3 • Peer-to-peer encryption and decentralised governance: A not-so-obvious pair

Published by

DeNardis, Laura, et al.
Concealing for Freedom: The Making of Encryption, Secure Messaging and Digital Liberties.
Mattering Press, 2022.
Project MUSE. https://dx.doi.org/10.1353/book.121036.

3

PEER-TO-PEER ENCRYPTION AND DECENTRALISED GOVERNANCE: A NOT-SO-OBVIOUS PAIR

IF THERE IS SUCH A THING AS A CONTINUUM IN SECURE MESSAGING APPLICATIONS, based on their technical architectures, the services examined in Chapter 2 would most likely situate themselves towards one of its extremes, and those addressed in this chapter would be located towards the other. Indeed, particular populations of users of secure messaging systems, especially those living in high-risk environments or involved in political activism, show an interest towards, and sometimes even a hope in, peer-to-peer architectures, as they see a coherence between their political and economic models, based on horizontal connections, mutual help, self-governance and participation, and the technical architecture of distributed networks. Peer-to-peer, as in previous instances in recent history, also promises less control by both governments and private corporations.

However, as the Introduction to this book has hinted at, this larger appreciation of decentralisation as a principle and a vision may itself become problematic; most notably, decentralisation may become an objective in and of itself. Decentralised protocols and applications are too readily assumed, because of their technical qualities, to bring about decentralised political, social and economic outcomes – 'architecture is politics, but should not be understood as a substitute for politics' (Agre 2003).

Peer-to-peer encrypted messaging faces a number of more specifically technical challenges as well. These include a 'vicious circle' between the adoption barrier and dependency on the number of users; the difficulty of managing users'

reputations and identities, as identities are unique but users usually find them hard to memorise due to the form they are presented in; and the mechanisms that lead to trust being invested in the client, which presents a lot of advantages censorship-wise, but may entail risks for users living in authoritarian regimes, where the main threat model remains physical pressure and device seizure.

'Furthermore, while the demand to redecentralise specific components of the Internet has become ubiquitous (Schneider 2019: 266), and despite a long history of tensions which we hinted at in the Introduction, the concept of decentralisation remains uncertainly defined. 'Despite increased research, there remains a great deal of conceptual confusion. Researchers attach a startling diversity of definitions and measures to the decentralisation concept' (Schneider 2003: 32). Practitioners are not on the same page either when it comes to defining what decentralisation means, technically and socially, and which of the many models to opt for when designing a messaging app.

This chapter explores the tensions between the potential and the challenges of decentralised architectures as applied to encrypted messaging, by discussing, in particular, the case of the application Briar. In doing so, it also traces a portrait of the particular populations of users that more frequently adopt these technologies – usually knowledgeable about **mesh networks**, or with a previous history of using decentralised technologies.

THE 'PROMISE' OF PEER-TO-PEER ENCRYPTION

As we have previously discussed in Chapter 1, the discourse linking encryption to peer-to-peer (p2p) is frequently associated with the 'promise' of this decentralised technology for the field of secure messaging. It is cited as such in a number of group chats that we have observed, with a particular focus on Russia[1] and France. These users, whom we have earlier classified as high-knowledge or tech-savvy/tech-enthusiasts,[2] regularly discuss the 're-decentralisation' of the Internet(s) (Rowe 2018). Two main aspects are underlined in these debates: the potential of p2p as a circumvention tool in the context of growing surveillance and censorship, and the particular kinds of metadata protection enabled by the technical features of p2p. Further, the potential of p2p to offer a certain

level of technical autonomy useful in case of shutdowns, or in remote areas,[3] as well as the technological 'elegance' of these solutions, are among the key arguments in its favour.

In countries such as Russia, where Internet governance is increasingly state-centred, centralised and authoritarian after a relatively open and decentralised earlier phase (Nocetti 2015), Internet activists suggest not only federation (which will be more extensively addressed in the next chapter), but also p2p as a possibly appropriate technical answer that can potentially help users to 'slip between the cracks' of state filtering and surveillance. In terms of the kind of metadata treatment enabled by p2p, users believe that decentralised solutions will have less impact on privacy compared to Google or Amazon-based solutions, and that metadata can be better protected within distributed or **mixnet**-based systems. Other discussions on re-decentralisation concern infrastructure, at both the physical and protocol levels, for example, how could the Domain Name System (DNS) be re-decentralised. An important place in discussions on re-decentralisation is held by 'alternatives' such as ZeroTier One, a portable client application that provides connectivity to public or private virtual networks, founded by Adam Ierymenko in 2014.[4]

In France, as well, discussions about the need to move away from proprietary and closed-source centralised services are spreading across tech-enthusiast communities. A new trend is developing, which is labelled as a 'relocalisation' of hosting and service providers. With the motto 'host local', a project called CHATONS[5] has been launched by the 'Degooglise Internet' collective, to map local independent hosting, email and XMPP providers. This collective suggests that instead of hosting data in a wide, centralised, remote and anonymous datacentre, it is more privacy-preserving to host it with someone you know personally. Trust relationships, and sometimes even 'IRL' encounters, give an additional layer of protection in addition to TLS and end-to-end encryption.

Mastodon,[6] a federated[7] version of Twitter, is gaining popularity in France (most of its instances are French). Diaspora, a decentralised social network, is also gathering important communities of French privacy enthusiasts, namely through an instance called Framasphere. As one of our interviewees from the French cryptoparty scene commented:

> I feel like recently there's a riposte of European services to USA-based ones. I don't really understand why we should give our data to giant datacenters somewhere across the ocean. It's like eating our local food … You like French cheese, French strawberries, why not French hosting? Or even better… you can grow your own strawberries [laughs] or run an instance at your place (A., informational security trainer, France).

