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

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

DELIVERABLE D2.1

INTERNET SCIENCE VOCABULARY AND KEY QUESTIONS

Carmela Troncoso (IMDEA), George Danezis (UCL), Marios Issakides (UCL), Harry Halpin (INRIA)

Beneficiaries:	**UCL** (lead), IMDEA, INRIA, CNRS
Workpackage:	**D2.1 Internet Science Vocabulary and Key Questions**
Description:	A list of key questions facing decentralized systems and initial agreed-upon common interdisciplinary vocabulary terms based on Internet Science

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

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

Contents

1 Introduction

2 Decentralization: The long road from 2001 to 2016

Decentralization is typically viewed as empowering to users and a cruicial component of one possible future of the Internet,[1] yet there does not exist a foundational treatment of decentralization or even a shared interdisciplinary definition of decentralization. Currently, due to interest in blockchains and resisting mass surveillance, there is a wave of interest in decentralization as exemplified by projects such as IPFS[2] and Ethereum.[3] By defining decentralization, schematizing the ways a system can be decentralized, and walking through the key design decisions in decentralized systems, the lessons of research can inform a whole new generation of collective awareness platforms and a new kind of future internet.

This is not the first time there has been a surge of interest in decentralization. As Cory Doctorow noted at the 2016 Decentralized Web Summit[4] "It's like being back at the O'Reilly P2P conference in 1999." Fifteen years ago O'Reilly published "Peer-to-Peer: Harnessing the Power of Disruptive Technologies" [177], which signalled a peak of interest around decentralized architectures. The 'hype' around decentralization was followed in the early 2000s by a mass of research and deployment activity around such systems, and the three main streams of this wave are still prominent on the Internet today: BitTorrent, Tor and - appearing much later - Bitcoin itself. Yet, whether these technologies will remain niche or will eventually become the dominant computing paradigm of the future is still very much an open question.

To a large extent, decentralized systems were a response to a particular threat: Censorship. Perhaps the first rallying cry for decentralization was the Eternity Service [9] by Ross Anderson: "I had been alarmed by the Scientologists' success at closing down the penet remailer in Finland; the modern era only started once the printing press enabled seditious thoughts to be spread too widely to ban. [...] So I invented the Eternity Service as a means of putting electronic documents beyond the censor's grasp" [9].

This anti-censorship resistance motivation is most clear in Tor's use of a decentralized network of anonymous relays as well as its naming infrastructure for hidden services based on a DHT[5]. Last but not least, Bitcoin [168] emerged as a censorship-resistant way to transfer funds to organizations like Wikileaks[6], having emerged at the end of 2008 from the ashes of the centralized e-Gold online currency, which had been shut down by the Department of Justice earlier that same year.

Similarly, Napster closed in 2001 after the RIAA challenged them to keep track of file copying, enabled by the Napster's centralized index. On the contrary, BitTorrent[7] succeeded as a peer-to-peer file sharing service due to not having a central indexing service like Napster. Furthermore, not only does BitTorrent not rely on a central index, but it also implements a tracker-less form as a distributed search and coordination service (using Kademlia). Both features make BitTorrent extremely resistant to attempts of closure by authorities.

Yet, despite the millennial fervour for decentralization, the mid-2000s are exemplified by the rise of the data centre and massively distributed – but *not* decentralized – systems becoming the dominant technical paradigm. The "Cloud" was associated with the popular and commercial rise of large internet service providers such as Google, Facebook, Yahoo and later Microsoft. Attempts to use open standards to federate and decentralize these platforms through efforts to create a "distributed social graph"[8] ultimately failed (with the possible exception of OAuth) or even led to these centralized services evolving into centralized platforms. Peer-to-peer and other

[1] For detailed scenario-planning the results of the "Internet Futures" EC study http://www.internetfutures.eu/wp-content/uploads/sites/48/2010/11/TAFI-Final-Report.pdf as well as the final document by the FP7 Paradiso project http://paradiso-fp7.eu/files/2012/02/PARADISO_refdoc_final.pdf

[2] https://ipfs.org

[3] https://www.ethereum.org/

[4] http://www.decentralizedweb.net/

[5] https://www.torproject.org/

[6] https://www.wikileaks.org/

[7] https://www.bittorrent.org/

[8] http://bradfitz.com/social-graph-problem/

decentralized paradigms did not displace more centralized models of interaction, but instead survived in niche spaces concerned with circumvention and so wary of the practical benefits of cloud computing.

One major event led to programmers being interested in reversing this trend: Snowden revealed that mass surveillance programs, operated by the US and UK, heavily relied on the technically centralized nature of communication and storage to get unprecedented access to information. These sort of attacks gave increased credence to long-standing privacy concerns brought about by the rise and popularity of centralized social networks such as Facebook and Twitter. Today, these concerns are widened into fears over how these platforms centralize political, economic, and cultural could abuse their quasi-monopolistic position.

The desire to preserve privacy, liberty, and the autonomous control of infrastructure and services have led to a call to decentralize – or "re-decentralize" – the Internet. A number of activities, such as "you broke the internet" have gathered attention and a wide range of developers are now proposing a number of alternatives to centralized infrastructures and services. In parallel, Bitcoin and the "blockchain" has received enormous attention and investment due to its promise to lower the barriers to innovating in the fin tech sector through a radical decentralized architecture.

It is of course important to not be either nostalgic about past work or fatalistic about future efforts: Today's networking and computing environments are vastly different from those in 2000: smart-phones have placed a powerful computer in many people's pockets; people are relatively well connected to the Internet; there is near infinite capacity in the backbone; clients, such as web browsers, are now mature end-used platforms with peer-to-peer communications enabled, e.g., using WebRTC [26]; and mobile code, in the form of Javascript, is a daily reality. The design space for modern decentralized systems is less restricted than it was in the past.

However, a number of fundamental challenges have not entirely disappeared: These are related to how people trust not only each other but their devices and the code that runs on them; the need for energy efficiency; the expectation of device independence and mobility; the fear of loss or compromise of devices and the continuing insecurity of platforms. Thus, today's advocates of rabid decentralization would do well to remember that the previous wave of enthusiasm, in the 2000s, did not replace centralized architectures. It is only by understanding the problems those systems faced, and devising appropriate solutions, that future decentralized systems could gain significant popularity.

Our key objective is to support future work on decentralized privacy systems by carefully reviewing and systematizing the evidence from the past 15 years of research – roughly the period between the publication of "Peer-to-peer" and mid-2016. In particular, the relative loss of public interest in peer-to-peer and other decentralized architectures in the mid to late 2000s has created a discontinuity that makes some of the earlier findings difficult to access or interpret in today's context. Thus, we take it upon ourselves to translate key findings and classic designs, but also the important problems faced by designers of past systems so as to inform the choices made by engineers today.

3 Epistemology, Methodology and Goal

3.1 Scope

Although there is a vast and wide use of the term "decentralized," we will restrict ourselves in this paper to discussing systems that claim to support *privacy* properties using *decentralized* architectures. We draw a distinction between *decentralized* and *distributed* architectures.

A distributed system makes use of multiple components that have their behavior co-ordinated via message passing without the use of a central clock [131]. These components are usually spatially separated and communicate using a network. Distribution is beneficial to support robustness against single component failure, scalability beyond what a single component could handle, high-availability and low-latency under distributed loads and ecological diversity to prevent systemic failures. However, all these benefits can be achieved with a

distributed system that is managed by a single root of trust or authority. Furthermore, decentralized systems are typically created for reasons of efficiency and scaling, not to enable privacy. So in many designs of distributed systems, the underlying distributed system is fundamentally trusted. Developments led by Google, ranging from BigTable to MapReduce, are distributed systems par excellence [68].

In contrast, decentralized systems are a subset of distributed systems in which multiple authorities control different components and no single authority is fully trusted by all others. In some decentralized systems, components may choose their own relationships of trust between each other autonomously but in other decentralized systems a component may also trust no other components. This has profound implications in terms of security and privacy: There is no single entity that can act as a reference monitor to enforce a global security or privacy policy; components in the systems need to consider adversarial behaviour not simply by external parties, but also by genuine components of the system controlled by different authorities. Thus, systems such as Bitcoin, BitTorrent and Tor are decentralized. Decentralized systems may not strictly be peer-to-peer insofar as not every component may communicate with every other component (See Section 4.1).

In many decentralized systems, the architecture is also *open* so that new authorities can join and leave the system, possibly at any time. This vastly differs from traditional distributed systems, where the number of components was assumed to be known and communicating over authenticated channels. While these kinds of closed distributed systems there is already a large amount known both around pratical engineering and formally proven results [226], very little is known about open systems and may not always be communicating over authenticated channels. In fact, successful decentralized software programs such as BitTorrent and Bitcoin seem to succeeded in their design more due to intuition than science, and what little formal results we have from "permission-less" decentralized systems are typically impossibility results [17]. Indeed, in decentralized open systems even standard consensus algorithms[132] will fail if not a majority of the participants are honest (i.e. non-adversarial). Not all decentralized systems are strictly open. For example, some decentralized systems function more like "clubs" that may allow new members in based on their meeting certain conditions or even after a possibly social process like voting.

To illustrate a distributed system that is not decentralized, consider the Telex design [254]. In order to provide censorship resistance, friendly Internet Service Providers deploy middle-boxes that recognize tags in encrypted flows from users behind national firewalls and redirect them transparently to the censored resource. However, while Telex relies on distribution to prevent censorship and scale, it is not decentralized since a single authority is envisaged to distribute tags to clients and manage the entire system (Multiple extbfllel Telex systems could exist but would not interact amongst themselves). In contrast, the Eternity Service design makes many copies of a censored resource across a distributed network of servers and ensures that the index of any given component server does not point to every single copy of the file. As no single server has a complete index and Byzantine fault tolerance can be used to reconstruct a file across multiple servers even if a large number are compromised, the Eternity Service is both a distributed and decentralized system.

Following the classical work of Baran [18], decentralized systems are conceived of as networks of interconnected components. Due to this, we will call the various components of a decentralized system *nodes*, which is roughly synonymous with the term "peer" although the term 'node' incorporates systems that may not be strictly peer-to-peer. Note that a system of nodes may have multiple sub-systems that are functionally independent networks with their own kinds of relationships between nodes.

3.2 Methods & Model

Due to the large amount of often unfounded claims by advocates of decentralized systems in terms of privacy, academic review of a system can be considered proof that some privacy claim actually may hold. To support this SoK we performed a systematic literature review: we scanned manually all papers published in the top-4 computer security conferences (IEEE S&P, ACM CCS, Usenix SEC, NDSS) as well as the specialized PETS, WPES and IEEE P2P conferences from the years 2000 to 2015 and compiled a list of works within scope. Specifically we considered within scope all works proposing or analyzing decentralized systems relating to privacy, including

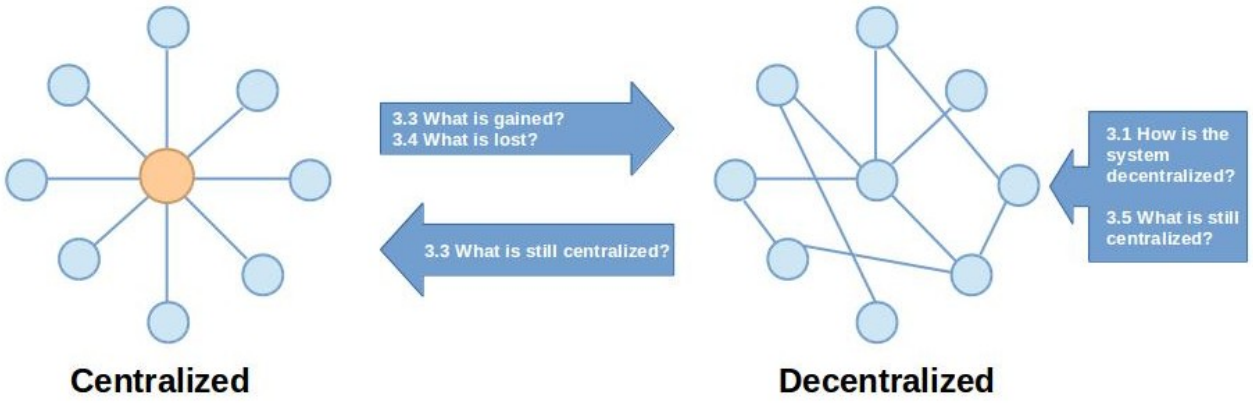

Figure 1: From Centralized to Decentralized Systems

those where privacy is supported by providing high-integrity services – such as key management – and claim to have high-availability. Given there are many (both closed and open-source) architectures that also claim to offer privacy and security properties that do not have academic review, these are only mentioned if they exemplify particular design choices.

For each work, system or issue discussed we attempt to answer one or more of the following questions:

1. How is the system decentralized?
2. How does decentralization support privacy?
3. What is gained from decentralizing?
4. What is lost when decentralizing?
5. What implicit centralized assumptions remain?

For all questions we have compiled a substantive body of sample systems that are characteristic of different issues and use them to illustrate different aspects of it. We also provide a quick review of other important systems and their relation to each question. Figure 1 illustrates our approach.

4 Decentralization and Privacy: Where are we?

This section runs over the key questions described in the previous subsection providing evidence of the current state of affairs in decentralized systems.

4.1 What type of decentralization?

A diversity of decentralized designs have tried to balance multiple factors such as integrity, availability and privacy in order to better serve the needs of their users. A number of key architectural decisions need to be made, including how to form the infrastructure of the network, how to distribute trust between nodes to establish authority and how to structure the routing of messages between nodes.

4.1.1 Infrastructure

When creating a decentralized system, one early architectural choice concerns the distribution of tasks across the distributed hardware needed for maintaining the service. The provisioning of the infrastructure naturally has ramifications for the entire design of the system in terms of trust and message routing. Historically, most

decentralized systems have had a federated model where a trusted authority is delegated by its users to serve as a node in a decentralized system on their behalf. Open federated models of infrastructure have been the most popular form of decentralization as shown by classic Internet protocols such as SMTP [183]. The popularity of federation may be due to the fact that large “always-on” computers are needed for communication, and were so especially in the early days of the Internet. In a strict “peer-to-peer” form of decentralization each node may serve as its own autonomous entity, making decisions about who to trust and how to communicate, and this model is exemplified historically by Bittorrent and Gnutella [99], which started off as a gossip based P2P system before introducing super-nodes to facilitate routing.

Users form the infrastructure In an effort to completely eliminate the need for a centralized authority, some decentralized systems called ”peer-to-peer systems” rely solely on users to contribute part of their resources (bandwidth, storage) in order to keep the service up and running. In Freenet [58] and Cachet [175] users are responsible for hosting encrypted data, with the latter supporting policy-governed write operations.

