded, real-time research and writing an STS-informed history of encryption protocols also allows us to understand the evolution of the debate about key obstacles to mass adoption of

encryption, as well as adversarial capacities and subsequent risks shared by (potential) users.

The making of standards as a making of (Internet) history

Standardization processes have been a long-standing concern of science and technology studies (STS). As noted by Lawrence Busch, there is an "intimate connection between standards and power", a power that "lies in their very subtlety" (Busch, 2011). Geoffrey Bowker and Susan Leigh Star had long since noted that standards play an important role in the making of public policy and, more broadly, of social order: "standards […], however imbricated in our lives, are ordinarily invisible […yet w]ho makes them, and who may change them? When and why do they become visible? How do they spread?" (Bowker and Star, 1999). A number of case study analyses of competing standards in information technology have contributed to shed light on these processes, including the birth of the QWERTY keyboard (David, 1985) and the VHS vs. Betamax controversy (Besen & Farrell, 1994).

Technical standards, according to STS approaches, are implemented at once in physical forms, in social and economic interaction, and in their intended or inferred use. Several standards are developed (and recognized as such) intentionally, and result from a regulatory action or voluntary adoption. Standards endowed with formality are developed in dedicated organisations, such as the International Organization for Standardization (ISO) or, for the Web, the World Wide Web Consortium (W3C), and take the form of documents that describe objects, their properties and the efforts to which they can be subjected without breaking or being compromised. One of these formats is the IETF's Request for Comments, or RfC, whose very evolutions over time

contribute to make visible the transformations of the IETF, its organizational forms and practices (see Braman, 2016). Internet governance literature in the political science and legal traditions has extensively examined "formal" standard development processes, addressing the particular mix of private regulation and governmental/intergovernmental intervention that characterizes most of them (see e.g. Weiser, 2001; Bygrave and Bing, 2009).

The meaning and pervasiveness of these standards in our everyday life often escapes our understanding as "common" users (Star, 2009), in a myriad of different situations. Given the attention that STS approaches give to the invisible and discrete processes that make a technical object what it is, it is not surprising that the making of standards -- the processes of standardisation themselves -- have been investigated by STS scholars in a variety of domains related to science and technology. The publication of a standard is only the beginning -- and indeed, as our case study for this paper will show, sometimes it is not even necessary; the adoption of something close to a 'standard' may occur de facto or accidentally, with seemingly minor decisions and actions becoming crucial for the development of a field in a particular direction.

In some instances, what determines the adoption of an object, process or protocol as a standard is its ability to circulate and get recognized: factors such as popular demand, perceived quality, and the credibility of its developers become crucial for a success. As Mendel (2006) points out, a number of dynamics support this circulation, from discussion forums to industry publications, as well as normative obligations to follow best practices: by not observing the standard, companies risk being locked out of their own sector (Murphy and Yates, 2009, 71). The stabilization of a standard -- usually seen as its widespread acceptance and its embeddedness in interactions -- has been variably identified by scholars as path dependency (Arthur,

1989), irreversibility (Callon, 1991) and trajectory (Bowker and Star, 1999), but such analyses point to one main result: successful standards, however they come about, are often difficult to modify or replace. One clear example of this, when it comes to the Internet, is the Internet protocol for identification and location of computers on the Internet, IPv4. For computers today, the adherence to the protocol is somewhat of a "self-regulation for its own good"; IPv4 is intrinsic to all operating systems for them to be functional. However, as Laura DeNardis (2009) has finely analysed, IPv4 is unable to face the steady rise in the global demand for connectivity; nonetheless, the transition to its successor IPv6, providing a virtually unlimited number of addresses, has proven complicated, is implemented only gradually, and is still ongoing.

Thus, the spread of standards through social, technical and institutional media can be understood as a complex and multifaceted process; mechanisms of standardization are economic, social and technical -- with different degrees of intentionality -- alongside the "official" institutional practices of certification and harmonization (Loconto and Busch, 2010).

STS approaches to Internet governance: a focus on "mundane practices"

A number of authors, some of whom with STS sensibilities but overall coming from a broader disciplinary spectrum, have examined in recent years how the concept and the practice of Internet governance may be reconsidered in light of an increasing number of informal uses, practices, norms that affect the distribution and the exercise of power on the Internet. While Michel van Eeten and Milton Mueller argue that the definition of governance should include "environments with low formalization, heterogeneous organizational forms, large number of actors and massively distributed authority and decision-making power" (van Eeten & Mueller, 2013), Sandra Braman

suggests that the definition of governance may go as far as including "decision making with constitutive (structural) effect whether it takes place within the public or private sectors, and formally or informally" (Braman, 2009). Governance, according to Hofmann et al. (2016) may even be just a "side effect of actions with non-governance-related aims".

Addressing the macro questions of politics and power related to IG requires unpacking the micro practices of governance as mechanisms of distributed, semi-formal or reflexive coordination, private ordering, and use of Internet resources (Epstein, Katzenbach and Musiani, 2016). Seemingly stable arrangements of IG heavily rely, for their functioning and their very existence, on a number of mundane activities and practices of internet design, regulation, and use, making Internet governance an "accomplishment embedded in everyday interaction" (West & Zimmerman, 1987, p. 125).

The STS-driven sensibility for social order as continuous and contested processes translates into a growing attention to the mundane practices of all those involved in providing and maintaining, hacking, developing and testing, and even using the Internet (Musiani, 2015), thus expanding the notion of governance in IG. These diverse practices are elements constitutive to articulating and challenging established, emerging or contested norms – the "doing" of IG. As such, at an analytical level, borrowing from the rich STS tradition of studying the scientific enterprise (e.g. Latour & Woolgar, 1986), this ensemble of invisible work and mundane practices is not treated as markedly separate from the designated IG institutions. Thus, IG as a continuously emerging and dissolving order, in this view, is – rephrasing John Law (1992, p. 382) – an effect generated by heterogeneous means. We will see how this focus on mundane practices is useful to unveil the informal dimensions of governance in processes such as

de facto standardization, licensing, etc. The attention to those mundane practices helps tracing the development of encryption protocols as intertwining "micro histories", where crucial decisions often happen during offline, off-the-record gatherings, or on small-scale mailing lists.