In this context of the creation of decentralised and federated projects, p2p solutions become part of a more global trend towards re-localisation, associated with a more responsible and even 'sustainable' attitude vis-à-vis the Internet. De-anonymisation of service providers paradoxically promises better anonymity and online privacy, which goes hand in hand with new protocol designs, often based on IRL contacts and key exchange. Several projects are moving towards a redesign of the backbone and propose a more direct and local, sometimes off-the-grid, device-to-device connection, in order to increase anonymity. Among the promises of p2p, data and infrastructure ownership is one of the most frequently discussed. Unlike centralised applications, which make users delegate part of their data (and therefore, part of their 'freedom') to a server, proponents of p2p submit that such a model guarantees more autonomy and privacy for users.

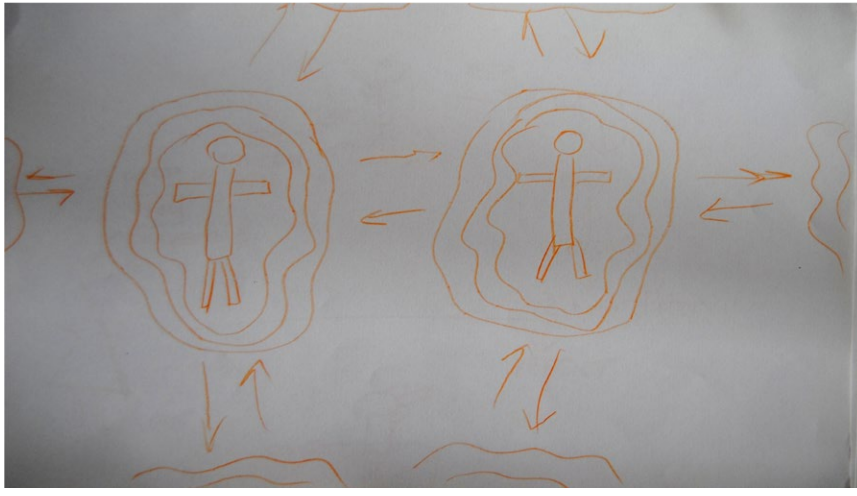

FIG. 3.1 A vision of 'safer Internet'. Drawing made by a feminist activist during a workshop in March 2017 in Saint-Petersburg

The shift towards decentralisation is recursively described as a solution to 'rescue' the Internet, with distributed architectures being considered as possible alternatives to the predominantly centralised, corporate and state-controlled Internet. Interviewees described wishing to 'turn to [the p2p] community to seek digital solutions that defend freedom' and evaluated mesh and peer-to-peer technologies not only as 'more secure' but also as censorship-resistant: 'community-run 'mesh' network … takes back control from corporations: everyone on the network can agree to keep all content open'.

Drawing on a long-term social and historical perspective on decentralised technologies, these statements can appear too technologically deterministic and blind to the big picture. However, what is most interesting about them from our standpoint, in order to account for the recent history of secure messaging development, is that such arguments are grounded in a perspective on decentralised architectures that sees them as intrinsically charged with a specific political and social vision: as Philip Agre has insightfully summarised, p2p is the epitome of a technical architecture that is seen as a 'substitute for politics' (Agre 2003). This chapter seeks to examine how this vision is entangled with the day-to-day technical challenges and opportunities posed by p2p technologies, and how developers and users attempt to embody the promise of such technologies in artefacts and procedures.

Re-localisation, data ownership, and 'utopias of resilience'

A first facet of the p2p promise concerns the intrinsic technical interest that developers have in decentralised technologies, coupled with their recognition that these technologies are accompanied by a number of challenges unseen in other architectural configurations. Indeed, the overwhelming majority of developers of secure messaging and email applications that we have interviewed express a general interest in decentralisation, and almost all, in the span of the same sentence, underline the technical difficulties related to the implementation of p2p protocols. It is perhaps due to the awareness of these complications that no one calls for the total replacement of centralised architectures in favour of p2p, even though much optimism and sociopolitical

promise is invested in distributed architectures. Take the following comment from Sarah, a developer:

> They (centralisation and decentralisation) both have the place in the wild. You'll always have to have collaboration on something centralised, but I think that technology and people are safest when power is distributed, and the way to distributed power is to create decentralised means of communication. My philosophy lies in that. I still see a need for centralisation in few areas, but many of the properties of centralised systems can be created within decentralised systems (Sarah, Ricochet developer).

A second facet of the p2p promise lies in what we call 'utopias of resilience': a number of projects propose solutions for communities in rural areas, or areas at risk (such as war zones), where Internet connections are non-existent, weak or dangerous (for example, if fully controlled by the state). Thus, a Syrian interviewee describes[8] his experience during the first months of the civil war in his country as follows: 'We all became hackers, as we turned to radio waves, walky-talky, mesh and other technologies to coordinate our actions in the absence of network coverage'.

Consequently, countries or areas with frequent Internet shutdowns are also targeted by p2p projects (Vargas-Leon 2016). Among them is Scuttlebutt,[9] invented by a sailor, and a project that proposes an 'off-the-grid' file-sharing, communication and blogging framework. As in Sarah's account above, understandings of power redistribution are once again related to control of data:

> In a database system, all the power is in the database. It's often called a 'single source of truth'. Who can do what is ultimately controlled by whoever administers the database. Here, we have no central database to decide for us what a given action means, instead when you make a post or a 'dig' or change your picture, the other peers (or rather, the software they run) interprets that. A social consensus.[10]

Traffic layer security, or metadata protection, is yet another feature that makes p2p a desirable alternative to models where the metadata collection and retention issue remains unsolved. However, all the developers working on decentralised projects agreed that peer-to-peer presents a number of important challenges usability-wise. They concur that a trade-off exists between the UI/UX features of centralised services that users have got used to, such as stickers, file sharing and calls, and the better level of anonymity offered by p2p-based applications:

> I feel like a whole bunch of people need it [anonymity]. It's really hard to recommend Ricochet over Signal [but w]here Ricochet is gaining it's because it's using Tor hidden services. Part of my goal is to make Ricochet more usable. Compared to Signal, Ricochet is more privacy-preserving because of no phone number. Ricochet is easy but it lacks a lot of features that other IMs have: like sending pictures and files, calls… Ricochet does not have that because this is hard to develop with respect to anonymity (Sarah, Ricochet developer).