Early implementations of structured peer-to-peer systems, inspired by Chord’s [219] use of a distributed hash table for routing, are a key example of users providing the infrastructure. MainlineDHT (MLDHT) [243] powers peer discovery in Bittorrent without out-of-band communication with central trackers. One typical problem with any peer-to-peer system is the lack of availability of peers, as peers may join or leave the network at any time, leading to “churn” in the network. In turn, this may lead to spare topologies and poor link performance. However, lookups in DHT can reveal the structure of a network and harm privacy [159]. One modern design for a peer-to-peer system is ShadowWalker [160], a low-latency P2P anonymous communications system that lacks any central servers; users route their own and their neighbor’s requests according to routing tables that they maintain locally and which are certified by “shadows” that verify correctness. ShadowWalker claims to cope with a moderate amount of churn and a sparse topology while preserving privacy.

The ability for peers to join a P2P system opens it up for Sybil attacks. Although solving Sybil attacks in general is still an open research problem, privacy-preserving P2P systems including X-Vine [161] and Drac [62] exploit social links to resist such attacks, with Drac also using cover traffic to provide anonymity against global passive adversaries.

Infrastructure is independent from users Alternatively, a set of trusted or untrusted entities may provide all or part of the system functionality to the rest of the nodes, i.e. act as infrastructure. Key motivations include an increased availability of the service, potentially a reduced attack surface, immunity to churn or the efficiency and confidence in managing fewer critical nodes. While the architecture is decentralized, there is a clear se extbftion between who provides the infrastructure to run the service (servers) and the actual users (clients). This design pattern underlies classic open federated protocols such as SMTP [183] and XMPP [10].

Unfortunately, historically these systems offer very little in the way of privacy from potentially malicious servers. One long-standing project still under development that minimizes trust of the infrastructure provider is the Tahoe-LAFS (Tahoe Least-Authority File Store) [224] for encrypted file storage. However, research into mix networks [50] and Private Information Retrieval [54] can be used to defend users from their own servers and so allow untrusted servers, although typically at the cost of scalability and performance as shown by systems like Dissent [251]. Riposte [59] is an example system that tries to combine elements of Dining Cryptographer networks (DC-nets) [51], Private Information Retrieval (PIR) [54] and secure multi-party computation protocols, like Dissent [251], to provide anonymous messaging that is resistant to traffic analysis attacks and denial-of-service attacks and claims to scale to millions of users. The core service is run by a number of untrusted servers and there is no direct client-to-client communication except via the servers, although the system is dependent on non-colluding audit servers.

Hybrid Systems The infrastructure of many decentralized systems falls between these models. In Tor, any user can use the infrastructure via a client, but another open group of users forms the infrastructure by contributing their resources to running relays, while another closed trusted group forms the directory authorities that the rest of Tor depends on [75]. A similar pattern is seen in Bitcoin [168], where users may use the Bitcoin infrastructure without running miners, but another group of users devote their computational resources to mining in order to provide the infrastructure of the blockchain.

In terms of privacy and security, these hybrid systems naturally have a diverse set of strengths and weaknesses, but new elements such as a distributed ledger could help decentralize traditionally centralized cryptographic protocols. For example, computation could be done locally and simply recorded to the blockchain with the support of secure multi-party computation protocols [266]. More generally two-party multi-party computation protocols assume direct communication between the computing parties, and so do protocols for secure multi-party computation (SMPC). However, real-world implementations of SMPC in practice distribute computations across a small number of stable entities, to ensure reliability and low-latency, as in the case of the Sharemind system [33]. The model behind blockchains can possibly be extended to handle SMPC that obviates the need for a trusted third party[11].

4.1.2 How is authority decentralized through the nodes?

A further key architectural consideration is the relation of nodes to each other in terms of performing the necessary tasks to deliver a service. While nodes may have their resources contributed by either the users themselves or third-parties, not all nodes are created equal and nodes may or may not have to render services to other nodes.

Peer-to-Peer: Nodes communicate directly In this design the nodes can communicate directly with each other as peers in order to complete an ad-hoc operation related to the parties involved only, ideally without an authority or even the participation of other nodes. Thus, nodes have no responsibility to carry traffic for other nodes and do not keep a consistent state once the connection drops. Of course, these kinds of systems are quite rare in practice, with exceptions for local area networks where both nodes can find each other with a simple gossip protocol such as the samba protocol. In practice, these kinds of decentralized networks can be utilized for real-time exchange of information (video calls, gaming). However, even if a direct peer-to-peer connection is used in the network, such as a WebRTC connection, usually there is a 'signalling' process to a third-party identity server in order to locate other peers [26].

For purposes of security and privacy, direct communication has a number of advantages: the channel can easily be made confidential between the two nodes and since network traffic is not sent via any other nodes, there is a high degree of privacy. If one ignores the problems of locating other peers and routing traffic efficiently, this direct model of peer-to-peer communication is the typical model in 'Alice' and 'Bob' idealizations in cryptography and security - a model that typically does not require a trusted authority. Direct peer-to-peer communication in group settings then requires that peers either re-send messages to each participant in the group or that they rely on group key agreement protocols for creating an ephemeral secret key shared among the users, and ideally participants are guaranteed the freshness of the key, since they have contributed to its generation [199].

Peer-to-Peer: Nodes must assist other nodes Most peer-to-peer systems organize as a network where all nodes are responsible for carrying out operations for all other nodes, rather than due to any pre-configured position of authority. Bittorrent swarms and Bitcoin miners follow this model. What contrasts this kind of peer-to-peer architecture from the preceding direct peer-to-peer communications is that nodes perform services for each other such as routing messages or storing blocks, usually according to their capacity. Thus, incentives in peer-to-peer systems must be aligned even for honest nodes to expend resources to benefit others, as nodes that act purely selfishly could disrupt the decentralized operations of the network even if they are not acting maliciously. Furthermore, these peer-to-peer systems are usually open with no admissions control, so the number of nodes in the network is dynamic and thus there usually must also be an incentive to join the network. Nodes also may have difficulty discovering each other without a centralized naming service, and so nodes often use a name that can directly identify their network address or a hash that can lead to a way for the service to authenticate itself, such as the hash of a public key in Tor Hidden services [30].

There are clear advantages in security and privacy in peer-to-peer networks, namely that information about the peers is not centralized in any node and messages do not even have to leave relaying sub-nets. Yet relying on peers for functionality poses a threat to anonymity in peer-to-peer systems using the classic *Sybil* attack, since user requests may be served by nodes controlled by an adversary [77]. Such nodes may simply passively

collect information on other nodes, and attempt to violate their privacy properties, or they may actively disrupt the operations of the network. Counter-measures to protect routing algorithms have been proposed, based on some trust criteria [119] and system-wide resource reputation mechanisms [61] that aim to marginalize adversarial nodes and content, but there is not yet a general solution to the problem of Sybil attacks.

Peer-to-Peer: Nodes assist only friends An alternative decentralized design is that nodes rely on a small number of other trusted nodes for operations. This design can also be peer-to-peer insofar as each node may choose its own set of nodes to co-operate with by virtue of trust, i.e. each node communicates via "friends," based on some social links. This maintains some advantages of a peer-to-peer system based on social networks but is less vulnerable to Sybil attacks as these nodes would be excluded from participating in the network or would be easier to detect [65]. Since trusted nodes must be reliably identified, a naming system directly based on key material is often used to authenticate their status as "friends."

A number of systems have been proposed that implement social-based communication to resist Sybil attacks. For example, Drac uses social-based routing of messages and hides metadata using cover traffic [62]. Mittal et al. proposed X-Vine [161], a protection mechanism that can be applied to existing peer-to-peer systems, backed by distributed hash tables, that is resilient to denial of service via Sybil attacks and requires only logarithmic state and control overhead at the cost of higher latency. Although there are few decentralized networks in practice that use social networks themselves to control the communication of messages between nodes, Freenet [58] version 0.7.5[9] lets users establish direct friend-to-friend connections with other freenet users they trust (Note the original FreeNet had no provision for exploiting social relationships in order to strengthen its DHT against sybil attacks). However, without cover traffic a passive adversary can monitor the network communications, de-anonymize users, and so easily discover their social graph.

Federated. In decentralized systems with a federated design, one set of nodes - the providers - are relied on as authorities to perform some operation by the rest of the nodes, who are typically users. Thus, users are directly connected with one or more providers, and providers are peers but users are not peers and can only communicate via providers. The providers then assist each other in order to deliver messages destined to users associated with other providers and the providers share resources to complete the operation. Each provider is responsible only for its own users, while users act solely as clients to their selected provider. In practice, these systems have been incredibly popular and characterize well-known standardized protocols such as asynchronous message delivery over email [183] and synchronous chat over XMPP [10]. Naming in federated systems is usually provider-based, which can lead to more 'human-readable' names based on the name of the provider, such as *user@domain.org*, although resolving domain names to network addresses is clearly centralized.

At first, federated systems may be thought more centralized and so to necessarily lead to weaker privacy and security as users are exposed to their provider and possibly other providers as trusted third-parties. However, relying on a provider does not necessarily result in weak privacy and security, as end-to-end encryption can help confidentiality [39], mix networking-based solutions may hide information from servers [63], and computation can be outsourced using secret sharing [194]. As shown by Alhadid et al. [6], when it comes to integrating horizontally partitioned data over the same set of attributes, providers can achieve ε-differential privacy to protect the privacy of their users from other providers. Despite these improvements, the primary weakness of federated systems is that users expect federated service providers to act honestly.

Accountability. In decentralized networks, monitoring the correctness of the operations of other nodes allows the network to minimize its dependency on the authority invested in other nodes. This particularly applies to the authority invested in providers in federated systems, where any user will need a transparent log of a provider's operations in order to ensure their honesty. The task is usually performed by a new type of peer, the auditors, or other providers when acting in lieu of their associated users.

It is clear that while the nature of the authority invested in an auditor is very different from traditional peer relationships, the security of peer-to-peer protocols such as Bitcoin critically relies on decentralized and independent

[9] https://freenetproject.org/assets/papers/freenet-0.7.5-paper.pdf

authorities [168]. Indeed, the new wave of interest in decentralization around blockchains is precisely due to the ability of a decentralized network to maintain a consensus over an ordered set of transactions. Nonetheless, there are difficulties in maintaining privacy in any distributed log, although the use of zero-knowledge proofs by Zerocash [24] shows that maintaining unlinkability is possible in auditing relationships.

In practice, Bitcoin maintains a consensus over the blockchain due to mining, but the general approach of third-party auditing has also proven useful in federated systems. Certificate Transparency [133] protocols rely on a decentralized set of services and auditors to keep track of issued and seen X.509 certificates and quickly detect potentially rogue or hacked certificate authorities to prevent targeted or mass interception. Similarly modern electronic election protocols [102] achieve robustness through proofs of correct shuffling of votes, using robust mix-nets such as Verificatum,[10] verified by auditors. CONIKS [156] is a key verification service in which clients can audit the consistency of a binding through time. The aim of the system is to make easily detectable the equivocation of a user's public key by her associated provider, collectively by the end users and by other providers, while concealing the identities and the number of users of the provider.

4.1.3 Network Routing and Topology

Decentralized networks may be may be implemented on top of a diversity of network topologies for routing messages, ranging from directories [75] to just using impromptu nearby nodes [168]. The structure of network routing may or may not mirror how authority is itself decentralized through the network, although it often does: Networks based on trusting only friends often use friends to route messages, but an otherwise decentralized network of anonymizing relays in Tor formed by users may need the help of a smaller set of pre-established directory authorities to construct the circuit to route messages [75], and even a system of federated messaging providers could in theory be reachable over a mesh network. However, the network topology is not independent from the security and privacy properties provided: The study of Diaz et al. [72] of low-latency anonymity networks demonstrates that network topologies have an important influence on the level of anonymity provided.

Mesh In mesh network topologies a peer can send a message to every other peer, where messages are relayed typically by flooding all of the network with their messages, although more sophisticated routing may be used. Flooding consumes bandwidth exponentially according to the number of peers and may overload the network, as well as offering no guarantees on message delivery. Mesh networks are designed for scenarios when a stable connection with other nodes is not an option, such as in mobile ad-hoc networking. Early versions of Gnutella also use a flooding-style broadcast in order to share files.

Historically mesh networking based protocols like Bittorrent do not preserve the privacy of their users [135] and popular mesh networking applications used in "internet-shutdown" scenarios like FireChat were not even encrypted.[11] However, there has been work on secure messaging systems such as the Briar project,[12] who choose this structure when they need to be functional during Internet blackouts. Mesh networks are unreliable and usually messages do not reach the destined recipient, but require no infrastructure and can be resistant to takedowns and DoS attacks, as well as to metadata and content surveillance.

Gossip Many decentralized systems rely on the broadcast propagation of messages from a node to the rest of the network. This is typically done via a gossip protocol, where a random subset of the nodes in the network are chosen to receive the messages, as opposed to flooding where all other available nodes receive the messages. With gossiping, these nodes then continue to broadcast the message via selecting independently another random subset of the network to relay messages. More efficient than flooding, gossip protocols have shown themselves to popular and are implemented in decentralized systems such as Bitcoin miners, Gnutella and Certificate Transparency.

In practice, the reliability of message delivery is questionable and information propagation usually experiences delays. These drawbacks could affect the overall system operation, as has been observed with Bitcoin, as the

[10] http://www.verificatum.org/

[11] https://citizenlab.org/2014/07/asia-chats-update-line-kakaotalk-firechat-china

[12] https://briarproject.org

slow information propagation, amplified by the delay of block verification at each node before its transmission, leads to transactions not being included in the blockchain and is the main reason behind blockchain forks [70].

Distributed Hash Tables Distributed Hash Tables (DHTs) allow more efficient network routing than gossiping or flooding without central coordination, as each node needs to know only routing to its neighbors and so does not need to maintain a global routing table. For any message sent to a node that is identified by a key, the destination is either known by the node given by the DHT or can be passed via a link to a node 'closer' to the intended address. Although a wide variety of DHTs have been devised, in general they allow a hash table to be distributed between nodes in a network, with DHTs having been shown to be scalable and fault tolerant of network churn.

DHTs do not by themselves grant much in the way of security, privacy and anonymity properties. Tran et al. [230] proved that low latency anonymity systems such as Salsa [169] are vulnerable to having large amounts of traffic captured by adversaries controlling a fraction of the relays. In order to improve DHTs, often nodes are grouped into quora that use some form of majority voting to help defeat adversaries that control a minority of the nodes [258].