Methodology and data collection

Data was collected within the frame of the H2020 CAPS research project NEXTLEAP (2016-2018). After completing a survey of 30 cases of encrypted messaging applications and a history of encryption protocols (Ermoshina, Musiani & Halpin, 2016), we have proceeded to select a few applications that could be studied in more detail and in a qualitative fashion, with a case-study methodology including in-depth interviews and both online and live ethnography. Signal has been selected as one of these cases, due to its central role in the encrypted messaging ecosystem in terms of number of users, media attention, available documentation on protocol development. Thus, our approach can be described as a multi-sited ethnography, inasmuch as we have undertook research in, and between, several online and offline locations as part of our study, and we have also explicitly conceived a protocol/system (Signal) as "part of a larger context that exceeds the boundaries of the field site" (Muir, 2011; see also a reflection on the method by first proponent George Marcus, pointing out that multi-sited ethnography has been "most creative, critical, and interesting where it has been involved with the [STS] study of distributed knowledge systems", Marcus, 2012).

STS methods allow to analyze the interfaces of messaging apps as "meeting points" between the intentional goals of developers and the needs of users (Oudshoorn & Pinch, 2005). STS aim at providing a fieldwork-driven sense-making of emerging systems and communities of practice, doing 'analytical thick descriptions' (for a recent

treatment of the concept see Ponterotto, 2006) of events, artifacts, organizations – in particular, moments of crises, debates, controversies – to try and understand the life of a technical artifact, from its creation to its appropriation and reconfigurations by users, to its becoming a subject of public debate, of governance, of lobbying. The primary methodology to achieve this goal is to observe, for relatively prolonged periods of time, specific case-study groups or communities, conducting on the side in-depth interviews with their members and reading appropriate documentation such as release notes or accounts of working sessions.

Just as we seek to have a nuanced understanding of developers' motivations and the representations they have of users and their needs, in the tradition of "user studies" developed within STS, we understand users not as a homogeneous and passive group, but as active contributors participating in innovation and co-shaping technologies, which is possible in software development via routes such as bug reporting, pull requests on code, mailing list comments, and in person contact of users with developers.

Interview subjects that were developers were mostly selected due to pre-existing personal relationships with the cryptographic research community of *** research team members. We also reached out to some developers via the GitLab and GitHub pages of the projects without personal connections (e.g. Ricochet, Conversations). In contrast, user studies were done with individuals that were selected more by chance via their attendance at training events in their local environments (both high-risk, in the case of Ukraine and Russia, and low-risk in the case of France, Germany, Austria and the United Kingdom) or conferences in pre-selected venues that were determined to be likely to attract high-risk users that lived in areas that, due to the level of repression, made it difficult if not impossible to interview them in their native environment, or would make it such that they could not speak openly there due to repression. This was

the case for users from Egypt, Turkey, Kenya, Iran, where the interviews took place in March 2017 at the Internet Freedom Festival and at RightsCon. All interviews were made between Fall 2016 and Spring 2017, for a total of 52 interviews. This article draws in particular from six of those, conducted with developers of Signal-based apps. Moreover, as the authors continue to be engaged within the field of cryptographic tools and protocols, they have been exposed to many ongoing debates in the community around Signal protocol and its implications for the field of encryption in secure messaging. This embedded research helps us put current and further developments of Signal protocol in a larger perspective.

The Signal protocol and "mass encryption": the historical turn of a field in the making

The field of end-to-end encrypted, secure messaging is a very diverse and complex one, both for technical reasons and socio-political and economic factors. This section of our article situates the birth and development of the Signal protocol in this context, addressing the changes brought about in the secure messaging field by the Snowden revelations, and examining the mostly pre-Snowden, recent-history debates that have taken place around Signal's predecessors OTR and PGP. We show how Signal seems to have drawn its success from a mix of continuity with previous protocols and ability to disrupt and innovate their well-known flaws; we also choose to "take the actors seriously" as they talk about Signal protocol as a new "norm", a reference for their own protocols and applications.

Pre-existing the Snowden revelations, but strongly reconfigured by them, secure messaging is a lively and constantly-evolving ecosystem of projects. Developers seek, in particular, to apply the technique of end-to-end encryption to messaging systems:

among the most widely known tools pertaining to this category are Signal, Telegram[5] and WhatsApp, each with different motivations and solutions for implementing encryption[6].

A recent systematization of knowledge paper on secure messaging argues that the field suffered from the "lack of a clear winner in the race for widespread deployment and the persistence of many lingering unsolved research problems", as well as discrepancies between "grandiose claims" and actual provided security (Unger et al., 2015). Part of the reason of the field's diversity and complexity is the relatively short life span of several projects, for a number of reasons including technical and academic experimentation that did not deliver as hoped or expected, the failure to develop an economic model, internal governance, and the inability to rally a critical mass of users around the app, often due to a lack of ease-of-use. The target audience of the applications, especially those born post-Snowden, is far from being limited to tech-savvy and activist groups; several projects are aimed at widespread use. A majority of members of the technical crypto community consider user-friendliness and usability as the main issue that stands between the wish of large-scale adoption and its realization in practice. Encryption software is often understood as an instrument in broader struggles to define the meaning of Internet freedoms (Hellegren, 2017); end-to-end encrypted messaging tools are the subject of a two-faced discourse, on empowerment and protection of fundamental civil liberties on one hand, and allegations of links to terrorism on the other, the latter being fueled by previous narratives about decentralized

[5] Interestingly, Telegram does not actually offer end-to-end encryption by default, but is nonetheless widely considered as part of the secure messaging market.