Our interviews help shed light on another facet of the p2p promise that may be showing an evolution in the history of decentralised technologies. Traditionally, these architectures have been thought of as best serving the needs of very specific groups of users: on the one hand, activists and strong defenders of civil liberties, members of anti-authoritarian, left-wing movements with a very high-risk profile who seek strong levels of data and metadata protection – so-called 'radical techies' (Milan 2013) – and on the other hand, tech-savvy people interested in 'playing with a new tool', in the more ludic dimension of technology development and testing (Coleman 2011).

However, our interviews with secure messaging users living or working in zones at high risk[11] show that, with the exception of tech workers, these users do not adopt p2p messengers, as they have not been trained to do so. They prefer offline communication on most sensitive issues, or using centralised tools with a deletion timer ('ephemeral messages'). Indeed, digital security trainers who advise Ukrainian or Crimean human rights organisations recommend centralised

apps, such as WhatsApp or Signal, as they are worried that p2p will be more difficult to understand and adopt and will present otherwise avoidable technical issues. After a risk assessment (an analysis of real threats and their probabilities in the given context, see Chapter 1), they often conclude that the threat lies on the client side, and the physical seizure or search of devices at borders is more likely to happen than traffic interception. However, digital security trainers or tech professionals like to 'test' new tools, including p2p tools, with their colleagues and friends.[12]

Based on our sample of user interviewees, heavily politicised users, primarily belonging to left-wing movements, are indeed interested in p2p messaging applications, as in their imaginaries of communication technologies a direct connection is established between social and technical decentralisation (Agre 2003); however, actual levels of adoption are very weak, a recurring problem in the history of decentralised technologies (Rowe 2018; see also Musiani 2015b). This aspect of p2p was criticised by a number of our developer interviewees, who share a belief in some of the positive aspects of distributed architectures but underline a number of unsolved challenges:

> People with […] anti-authoritarian politics bend them [their politics] to a decentralised model and they believe very strongly that all of the technology must follow a decentralised model. And our critique was that … there's a lot of technical problems with decentralised model (Elijah Sparrow, LEAP developer).

BATTERIES AND BUSINESS MODELS: THE CHALLENGES OF P2P ENCRYPTION

Among the problems or challenges of p2p, there is, first of all, a vicious circle of barriers to adoption and a dependency on a critical mass of users (also framed in the past as a 'chicken and egg problem'; see Musiani 2015b and Musiani 2021). Indeed, the more people are using a p2p tool, the better quality of service it can offer; however, it is hard to motivate people to use a relatively unpopular messaging app, due to the poor quality of service in the bootstrapping period. Secondly,

projects that offer better levels of anonymity, such as mixnets (for example, POND), have latency issues. These two challenges are very well described by Roger Dingledine, Tor lead developer, in the interview we conducted with him:

> Part of the challenge was that… should we work on this low latency, low security system called Tor or should we work on this high latency, high security system called mixminion? We have a choice – which one is better for the world? And then we did more economic analysis and we realised mixminion will have approximately no users, so while in theory it must be safe, in practice it will not be more safe. So, the answer [to the initial question] was evident (Roger Dingledine, Tor lead developer).

Another problem of p2p tools is the difficulty of managing users' reputation and identity that is often presented as a 'long hash' (as in Ricochet, which uses Tor 'rendez-vous' points). In this context, identities are unique, but users usually find them hard to memorise. The form that identifiers take in a messaging system is most often the result of a trade-off between different properties, as explained by Elijah Sparrow from LEAP:

> (In p2p environments) user IDs are long strings that are hard to remember… There's something that is called Zooko's triangle. For any identity system you get to pick two of the following three choices: you can have something where the names are globally unique, you can have something where the names are globally memorable, and you can have something where the naming system is decentralised. The problem is that everyone wants to get all three, but you have to pick two (Elijah Sparrow, LEAP developer).

These identity management problems result in reputation management issues. This makes it highly problematic for users of p2p environments to be able to authenticate whom they are communicating with – which developers identify as a core issue of today's Internet due to the numerous problematic and potentially damaging practices it hosts (Badouard 2017):

> Certain usability properties of identities are very difficult to do in a peer-to-peer decentralised model. And a decentralised model also has issues with Sybil attacks, the question of how you control access, how you establish reputation when there is no barrier to entry. There's essentially no good way for a p2p model to have reputation, which is a very big problem in any online communication setting because there is so much trolling (Elijah Sparrow, LEAP developer).

Another challenge of p2p concerns the trust that is placed on the client side. For example, Beaker Browser[13] promises to turn users' browsing experience 'inside out' by hosting websites on users' clients and using a specific protocol for file sharing. The URLs generated with this process are said to be 'unguessable' and are never sent over the network, the URL itself therefore being a public key helping to decrypt files. While this model presents a lot of advantages for efforts to circumvent censorship, as it makes it almost impossible to block or delete any of the Beaker websites or files, it may present risks for users living in authoritarian regimes where the main threat model remains physical pressure and device seizure (see Chapter 1). Secondly, the p2p architecture requires the user's device, by design, to be constantly online (as every device is also a 'server'), which has significant consequences for battery consumption:

> [The] peer to peer promise [says that you] have to trust your device all the time and you have to deal with identities with these long hashes and you have to deal with burning of your battery, memory and mobile device (Elijah Sparrow, LEAP developer).

Improving this aspect is one of the core ongoing tasks for p2p projects, whose developers are being creative in exploring alternatives to this major design constraint, such as whether the client can remain connected to an anonymity network without constantly exchanging data.