Content-centric Networking Content-centric Networking in which the content can be located directly rather than via a network address could be well-suited for decentralization, although obviously aligning the location of data on a network with the content of the information to be retrieved presents a large security risk. Other proposals focus on anonymous routing, such as ANDaNA [73], which imitates onion routing aiming for communication privacy in content-centric networking systems, although all such proposals are currently immature.

Super-nodes One open question in peer-to-peer systems is whether or not nodes that offer more resources to the system and can maintain connections with other nodes eventually transform over time into 'super-nodes' that essentially play the role of 'providers' in federated systems. Most peer-to-peer systems such as Bittorrent eventually developed super-nodes [69], and the amount of traffic sent through Tor relays is far from uniform [120].

These super-nodes are then targets for adversaries that wish to control or even observe network traffic to de-anonymize users. One attempt to prevent this is to use, as mentioned earlier, a trusted social network to route traffic, as is done in Tribler [184]. Tribler hopes to improve file-sharing by noting that friends have similar tastes in media, and use this similarity in order to improve performance, content discovery, and downloading. Likewise social-based trust relations have been used to attempt to address the security concerns with DHTs. Nasir et al. [172] designed a socially-aware DHT to be used in decentralized online social networks, which reduce latency and improve the reliability of the communication. However, since social networks then become 'scale-free', network-routing based on social trust may then just inherit the super-nodes implicit in the social graph, and routing via trusted social contacts can be used to reveal the social graph of users and de-anonymizes participants.

Hierarchical. Unlike DHT-backed networks, in hierarchical networks nodes are more tightly bound to an assigned role. The vast majority of traditional network routing is done in an hierarchical manner, including spanning tree protocols such as in BGP[190] in the current Internet as well as 'next generation' designs like SCION [263]. So realistically, it is often easier to build a decentralized system on top of a hierarchical network routing level, as is typically done by federated systems. For example, the federated system of number assignment in IP addresses allows for easier, if not entirely decentralized, routing.

From a security and privacy perspective, the networking hierarchy itself then becomes a key point for attack, even if the rest of the system is decentralized. For example, BGP assumes a trusted relationship between BGP peers when it is updating its routing tables in order to pick the most efficient routing path. As traditionally BGP is unauthenticated, an attacker can easily impersonate a BGP peer [190]. By introducing hierarchical public key infrastructure for BGP in the form of RPKI, these attacks can be ameliorated [138].

Stratified Many decentralized systems use a *stratified* design where different parts of the network have well-defined roles that they are assigned in terms of routing (and in terms of authority), and co-operate to maintain the operation through a diversity of routing mechanisms. As illustrated in Figure 2, Tor clients can autonomously choose circuits through the Tor network by choosing relays from an open-ended number Tor relays, yet a global

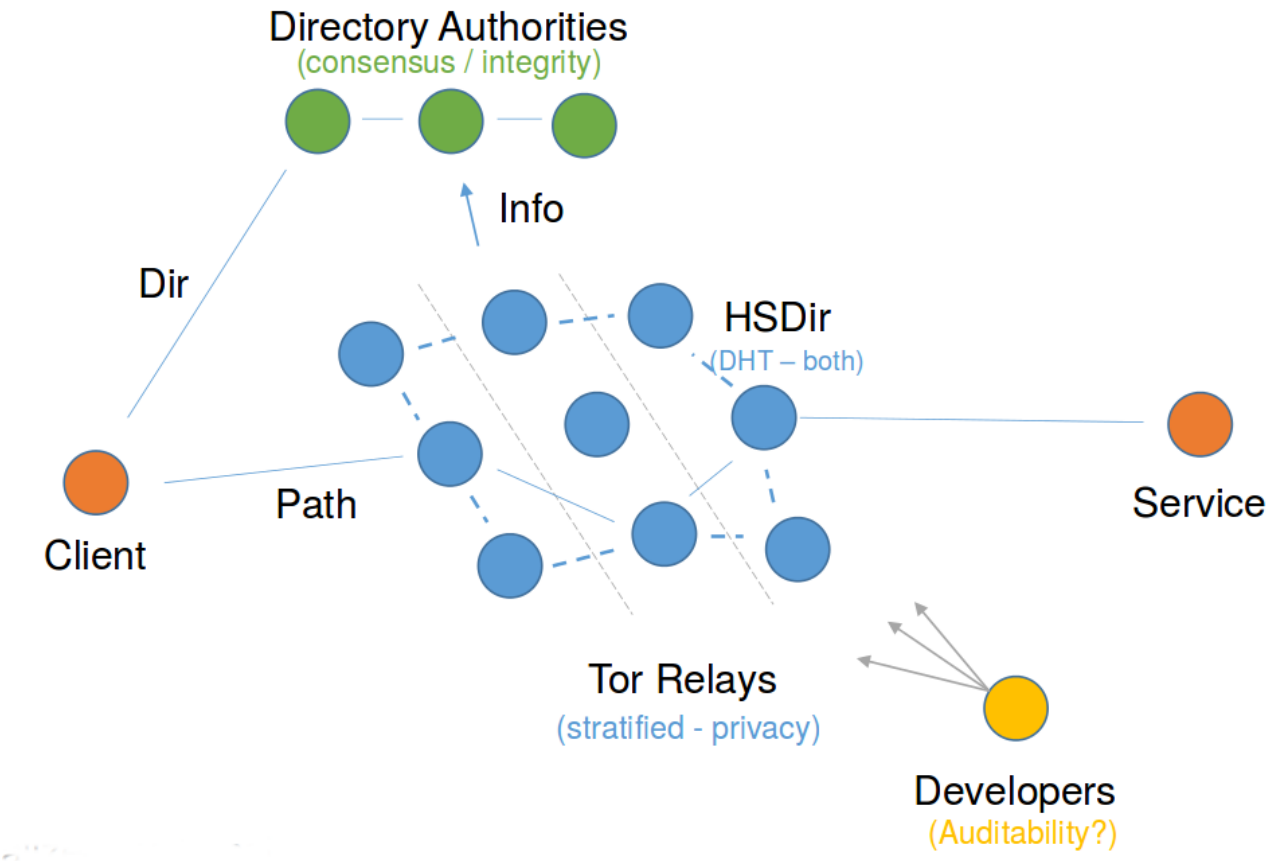

Figure 2: Multiple Roles and Authorities in Tor

list of the relays is maintained in a view with integrity of the entire Tor network created through consensus by a closed group of Directory Authorities. In contrast, Tor hidden services, which can only be reached via circuits in the Tor network, are reachable through a Hidden Service Directory maintained by a DHT without the knowledge of the directory authorities. Lastly, one hidden point of centralization in Tor is that the developers centrally control the software that each node runs, although they may be audited via their open source code for integrity, a perhaps hidden and "centralized" authority in Tor as explored in 4.5.

4.2 What do we gain from decentralization?

There are a number of perceived intrinsic architectural advantages of decentralized architectures. Although intertwined, each of these perceived advantages need to be discussed in turn: security, availability and integrity.

4.2.1 Flexible Trust Models

An intrinsic advantage of decentralized architectures relates to the existence of multiple authorities, which in some contexts could be assumed to not be working jointly with the adversary, or at least making independent decisions that an adversary cannot predict. The general point is to create a distributed trusted computing base that ensures that a subset going rogue does not compromise the security properties of the system, so that no single compromised component can revoke the security properties of the entire system.

Distributed Trust Decentralized systems leverage the existence of multiple independent authorities, and can transform this into a security assumption: for example, all forms of threshold cryptography guarantee that if some fraction of participants faithfully follow the protocol and do not leak information to the adversaries, that

some security property can be guaranteed. This principle can also be applied to distributed key generation, public randomness and threshold-based decryption, and signing.

One such system is Vanish [98] that illustrates how a multi-authority system can be used to implement properties that would otherwise be impossible, or implausible, to securely implement by a single entity, in particular guaranteed deletion after a pre-set expiry date. The Vanish system distributes secret shared [204] digital content across a distributed hash table (DHT), along with an expiry time, and each peer promises to delete the share after this time. As long as a critical number of peers are honest, the material will not be available for retrieval after that date. Other such systems include traditional anonymous channels such as mix-systems [51] or Tor [75] that depend on a set of non-collaborating relays to guarantee meta-data privacy. However, such non-collusion assumptions need to be studied critically, as the DHT instantiation of Vanish was shown to be vulnerable to a Sybil attack [250]. Reliance on multiple authorities to regain a degree of privacy has also been proposed for commercial clouds: a multi-cloud storage [215] system may be used in that context to ensure privacy in case some providers are dishonest.

No natural single authority In some systems there is intrinsically no central authority and thus a decentralized architecture is a natural choice. This setting has been traditionally studied in the contexts of decentralized access control, as in TAOS [249] and SDSI [86], and 'trust management', such as Keynote [32]. In such systems a set of distributed principals make claims about users and each other, and those claims may be assembled and used to resolve access control decisions using a specific logic. Bauer et al. [20] show that the task of resolving access control decisions is faster in a decentralized setting, as compared to a centralized authority performing a similar task.

Leveraging existing distributed trust networks In some cases a decentralized infrastructure embeds or expresses a pre-existing set of trust relationships that a system may reuse to support security properties. Kusters et al. [130] offer a theoretical foundation for accountability and verifiability in general, and in particular applied to contract signing, voting and auctions.

For example, one anonymity system leveraging friendship relations in a decentralized system is Drac [62]. Drac builds an overlay anonymity system over a peer-to-peer network, by connecting users to their immediate close friends, and creating paths to further away contacts through paths in this network. As a result it provides anonymity and some degree of unobservability due to cover traffic. The security argument in Drac crucially depends on the underlying social structure, and the lower likelihood of users being betrayed by their 'friends'. This is an instance where a decentralized topology was chosen to match this underlying social trust network.

More direct example systems use the underlying social trust structure to build overlay privacy-friendly social network services, as surveyed by Paul et al. [179]. As an example, the Frientegrity system [90] provides a social network platform using untrusted providers seeing only encrypted data, where users can exchange information with 'friends' protected by cryptographic access control. This use of encryption to defend against the providers themselves is not the case for systems like Diaspora [28], an open-source project that takes a different approach: users connect to a provider they trust – that gains full visibility of their activity – and attempts to reclaim some privacy and user control purely through the decentralized architecture.

4.2.2 Allowing the network to be easily deployed with high availability

The central premise of decentralized peer-to-peer networks is that having users or institutions contribute their own resources without the use of a central authority reduces costs and may even help ease deployment. Costs are lowered as a decentralized architecture may be able to access and use spare capacity in the existing infrastructure, as exemplified by underutilized resources given by users such as the early SETI@home project and the use of users' storage in Freenet.

In a privacy context, Torsk [155] is a peer-to-peer onion routing design, backwards compatible with the popular Tor, that uses peers as directory servers to distribute and reduce the load associated with accessing directory

authorities and keying information. While PIR could be used, one key element in this case of decentralizing fully the design was to use spare resources, and facilitate deployment given the existing Tor nodes.

Decentralized architectures by virtue of usually also being distributed, and potentially exhibiting fewer correlated failures, can be leveraged to provide high-availability, sometimes in conjunction with privacy properties. As an example the Cachet privacy-friendly Online Social Network system [175] uses a pool of untrusted peers as a storage back end of a decentralized social network. Using appropriate information dispersion codes, the system helps to ensure that items will be available despite node churn – while appropriate cryptographic controls also guarantee integrity and confidentiality. Vanish [98] uses a similar design to ensure files disappear.

4.2.3 Availability when resisting formidable adversaries

Censorship Resistance The original motivation for many decentralized systems was censorship resistance [9]. Decentralization can also find resources that would be difficult to centralize, such as the network location diversity needed for bridges to bypass censorship both on the network and legal levels. Proximax [154], for example, manages and distributes volunteer IP addresses to potentially censored hosts to use as proxies. A number of designs take advantage of this, like Publius [242], in order to resist powerful adversaries trying to censor information stored in the network.

Survivability Decentralized architectures have the potential to survive attempts to take them down or inflict massive damage, in a way that centralized systems cannot resist 'decapitation' attacks [255]. This property has been used to build highly robust botnets using a peer-to-peer architecture [195]. Although these systems are decentralized on the technical level, there is of course a need to maintain central but covert command and control (C&C). Those botnets have indeed been harder to take down using conventional techniques, but are also vulnerable to new threats that result from their decentralization, such as poisoning and enumeration of nodes. A further discussion of such topological issues in the wider 'Darket' is provided by Zhou et al. [143].

Separate Development from Operations In high risk threat models, the developers of systems may reasonably be concerned about being coerced to either reveal private data on users of the system or to deliberately introduce privacy weaknesses into their systems. In this context decentralized architectures may mitigate this risk, by clearly separating authorities that provide public code – and that have no access to operational data and secrets – and those that run the code to serve users, each of which can audit any open source code and has partial visibility into user data.

A key architecture embodying this benefit is the Tor [75] project and the onion routing network using their code, as the code and the underlying network is run by independent operators. The core development team updates the code, that is publicly visible and auditable, but that is then run by independent relay operators. As a result, attempts to coerce the Tor development team can only have an indirect and possibly highly visible effect – rendering such attempts less effective. On the other hand a vast decentralized architecture guarantees that coercing a critical mass of operators, to reliably trace users, is extremely expensive and requires a multi-jurisdiction attack. Similar designs are put forward in other peer-to-peer anonymity systems like Tarzan [93].

4.2.4 Public verifiability of integrity

Due to the availability of multiple independent authorities, decentralized systems may be used to implement security properties through 'auditability' mechanisms that allow these multiple authorities to have their integrity publicly verified. In such settings adversaries are disincentivised by ensuring that attacks will leave an observable effect to be used against the adversary later on or that such cheating will be caught before it has the desired negative effect.

Transparency can be used to help enable privacy as ensuring that actions are transparent enables users to know what happened with their data. Similarly, Pulls et al. [185] use decentralization to support transparent audits of personal data accesses. Auditability is also a key feature of secure electronic election systems such as the

Helios system [1]. Such systems rely on the existence of multiple authorities in a number of ways: threshold cryptography is used for parameter and ballot generation, anonymization and threshold decryption. However, a number of properties can also be provided through auditability: every party contributing to Helios generates a cryptographic proof that it operated correctly, i.e. in a manner that will not affect the correct election result. It is crucial that other parties in the system audit those proofs and raise an alarm in case of deviation. It is their collective proving of correctness and auditing of correctness, and the assumption that some will be honest, that guarantees the integrity of the results.

4.3 How is Privacy supported?

Decentralization works when there is no trusted third-party. If a user does trust a single third-party provider, having data at a single third-party with privacy-enhancing measures (such as adequate cover traffic) could be considered more resistant to traffic analysis by a global passive adversary than spreading communication throughout a peer-to-peer network without privacy-enhancing measures.