[6] Elements in the following two paragraphs have previously been discussed in more detail in (Musiani & Ermoshina, 2017).

technologies and peer-to-peer as technologies favoring both empowerment and illegal practices (Musiani, 2013). These issues are taking place in the broader context of discussions about governance by infrastructure and civil liberties (Musiani et al., 2016).

Indeed, after the Snowden revelations, several companies, in particular those based in the United States, have implemented a number of cryptography-based organizational and technical responses aimed at restoring user trust in their cloud-based services. This dynamic has been identified as a "cryptographic turn" opening up new issues and questions from both legal and political standpoints (Rubinstein & Van Hoboken, 2014) and is considered a new phase of the 1990s "Crypto Wars" (Froomkin & McLaughlin, 2016), where cryptographic features of various types of protocols behave as micro-instruments of governance. They determine, at a technical level, the limits and possibilities of collaboration with governmental and private actors. For example, server-side encryption does not offer the same conditions as end-to-end encryption[7] in case that a "forced decryption" is required by law enforcement procedures. Cryptographic properties such as forward secrecy or non-repudiation, as well as key management[8], technically define terms and conditions of possible interactions and data exchange with third-party actors, would it be private or public sector.

[7] Server-side encryption means that data is encrypted on the server (of the company providing the messaging services). End-to-end encryption posits that only the communicating parties can read the message, which is encrypted in transit *and* on users' terminals.

[8] For the definitions of forward secrecy, see above. Key management includes all operations related to the management of cryptographic keys in an encrypted system, including their generation, exchange, storage, use, destruction and replacement. Non-repudiation is the assurance that someone cannot deny the validity of a particular operation; in cryptography, the concept refers to a service that is able to provide proof of the origin of data as well as their integrity.

In this context, many actors in the field share a tacit agreement that the Signal protocol's Double Ratchet[9] is currently the leading protocol for instant messaging. In order to better understand how the Signal protocol interacts with previous and further developments of encryption protocols, one should look at the historical debates around two major protocols, that have been somehow dominating the ecosystem for many years before Snowden: OTR (Off-the-Record, used to encrypt instant messages sent over XMPP) and PGP (Pretty Good Privacy, used to encrypt emails).

The main problems of PGP discussed in the cryptographic community, and shared by advanced users, could be resumed to two main aspects: complex key management and lack of repudiation and forward secrecy. The crisis of "public key infrastructures"[10] and of the very concept of cryptographic keys and signatures was also highlighted by digital security trainer communities and international NGOs promoting privacy-enhancing technologies, such as Tactical Tech and the Electronic Frontier Foundation (Musiani & Ermoshina, 2017). While still offering one of the most cryptographically robust solutions, PGP clearly faces usability challenges.

The OTR protocol started as a research project at UC Berkeley in 2002, and was first released in 2004. OTR developer Ian Goldberg described himself, in our interview with him, as a very early PGP user and mostly an "email person"; it was his student,

9 The Double Ratchet algorithm is a key management algorithm developed by the creators of Signal, Trevor Perrin and Moxie Marlinspike, in 2013, which manages the ongoing renewal and maintenance of short-lived session keys after a first key exchange. It is a "double" ratchet because it combines a cryptographic component with a key derivation function.

10 Public-key (or asymmetric) cryptography is a cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys known only to the owner. PKI is the set of roles, policies, and procedures needed to create, manage, distribute, use, public-key cryptography.

Nikita Borisov, who drew Goldberg's attention to the growing field of social networking and instant messaging, thus identifying a new security gap to be filled. Ian's description of the mission behind OTR relates this protocol to preexisting technologies, as a way to respond to challenges that were not properly addressed by PGP:

> "your choices at that time were either completely unprotected communication neither encrypted nor authenticated, or PGP in which case it's confidential and authenticated unless your key leaks in which it's not confidential and the authentication with digital signatures leads to non-repudiation" [Interview with Ian Goldberg, May 2018]

While OTR's solution to use per-conversation keys offered good repudiation and forward secrecy, it did not permit group chat encryption, as, in Goldberg's words, "*that was how instant communication worked back then: you had to be online at the same time*": OTR's design was inspired by Aim and other existing tools, that required synchrony. Moreover, while popular in high-risk activist communities, OTR-based applications (such as Jabber) were widely criticized for the lack of multi-device support and other usability problems.

These shared concerns were addressed by the Signal protocol, according to Goldberg: "*Text Secure, and later Signal, basically took OTR protocol and added basic features to it to make it work in an asynchronous setting*". Using per-conversation key material in a similar manner to OTR, it did not force complex key management on the users. It maintained properties of repudiation and forward secrecy by virtue of the Axolotl Diffie-Hellman key ratchet[11], but added "future secrecy" so that messages could not be read at any point in the future in the case of a key material compromise (Cohn-Gordon et al., 2016; see also note 4). It solved the asynchronous messaging problem by

11 https://github.com/trevp/double_ratchet/wiki; https://signal.org/docs/specifications/x3dhratchet/wiki

virtue of allowing longer-term pre-keys managed by the Signal server, and offered group messaging implemented as point-to-point messaging.

As the Signal protocol initiated a dialogue with the previous crypto protocol "tradition" (mainly by addressing the aforementioned limits of OTR and PGP), it quickly attracted the attention of the academic cryptographic community, and only minor flaws were found (Frosch et al., 2016). Although alternative approaches were developed and widely deployed, like MTProto by Telegram, these protocols developed their own cryptographic primitives and so received less attention from the academic community, although a number of bugs and usability problems were revealed (Jakobsen and Orlandi, 2016; Abu-Salma, 2017). Interestingly, while Signal has deeply influenced the crypto protocol field, it did not depart from previous efforts, but drew from their well-known flaws: it is this continuity that can partly account for the interest of crypto experts.