A related feature concerns the possibility of planning regular account backups. In distributed applications, it is difficult, by design, to use any kind of cloud platform or other automated or regular backup solution. This feature

can be a positive in high-risk situations (deleting an account from a client deletes it 'forever', as no servers are involved). However, it may be a complication for users who prefer to rely on cloud-based solutions or need to keep archives of their communications. Michael Rogers, the lead developer of Briar,[14] which we will be examining extensively later in the chapter, notes in this regard:

> Briar is in a worse situation than some tools, by the moment your own account is stored on your device. If you destroy the device or uninstall Briar, you lose all your contacts and messages (Michael Rogers, Briar lead developer).

To summarise, in the view of the developers we interviewed who are either working on p2p-based tools or considering whether to do so, adoption of p2p systems in the field of secure messaging seems to be limited because of their insufficient usability levels, restricted multimedia sharing capacities, memory and battery concerns. This makes p2p applications harder to adopt in areas where people use older and less powerful phones with lower quality components, smaller memory and shorter battery life. There is therefore the potential for this architecture to contribute to the digital divide (Howard 2007). Developers underline users' 'dependency' on UI/UX features, such as stickers, and agree fact that peer-to-peer solutions cannot compare to centralised applications in terms of usability.

A related question is why p2p solutions are lagging behind centralised applications, when the search for suitable and sustainable business models has been a long-standing issue for decentralised architectures-based applications (see Musiani and Méadel 2015). An immediate answer is that p2p systems have no central intermediary entity that could track – and monetise – social interactions in order to fund the development of applications. Peer-to-peer architectures traditionally attract sizable attention within academia, with a growing number of conceptually complex and promising projects; however, there remains the problem of the 'knowledge gap':

> [There is an] enormous conceptual gap between what the designers of an encryption tool think that everybody knows and needs to know in order to make a system work, and on the other hand what a user actually tries to achieve through the use of it (Michael Rogers, Briar lead developer).

In this sense, while usability seems to be less of a burden for centralised systems, users have not yet formed proper 'mental models' to embrace distributed secure communication:

> With a certain technical structure that is more centralised, it is definitely achievable […] But now the question is: can we also bring decentralisation into that picture without breaking all of those mental models that users have and without asking them to learn a lot and make a lot of theoretical effort before they can use that tool. […] What we're trying to achieve is a balance between asking a user to understand how the system works, which is obviously a burden, or having a system do surprising things because it works differently from what they expect (Michael Rogers, Briar lead developer).

Despite extended critiques by tech and trainer communities, peer-to-peer encrypted messaging apps are developing and some of them are gaining users, funding and media attention. The second part of this chapter turns to analyse the case of Briar, a peer-to-peer, end-to-end encrypted, instant messaging app using Tor hidden services.

BRIAR: RETHINKING ANONYMISATION AND RESILIENCE

Briar is born out of a problem that is activist and academic at once: how to increase anonymity and move communications off the backbone, in the context of Internet shutdowns (Vargas-Leon 2016) and increased governmental control over the network in a number of countries around the world:

> I was working on p2p communication networks for my PhD and I reached a point where I realized that being able to observe the Internet backbone gives

> you the ability to observe all of the endpoints and their interconnections; this shapes the possibilities for having private communications over the Internet […] if you can see the end points, you cannot get the anonymity (Michael Rogers, Briar lead developer).

In the late 2010s, the lead developer of Briar, Michael Rogers, was contributing to LimeWire, a peer-to-peer file sharing service. In 2009, LimeWire developers were contacted by Iranian journalists from the Green Movement. Activists were wondering if LimeWire would be suitable for use as a communication tool in Iran:

> The guy who contacted us worked for BBC Persian service. He had a principal interest [in] getting news from BBC into Iran, but the question was essentially: what can we do to support a movement like this? One part is getting news from the outside world, another part is disseminating news to the outside world, and the third way is internal communication. And those are all things that we kept going as strands within Briar, how do we look at those different use cases (Michael Rogers, Briar lead developer).

At that time, LimeWire was not suitable and secure for high-risk communication; however, Michael suggested building another system with a greater focus on security features. Together with activists, he sketched the rough idea of a network built over social connections, relying as much as possible on local network connections. This technical solution was relevant to the local political context: international connections in Iran were heavily monitored and filtered. In this context, Michael and his team opted for an off-backbone communication: this collective effort took shape, eventually, as Briar. At the time of our fieldwork (late 2017), the team counted four members, with two developers, a UX/UI designer and a security/usability researcher who was also responsible for communications and outreach for the project.

As with most software development projects, its name sheds light on its history and on the *zeitgeist* of its developers. 'Briar' is an organic metaphor: it

is a distributed, rhizomatic and ramified structure, which, despite its seemingly hostile appearance, can create a protective environment:

> Briar, as far as I understand, means a thorny plant, and there's a fairy tale about a little rabbit thrown into a briar patch who knows how to avoid it, because for him it's not dangerous, he was born and raised there. So it gives this idea for agility and resilience to escape dangers. And I like the little story behind it (Thorsten, Briar developer).
>
> Indeed, it's an American folk story: it's about a fox that catches a rabbit and says: I am going to tear you into pieces. And the rabbit starts crying: Oh, please, do everything you want to me but please don't throw me into the briar patch! So the fox eventually throws the rabbit into the briar patch, the rabbit runs away in the briar laughing: 'I was born and bred in the briar patch, you know?' […] In order to communicate privately we have to move away from these centralised services and rely on our social networks, and we have to fall back on these much more difficult structures to communicate (Michael Rogers, Briar lead developer).