The primary advantage of decentralized systems from the perspective of security and privacy is that they remove central trust. Centralized systems can be transformed into decentralized systems both in order to eliminate the single point of failure in terms of availability and to reduce the risk to this trusted party of being coerced to harm privacy. One key problem with privacy-preserving systems is that many of them require a trusted third party to ensure service integrity (see Section 4.5.3).

There are two general methods to decentralize a system. In the first, a trusted third party is substituted with a decentralized protocol. An example is the decentralized anonymous credential scheme proposed by Garman et al. [96] in which the authors build on e-cash public ledgers to enable the decentralization of the Credential Issuer authority, replacing a “central bank” with joint oblivious functionality. Another example of trusted third party decentralization is Adeona [192], where storage of user locations is decentralized to gain robustness, reduce the cost and also increase privacy.

The second method is to use a trusted third-party but check its integrity by public forcing of transparency. In Certificate Transparency, all observed certs (from central certification authorities) are logged and checked for conflicts, with multiple logs communicating via a gossip protocol [133].

In this section we survey the privacy properties that can be obtained through decentralization. We use the widely accepted properties defined by Pfitzmann and Hansen [180] to guide our study. Not all architectures provide these benefits, as we will see in Section 4.4.

4.3.1 Confidentiality

We distinguish two ways of obtaining content confidentiality through decentralization. The first provides users with data privacy towards a system or an observer, the second provides data privacy also with respect to other peers participating in a decentralized operation such as a computation.

Confidentiality from third-parties. Some designs employ a decentralized architecture since such an architecture may not feature a centralized attack surface that needs to be surveilled to gather intelligence about online user activities, as data is stored in peers and not a centralized datacenter. In this first category the most characteristic systems are those that exploit threshold encryption [204] in order to trade off information confidentiality and information availability, such as the PASIS [256] architecture. This scheme splits the data in n “shares” and distributes it among peers in such a way that recovering m shares allows one to recover the data, but having less pieces provides no information. Similar solutions are provided by POTSHARDS [220] or Plutus [124].

Confidentiality from peers. Another scenario is that in which nodes need to perform a joint computation but they do not trust each other nor a third party with their data. In this case, decentralization enables them to exchange encrypted data and obtain the sought after result without relying on any particular entity to preserve

their privacy. The P4P framework [79] is such a system, in which further zero-knowledge proofs are integrated to protect computations against malicious users.

Covertness. Other systems defend even the existence of the participation of nodes in the decentralized network from outside observers. The Membership Concealing Overlay Network (MCON) [235] leverages this to provide strong forms of covertness. All peers in MCON only have links with trusted friends, and a complex routable overlay network is jointly created that allows all peers to communicate indirectly with all others. Yet, any peer only connects to other locally trusted peers, resisting attempts to enumerate all users by malicious peers.

4.3.2 Anonymity

One of the main advantages of decentralized systems is that they can provide support for the decoupling of actions from the identity of the users that perform them in order to provide anonymity.

Pseudonymity. Before diving into the analysis of systems that aim at completely hiding the identity of the originators of actions, it is worth mentioning that some designs enable pseudonymity. Even though this is a weaker concept than anonymity [180], e.g. it allows linkability of actions which may have a negative impact on privacy (see Section 4.3.3 below), in some scenarios it may have advantages, whether to enable functionality (e.g. detecting returning users) or reduce the complexity of the system. An outstanding example is Bitcoin [168], where every transaction is linked to a pseudonym and stored in the blockchain. This allows to trace the money and avoids double-spending, but on the downside if a pseudonym is ever linked to an identity (e.g. [29]), all actions from the person behind the pseudonym would be de-anonymized.

Anonymous communications. Probably the most textbfdigmatic intention for using decentralization is to protect communication patterns (browsing habits, communicating patterns, etc.) from network-level adversaries. As mentioned earlier, the Tor network [75] has messages relayed through several decentralized nodes before being delivered to their destination. In Tor decentralization and diversity enable users to avoid a global adversary who can observe both sides of a communication and thus breach their privacy. Many other systems, both deployed [116] and in the literature [93, 169, 155, 161], leverage this approach to provide anonymous communication.

Another decentralized building block for implementing anonymous communications is group signatures [52], where all members of a group obtain a signing key that they can use without the need of coordination, in such a way that given a signed message it is not possible to identify the group member that has issued the signature. For instance in AMOEBA [196], a privacy-preserving vehicular communication system, nearby vehicles agree on a group signature key that they use to sign messages while being anonymous within the group of nearby vehicles. This approach has also been leveraged to allow private multi-party messaging in which it is not possible to say who sent which message without relying on a trusted party [101].

4.3.3 Unlinkability of user operations

. Due to the distributed nature of resources in decentralized networks, no entity can observe all actions happening in the network nor track all activities from a user. This supports privacy in several dimensions as follows:

User Freedom. This property enables a subject to safely and believably deny having originated an action, so as to shield her from any responsibility associated to performing such action. The fact that actions cannot be linked back to a user, equips users with freedom to perform actions without fear of retaliation. For instance, in Freenet [58] requests can not be linked to their originator, thus users can freely search for information without revealing their preferences. Similar freedom is provided by censorship-resistant systems [74, 241] that, in order to avoid blockage and enable the use of online resources without fear of punishment, preserve a user's privacy by hiding which users have performed a given action online.

Anonymous Publishing. Plausible deniability is crucial in facilitating anonymous and censorship-resistant publishing. This was the motivation behind the original Eternity service [9] and well-known designs such as Publius [242]. Another example is Tangler [241], a scheme that achieves censorship resistance, that publishes content in a decentralized manner by replicating it in several blocks "entangling" different information. This replication method not only hampers deletion, but also provides anonymity to publishers (entangled replication hinders the identification of the content originator) and readers (entangled replication hinders the identification of the content being accessed since several files are downloaded every time). Likewise, modern underground market places increasingly use decentralized Tor Hidden Services [75] for anonymous publishing. Anonymous notifications can also be published: Lincoln et al. [144] propose to use intermediate repositories to "mix" security alerts in such a way that collaborative analysis of information security threats can be done but alerts cannot be linked back to contributors.

Unobservability. An important privacy property that can be achieved through the decentralization of communications and information storage is that of unobservability, i.e. the hiding of the very existence of actions. For instance Drac [62] not only achieves anonymity in terms of who speaks with whom by relaying traffic in a decentralized manner, but as nodes always participate in the building of communication tunnels and relaying of packets, an adversary cannot distinguish whether they are communicating with a friend or not. The Dissent [251] anonymity system also provides some degree of unobservability through the use of Dining Cryptographers networks. Another way this privacy property can be implemented is to hide the social graph in social networking applications like Xbook[208] and presence information in chat applications as done by PIR in DP5 [34]

Analytics prevention. A first advantage is that no entity can monetize an analysis of the data that could be extracted by such a centralized tracker, since this authority does not exist. Such a property is one of the primary motivations behind decentralized online social networks as surveyed by Paul et al. [179]. By hampering analytics on the level of the individual user, decentralization reinforces the provision of anonymity, with decentralized and privacy-enhanced analytics still an active area in need of more research.

Private retrieval. Decentralization can also help breaking the link between a query and its response, for instance in information theoretic private information retrieval [100]. This property is leveraged by PIR-Tor [162] to obtain Tor nodes in a private manner without having to download the full directory service. Other systems use this approach to implement private presence services [34] or to enable private location-based services [176].

4.4 What do we lose because of decentralization?

Sadly, there is no free lunch in decentralization. As mentioned in the two previous sections decentralizing brings many advantages, but there is no guarantee that the properties and features of centralized systems are maintained in the process. This section summarizes problems stemming from decentralization which should be taken into account when moving away from centralization. A further critique of decentralized systems, focusing on personal data, is provided by Narayanan [171].

Increased attack surface A first problem is that decentralizing systems across different nodes inherently augments the number of points (attack vectors) that an adversary could use to launch an attack or use to observe the the traffic of users.

Internal adversaries As opposed to centralized systems, where system components can be monitored and evaluated by a trusted entity and therefore a malicious insider can be more easily detected, in a decentralized system it is easier to insert a node with adversarial intentions. For instance, the predecessor attack [252, 253] uncovers communication partners in many anonymous communication schemes [51, 75, 189, 222], while the Sybil attack on decentralized systems allows an adversary to insert many nodes in the system, which could be used to bias the reputation scores in her favour [78] or corrupt the information exchanged inside collaborative decentralized intrusion detection systems [117]. Also, when messages are relayed through other nodes in order to gain anonymity, the content of these messages is exposed to more adversaries, e.g. Crowds anonymous web transactions [189] or distributed search engines. Even if messages are encrypted, other techniques can be

used to reconstruct their content, as in the case of VoIP conversations [247]. Open federated systems also fall for internal adversaries, as exemplified by the widespread use of spam in email.

Traffic analysis Second, decentralization inherently implies that information will be sent around through the network. So even in the presence of encryption, metadata is available to external adversaries with access to the network (e.g. ISPs, Law enforcement agencies) that can be used to compromise the privacy and security of a system. A classic example of this is anonymous communications networks, where it has been repeatedly shown that both passive local [159] or (partially) global [121, 166], as well as active adversaries [246], can break anonymity by looking at traffic patterns. Therefore, resistance to traffic analysis is one of the largest problems facing all decentralized networks.

Attacks on inconsistent views The fact that decentralization typically implies that nodes have a partial view of the network can impact systems security, as such views may be non-consistent and so lack integrity. These non-consistent views allow adversaries to "cheat" without being detected. For instance, in the Bitcoin decentralized peer to peer network adversaries can force non-consistency through fast operations [126] or eclipse attacks [109] in which the adversary gains control over all connections of a target node thus isolating her from the rest of the network, and exploit this lack of consistency to perform double spending. Similarly, a non-consistent view of the relays in anonymity systems opens the door to epistemic de-anonymization attacks [66] as well as in the Bitcoin network [29].

Inconsistency in Privacy Besides routing problems, the lack of a global view brings problems in terms of privacy. Since users act in a decentralized manner they do not necessarily make the best choices with respect to optimizing their privacy and instead may only optimize based on their relationships to other local nodes but leave themselves open to adversaries that observe the network. This issue has been studied both in the context of anonymous communications [74] and location privacy [94].

Denial of Service Although a single system may fall victim to denial of service attacks, mitigations against denial of service (DoS attack) are typically focused on a single network resource rather than an open-ended decentralized network. So another consequence of decentralization is that it eases the deployment of denial of service attacks to disable the system partially as groups of nodes, rather than a central system, may be attacked. These attacks can be launched by internal adversaries, as in the case of Daswani et al.'s attacks [67] on Gnutella [99]; as well as by external entities that can send messages to the system nodes [57, 118]. Besides being harmful in itself, DoS has also been shown to facilitate other attacks, in particular de-anonymization [29, 118, 35].

4.4.1 Cumbersome management

An obvious problem of decentralization is that there is no entity who has a global vision of the system, nor does there exist a central authority that can either act as controller to make optimal decisions, or provide nodes with information that allows them to make these decisions on their own. This makes the availability of a decentralized network often more difficult to maintain.

Routing difficulties A straightforward consequence of this lack of centralized control is increased complication in routing data since it becomes difficult to get an overview of the network and its capabilities [221] and therefore to optimize routing decisions [261]. This is even harder given some of the intrinsic properties of distributed networks such as highly diverse nodes [91], the existence of churn [12] and the fact that discovering the network relies on possibly malicious nodes [245]. Solutions to these problems include using complex routing algorithms to enable secure and private discovery of nodes [155, 163, 160], decentralized virtual coordinate systems that open a new door to insider attacks [261], or resorting to a centralized directory (see Section 4.5.1). An additional challenge for these routing algorithms is to prevent adversaries from learning the network topology through traffic analysis, since it has been shown to be of great help for de-anonymization [170].

Performance Loss The lack of centralized routing information also impacts performance since no entity has a global view of the network and thus choosing optimal routes and ensuring load balancing is difficult. We can

find two approaches to alleviate this problem in the literature: those that use local estimations to improve performance [7, 8, 227], or those that aim at helping users to make better decisions about routing individually [209], although the latter also opens the door to new attacks [111, 165].

Difficult attack prevention Decentralization not only opens the door to new attacks, as mentioned before, but the lack of centralization also hampers the establishment of effective protection mechanisms. For instance, it has been shown that the lack of a consistent view of the network hampers the use of collaborative approaches to detect incorrect information, calling for more complex solutions [125]. Another prominent example of this problem is sybil attacks. Since there is no entity that can observe all nodes and have complete information about their structure and behaviour, it becomes extremely difficult to decide which nodes are the sybils. Thus, defenses must leverage local information, e.g. defenses based on social networks [65, 259], or implement collaborative approaches that combine information from several nodes to identify replicas [178].

Challenging collaborative computation Though collaborative approaches are common in defending the system from many attacks, their implementation in a decentralized setting brings new challenges. Not only it is difficult to know which peers to rely on, as we discuss in the next subsection, but reliance on ephemeral peers makes it difficult to trust the integrity of collective computations. Thus, additional mechanisms are needed to harden distributed computations and verify their results [157, 223], increasing the cost incurred by the system. The use of the blockchain to maintain distributed integrity in Bitcoin [168] also shows that while such a system to maintain the integrity of collaborative computation may be decentralized, it may incur large costs such as maintenance of the blockchain and hashing power.

Network diversity Finally, it is very common that nodes in a decentralized system have hugely varying capabilities (bandwidth, computation power, etc.). This increases the management difficulties for all the above points. For instance, heterogeneity in bandwidth or computation [91, 239] or the presence of networks of a different nature [145] increase the difficulty of routing. The emergence of superpeers to deal with peer diversity makes them attractive targets [158], thwarting the deployment of many defenses.

4.4.2 Lack of reputation

Decentralization is also an obstacle to the implementation of accountability and reputation mechanisms. Thus, nodes cannot be threatened with retaliation in the case of misbehaviour, which in turn has a negative impact on security and privacy. The negative effect can be amplified when privacy and anonymity mechanisms are in place, since it becomes even more difficult to identify misbehaving entities [112].

Information integrity A first effect of this lack of accountability is that nodes have no incentive to behave correctly and can try to distribute fake information that provides them with some advantage in the system (e.g. better performance). This problem has been identified in many scenarios such as P2P file sharing [262], multicast communication [260], or reputation [112]. In particular, the presence of churn, which make nodes short-lived and difficult to track over time (e.g. in ad-hoc networks) makes the establishment of reputation to guarantee veracity a very challenging problem [187], even more if privacy has to be preserved [198].