With minor variants implemented in the vastly popular WhatsApp messenger, the core Signal Protocol seems well on its way to clearly replace the use of XMPP+OTR and even a competitive, if somewhat "boutique", feature for mainstream messaging services (as shown by the adoption of the Signal protocol as an option by both Google Allo and Facebook Messenger). Encrypted messaging applications like WhatsApp, Telegram, and Signal are now the default application of this kind for users that consider themselves to be high-risk. Usability studies have shown that although Signal (similar to OTR) is easy to set up and use, even highly-skilled users fail to use verification correctly. Currently, Signal is centralized, as a single server mediates the

setup of the protocol in most widespread deployments (WhatsApp, Google Allo, Facebook Messenger, Wire)[12].

Open-source alternatives that claim to use the Signal protocol or its forks[13] exist, such as the centralized application Wire that uses a fork of Axolotl protocol called Proteus. Parts of the Signal protocol were copied by a draft XMPP Foundation standard called OMEMO, for use by applications such as Conversations and ChatSecure, which led to usage of Signal's double ratchet in federated projects. Another decentralized project called Matrix has reused parts of the Signal protocol to integrate them into their own cryptographic library called Olm[14]. While Signal appears to be widely adopted and considered an improvement over both OTR and PGP, the core Signal protocol remains officially unstandardized, even though the protocol's creators Trevor Perrin and Moxie Marlinspike have produced an informal draft after considerable "*community pressure*" [as Matrix.org lead developer Matthew Hodgson puts it].

A "quasi-standardization" process

As we have seen earlier on in the literature review discussion, standardization can happen in ways and arenas other than formal working groups and bodies through codified procedures. This section delves into the ways in which the process which we

[12] It is interesting to note that, although we will not focus on it in this particular paper, centralization *per se* has become another controversy, promoted by Moxie as a better governance option for secure messaging apps, compared to decentralized architectures considered as less stable.

[13] Forking a piece of software during its development process means that developers take a copy of its source code and start independent development on it, creating a separate piece of software. An act of forking is generally not merely a technical issue, but involves a (governance/organizational) change, possibly conflictual, in the developer community.

[14] https://matrix.org/git/olm/about/

have called ‘de facto standardization’ (or quasi-standardization, or standardization ‘by running code’), happens in the Signal case. We see how this dynamic is boosted by a number of factors, which include a relative mistrust in standardizing bodies (not necessarily in their trustworthiness, but in their suitability to produce results adapted to the encrypted messaging case), a perceived need for “best practices” from which standardization would follow, and the necessity to build a ground for common knowledge before attempting to “box” it.

While most users we have interviewed, including high-risk users, do not appear to have standardization as an explicit priority, developers care deeply about standards as “something they would eventually be working on”, namely for increasing the ‘dialogue’ between applications and reduce the silo effect:

> In the long term I am not opposed to the idea of standardizing, it’s great to have a reference for interoperability. [Michael Rogers, Briar lead developer]

Standardization serves as a *reference* and thus as an important communication or mediation instrument, that helps the security community understand each other and build a ground for common knowledge (such as cryptographic libraries), and also guarantees a smoother development of new applications on top of standardized protocols. The approach to standardization as a tool for cooperation is, for instance, largely promoted by the decentralized secure messaging application Delta.Chat, which reuses a set of open standards (namely, PGP and SMTP/IMAP) and advertises the usage of standards as a proof of transparency, interoperability and openness, while criticizing Signal approach for its “silo effect”[15] and “control” over users.

15 One of the Delta Chat’s core developers, Holger Krekel, uses the term “messaging silos” to describe the effect of non-interoperability of the new secure messaging applications that

Yet a widespread discontent with existing standards bodies is expressed by developers, for several reasons. Developers underline recent transformations of these organizations, referring to a previous 'golden age' of standardizing bodies, when their mode of existence was closer to that of FOSS communities. Our respondents note the growing importance of private actors as stakeholders within standardizing bodies.

> "My impression of the IETF is that it's not the same beast it was in the early days. There was a time when it was a group of enthusiastic people who would come to the IETF with an idea that was sort of halfway finished and they'd say look I wanna let everybody know about this, let's knock it into shape and we'll all build on it. I think it's become a much slower moving and more adversarial environment. This area of technologies has attracted more money and more corporate participation and therefore, conflicts of interest". [Michael Rogers, Briar lead developer]

This institutionalization of standardizing bodies and their progressive removal from coding communities creates an environment that is less suitable for experiments and unfinished projects:

> "I think that [an automated, periodical clearing out of message history] is something that we will implement, it just probably will not be standardized because the XMPP community is very conservative. I don't think… they don't fully get it. It's something that users want… so why?... I don't know. They end up in that old school stuff". [Chris Ballinger, ChatSecure developer]

As Callon (1986) would put it, standardization implies the "translation" of a protocol as a sociotechnical experiment into a pre-standard, able to "enroll" and convince various agents within evaluation bodies. Standardization involves collective work that opens up the core-set of protocol authors to include external experts from

> build their own protocols instead of reusing existing standards that could, according to him, foster interoperability and give users "more freedom".

standardizing organizations, some of them being far from users' experiences and needs, and from the "real" economy of the encrypted messaging field -- a process that is hardly appealing to some developers as it is seen as time-consuming in early stages of project development:

> "I wouldn't really think about submitting something to the IETF on early stage these days because I think that would probably involve a lot of work to convince other people to allow it to become a standard… and obviously everybody would have their own thoughts of how better to work." [Michael Rogers, Briar lead developer]

Instead, most developers share the philosophy that they would build the application first, and then focus on standardization and decentralization via the use of open standards:

> I used to work with W3C a long time ago and I am very aware of how they work and that they may have some limitations. We want to get Matrix as mature enough and solid and stable enough, then we can pass it over to a proper governance organization but right now it's still evolving very rapidly. [Matthew Hodgson, Matrix.org lead developer]

In the case of secure messaging, it is still felt that more development is needed on the code, and standardization would only slow down existing development efforts.