The Briar Patch is also a specific region of space featured in Star Trek. According to the plot of the 1998 film Star Trek Insurrection, the Briar Patch emanates a specific 'metaphasic radiation' that is concentrated in the planet's rings, continually rejuvenating their genetic structure – it is a region of space that star-ships usually avoid because of various radiation sources and energy fluctuations that impair communications systems and make it difficult for vessels inside the nebula to make contact with those outside it. This description bears a close resemblance to Briar's architecture and technical features, these being designed for situations where communication with the 'outside' Internet is hard to maintain. Briar focuses on a specific context of state-driven blocking and filtering measures (Bendrath and Mueller 2011; Mueller, Kuehn and Santoso 2012), as well as extreme situations such as a significant Internet blackout or shutdown (Vargas-Leon 2016). Connections in Briar are made over Bluetooth, Wi-Fi and Tor. In this sense, Briar is designed both as a circumvention and an anti-surveillance tool:

> What we had in mind specifically was how to get information in and out of the country in times of unrest when it might be blockaded, and it might be particularly difficult to reach Facebook [and other international sites]. One of the problems is how do you tunnel information outside or within the country and then let it spread widely outside the narrow tunnel. And that remains a question that people in Briar think of. People need to use it in conjunction with other tools and especially when they need to reach people who are not part of a movement or whatever social group it is, and who are not using Briar. We need to think about bridges. So we have an import feature to import a blog from a web (Michael Rogers, Briar lead developer).

Briar sees its users as people who are aware of their own need for security and mindful of surveillance-related threats. Briar's threat model sees governments as the main threatening group of actors and attackers, performing filtering and interferences as well as blackouts – not only reading and intercepting communications and metadata. Briar is also intended to be a solution for crisis mapping and disaster response, and as such is aiming to collaborate with humanitarian organisations. Briar's UX/UI and usability concerns are informed by the experience of lead developer Michael Rogers, who previously worked as an informational security trainer for journalists and, in his words, had witnessed in this role the conceptual gap between the expectations of encryption tools' designers in terms of users know and need to know in order to make a system work, and actual user expectations concerning the tool (see also Abu-Salma et al. 2017a and Dechand et al. 2019). In this sense, one of Briar's concerns is to make a usable peer-to-peer tool for secure communication.

From the protocol to the application: A framework for a decentralised alternative

The Briar project consists of two parts: the underlying protocol, called Bramble, and the actual user-facing application, called Briar.

> I would say Bramble is a framework, or a library that gives you these peer-to-peer connections with people, without any intermediaries, and it gives you also the notion of groups and of contacts that can interact. And on top of this you can build different applications, and Briar is just one of them, it's a showcase of what the technology can do (Thorsten, Briar developer).

Recently, Briar has been shifting its efforts from the user-facing application to the codebase and infrastructure, and is expecting to guarantee the sustainability of the project without dependency on users – or, to be more precise, in shifting from end-users to 're-users', power-users or lead users (von Hippel 1986) with above-average technical and computing skills who can adapt the protocol to their needs and develop other projects on top of it. The sustainability of Briar is supposed to be guaranteed by separating the protocol from the application, the reason being that, while maintaining pieces of infrastructure is easier in the open-source world as there are several positive precedents (Powell 2012), proper maintenance is more difficult for a user-facing application:

> That's partly why we want to make this separation because the user-facing app will probably have to be maintained with crowdfunding from users, or hopefully it can be maintained on a volunteer basis because most of the difficult technical plumbing will be moved into the infrastructure project, where the users don't have to maintain it (Michael Rogers, Briar lead developer).

The Bramble protocol is not yet standardised, and won't be in the near future, because developers see standardisation as the last step in the chain of releases, after the beta-version of Briar application is properly tested, and the final release is published: 'If you standardise it, you need to know that it's the best way to do it', as Thorsten puts it (see Chapter 2). However, the Bramble protocol can follow Signal's route of '*de facto* standardisation', if it is adopted by a sufficient number of other projects. In this way, as with Signal, while the protocol itself is open source, expertise may be needed to implement it. Providing this expertise as a service is now considered a way of providing the project with a certain degree of financial sustainability, which would make the sociotechnical goal of

promoting resilient and distributed networks easier to achieve. Indeed, Briar's lead developer sees in this initiative of charging for expertise an embryo of a possible business model for Briar, that echoes Signal's 'non-standardisation as a business model' (see Chapter 2 and Ermoshina and Musiani 2019):

> The idea is that people can build other kinds of resilient networks on top of the same protocol stack and hopefully we can make a sort of consulting business for people who need to communicate with devices out in the field or to communicate within teams that deploy in remote areas, that can be interested to use this kind of networking technology (Michael Rogers, Briar lead developer).

Therefore, the Bramble protocol aims to prepare a framework in order to build distributed alternatives to existing centralised services. In the words of Thorsten, 'I would like as many of the services that people currently use to be transformed to this peer-to-peer model when we don't need anybody in the middle anymore'.

Working at the margins: Threats to metadata and Internet shutdowns

Briar's main 'killer feature', as described by its developers, is intended to be the close attention it pays to metadata protection. As developer Thorsten puts it, Briar 'solves interesting problems that are not solved by other tools on the market, for example, it enables people to chat without needing any servers and without leaking any metadata'. An interesting consequence of this, and of the fact that Briar uses Tor hidden services even for feedback submission and crash reports, is that the Briar team itself does not have any precise data about the number of users, or exact usage statistics:

> All the data only exists on the users' devices. There's no Briar server that can store anything. If two people use Briar in a village in Chad in Africa, we don't know about it, there is no connection made between them and any of the computers we control. The only connection ever made to other people, they add themselves. And we will never know these people even use Briar.

> We don't store anything, because it's from person to person (Thorsten, Briar developer).

As mentioned, Briar uses Tor as a 'very well-designed backbone that's designed to know as little as possible on what we do'; however, the team is currently reflecting upon Tor's limitations and security flaws – a concern that mirrors a broader preoccupation in privacy research (Manils et al. 2010). Briar has been conceived to be deployable on any kind of infrastructure; it is not, by design, attached to Tor and could be migrated to a different kind of distributed backbone:

> I think Tor is starting to show its age. Some of the attacks we heard about as theoretical actually went very practical, and we need to think about anonymity infrastructure, privacy infrastructure that is not operated by someone in your house or on your street (Michael Rogers, Briar lead developer).