Poor Incentives Without reputation and retaliation it becomes a challenge to establish incentive schemes for nodes to not be selfish, in particular in a privacy preserving manner. A solution to this problem is increasing transparency of actions, e.g. by permitting witnesses to report on malicious nodes in a privacy-preserving manner [264]. However, the most popular approach is the use of (anonymous) payments that incentivize good and collaborative behavior that benefits all users in the network [23, 53, 129]. Bittorrent uses a tit-for-tat strategy when users share blocks to incentivize sharing.

4.5 What is still centralized in decentralized designs?

In many decentralized systems there are often “hidden” centralized assumptions in the form of parts of the design that need to be centralized for the design to operate correctly but that are not made explicit. These centralized

components are not elaborated in the original design and so left implicit, such as the power of the developers in projects like Tor and Bitcoin. There is also the case that components could be or have been decentralized but that over time have eventually evolved into centralized components, or a centralized option was chosen for some reason over a decentralized option, such as the use of Facebook Connect over the federated OAuth standard for authorization [108]. Such assumptions highlight research directions to achieve more effective decentralization, which we elaborate in Sect. 5.

4.5.1 Centralization of Network Information

It is a challenge in any decentralized system to route packets across the network to get to a destination for operational and privacy reasons. Typically routing can be divided in two main tasks: first how to find candidate nodes to relay traffic, and second how to select among these nodes. While as detailed in Section 4.1.3, there are many decentralized algorithms to choose the route, actually finding candidate nodes often still requires centralized work that can inform clients about possible routing choices.

4.5.2 Centralized Directories

A solution for the first problem is to assume that there exists a centralized directory that knows all members of the net. The most prominent example is the Domain Name System (DNS) that allows to resolve IP addresses associated to easy-to-remember domain names to allow finding hosts in the largest known decentralized system: the Internet. Though distributed, this centralized service has serious security implications, e.g. for privacy [164] or availability [236], and thus several alternatives are being proposed [240] and deployed [76]. Another example are Tor Directory authorities [75] that provide Tor clients with the full list of onion routers. These directories solve the routing problem but have become a bottleneck for the scalability of the system. Decentralizing these authorities in an efficient, privacy-preserving manner is an active area of research [162, 155]. Centralized directories are also present in Bitcoin in the form of the DNS seeds that are used upon a first connection to Bitcoin to discover other nodes. In all of these cases, a directory of what nodes are 'part' of the decentralized network is kept by a centralized authority.

Path selection Once routing alternatives are known the question remains: Which route to choose? Although DHTs could be used, it is more common to have a centralized server that can "rank" these options to allow for path optimization with respect to adversaries [5, 16, 82, 123], performance [206, 209, 238], or with respect to users' reputation [244]. Such a centralized ranking approach has been shown to be vulnerable to attacks [19, 30].

4.5.3 Trust establishment

Another hard challenge when decentralizing networks is how to ensure that nodes can be trusted to perform the actions they are assigned or can authenticate themselves as the intended receiver of a messages. Thus, designers often rely on centralized authentication or authorization services to establish trust relationships in distributed systems.

Authentication In general, PKI and certificate infrastructures are not decentralized. Some decentralized systems rely on centralized certification authorities to authenticate nodes that can be used for secure routing [48, 219], user authentication [38], or to enrol users in the system in the context of anonymous credentials [42, 43, 22], a privacy-preserving alternative for authentication to allow selective access to resources without requiring identification of the users. Such centralized authorities are simpler for deployability or usability, but become a single point of failure as pointed out by Lesueur et al. in [141]. They also introduce an imbalance of power unnatural for decentralized environments since they allow a single entity to revoke peers' authentication credentials. We also note that many decentralized designs do not address authentication (e.g. [175, 205], see [179] for more details).

Authorization Authorizing whether one node can access capabilities or data on another note can also be centralized. OAuth was designed to be federated in terms of authorization, but in practice only a few large providers use the standard for authorization [203]. Thus, if an adversary compromises a user's single authentication method such as a password, it could allow their node to be compromised across multiple decentralized systems. Assuming the existence of a centralized entity is also common when it comes to storing and enforcing authorization policies, as highlighted by numerous efforts to decentralize policy management and enforcement [139, 142, 248].

Abuse prevention When systems are decentralized and in particular when users are anonymous, accountability becomes a challenge. Hence, existing abuse-prevention schemes end up relying on centralized parties, often determining global reputation scores. Solutions based on blacklistable credentials (anonymous credentials for which authorization can be selectively revoked) use a centralized authority for enrollment [231, 232], or to store blacklists [122, 233]. Similarly, identity escrow [31] or revocable anonymous communication solutions [55, 128], that allow for re-identification of misbehaving users require a centralized party that stores those identities. In practice, spam prevention in federated email systems also uses centralized lists of known spammers.[13].

Payment systems In many applications of decentralized services it could be desirable to count on a payment system to reward peers for their contributions. While many alternatives have been presented in the literature specifically aimed at peer to peer systems, e.g. [23, 44, 257], they inherently rely on a centralized authority that opens accounts (the *bank*) and sometimes even on other authorities that can act as "arbiters" in case of dispute [23], or on authorities that record transactions to help taxation on the operations run in the system, even if the transactions are anonymized [225].

4.5.4 Computation is centralized

A number of decentralized systems are designed with the assumption that there is a central entity that performs computations on the data collected by the nodes in the system. Paradigmatic examples of this behavior are decentralized sensor networks [49, 88, 265] where the challenge is to send decentralized measurements to a "master" node, but there exist other applications such as distributed network monitoring for intrusion detection [186], anonymous surveys [113], or private statistics [83] in which, even though nodes perform decentralized computations, interaction with a central authority is needed to produce the final result. Another example is eVoting in which not only the final tally but also other events key to the security and privacy of the system are centralized, such as voter registration or ballot printing [182]. Despite the gains made in secure multi-party computation, in general the performance loss is still too high for practical systems.

Correct Protocol Implementation Often designs focus on limited aspects of a system, implicitly assuming that the parts that are not addressed will not affect the security or privacy properties obtained through decentralization. This is exemplified by de-anonymization techniques based on bad node configuration [152], or unobservability broken by lack of fidelity to a protocol [114]. Another example is eVoting protocols, that assume that voters' interfaces, networks, etc. are secure, while this is not always the case [210]. Finally, it is generally assumed the user themselves, who may give resources or otherwise control their own end-point, is secure, but this is often not the case.

Incentives are aligned Finally, decentralized schemes' security properties many times rely on the assumptions that nodes will act as they were intended to by the designers of the system. Yet, even if nodes are not malicious per se, they may not have incentives to provide enough resources to the network, i.e. the nodes may be selfish, which in turn affects the security properties [89, 174] of the entire decentralized network.

Trusted Developers and Protocol All decentralized systems work in virtue of having the nodes communicate via the same protocol. Thus, the actual software can often inevitably be a centralized point of failure if there are flaws on the level of the protocol. If the protocol is standardized, the implementation of the protocol itself may be a failure. Furthermore, the developers themselves could be compromised. One solution is to apply the

[13] http://spamhaus.org

technique of forcing public transparency and auditing of the integrity of the development process. Thus, open-source development with all development done in public repositories is increasingly required in decentralized system design as exemplified by Tor, with integrity ensured via deterministic builds to ensure all can verify the genuine binary, so that the authority to run new versions of the software remains in the hands of the operators. This is shown in Tor and increasingly in Bitcoin, where the choice to deploy particular open-source code is up to miners.

5 Decentralize all things: Vision and Roadmap

In the previous section we reviewed the state of the art in decentralized systems in terms of architectures, properties they bring and their limitations. Based on this state of the art, we provide an overview of the challenges that need to be addressed in the next years in order to design, implement and deploy secure decentralized privacy systems.

5.1 Address the shortcomings of decentralization

A number of designs we review consider decentralization as a goal and virtue in itself and do too little to address the challenges of such designs. In particular we studied in Section 4.4 a number of challenges intimately associated with decentralized architectures: an increased attack surface, with corrupt insiders; susceptibility to peers violating privacy and vulnerability to traffic analysis, integrity and consistency attacks; expensive and fragile routing; potential degradation in performance; loss of central choke points to enforce security controls; peer diversity and lack of incentives. These challenges are serious and real, and designers that do not acknowledge them and confront them head on will create weak systems that cannot credibly compete with centralized solutions.

Addressing these challenges in future designs requires both design and implementation discipline as well as fundamental research advances. A key question relates to how to reduce the attack surface across peers, and systematically protect the communication layer, including content and meta-data, as well as providing routing security in a systematic manner. Systems such as Freenet and P2P, as well as Tor Hidden Services, attempt to provide application platforms to an ecosystem of privacy applications. More research is required looking at those systems as platforms rather than purely as channels, including understanding their interfaces, their required performance and quality of service guarantees and their security properties as a whole system of interlocking parts.

Further work is also required to radically simplify the deployment and management of decentralized applications, either on larger platforms or as stand-alone distributed systems. Deployability and usable application life-cycle support is at the heart of the current centralized Cloud-based dev-ops revolution, and has made centralized app stores and Web applications as popular as they are. Yet, there are no equivalent tools or technologies to facilitate the deployment, management and monitoring of decentralized systems, let alone their continuous updates, application life-cycle management, and telemetry. The complete lack of such toolchains lowers the productivity of developers and makes the engineering and maintenance of decentralized systems very expensive. Yet, building such toolchains – without introducing any central control – is largely an open research problem. Successful projects such as Tor and BitCoin have developed best practices and running code in that space such as open-source development and deterministic builds to address security concerns that may be generalized.

Finally, there has to be a deeper acceptance that even honest users and peers in decentralized systems will have to be incentivised to participate and behave cooperatively. This is particularly true when stronger privacy protections are implemented and reputation based on repeated and iterated interactions cannot be leveraged. In those cases standard platforms must be developed to combat Sybil attacks and establish privacy preserving reputation to curtail abuse, and accounting and payment mechanisms need to be devised to ensure that those that do work are rewarded to sustain their operations. Systems that do not provide incentives for participation

in the infrastructure will fall foul of the tragedy of the commons and will remain small proofs of concepts – David Clark (IETF) called those 'Peter Pan Systems', namely 'systems that refuse to grow up'.

5.2 Towards full decentralization

In Section 4.5 we show how, despite the advances in the field, many aspects of decentralized systems are still carried out by a central entity. This reluctance to decentralization may respond to performance loss, security concerns or lack of adequate solutions.

From our study of the literature, we have shown a number of key functions of decentralized systems often fall-back to centralized models in practice. First, directories, key management and naming often remain more centralized. Thus the design of collective high-integrity infrastructures to support the directory, node discovery and key exchange needs of decentralized designs are needed. These designs will need to scale up and remain decentralized, while not being open to corruption or inconsistencies. The Bitcoin protocol and subsequent distributed ledger platforms – such as Ethereum[14] that supports smart contracts – have recently become popular and might be leveraged to ensure decentralized integrity. These can also address the need for payments – which are often assumed to require a central entity to act as the 'bank.'

Second, reputation and abuse control often require centralized entities: even advanced privacy-preserving techniques such as anonymous blacklisting assume that centralized services will issue and bind identities, and e-cash protocols rely on a bank to issue coins and prevent double spending. More work is required in establishing reputation in decentralized systems and preventing abuse without resorting to central points of control.

Finally, it is important to make credible assumptions about the platform security and computing environment of end-users or other peer devices. It is too facile to heavily rely on end-user systems keeping secret keys and data, and ignore that they are often compromised with malware and spyware – therefore exposing users to greater risks than if their data was kept away from those devices on centralized provider systems. Achieving perfect end-point security is an ambitious goal – and probably beyond the strict remit of building secure decentralized systems. However, architectures that display or limit the effect of compromises, and which may 'heal' and recover privacy properties following hacks, should be preferred to those that fail catastrophically under those conditions, particularly if they fail silently.

5.3 Develop design strategies

Aside from having a common definition of the privacy and security properties that can be sought through decentralization, as well as the means to quantify the extent to which a design supports them, robust decentralization engineering also requires the development of design strategies.

Section 4.1, for instance, illustrates the variety of different design alternatives available in the decentralized design space. Currently, there is no guidance as to how to search this space and choose the best combination to achieve a certain functionality as well as the desired level of privacy protection.

5.4 Develop systematic evaluation tools

A key missing piece in the puzzle of privacy-preserving decentralization is the lack of systematic means for evaluating the privacy and security properties provided by a given system.

As evidenced by our studies, decentralization can support privacy in a wide range of manners (Section 4.3), as well as supporting other properties too (Section 4.2). We observe that systems are often designed with one particular privacy goal in mind and designers craft ad-hoc evaluations to prove that indeed the goal is achieved.

[14] https://www.ethereum.org/

Moreover, not only evaluations are tailored to the designs but also the privacy goals are often redefined in these works. More often than not other properties are swept under the carpet, rarely mentioned even less evaluated.

A similar trend is observed in terms of measuring the severity of disadvantages introduced by decentralization. Though, as we show in Section 4.4, many weaknesses arise from decentralizing, few works evaluate their implications, and if they do so it is again following a procedure specific to their design which is difficult to extrapolate to other systems or environments.

A consequence of this custom-made design & evaluation procedure is that it becomes extremely difficult to compare systems and find promising directions that should be followed by posterior designs. In other words, it prevents the development of robust decentralized systems that can be built from the best bits and pieces of other designs.

It thus becomes apparent that the community needs to put effort into systematizing the definition of privacy properties sought by decentralized designs, as well as other properties that are needed to achieve privacy. More importantly, given that decentralized systems designers cannot be expected to have great expertise in security or privacy issues, there is a need to develop systematic means to evaluate how well these properties are achieved. This challenge shall not be solved simply once there is agreement on common definitions for privacy properties, but also requires the formalization of privacy properties into measurable objectives as well as the development of algorithms that are able to quantify privacy in systems. These algorithms, furthermore, must be scalable since they may often be applied to systems that count with thousands, or even millions, of nodes.

6 Open Technical Questions for Designing Decentralization

To build the next generation of decentralized systems, good will, slogans, and demands are not enough. Most designers of decentralized systems simply do not have a firm grasp on what properties their systems actually provide and as a result build systems that are likely often worse than centralized systems. For example, does it matter if you *control* your own data, if your data is not encrypted end-to-end so that it has no integrity, i.e. it can be altered at any time without a trace? Or do you really want to use a peer-to-peer system or blockchain-based system where your privacy is almost always sacrificed to any adversary that can observe the traffic between nodes, or even more easily simply inspect the public audit log? Lastly, why would you host any data of value on a decentralized node of a network such that your data may not be available when you need it? Indeed, given these constraints it seems almost logical that users return to the Cloud for real-world applications.