Indeed, a new way of '*quasi-standardization*' or 'standardization by running code' is being practiced in the field of end-to-end encrypted messaging applications, around the Signal protocol. In this process, a quasi-standard is defined as "s*omething that works*" and that's been iterated and redeployed by others. In this sense, all of the various Signal protocol deployments (e.g. Wire, WhatsApp and OMEMO-based apps such as Conversations and ChatSecure) work as crash-tests for the protocol, where the protocol gets forged by usage. As Michael Rogers explains it, Signal's approach

consists in first developing the protocol until the stage of "something that works" is reached. Only then comes the phase of documentation, in case there is a demand for interoperability or reimplementation from the developer community. Standardization, if it arrives, happens at a much later stage.

The Signal protocol, characterized by the double Diffie-Hellman ratchet, is now considered the best practice in the field and becomes a trend-setter for other projects in terms of privacy and security features (e.g. forward secrecy and future secrecy for example). Developers, even those working in federated (e.g. Conversations) or peer-to-peer (e.g. Briar) projects, see the Signal solution as one of the best designs available, even if it is not fully standardized.

Forking the Signal protocol: Licensing problems and non-standardization as a business-model

The history of the Signal protocol interestingly illustrates how, in the field of secure messaging (as it is, of course, the case in other areas of software development) there is a complex and closely intertwined relationship between the culture and ethos of developers, the possible licensing choices, and the elaboration of business models. This section addresses how this nexus of issues has unfolded around Signal. In particular, it shows how some forks of the Signal protocol, such as Wire, have travelled a bumpy road towards re-implementation of the protocol, primarily due to Signal's intentional non-release of complete specifications as part of its business model, and to a controversy around licensing. Indeed, the reason why standards are different than 'mere' code is that they describe the specifications for code, and this code may then be independently implemented in conformance to the specification with varying licensing options, ranging from open source to proprietary options. Thus, licenses are relevant to

business development in relation to protocols, and the choice not to release full specifications shapes the field at various levels, including the politics of open/closed source and the structuring of the crypto community.

The field of instant messaging applications has been deeply transformed with a number of implementations of the Signal protocol, but also because of the growing popularity of other secure messaging tools, such as Telegram, Threema or Wickr that use their own protocols. The turn to encryption has modified the market and brought considerable changes on the level of governance, engaging important private sector players in the game:

> What is happening last two years [2014-2016] is fantastic, with a number of messengers popping up and also greater publicity around Axolotl or Signal… Snowden also talking about it... So this is something that is really good for the industry. And we've seen it's triggered even the big ones who started using encryption [Alan Duric, CTO of Wire]

One of the most well-known and popular forks of the Signal protocol is called Proteus and is used by the chat application Wire. Wire was launched by ex-Skype developers, with a desire, according to its CTO, to respond to "one of the biggest gaps that was missing on the market, related to privacy and security". Wire's primary targeted user group is identified as "privacy-aware consumers".

As Wire is not aimed at activists or at a tech-savvy audience, but at the average user, one of their main concerns was to build a usable interface and integrate new features that would distinguish them from other end-to-end encrypted messengers. Thus, Wire supports drawings, GIF exchange, large end-to-end encrypted group chats, multiple-participant group video calls, disappearing timed messages, file transfer. A number of our interviewees have underlined the aesthetic aspect of Wire's UI as an advantage, favoring Wire's widespread adoption as opposed to Signal. Another of

Wire's selling arguments is the voice calls quality and encryption, as it offers end-to-end encrypted voice calls using a specific protocol based on constant bit rate encoding[16].

The underlying Wire encryption protocol, called Proteus, is a fork of Axolotl with "*pieces that were needed to have support for the multiple devices as a standalone*", in the words of Alan Duric. However, difficulties and tensions have been observed around Wire's attempts to reimplement the Signal protocol. Some of these difficulties are due to the lack of specifications (documentation)[17]:

> The problem there was with Axolotl, if you wanted to build it completely from the specification, there was, I would even say on purpose, not enough available documentation. […] I was very naive and went to Moxie last year in June and asked him to review our implementation and we would pay him a very good money for that. Instead he said you can pay 1,5 million, and I will keep your binaries and will help you to get going the implementation. And then I was like… yeah, exactly… What happened afterwards – he said he would sue us and started threatening about it. And you know that's a threat, and our legal guys said that's a threat and it needs to be legally handled. So we sued him for a threat. And then it was just settled. He dropped his charges, we dropped our charges and we are using Axolotl the way we do and how we would like. [Alan Duric, Wire]

Therefore, the lack of specification somehow becomes a business model and obliges developers to re-code the protocol from scratch sometimes using other programming languages. However, since then, Signal has published a draft specification for its encryption protocol[18]. As our respondents explain, this would likely not have happened without the pressure from other developer communities. Indeed, the

16 https://medium.com/@wireapp/a-major-upgrade-to-calling-9ac8780741a1

17 This independent audit of OMEMO protocol includes a dedicated part on Signal protocol and refers to lack of documentation: https://conversations.im/omemo/audit.pdf

18 https://whispersystems.org/docs/specifications/xeddsa/

Matrix.org team, as well as other developers, reached out to Signal with demands to provide better documentation:

> OWS [Open Whisper Systems, the company managing Signal] did not prioritize standardizing [the protocol] both because it gave them flexibility to change it as well as allow it to be more valuable to them as intellectual property. However, they have just finished standardizing a lot of it, […] and I think to some extent that was because of the pressure coming from community like us [Matthew Matrix.org lead developer]

Thus, standardization is understood at the same time as an "obstacle" (as it reduces the ability to quickly modify the protocol if needed), and as an enabler, that creates possibilities for interoperability and the evolution of the whole field of secure messaging, as it fosters development of new tools and improvement of protocols.