The Tor vulnerability mentioned by Rogers concerns the exit nodes and is related to the connection point between the onion network and the 'normal' Internet. The traces left by the exit nodes can provoke serious problems for the node administrators – a fact well-known to Dmitry Bogatov, arrested on 10 April 2017 because his exit node was used to post messages judged by a Russian court to be 'extremist' (Hatmaker 2017). The critique of Tor vulnerabilities has led the Briar team to imagine a separate, resilient network, independent from the Internet infrastructure: in the words of Rogers, 'I was looking for something that would work in a sort of partially disconnected environment'.

Briar's particular inspiration comes from the Internet precursor Usenet, when the historical network was running on dial-up connections and supporting early publish-subscribe systems on top of a patchwork of different technologies, before the era of IP addresses (Paloque-Bergès 2017). Some of these ideas had already been developed within the now-dormant Pond project – itself a delay-tolerant, mixnet-inspired messaging system that introduces noise and latency to increase privacy and hide metadata.[15]

One of Briar's central use-cases, which was tested in the field in Brazil with local bottom-up activist communities, is the case of Internet shutdowns. In these

scenarios, Briar is meant to still operate using either Bluetooth or any other network that is not connected to the global Internet. Briar received attention from Brazilian activists because of the recent Internet shutdowns and mobile network jamming used by the police during rallies (Internet Without Borders 2018). Briar is also thinking of deploying it in Cuba, where it is common for networks to be disconnected from the global Internet. By not relying on servers, Briar need not be dependent on the Internet as a backbone, and can potentially be run on any kind of autonomous community network:

> When I tried to send you a message on Signal before it did not work because the Internet was down, and the message needed to first go to the Signal servers and then from the Signal server it came back to you. With Briar we make a direct connection in here, in this network. We are all in this network, we have IP addresses and Briar uses these IP addresses to connect to you. It can also use Bluetooth, or other technologies. For example, we have a mechanism when you can use USB sticks, USB hard drives or SD cards. You plug this in your computer, you say for whom it is, who is the contact and it will synchronize or this contact (Thorsten, Briar developer).

For the Briar developers, their interest in not running exclusively on Tor hidden services (which is the case, for example, for Ricochet[16]), is that then Tor becomes one of several possible ways of transporting the data. So, if in some countries Tor is temporarily blocked (as has happened, for example, in China; Winter and Lindskog 2012) users have alternatives, including some that may be developed in the future.

Group chat: A 'social-based paradigm'

Briar's group chat architecture and key discovery processes draw heavily from the observation of social interactions among social movements and grassroots communities. The Briar protocol is, in some way, a 'modelization of social phenomena such as friendship links, affinity-based community formation, attribution of trust' (Musiani 2010: 193), something that has been a longstanding

concern of many innovators tackling the development of 'next-generation' P2P applications since the mid-2000s. This 'social-based paradigm' (Pouwelse et al. 2006) works towards achieving trust by relying on both the technical features of the protocol and the social aspects of the 'human' community itself.

Briar's identity management and key discovery are linked to the structures of social movements and to offline communication structures. In this sense, Briar seeks to redistribute the trust relationship between human and non-human agents:

> Social networks are the foundation of all-powerful social movements, so by emphasizing it we bring the attention back to the fact that all the security relies on the people that you can trust, by bringing those trust relationships to the fore… This very difficult constraint can turn into a strength. And I feel again that we are in the position of the rabbit, we're thrown in a supposedly hostile environment that actually is the place where we were born and bred (Michael Rogers, Briar lead developer).

Many interactions are happening offline and face-to-face. The key discovery and contact exchange, for instance, is happening **out of band**. Key discovery in Briar happens in two different ways: directly, by QR-code scanning, and indirectly, on the suggestion or invitation of another user. The first configuration postulates the co-presence of the two users in the same physical space; this use context is considered as the most secure and the 'trust level' shown by the application is 'green'. The second case supposes that two users have one contact in common; the trust level is set to 'yellow'. Yellow can later be transformed into green when the two users meet and verify fingerprints by scanning QR-codes out of band. The 'red' trust level designates participants in a group chat with whom no key exchange has been established. However, Briar also tries to minimise users' interaction with keys and make it as smooth as possible:[17]

> All is end-to-end encrypted by the keys that are automatically created when you add your contacts. You need to be face-to-face to add each other. And when you do, you can be sure that no one is in the middle messing up with

> your keys. And you don't see your keys, never. It's encrypted but encryption is invisible to you (Thorsten, Briar developer).

By choosing out-of-band key discovery, Briar tries to solve the **Man-In-The-Middle** problem. However, the QR-code model showed its limits – for very material and physical reasons – in the real-life crash test that we, the authors, performed in August 2017, during a rally on Dvortsovaya square in Saint-Petersburg. A group of 12 activists had installed the Briar application before the rally. We met on the square and had to physically scan QR-codes in order to be able to add each other in the contact list, create a group chat and start the testing. However, the scanning of QR-codes turned out to be hard due to the sunlight. Some users had to hide under their coats in order to scan their codes, and this attracted unnecessary attention from the police and other participants. Other users had their phone screens broken (a very frequent case among left-wing activists), and this has also made the process harder and slower. Moreover, this dependence on offline face-to-face contacts, emphasising the local and the proximity dimensions of the p2p application, made Briar hard to use for coordinating an international movement, or even country-wide one. When we issued a call for Briar user testing in Russia using our contacts in different tech and activist communities across the country, several dozens of people were interested in testing it, but they could not 'add' each other and create a common group chat. Vast distances and decentralised communities with one or two people per city made it harder for the out-of-band system to work.