6.1 Integrity, Availability and Privacy

Many developers want users to return to a lost golden age where everyone ran their own web-server: Alas, this was only true of a few self-selecting engineers, and the vast popularity of services like Facebook and Gmail is testament to the fact that most users do not have the time or skills to host their own node in a decentralized network without a powerful incentive like file-sharing. Worse, we cannot demand that ordinary users magically transform themselves into systems administrators, and neither can we pretend to ourselves that even the most skilled of systems administrators in the 1990s did not have difficulties maintaining systems against security bugs and adversaries. Indeed, the immense reach of pervasive surveillance programmes demonstrates in spades how difficult it is to secure a system against a powerful adversary, and how often basic protocols that the entire Internet depends on have major flaws. Building successful decentralized systems that do not betray the security and privacy of their users is hard, and no job for amateurs who simply want to tack on a blockchain or P2P network to a pre-existing problem. However, has our analysis shown there is some fundamental trade-off between integrity, privacy and availability in decentralized systems?

The fundamental economic problem of building and maintaining such systems naively is that a good solution for one security property is an unsafe design pattern for another. The three primary decentralized systems all demonstrate this. Bitcoin comes with high-integrity at the cost of a public ledger with little privacy. Tor routers

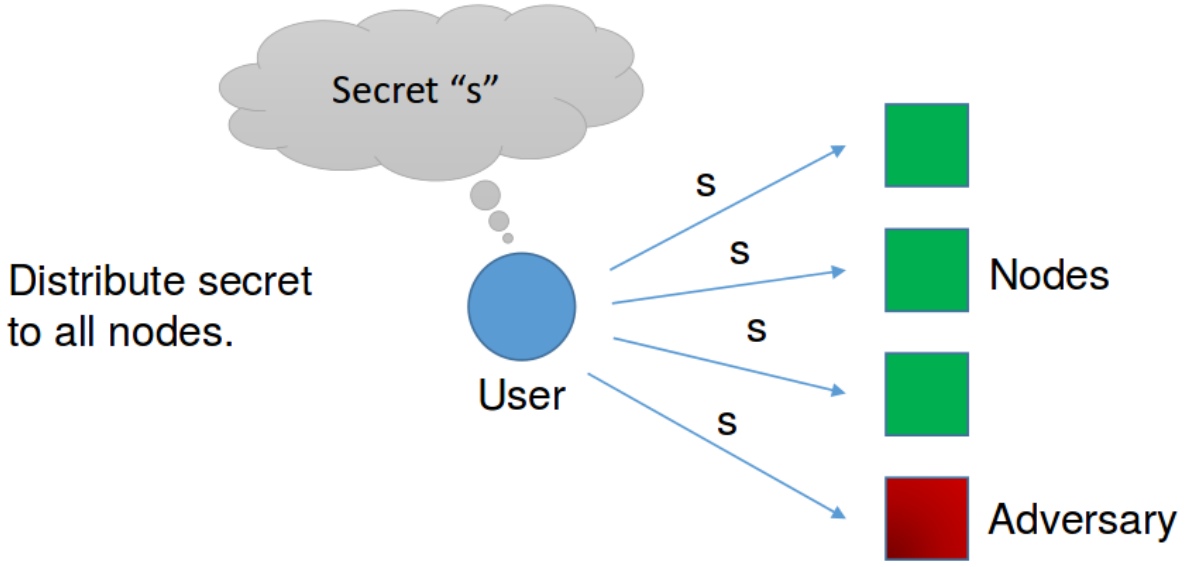

Figure 3: Privacy in Decentralized Systems

provide high-privacy at the cost of no available or correct collective statistics to ensure the integrity of the entire system. BitTorrent provides high availability in downloading files, but fails to provide privacy and integrity to its users against powerful adversaries.

We can broadly characterize privacy, integrity and availability properties into patterns of “fragile” decentralization. Privacy depends on sharing a secret s across a number of nodes in a decentralized network, so that if any part of the decentralized system is corrupt you lose privacy. Thus, the only design pattern that makes sense is to split the secret across all (or at least a large enough subset) of nodes as given in Figure 3. In terms of integrity, when storing a value of a function $f(p)$ as in Figure 4, if any part of the decentralized system is corrupt a node can lose the integrity of the response. Thus, the one safe design pattern is to make sure all nodes agree on the value, as can be done in a decentralized manner via blockchain technologies. Lastly as given by Figure 5, if any part of the decentralized system is unavailable you lose service. One solution is to rely not on the entire network, but only on a small agile subset of nodes. Yet this trade-off seems to go against integrity.

However, it is not pre-ordained that there is a trade-off between privacy, availability, and integrity in decentralized systems. Unlike Bitcoin, Zerocash combines both high-privacy and high integrity due to its complex cryptographic assumptions around zero-knowledge proofs. Likewise, many of the systems from the academic literature that build privacy into a P2P network, such as Drac[62], remain unimplemented but plausibly solve the issues of traffic analysis to defend privacy in a P2P network, and work on scalability shows that many cryptographic designs can scale if they make the right engineering decisions. Lastly, one of the surprising reasons why Bitcoin works is that it takes advantages of selfish incentive structures outside of traditional computer science: The desire for people to transfer around money and avoid paying fees. The same held for BitTorrent as well, whose immense popularity rested on the desire for people to share files. These incentives may not always be selfish: Tor’s uptake of Tor relays, in contrast, is run on the altruism of people who want to defend the right to privacy. So to build good secure decentralized systems, one needs:

- Experience in building *distributed systems*, as decentralized systems are by definition distributed.
- Deep knowledge of *cryptography*, as complex cryptographic protocols are necessary to achieve simultaneously privacy, integrity and availability.
- Understanding of *mechanism design, game theory and sociology* in order to understand the motivations of possibly selfish or unmotivated actors and build them into the decentralized system.

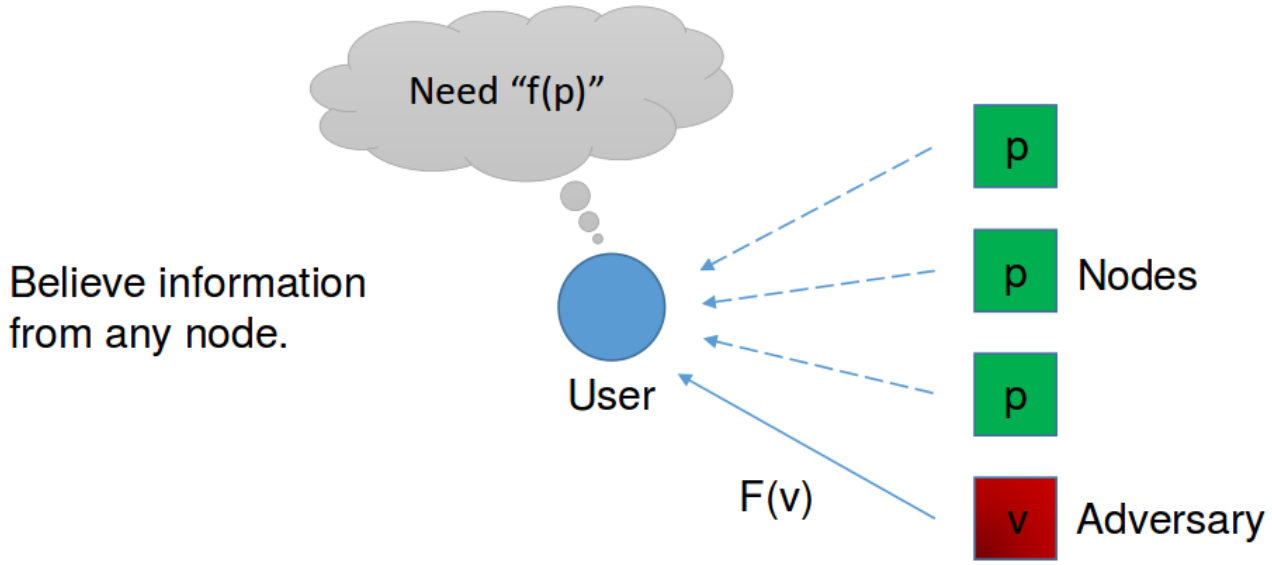

Figure 4: Integrity in Decentralized Systems

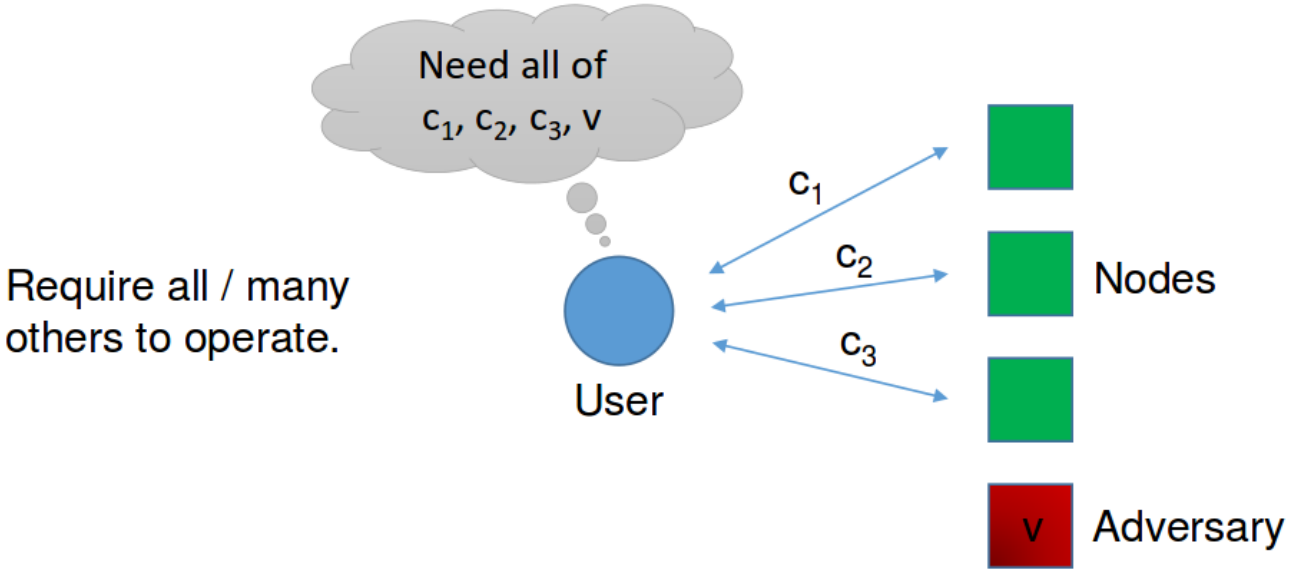

Figure 5: Availability in Decentralized Systems

It should be no surprise that there are not more deployed decentralized systems and very few people in the world exist that combine these skill-sets. Indeed, the amount of knowledge needed to build decentralized systems is simply greater than the knowledge needed to build a simple centralized web service. Furthermore, as shown by the massive successful attack on the Distributed Autonomous Organization built on Ethereum,[15] there is almost no application of the large amount of academic research in verified and provable security properties in distributed programming languages, languages one would assume would be naturally suited to building decentralized systems [92].

6.2 Scalability Concerns

A core central engineering challenge is that decentralized systems seem not to scale and are typically inefficient in comparison to centralized systems - and in practice, in a world with limited resources and investment, inefficient decentralization leads to a failure of decentralization. This problematic dynamic is typically built into decentralized designs: Maintaining high-integrity requires a majority to honestly participate in decisions. Although one could point to Bitcoin as a success, the larger Bitcoin grows the less it scales, as all miners need to detect and verify new blocks. This becomes even worse in Ethereum, where to "decentralize" a smart contract, the program must be executed on each node in the network: A less scalable design is almost unimaginable. In both Bitcoin and Ethereum, the larger the decentralized network, the more work each peer needs to do. Thus, Bitcoin and Ethereum will not scale without major design changes. One possible design is to require enough separate authorities to ensure diversity, but as few as possible to ensure efficiency and scalability. This principle could already be in action, such as in the way mining pools concentrate Bitcoin mining. Furthermore, it could also be at work in some of the centralizing tendencies in decentralized systems that we see in everything from BitTorrent super-nodes to market centralization. Yet, it does seem that a single authority is dangerous, and most successful institutions seem to have a separation of powers, even if only by three: From the European Commission, the Parliament and the Council to Tor entries, exits and guards. More empirical work is needed to work out the details of how decentralized systems can scale on both the technical and social levels.

6.3 Open Technical Questions for Decentralization

The ultimate security bet of decentralized systems is also ultimately unknown: Is being vulnerable to a random subset of decentralized authorities better than being vulnerable to one? In all honesty, the answer to this question is unknown and rests itself on assumptions about the adversaries being likely to compromise or control some of the authorities, or some of the authorities being naturally untrustworthy, assumptions that are fundamentally social. Ultimately, decentralization is the natural result of a breakdown in trust in centralized institutions, but we do not yet understand how to build decentralized institutions to support decentralized systems despite all the promise of Bitcoin to produce algorithmic monetary policy, and even more the hysteria around Ethereum to produce modern civilization with a scripting language with dubious security properties. As studied by projects like P2PValue,[16] there have been various successful commons-based projects whose governance structures survived the original "peer-to-peer" boom in 2001 such as Wikipedia. Due to advances in mechanism design in understanding the incentive structures behind human co-operation [181] and the rise of computational social science that can measure empirically social incentives [134], an emerging art of building decentralized social structures is now possible. To summarize, open questions remain such as:

- How to integrate strong integrity, availability and privacy via cryptographic protocols despite wide decentralization?
- How to make decentralized systems scale up so that more participants can provide more capacity and value to the entire network?
- How to co-design institutions, incentives, usability and governance in vast decentralized systems?

[15] http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

[16] https://p2pvalue.eu/wp-content/uploads/legacy/files/u28/D12_31July_TheoreticalFindingsA%20(1).pdf

7 Social and Philosophical Perspectives on Decentralization

The technical work on decentralization, including properties such as integrity, availability and privacy need to be complemented with an investigation of the socio-economic and political aspects of decentralization. Indeed, a quick glance to the current debates on decentralized technologies show how their effectiveness, visibility and sustainability over time are intimately entwined with issues of governance, management and cooperation not as separate, "juxtaposed" phenomena, but as they are embedded in the architecture itself. The promise of technical decentralization and the inscription of openness and end-to-end empowerment in Internet architecture is still holding its own, with a variety of research projects and start-ups building on the Bitcoin blockchain[17] or hoping to creating their own.[18] The most ambitious attempt, Ethereum, was based on a "Turing-complete" programming language for smart contracts called Solidity. Excited, small investors poured over 150 million USD into an attempt to build a "Distributed Autonomous Organization" (DAO), a sort of stateless corporation that could act as an investor-directed venture capital fund that would act on behalf of its investors to offer high integrity and decentralization for its 11,000 investors. The integrity would be guaranteed by the open-source code of the protocol, which would allow investors to have trust that the DAO was acting as designed to invest their funds in a transparent manner.