One of ChatSecure's developers explains this conflict as a consequence of a specific licensing politics, that lead to tampering and modifications in the legal terms and agreements between the Signal team and other implementers:

> "Signal protocol is open source under the GPL, that means you can't integrate it into a commercial product; that's why OWS were getting large licensing agreements from Facebook and Google and WhatsApp to integrate without opening up all of their source code. Part of that they were incompatibilities with GPL and AppStore specifically. So we needed to get some of the legal language exempted [...] Moxie [Signal lead developer] needs to protect his revenue. Part of his arguments with Wire was that they [Signal] hadn't documented Signal protocol, so there was no open specification, so if you wanted to write a compatible reimplementation, you would have to read the source code which would create a derivative work, which would not allow you to use it commercially because he would argue he still has copyright of the majority of the work". [Chris Ballinger, ChatSecure developer]

The Signal developers are concerned about the technical competence of having third-party developers standardize or deploy forks of their protocol; one of them

remarks that "*Moxie is a very good coder and his standards are very high*", although it is likely that for Signal's main developer, avoiding the forking of the system in ways that could make it less secure is about more than high standards, but also about maintaining the integrity of the system in accomplishing what it is meant to do. Indeed, the Signal team is also concerned about the possibility of not being able to update the protocol rapidly enough in response to research and bugs. This makes it possible to use the non-standardization of the protocol as part of Signal's business model, where the expertise and specification necessary for a proper deployment of the protocol can be offered by the Signal team as a service:

> "You can say OK we will license this technology which is not something I am interested in because I would like it to remain free software. But you can also say 'we are the people who understand this technology, it makes sense to hire us if you want to deploy it.' If people build systems on top of it, then they pay somebody to contribute changes down into that codebase" [Michael Rogers, Briar lead developer]

As we have seen from our user survey and observation of security trainings, open-source and licensing choices are less covered in high-risk trainings, as high-risk users do not always associate open-source with security. Open-source is often perceived as a less important criteria in the context of an immediate physical threat, as when a proprietary but "efficient" and "easy to explain" solution exists, trainers will give priority to it. The primary task in high-risk contexts with low-knowledge users is to help them quickly abandon unencrypted tools as well as tools that collaborate with their adversaries. However, users do care about sources of funding and business models of end-to-end encrypted messaging applications. It was and is the case for Signal, as well; in particular, questions about business models were very frequent on different chats on cybersecurity that we have been observing since September 2016. Users ask for

transparency of funding but at the same time show a certain scepticism regarding crowdfunding models (donations) that seem not sustainable enough for an application to be properly maintained.

Recent critiques addressed to Signal concern their dependency on US government funding:

> Signal was created by the same spooky regime change outfits that fund the Tor Project. The money primarily comes through the federal government's premier Internet Freedom venture capital outfit: Open Technology Fund, which works closely with the State Department's regime change arm and is funded through several layers of Cold War CIA cutouts — including Radio Free Asia and the Broadcasting Board of Governors.[19]

Telegram creator Pavel Durov's critique of Signal goes in the same direction, noticing that no US-government funded application could be trusted.

To summarize, the development and deployment of the Signal protocol show for encrypted messaging – as the history of software development has revealed for other subfields – how licensing choices, business models and politics of open/closed source are complex socio-technical processes, embedded in both community-related interactions, economic context and legal arrangements.

A feedback loop: futures of crypto protocols after Signal

The Signal protocol has deeply influenced the crypto protocol field by introducing a combination of properties, such as forward and future secrecy and non-repudiation, combined with a modern interface, that have become a new minimum

19 https://surveillancevalley.com/blog/government-backed-privacy-tools-are-not-going-to-protect-us-from-president-trump

required for a secure messaging application, without being an actual, formal standard. As the inventor of the OMEMO protocol mentions, “If you are designing a new protocol for end-to-end encryption now, or even two years ago, for instant messaging having forward secrecy in it is just a good practice. It’s just what all the other IM encryption schemes are doing as well. Signal does it, WhatsApp does it”.

In light of the recent history of secure communications and the debates within the developer community, we can identify an effect that can be labeled as “feedback loop” effect: boosted by Signal’s innovations, older protocols have been refurbished, and new standards are now being discussed that aim at bringing some of these properties to a documented and stabilized form.

In the field of email encryption, a new specification was proposed around 2016, called Autocrypt, that aimed at facilitating the key management by putting public keys into email headers. Following the trend of bringing end-to-end encryption to the masses by taking the responsibility for the key management away from users, Autocrypt aims at rejuvenating email encryption by making PGP more accessible for non-technical communities. PGP has recently made a comeback in the field of instant communications, with the rapid growth of “chat over email”; for instance, Delta.Chat, that combines Signal’s with Autocrypt specifications and rPGP (a memory-saving optimized version of PGP).

The OTR protocol has also been transformed and updated under the influence of Signal: in the OTR v3, the multi-device (desktop and mobile) problem has been fixed. And the newer OTR v4, according to its author Ian Goldberg, has features that address specifically shortcomings of Signal, such as deniability[20].

[20] The encryption technique that allows to ‘deny’ the existence of an encrypted file or message, in the sense that an adversary is unable to prove that the associated data exists.

Other standards were designed, such as the OMEMO protocol, that brings asynchrony into OTR and combines some of the properties of OTR and Signal protocol, adapted for use in federated messaging systems based on XMPP. Signal's success has also brought attention of the cryptographic community to a few unsolved problems, such as metadata exposure, and other issues related to the centralized nature of Signal and other popular IM tools. This leads to a certain "revival" of federated systems, with projects such as Matrix and Delta.Chat (and other "chat-over-email" apps), and the rise of mixnets and peer-to-peer solutions such as Briar, as a way to protect users' metadata.