The latest release of Briar, however, implements a solution to the problem of adding contacts remotely, by making it possible to share a special 'Briar link' (Figure 3.2).

Briar's group chat model is, by far, one of the most interesting across the end-to-end encrypted messengers that we have examined. Briar's group chat structure takes the shape of a star: everyone is connected only via the creator of the group. This offers some degree of metadata protection to the Briar group chat participants. However, once again, the development of this feature was the result of complicated trade-offs:

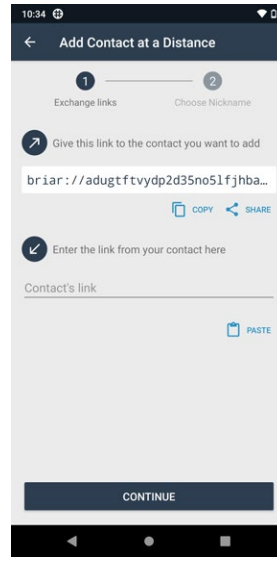

FIG. 3.2 Adding a remote contact on Briar (source: https://briarproject.org/manual).

> Doing the group chat was a surprisingly challenging task. We had to make some compromises there. For example, it was a difficult decision – who can add new people to the group? If you allow everybody to pull new people in it, it's a mess and you have a question who can kick them out again. So to simplify things we decided that only the creator of the group is able to add new people. So if you are in a group and you're not a creator and you want your friend to be part of the group, you would ask the creator please invite my friend – it's a social way of doing it (Thorsten, Briar developer).

Human trust is an important aspect that minimises some of the risks related to 'social centralisation' in Briar's group chat architecture:

> In a way, there is a certain centralisation aspect again [in the group chat model]. The creator has more control on the group than other people have. But you are not forced in the group, you are invited and you can join when you trust the creator sufficiently to handle this group, not invite bad people into it, and secure their phone properly (Thorsten, Briar developer).

It is the fact that Briar puts metadata protection at the heart of the protocol that led them to develop this unique group chat model. However, a protocol feature has been implemented in Briar that makes it possible to redistribute the power and remove part of the technical and social responsibility from the group creator:

> Because there's no server that distributes messages to everybody, participants need to exchange messages with themselves. So, if there is a creator, she needs to have connections with all the members of the group for the messages to travel, as she distributes the messages. But the implemented mechanism makes it easier because what if the creator loses the phone or is offline? The whole communication stops. So we came up with a solution that allows people to reveal their contact relationships to the group: if we are in the group and Ivan is also in the group and you want to be able to keep the communication running when I, the creator, am not here, you can decide to reveal the fact that you are friends with Ivan to the whole group, and exchange messages with Ivan directly (Thorsten, Briar developer).

However, while this feature continues to be effective in Briar's beta for Android, our observations of user testing in Russia, as well as discussions with testers from around the world, have shown that users do not have a clear understanding of the 'contact reveal' feature. Users tend to think that this involves sharing their social graph and consider this function as insecure and dangerous for them and their community. However, Briar's threat model does not intend that anybody should know who the contacts of a particular user are, and this is why the user needs to opt-in to reveal her contacts so that she can communicate without the creator. The Briar team is using real-life use-cases to explain this feature to their testers:

> Everyone is connected to each other through the creator. But when you reveal the contact it does not mean that you receive a notice saying you are friends with Ivan, it just means that you can exchange group messages also directly with Ivan without the creator being connected. It can happen

> that people need to make more connections to each other so that they can exchange messages more fluidly: if you are in a bigger group at a protest and you send 'the police is coming, you need to run away', this message will not arrive to all the members if only the creator is distributing messages (Thorsten, Briar developer).

Enrolling users: Community-building on top of research

A number of theoretical problems have been revealed during the work on the beta version of Briar's group chat, for example '**gossiping**', i.e. making sure that data are disseminated to all members of a group after a user has left it (the nature of peer-to-peer makes it hard to let all users know about it at once, and this may confuse users). The Briar team is looking at both academic work and other projects for solutions but has concluded that existing efforts do not solve these problems. A possible solution may lie in the collaboration between Briar and other anonymity-centred projects, such as Panoramix, Loopix or Pond. At present, Briar does not collaborate closely with any project, but it is following the overall 'galaxy' of encrypted messaging applications, even centralised ones, to keep track of the UI/UX features that users want to have. Briar wants to learn from these popular messaging apps and propose a smoother way for users to 'migrate' from centralised messengers to Briar:

> We usually look at Signal, WhatsApp and Telegram, simply because in this space these are the biggest three apps that fit… With Signal, since the source code is open, we look at it and use some of it, for emoijs for example. If we want to solve some UI problem, we do look at how other projects do it. Because if we have our own way, it may be confusing for people because they are already used to other ways. [We want] for the users to have it easier switching from other apps to our app (Thorsten, Briar developer).

However, Briar has a very different approach to the protocol governance and centralisation/decentralisation debate. Briar is actively distributing its Android

Package, which allows the distribution and installation of their application, via F-Droid; however, they cannot at this stage give users absolute freedom to modify the protocol, for interoperability reasons. Thorsten explicitly states that Briar has a different philosophy to Signal, inasmuch as Briar does not attempt to centralise the protocol and distributions of its apps: 'They [Signal] are very strict in having control over it [the protocol and distributions] while we encourage other implementations'. (Thorsten, Briar developer)

The Briar project is now experiencing a transition phase as the team is choosing which path to take in the near future; during this transition phase, primary concerns include the aforementioned separation of the Bramble protocol from the Briar app, and the search for alternatives to Tor as a backbone.