There was a dangerous error in the code of Solidity, the language used by Ethereum that ran the DAO. In effect, the DAO was constructed so that investors could split into a new DAO - a "child DAO" - after executing a "smart" contract. This was due to the fact that while the DAO was a decentralized high availability ledger, a small number of people known as "curators" centralized the ability for funding proposals to go to the DAO. Splitting a DAO would allow investors to create a new DAO with potentially a new board of centralized curators. However, this underlying vulnerability allowed a smart contract to execute and spawn a child DAO but before the funds were withdrawn and balance updated. Thus, before funds could be put into the new child DAO, the smart contract would propose yet another split. Thanks to Ethereum's language for smart contracts being Turing-complete, this split could be done recursively, and before each new child DAO was created the funds would be drained from one DAO to another - infinitely, or at least until the money ran out - before the final balance was updated.[19] This allowed a "Dark DAO" controlled by a malicious adversary to siphon off 50 million USD of the funds in the DAO, crashing the price of Ether. In panic, the Ethereum developers proposed a "soft fork" that would exclude the stolen funds from future transactions, as well as a "hard fork" that would simply roll back the transactions: "Thus, politics - the discipline of collective management - reasserts itself as having primacy over human affairs" as the coders would simply seize back control of their own decentralized system and impose arbitrary social decisions.[20] In this regard, the implicit centralization of power in the developers over what they advertised as a perfect - due to its supposed absence of human interference - decentralized system like Ethereum was revealed. The DAO hack gave proof of the social embedding of decentralized systems, and the necessity for any study of decentralization to take on board the full social context seriously, from the values of the developers to the motives of hackers and honest users.

Indeed, as we have argued elsewhere within the frame of the P2PValue project,[21] it is appropriate to consider the variety of levels at which (de-)centralization operates in a networked system – ranging from architecture to governance, ownership and value redistribution - in order to comprehensively assess the benefits and downfalls of decentralization for different types of systems [81]. Thus, as a necessary complement to the technical discussion, this section provides an overview of the literature that addresses network architecture and design, and more specifically the dialogue between centralization and decentralization, from the perspective of social, information and economic sciences, in particular science and technology studies (STS). As Susan Leigh Star has effectively put it, social sciences can and should contribute to "surface invisible work" [211] underlying networked

[17] Blockstream being representative, having received 21 million USD in funding for side-chains off the Bitcoin blockchain https://www.blockstream.com/

[18] Such as Freecoin from the D-CENT EC Project: http://freecoin.ch/ that wanted to create a blockchain for the social good, although it currently has little adoption among users.

[19] http://hackingdistributed.com/2016/06/17/thoughts-on-the-dao-hack/

[20] https://www.benthamsgaze.org/2016/06/17/smart-contracts-beyond-the-age-of-innocence/

[21] https://p2pvalue.eu/

practices, uses and exchanges - as an integral part of the “processes of constitution, organization, and change of ... the network society” [47] (p. 693).

Within the larger body of literature that addresses the role of technology as a political and social tool, a small subset of scholars working in social studies of technology have addressed the question of how particular forms of distribution and decentralization (or their lack) in network architecture impact specific procedures, practices and uses. As Barbara van Schewick has suggested, architectures should be understood as an “alternative way of influencing economic systems” [234] (p. 3), indeed, the very fabric of user behavior and interaction. Most notably, the status of every Internet user as a consumer, a sharer, a producer and possibly a manager of digital content is informed by, and shapes in return, the technical structure and organization of the services (s)he has access to: their mandatory passage points, places of storage and trade and required intersections. The architecture of networking applications is here studied as a “relational property, not as a thing stripped of use” [214] (p. 113), “as part of human organization, and as problematic as any other” [212] (p. 116). Such an approach provides an added value to the study of those communities, groups and practices that, by leveraging socio-technical dynamics of distribution, decentralization, collaboration and peer production, are currently questioning more traditional or institutionalized models of content creation, search and sharing.

7.1 Social sciences and the integration of architectures and practices in qualitative methods

The architecture of a network or an application is its underlying technical structure [234], designed according to a “matrix of concepts” [2]: its logical and structural layout, consisting of transmission equipment, communication protocols, infrastructure and connectivity between its components or nodes. The choice of taking architectures, artifacts transparent to end users by fiat of their creators, as the starting point - or at least as an important and integral part - of a study of practices and uses of network media raises a number of challenges, as well as great promise.

As Barbara van Schewick points out, the compartmentalization of disciplines may have led in the past to a general understanding of architectures as artifacts that are “relevant only to engineers”, and as such, should be exclusively left to their purview [234] (p. 2). However, in relation to network media, software, code and cyberinfrastructure, studies have recently taken up the challenge of interdisciplinarity [95], drawing on past endeavours in the field of the sociology of technology and science, exploring the social and political qualities of infrastructures [211]. In addition, some authors experimenting at the intersection of computer science, sociology, law and science & technology studies explore innovative methodological approaches to architectures, working on the integration of architectures and practices in their analyses. These bodies of work will now be addressed in some more detail.

7.2 Internet architecture, a cross-disciplinary research object

Literature in computer science and computer engineering has, perhaps quite obviously, paid a great deal of attention to architectures of Internet-based applications and networks: their definition [201] [200], their technical advantages and disadvantages in a comparative perspective (e.g. client/server vs. peer-to-peer architectures [237] p. 11-16) and their application to specific projects serving a variety of uses [177] (p. 67-159); these “purely” technical aspects of such systems are seldom addressed in relation to their societal, relational and organizational properties [228] (p. 113-115). In some cases of highly publicized and debated applications - as with some P2P systems - engineers have at times sought to present a technical perspective on the limits and advantages of specific architectures within at-large political and public debates [13] [137] [136]. Other scholars, interested in the metrology of networks, seek to model interactions by means of large-scale graphs, so as to study patterns of information propagation, the robustness of networks and the forms of exchange and sharing [3]. Their aim is to build measuring tools that are better adapted to the ever-increasing size and complexity of networks and more able to face the increasing inadequacy of traditional statistical and sampling methods to account for the magnitude of this scaling process [15].

On the other hand, as of today, an important number of works in economic and social sciences have sought to explore the practices of sharing, cooperation and interaction facilitated or enabled by online environments: as is the case with many contributions exploring new forms of organization, contribution and collaboration, like social networks [36] [45] or online communities [14], be they composed of fans [110], contributors to wiki projects [188], or specialized professionals [146].

The body of work on the law of network technologies has been extensively addressed, around - again, perhaps unsurprisingly - the dynamics of file-sharing practices by means of direct-exchange networking technologies, and has focused the debate on the ways in which innovative networking practices may be assimilated, by analogy, to mechanisms of remuneration and compensation similar to those in place for material, private copies (e.g. [97]). As pointed out by Melanie Dulong de Rosnay [80], as of now, only a comparatively small number of works have been devoted to the ways in which law can take into account the objects and sources of value (such as metadata and personal data) produced by new technical configurations.

Some examples in recent literature open very interesting paths by undertaking the next step in the experimentation with interdisciplinarity. These authors, coming from a variety of different backgrounds, approach architectures in innovative ways by integrating the link between architectures and practices in their analyses.

7.3 Science and Technology Studies

Perhaps the most notable attempt in this direction is constituted by the work, carried out during the last fifteen years by Susan Leigh Star and colleagues within the field of STS, on infrastructures as constantly evolving socio-technical systems, informed not only by physical elements invisible to the end user, but also by factors such as social organization and knowledge sharing [214] [173] [211] [212] [213]. Through her “call to study boring things,” Star effectively conveys the idea that architectural design choices, technical specifications, standards and number sequences are no less important to the study of information systems because they are “hidden mechanisms subtending those processes more familiar to social scientists” [211]. As she writes in a seminal article on the ethnography of infrastructure:

It takes some digging to unearth the dramas inherent in system design creating, to restore narrative to what appears to be dead lists. ... Much of the ethnographic study of information systems implicitly involves the study of infrastructure. Struggles with infrastructure are built into the very fabric of technical work ... However, it is easy to stay within the traditional purview of field studies: talk, community, identity and group processes, as now mediated by information technology. ... Study an information system and neglect its standards, wires and settings, and you miss equally essential aspects of aesthetics, justice and change [211] (p. 337-339).

This “relational” approach brings about considerable changes in methods, as the scope of the fieldwork enlarges to include arenas where the shapes of architecture and infrastructure are observed, deconstructed and reconstructed, and decisions are made about codes, standards, bricolages and reconfigurations [213] (p. 151-152), where the scholar undertakes a combination of “historical and literary analysis, traditional tools like interviews and observations, systems analysis and usability studies” [211] (p. 382).

Emergent bodies of work such as software studies, critical code studies and cyberinfrastructure studies [148] [95] [151] [191] owe a lot to the STS approach, seeking, as Matt Kirschenbaum [127] puts it, to balance “the deployment of critical terms like ’virtuality’ ... [with] a commitment to meticulous documentary research to recover and stabilize the material traces of new media”. The materiality of software, code and so-called virtual elements of the Internet user’s experience is reaffirmed, and the relationship between these layers (or “levels”, as defined by Mark Marino) explored: Meaning grows out of the functioning of the code but is not limited to the literal processes the code enacts. Through CCS, practitioners may critique the larger human and computer systems, from the level of the computer to the level of the society in which these code objects circulate and exert influence [151].

7.4 Quantitative sociology and information studies

On the side of computational and quantitative sociology, David Hales and colleagues have sought to explore features of particular groupings that he calls “virtual tribes”, such as dynamic formation and dissolution overtime, cooperation, specialization, reputation systems and occasional antagonist behavior; he considers that a thorough understanding of such phenomena is a necessary precondition for the construction of robust and resilient software systems, both today and in the future [103] [104] [149].

Information studies scholar and Internet pioneer Philip Agre has, ahead of his time, explored the relationship between technical architecture and institutions, notably the difference between “architecture as politics” and “architecture as a substitute for politics” [2]. He argues that technologies “often come wrapped in stories about politics”, and while these stories may not explain the motives of the technologists, they are indeed useful to account for the energy that makes a technology an inherently social one, and projects it into the larger world (p. 39). Defining architectures as the matrices of concepts (e.g. the distinction between clients and servers) designed into technology, and institutions as the matrices of concepts that organize language, rules, job titles and other social categories in particular societal sectors, Agre suggests that the engineering story of rationally distributed computation and the political story of institutional change through decentralized architecture are not naturally related. They reconfigure and evolve constantly, and for these reconfigurations and evolutions to share a common direction, they need work:

Decentralized institutions do not imply decentralized architectures, or vice versa. The drive toward decentralized architectures need not serve the political purpose of decentralizing society. Architectures and institutions inevitably co-evolve, and to the extent they can be designed, they should be designed together... Radically improved information and communication technologies do open new possibilities for institutional change. To explore those possibilities, though, technologists will need better ideas about institutions. [2]

7.5 Governance, Law and Policy Studies

At the crossroads of informatics, economics and law, Barbara van Schewick has recently put forward the idea that the architecture of the Internet, and of the applications running on it, is relevant to economics. Her work seeks to examine how changes, notably design choices, in the Internet’s architecture (that she defines operationally as the “underlying technical structure” of the network of networks) affect the economic environment for innovation, and evaluates the impact of these changes from the perspective of public policy [234] (p. 2). According to van Schewick, this is a first step towards filling a gap in how scholarship understands innovators’ decisions to innovate and the economic environment for innovation: after many years of research on innovation processes, we understand how these are affected by changes in laws, norms and prices; yet, we lack a similar understanding of how architecture and innovation impact each other (p. 2-3). Perhaps, van Schewick suggests, this is due to the intrinsic appeal of architectures as purely technical systems:

Just as the architecture of a house describes its basic inner structure, the architecture of a complex system describes the basic inner structure of the system - its components, what they do, and how they interact to provide the system’s functionality. That such a technical structure may have economic consequences at all is a relatively recent insight. Most people still think of architectures as technical artifacts that are relevant only to engineers. Thus, understanding how the Internet’s architecture affects innovation requires us to think more generally about how architectures affect innovation. [234] (p. 4).

Traditionally, she concludes, policy makers have used the law to bring about desired economic effects. Architecture de facto constitutes an alternative way of influencing economic systems, and as such, it is becoming another tool that actors can use to further their interests (p. 389).

Along the same lines, within a large-scale project investigating how the corpus of Requests for Comments (RFCs) of the Internet Engineering Task Force provides indications on the ways in which the Internet’s technical designers understood and engaged with law and policy issues, Sandra Braman has recently [37] explored how the core problem in the Internet’s technical design was to build structures that not only tolerated, but actually

facilitated change in ways not envisaged by its initial developers. By addressing the ways in which change and stability themselves were conceptualized by Internet designers, Braman argues that undertaking research on architectural “design for instability “ as applied to the Internet provides insight not only into the Internet itself, but into its social, legal and technical relations with other information and communication technologies (ICTs).

Drawing on pioneering works such as those of Yochai Benkler on sharing as a paradigm of economic production in its own right [25] and of Lawrence Lessig on “code as law” [140], the relationship between architecture and law is further explored by Niva Elkin-Koren [84] [85]; a common trait of her works is its underlying perspective on architecture as a dynamic parameter, and she treats it as such while studying the reciprocal influences of law and technology design in information and communication systems. Elkin-Koren argues that the interrelationship between law and technology often focuses on one single aspect, the challenges that emerging technologies pose to the existing legal regime, thereby creating a need for further legal reform; thus, she notes how juridical measures involving technology both as a target of regulation and as a means of enforcement should take into account that the law does not merely respond to new technologies, but also shapes them and may affect their design [85].

7.6 Social science and Decentralized architectures

The Internet’s current trajectories of innovation are making it increasingly evident by the day: the evolutions (and in-volutions) of the “network of networks”, and at a broader level of electronic communications, are likely to depend in the medium-to-long term on the topology and the organizational/technical model of Internet-based applications, as well as on the infrastructure underlying them [4].