Finally, a recent effort called MLS (for Messaging Layer Security) has been launched to develop a standard offering message confidentiality, integrity and authentication, asynchronicity, forward and future secrecy and scalability. MLS aims at providing possibilities for federation between various encryption protocols for key establishment, authentication, and confidentiality services. As the draft standard mentions, MLS "draws on lessons learned from several message prior message-oriented security protocols [such as] S/MIME, OpenPGP, Off the Record and Double Ratchet[21].

Signal's impact goes beyond "linear" leap from older to modern protocols. The turn to mass encryption has shaken the secure messaging community, instigated protocol renewals and raised new challenges, many of them still being unsolved. It has also inspired new approaches, such as, for instance, the new trend of "chat-over-email" (with Delta.Chat as one of its popular examples), that mainly criticizes Signal for using phone numbers as user identifiers, and suggests to turn to a more privacy-preserving email-based identification.

21 https://datatracker.ietf.org/wg/mls/about/

Conclusions

Addressing the adoption of Signal as a "quasi standard" or de facto standard, this article has showed that, as encryption becomes much more of a public concern than it was a few years ago, several end-to-end encrypted messaging developers are growing skeptical of traditional arenas of exchange and dialogue on potential standards, such as the IETF, XMPP Foundation, W3C or NIST, which they consider less effective (or more 'compromised') than a development-based approach. Does this mean that, as far as widespread adoption of encryption in secure messaging is concerned, we are looking at the "end" of the standardization era? Will governance of encrypted messaging happen by infrastructure and by code, by 'something that works'?

Indeed, the capacity of a tool to appeal to, and be mobilized by, a large pool of users as "something that works" seems to be a core indicator of success and adoption as a de facto standard. The IETF itself has recently acknowledged an "opportunistic turn" in encryption (IETF, 2014) that gained momentum in 2014 after the Snowden revelations, and consists in a progressive move of the crypto community towards making encryption "seamless", with almost no efforts required from users.

In some cases, it is strikingly interesting to observe how the actual cryptographic protocol and security and privacy properties lose their importance for users, compared to the user interface features and he reputation of the app's creator. Thus, motivations for adoption of privacy-enhancing tools are also dependent on the reputation of their creators, as well as more traditional "governance" dynamics such as shifting geopolitical alliances that may affect the reach of government agencies.

Interoperability and de facto standardization processes are one of the several dynamics that speak to Internet governance in a STS sense, as "mundane" and informal activities that come to be invested of a clear socio-political value. Other such dynamics

are the tensions between centralization, federation and decentralization of technical architectures -- and of communities; concentration of leadership, and controversies between prominent albeit informal leaders; and last but not least, the openness of code, which is linked to both geographical differences and to the variety of user threat models, and is a concern of varying importance for different actors.

The analysis of how interfaces and underlying protocols and architectures are created and 'stabilized' helps us to address important questions that speak to Internet governance. We have analyzed here how long-term changes in infrastructure via standardization, whether it happens through more informal means or more traditional avenues such as the IETF, requires inspecting the attitudes of developers towards adoption, their relationship to institutions, as well as their business models (or lack thereof).

Further research may address, for example, how an analysis of the technical design choices made by developers can both provoke new questions in the cryptographic research community, and lead to the revisiting of previous design choices that the secure messaging developer community may have made that are at odds with user expectations. Also, we expect to address how the choice of centralization, decentralization, federation can be qualified as a tentative of "governance by infrastructure" (Musiani et al., 2016) -- an attempt to stabilize a governance model, both for the technology and the communities managing it, through the technology itself.

Funding Details

This work was supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation (H2020-ICT-2015, ICT-10-2015) under grant agreement nº 688722 – NEXTLEAP.

No financial interest or benefit has arisen from the direct applications of this research.

References

Abu-Salma, R., Krol, K., Parkin, S. et al (2017) The Security Blanket of the Chat World: A Usability Evaluation and User Study of Telegram. In *Proceedings of the 2nd European Workshop on Usable Security (EuroUSEC)*, Paris, France.

Arthur, W. B. (1989). "Competing Technologies, Increasing Returns and Lock-in by Historical Events", *The Economic Journal*, 99: 116–131.

Besen, S. M., & Farrell, J. (1994). Choosing how to compete: Strategies and tactics in standardization. *The Journal of Economic Perspectives*, *8*(2), 117-131.

Bowker, G. C. & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*. Cambridge, MA: The MIT Press.

Braman, S. (2016). Instability and internet design. *Internet Policy Review*, 5(3). DOI: 10.14763/2016.3.429

Braman, S. (2009). *Change of state: Information, policy, and power*. Cambridge, MA: The MIT Press.

Busch, L. (2011). *Standards: Recipe for Reality*. Cambridge, MA: The MIT Press.

Bygrave, L. A., & Bing, J. (Eds.). (2009). *Internet governance: Infrastructure and institutions*. Oxford: Oxford University Press.

Callon, M. (1986). "The Sociology of an Actor-Network: The Case of the Electric Vehicle", in Callon, M., Law, J. & Rip, A. (eds.) *Mapping the Dynamics of Science and Technology: Sociology of Science in the Real World*, London: Macmillan Press, 19-34.

Callon, M. (1991). "Techno-economic networks and irreversibility", in Law, J. (ed.) *A Sociology of Monsters: Essays on Power, Technology and Domination*. London and New York: Routledge, pp. 132–161.

Cohn-Gordon, K., Cremers, C. and Garratt, L. (2016). On post-compromise security. In Computer Security Foundations Symposium (CSF), 2016 IEEE 29th, pages 164–178.