An important issue to underline is that Briar has become available to users only very recently; until spring 2018, test builds were available for Android devices on request, and the Briar team was organising usability workshops to test different features of the tool.[18] Thus, the 'chicken and egg' problem of getting a critical mass of users interested and motivated has not yet had to be fully confronted. However, the decentralised tool project, though unused by the general public thus far, has been tested in 'field' conditions in remote rural areas, where participants could communicate successfully in the Briar mesh at a limited distance. Furthermore, even though Briar does not yet have an actual user base, it is an interesting example of a project that is driven at the same time by research interests (usable p2p encrypted instant messaging in the context of resilient communications and blackouts) and by activist- and community-based motivations (the team members are frequent participants of Circumvention Tech, now Internet Freedom Festival, and are collaborating with the Guardian project, GNUNet, Unlike Us and Open Internet Tools). This community-building dimension with relevant actors is understood by the Briar developers as a dynamic that will structure developments in the near future, as an important motivation for developers:

> The sense of community is really important to have everybody motivated to work on these projects that are very open-ended, and somehow against the flow that society in general is taking … where there is less and less privacy

> and more and more social control. It's nice to be reminded to know that other people are going in the same direction (Michael Rogers, Briar lead developer).

Ultimately, Briar is a sociotechnical experiment (alongside other projects from the galaxy of p2p encryption tools, such as the MIT-based Vuvuzela) and as such, illustrates important questions about the limitations and problems of p2p-based secure messaging, while at the same time showing its potential.

CONCLUSIONS: THE DIFFICULT DAY-TO-DAY PRACTICE OF THE 'PROMISE OF INTERNET EQUALITY'

In the galaxy of end-to-end encrypted messaging tools, decentralised ones appear to be subject to the highest degree of experimentation. If we consider this in a historical perspective, we see that the nexus between p2p and secure messaging today is an important manifestation of a longstanding and complex tension related to decentralised technologies – between their alluring promise of interoperability, horizontality, mutual help, self-governance, participation, reduced control by governments and the private sector, and, on the other hand, the multiple technical and economic challenges standing in the way of its widespread implementation, including the 'chicken and egg' problem of user motivation vs technical dependency on the number of users, and the difficulty of how to manage users' reputation and identity.

As the 'promise of Internet equality' (Agre 2003) of p2p technologies remains strong, particularly in specific activist and academic settings, seeking solutions to the different challenges posed by decentralised architectures to encrypted messaging is at the heart of a substantial portion of current privacy and anonymity research. However, there seems to be a gap between academic research fields and activist needs and questions:

> It's one of the greatest unsolved mysteries [...] The computer science problems that activists care about are not necessarily close to the computer science problems that are prestigious to work on in computer science. For

> me, as an activist working on usable communication this is a great unsolved problem (Elijah Sparrow, LEAP).

Some projects, such as LEAP, Delta Chat (see Chapter 4) and Briar are trying to work in between the two; the NEXTLEAP project, via the work of the authors and other members of the consortium, has operated in the same direction, with our focus on activist use-cases (both high- and low-risk) and collaboration with open-source developer communities (Autocrypt).

We have selected the Briar case among decentralised secure messaging projects as we believe it demonstrates well the new potential of peer-to-peer encrypted messaging applications, as well as the challenges presented by p2p. However, in a broader context where net neutrality is put under increasing strain, and Internet censorship around the world is growing and becoming more pervasive, it is important to acknowledge that some of Briar's technical and social solutions can be reused, and possibly improved upon, by other projects. Alongside Briar, several projects sharing its foundational interest in decentralised architectures show a trend towards reinventing the Internet backbone itself, and migrating to other networks – seeking a freer, more decentralised Internet that would be less controlled by both governments and private corporations. New projects develop and define themselves as real 'ecosystems' suitable for any kind of data exchange, such as Matrix.org (see Chapter 4), CJDNS,[19] i2p[20] and Yggrasil,[21] decentralised and encrypted network protocols that have a growing user base in countries like Russia, as a response to the country's current trend towards more centralised control over the Internet.

As the secure messaging field grapples with the issue of delegating too much trust to the creators of centralised IMs (an illustration of this has been the Pavel Durov vs Moxie Marlinspike case, described in Chapter 2) or to the infrastructures they manage, p2p messaging applications seek to somehow *re-distribute* the trust between humans and protocols. However, our research on this type of secure messaging systems has shown that many users are still very much bound to centralised architectures, except for specific technical communities or openly anti-authoritarian activists.

NOTES

1 E.g., Telegram chats of Pirate Party Russia, Cybersecurity chat, internal Rublacklist chat

2 And questioned/challenged this label in Chapter 1.

3 See projects such as Secure Scuttlebutt: https://scuttlebutt.nz.

4 https://zerotier.com.

5 https://chatons.org.

6 https://joinmastodon.org.

7 See Chapter 4.

8 Discussion at the Citizen Lab Summer Institute, session on armed conflicts and information control, July 2017.

9 See above: https://www.scuttlebutt.nz, a 'decentr(alised) secure gossip platform'.

10 https://github.com/ssbc/handbook.scuttlebutt.nz/blob/master/principles/legacy.md.

11 E.g., in the frame of a 'corollary' project of NEXTLEAP, in January 2018 we interviewed 28 Ukrainian and Crimean journalists, tech workers and activists.

12 See, e.g., the 2017 wave of interest in Tox messenger among the tech community in Kyiv, Ukraine.

13 https://beakerbrowser.com.

14 https://briarproject.org.

15 https://youbroketheinternet.org/secure-email#pond.

16 https://ricochet.im.

17 We will discuss in the final chapter how this and other similar strategies fall into a tendency which we define as the 'opportunistic turn' in encryption.

18 The authors participated in a usability workshop for Briar at University College London in February 2017. Eleven people took part in the workshop, all of them being UCL PhD or postdoctoral students in computer science or usability. We tested several functionalities, such as **key exchange**, invitations for a one-to-one chat, group chat creation, blacklisting and changing the 'trust level' of contacts.

19 https://github.com/cjdelisle/cjdns.

20 https://geti2p.net/en.

21 https://yggdrasil-network.github.io.