David, P. A. (1985). Clio and the Economics of QWERTY. *The American economic review*, *75*(2), 332-337.

DeNardis, L. (2009). *Protocol Politics: The Globalization of Internet Governance*. Cambridge, MA: The MIT Press.

Epstein, D., Katzenbach, C. & Musiani, F. (2016). Doing internet governance: practices, controversies, infrastructures, and institutions. *Internet Policy Review*, 5(3). DOI: 10.14763/2016.3.435

Ermoshina, K., Musiani, F., & Halpin, H. (2016). End-to-end encrypted messaging protocols: An overview. In: Bagnoli, F., et al. (eds.), *Proceedings of the Internet Science Third International Conference*, INSCI 2016, 244–54. Florence, Italy, 12–14 September, Springer. DOI: https://doi.org/10.1007/978-3-319-45982-0_22

Froomkin, D., & McLaughlin, J. (2016). FBI vs. Apple establishes a new phase of the crypto wars. The Intercept, 26 February. https://theintercept.com/2016/02/26/fbivs-apple-post-crypto-wars

Frosch, T., Mainka, C., Bader, C., Bergsma, F., Schwenk, J. and Holz, T. (2016). How secure is TextSecure? In European Symposium on Security and Privacy (EuroS&P), pp. 457–472.

Hellegren, Z. I. (2017). A history of crypto-discourse: encryption as a site of struggles to define internet freedom. *Internet Histories*, *1*(4), 285-311.

Hofmann, J., Katzenbach, C., & Gollatz, K. (2016). Between coordination and regulation: Finding the governance in Internet governance. *New Media & Society*. http://doi.org/10.1177/1461444816639975

Internet Engineering Task Force (2014) Request for Comments 7435, Opportunistic Security: Some Protection Most of the Time, https://tools.ietf.org/html/rfc7435

Jakobsen, J. and Orlandi, C. (2016). On the CCA (in)security of MTProto. In Proceedings of the ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 113–116.

Latour, B., & Woolgar, S. (1986). *Laboratory life: The construction of scientific facts* (2nd edition). Princeton, N.J: Princeton University Press.

Law, J. (1992). Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. Systems Practice, 5(4), 379-393.

Loconto, A. & Busch, L. (2010). "Standards, techno-economic networks, and playing fields: Performing the global market economy", *Review of International Political Economy*, 17(3): 507–536.

Marcus, G. E. (2012). "Multi-sited ethnography: Five or six things I know about it now", in *Multi-sited ethnography*, London: Routledge, pp. 24-40.

Marlinspike, M. (2016). "Reflections: The ecosystem is moving." *Signal Blog*, 10 May 2016. https://signal.org/blog/the-ecosystem-is-moving/

Mendel, P. (2006). "The Making and Expansion of International Management Standards: The Global Diffusion of ISO 9000 Quality Management Certificates", in Drori, G., Meyer, J., and Hwang, H. (eds.) *Globalization and Organization: World Society and Organizational Change*, Oxford and New York: Oxford University Press, 137–166.

Myers West, S. (2018). Cryptographic imaginaries and the networked public. *Internet Policy Review*, 7 (2). DOI: 10.14763/2018.2.792

Muir, S. (2011). "Multisited ethnography", in Southerton, D. (ed.) *Encyclopedia of Consumer Culture*, London: Sage, http://dx.doi.org/10.4135/9781412994248.n375

Murphy, C. and Yates, J. (2009). *The International Organization for Standardization. ISO: Global governance through voluntary consensus*, London and New York: Routledge.

Musiani, F. (2013). *Nains sans géants. Architecture décentralisée et services Internet*. Paris, Presses des Mines.

Musiani, F. (2015). Practice, Plurality, Performativity and Plumbing: Internet Governance Research Meets Science and Technology Studies. *Science, Technology and Human Values*, *40*(2): 272-286.

Musiani, F. and Ermoshina, K. (2017). What is a Good Secure Messaging Tool? The EFF Secure Messaging Scorecard and the Shaping of Digital (Usable) Security. *Westminster Papers in Communication and Culture*, 12 (3), 51-71.

Musiani, F., Cogburn, D. L., DeNardis, L. & Levinson, N. S. (2016, eds.). *The Turn to Infrastructure in Internet Governance*. New York: Palgrave/Macmillan.

Oudshoorn, N. and Pinch, T. (2005}. *How users matter: The co-construction of users and technology*, Cambridge, MA: The MIT Press.

Ponterotto, J. G. (2006). Brief note on the origins, evolution, and meaning of the qualitative research concept thick description. *The Qualitative Report*, *11*(3), 538-549.

Rubinstein, I., & van Hoboken, J. (2014). Privacy and security in the cloud: Some realism about technical solutions to transnational surveillance in the post-

Snowden era. NYU School of Law, Public Law Research Paper No. 14–46. Available at: https://ssrn.com/abstract=2443604.

Star, S. L. & Lampland, M. (2009). "Reckoning with Standards", in Lampland, M. & Star, S. L., *Standards and their stories: How quantifying, classifying, and formalizing practices shape everyday life*, Ithaca, NY: Cornell University Press, 3–24.

Unger, N., et al. (2015). SoK: Secure messaging. In: 2015 IEEE Symposium on Security and Privacy, 232–49. IEEE. DOI: https://doi.org/10.1109/SP.2015.22

Van Eeten, M. J., & Mueller, M. (2013). Where is the governance in Internet governance?. *New Media & Society*, *15*(5), 720-736

Weiser, P. J. (2001). Internet Governance, Standard Setting, and Self-Regulation, *Northern Kentucky Law Review*, *28*(4), 822-846.

West, C. and Zimmerman, D. H. (1987). Doing gender. *Gender & Society*, *1*(2), 125-151.

Whitten, A. and Tygar, J. D. (1999) Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In Usenix Security.