The development of services based on distributed architectures is currently affirming itself as one of the Internet’s most important axes of transformation. The concept of distribution is somehow shaped and inscribed into the very beginnings of the Internet - notably in the organization and circulation of information fluxes - but its current topology integrates this structuring principle only in very limited ways [177]. The limits of the “classic” urbanism of the Internet, which has been predominant since the beginning of its commercial era and its appropriation by the masses, are becoming evident with regards to phenomena such as the widespread success of social media [197]. While Internet users have become, at least potentially, not only consumers but also distributors, sharers and producers of digital content, the network of networks is structured in such a way that large quantities of data are centralized and compressed within specific regions of the Internet, and at the same time are most suited to a rapid re-diffusion and re-sharing in multiple locations of a network that has now reached its full globalization.

The current organization of Internet-based services and the structure of the network that enables their functioning, with its mandatory passage points, places of storage and trade, and required intersections, raises many questions, both in terms of the optimized utilization of storage resources, and of the fluidity, rapidity and effectiveness of electronic exchanges. Other interrogations, on the security of exchanges and on the stability of the network, must also be added to these issues: a series of malfunctions and breakdowns with important consequences at the global level draw our attention to questions of security and data protection, inherent to the Internet’s current structure.

These questions impact largely the balance of powers between users and network providers, and reach questions of net neutrality. To what extent can network providers interfere with specific uses? Can the network be optimized for specific uses? As Barbara van Schewick points out, by enabling users to use the Internet in the way that creates the most value for them, changes in architecture are not only likely to impact the value of the Internet for users, but also to increase the Internet’s overall value to society:

But the social value of architectures ... goes beyond that. The Internet has the potential to enhance individual freedom, provide a platform for better democratic participation, foster a more critical and self-reïňĆective culture, and potentially improve human development everywhere. The Internet’s ability to realize this potential, however, is tightly linked to features: User choice, non-discrimination, non-optimization [234] (p. 387), that may

be achieved in different ways by designing its underlying architecture in different ways. Resorting to decentralized architectures and distributed organizational forms, then, constitutes a different way to address some issues of management of the network, in a perspective of effectiveness, security and digital "sustainable development" (better resource management), and of maximization of its value to society. This idea is further explored by Michel Bauwens [21] who, proposing a vision of the P2P model that is based on but goes beyond computer technology, puts forward a P2P theory as a "general theory" of collaborative and direct human interaction, an emerging, pervasive and inherently social phenomenon that may be profoundly transforming the way in which society and human civilization is organised.

7.7 Architecture as politics, architecture as a substitute for politics?

Social scientists should watch out for the traps that an architectural model with strong a priori connotations of equality and decentralization may set up. As noted by Philip Agre in the case of P2P, it is particularly easy to juxtapose architecture to the stories of institutions, individuals and groups, assuming that one determines the other - but this may lead to a misleading shortcut:

In the case of P2P technologies, the official engineering story is that computational effort should be distributed to reflect the structure of the problem. But the engineering story does not explain the strong feelings P2P computing often evokes. The strong feelings derive from a political story, often heatedly disavowed by technologists but widespread in the culture: P2P delivers on the Internet's promise of decentralization. By minimizing the role of centralized computing elements, the story goes, P2P systems will be immune to censorship, monopoly, regulation, and other exercises of centralized authority. This juxtaposition of engineering and politics is common enough, and for an obvious reason: engineered artifacts such as the Internet are embedded in society in complicated ways ... the case of P2P computing (is good) to analyze the relationship between engineering and politics - or, as I want to say, between architectures and institutions... The P2P movement understands that architecture is politics, but it should not assume that architecture is a substitute for politics [2] (p. 39-42).

Decentralized socio-technical systems may be better analyzed and understood with an approach that addresses, studies and explores architecture as the very fabric of those interactions and examines how these shape, in return, subsequent negotiations and redesigns of the system. Scholars interested in networking technologies of communication and exchange need to "learn to read these invisible layers of control and access. In order to understand how this operates, however, it is necessary to 'deconstruct' the boring, backstage parts ... to disembed the narratives it contains and the behind-the-scenes decisions ... as part of material information science culture" [211] (p. 110).

A social science-informed gaze placed upon architectures allows to delve into the dynamics of articulation between local and global dimensions in a distributed application; of sharing of disk space and bandwidth as the cornerstone of a socio-economic model; of deployment of technical uncertainty and social opportunity at the "edges" of the network, where under-utilized resources, both human and material, can be leveraged.

7.8 Towards the case studies on decentralized software

Thus, the elaboration of ethnographic and social scientific case studies on decentralized messaging applications entails a plural approach, that follows on one hand the innovators, trying to identify their strategies in the construction of the technologies, as well as their valors, cultures and imaginaires of reference, and on the other hand, the role played, where possible, by the first users of the systems. The objective is threefold: retracing and breaking down, in developers' and users' narratives, the actions and dynamics that represent at once the technology and the changes it purports; following, by means of onsite and online ethnography, how innovators manage the economic, political and social "relapses" of technical changes and development processes; tracing how discussions and controversies that take place on technical forums between developers and users, and among users themselves, progressively shape directions of mobilization for and by means of decentralized applications.

For all these reasons, it proves useful to avoid considering “decentralized messaging” as a pre-defined object. Adopting a pragmatic approach, the starting point for the fieldwork becomes the observation that, in the ICTs domain, there currently exist a variety of research projects and applications that, in different manners and for different purposes, take up with “decentralized messaging”, that is defined in a transversal way as a distributed, private, social and user-centered alternative. A name and four adjectives that become the entry point into the fieldwork, which to observes the (re)configurations and (re)compositions in the hands of the actors and the shaping of the systems.

7.9 A real-time sociology of innovation

An empirical inquiry carried out by means of this approach helps identify “live”, and in a manner transversal to the different cases, uses and technologies “in the making” [40] [41]. At the same time we are trying to obtain a common vision of the directions of appropriation of decentralized messaging technologies. What we have called a “real-time sociology of innovation” [167] proves a viable method to apprehend variable, multi-dimensional situations, and attempts to draw some conclusions on their possible developments and applications. At the same time, there is a need to address the more ideological and utopian dimension of these “alternatives” - that which speaks of an Internet ideal of decentralization and autonomy - that are taken as a subject of inquiry, to try and show how it leads to ways of doing things, explains choices and validates assumptions. Along these lines, and once again following an STS-based tradition, the observation of transformations, passages, negotiations, modifications of objects, and of the moments where these are put on “trial” beyond the scheduled phases of development, are of special importance.

A particularly stimulating aspect of this approach is the consideration of how techno-legal forms take shape, in the pursuit of three objectives. Firstly, in order to successfully define the “legality” of such services, strictly linked to their constantly evolving architecture that is often only partially accounted for in written juridical documents. Secondly, to try and rise above a conception of the relationship between law and technology that all too often focuses on one aspect: the fact that emerging technologies pose challenges to existing legal regimes, creating a need for reform of these regimes. Thirdly, so that the objects and the resources enabling decentralized and private messaging may be understood as instruments of definition and of protection of the rights of users of Internet-based services.

In short, the acknowledgment of the importance of architectures calls for a process of methodological readjustment. It implies delving as much as possible into the technical functioning, especially encryption, and takes it as a core feature (even if not necessarily the cause) of the types of exchanges that bake place within a service, of their effectiveness and of their directness. It implies addressing the total or partial removal of technical “intermediaries” [85], as a structuring dynamic in new-generation participative instruments. It means understanding where in the “fringes and materialities of infrastructures” [212] a password is stored, a file is indexed and encrypted, a download starts and ends, so as to understand how new dynamics for the protection of personal liberties and rights are taking hold - or are endangered. In short, learning to read the “invisible layers” of decentralized messaging applications is as much a challenge as it is an opportunity to explore collaborative practices carried out in, on and through them, and to observe how these practices inform the architecture in return, the sharing of resources it entails and its medium- and long-term socio-technical sustainability.

However, in a connected world where more applications than ever want to use the network, send packets and consume bandwidth - thereby placing new strains and tensions on the Internet’s architecture - social scientists need to accept the challenge just as much as the technical people who are working on the future topology of the “network of networks”. It is, likely, one of the most promising ways to shed new light on dynamics of content creation, sharing, publishing and management that are shaping, and being shaped by, the future Internet - one of the best ways to contribute to its future sustainability.

7.10 Open Questions for the Sociology of Decentralization

Our goal in NEXTLEAP of "caring about the plumbing" means "[f]inding the invisible work ... in the traces left behind by coders, designers, and users of systems" [211]; the inclusion of the lower layers in the analysis means doing a sociology of networks that is not afraid of its subject of study. A consequence of this approach is a specific attention to an aspect of networks that is not only very discreet, but even invisible to the eyes of the users: their architecture. Of course, social scientists should remain social scientists: this interest in architectures derives from the hypothesis that particular forms of distribution call for specific procedures, particular uses and peculiar "user portraits". In doing so, one is able to flesh out how some attributes of technology, of which users often lack a direct knowledge or awareness, are bound to fully influence and inform issues that are often crucial for uses and practices, such as the treatment and physical location of data, the management of computing resources, the shape and results of queries to search engines. The shaping of links, nodes, mandatory transit points, information propagation protocols - in a word, the architecture - can tell social scientists many things about the socio-political specificities and promises of decentralized messaging applications, the challenges they face and the opportunities they may present for the medium-term evolution of the Internet model.

- How does the development of decentralized and/or encrypted messaging systems unfold, and to what extent are users active part of this development? What kinds of users are they – Power users, tech-savvy users, activist users – or is there more?
- What are the motivations and incentives of different user communities to participate in decentralized and/or encrypted messaging systems?
- How does decentralization unfold at different levels of the systems, and how do these different levels of decentralization (technical, social, political) influence one another?

7.11 The Philosophy of Decentralization

The socio-technical science of decentralization so far rests upon a cluster of terminology: Decentralization, network, node, peer, adversary, trust, and so on. Without defining these terms as done in Section 9, no new internet science could be created. While we find the entire theoretical edifice of decentralization is constructed on terms with seemingly "common sense" meanings, in reality there is much terminological confusion, which if left unchecked would lead to scientific confusion. For example, Baran in his original internet network routing design used the term "decentralized" to mean what has been earlier called by ourselves "federated," and Baran defined "distributed" to mean what we call "peer-to-peer" [18]. The security and privacy properties are phrased in terms of words such as "adversary" and "trust," whose ultimate definition is often in the context of particular social situations embedded within a historical contexts.

In order to create a new science of decentralization, we must ground this new kind of internet science in a philosophy of decentralization that reveals and clarifies the conceptual constellation that lies behind the use of this socio-technical vocabulary. As put by Marcuse, "exactness and clarity in philosophy cannot be attained within the universe of ordinary discourse" (p. 184) as philosophy must maintain an ultimately critical - and political - perspective that rather than taking everyday "concepts" as what they are, but returns to "the qualitative difference between that which things really are and that which they are made to be" (p. 188), a sensibility that would also appeal to many programmers [150]. So the ultimate fusion of the social and the technical concepts must then by necessity happen on the level of philosophy. Philosophy can fuse the conceptual structures of the disparate interdisciplinary vocabulary into a new whole that not only grounds a theory of decentralization and does justice to existing systems, but can open up new kinds of spaces of design to empower different needs that go outside the needs imposed by centralized services such as Gmail and Facebook.

As physics originally began as a branch of philosophy in the time of Aristotle that prepared one for the "first science" of ontology, at the dawn of the digital age internet science must return to ontology in order to grasp the technical changes wrought by the Internet on our technological, social, and even biological structures. The point of philosophy is that, due to our own epistemological limits, the ontological structures of "what are" can

be hidden and need delicate philosophical work to uncover and so become generative of new concepts again. Although the vocabulary in Section 9 is our agreed upon starting point for a conversation, the nature of these concepts will be transformed as the interdisciplinary work of the NEXTLEAP project around decentralization continues.[22] This should be expected, as our current understanding of most disciplinary terms comes from before the widespread ubiquity of the Internet, if not the early stages of the Internet and the digital era. What has been termed "philosophical engineering" acknowledges that engineers are now bringing via their creation of Internet protocols into concrete and material being new kinds of ontologies that are radically different from the ontologies of the pre-Internet era [107]. So a new kind of philosophical practice that takes technical invention as a "first-class citizen" is needed in order to make explicit the social and ethical commitments in "value-by-design," but goes deeper into the metaphysical and ontological ruptures that are creating entire ways of being and thinking in the Internet era.

Decentralized systems also hold implicit the promise of a new kind of intelligence that combines human intelligence with technical devices: Collective intelligence is central to our being in the Internet era. As put by Andy Clark, there is no reason for us not to give "cognitive credit" to artifacts in the world if they are necessary for us to accomplish certain kinds of cognitive tasks: The classic example being a person with severe memory loss being dependent on their notebook to find their way around[56]. Today, this situation is common-place, but not only for those with severe memory loss; increasingly, all of us are becoming dependent on Google Maps and our smartphone to find our way around the world [105]. Although this dependency may be viewed as dangerous, it seems that it has long been part of human history to essentially co-evolve with our techniques [216], and that this co-evolution now takes a fundamentally cognitive character. What is not understood well is what new kinds of Internet-enabled technically-embedded capabilities - including on the social and economic level as explored by Amartya Sen [202] - mean on the collective level of society, especially given the rise in digital social innovation where the Internet allows the "scaling" of these capabilities due to increasingly universal access to information and low-latency global communication [106]. Collective intelligence is how we extend our cognition not just via our devices, but via each other.

One hope, engendered by pioneers of Engelbart but woefully under-theorized, is that these new kinds of capabilities would finally allow the solution of global problems that escaped even the most intelligent of individuals: "By augmenting human intellect we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems. Man's population and gross product are increasing at a considerable rate, but the complexity of his problems grows still faster, and the urgency with which solutions must be found becomes steadily greater in response to the increased rate of activity and the increasingly global nature of that activity"[87]. The kinds of global problems we face today, ranging from the financial crisis of value to the ecological crisis of climate change, may require new kinds of thinking and analysis that require Internet-driven collective intelligence on a global scale. As noted by Deleuze, although we have always thought *with* milieus and via assemblages of objects (such as a traditional philosopher writing with pencil on paper after having read related literature in the form of books), collective intelligence form new kinds of techno-social assemblages with new capabilities that are inherently autopoeitic with regards to their technical milieu; in other words, a collective intelligence that is embedded in the environment that is self-sustaining in terms of its organization due to low-latency communication [153].

Collective intelligence is profoundly different than externally and centrally-directed crowd-sourcing, as collective intelligence is decentralized (and so has multiple sources of authority) insofar as a collective intelligence is composed of autonomous individuals that collectively agree to set their own collective goal and so be able to solve certain new kinds of problems that the component individuals could not solve before. One example is the Polymath project: hundreds of professional and amateur mathematicians collaborated using the Internet to create genuinely novel mathematical proofs; so exploring logical pathways beyond the cognitive resources of any lone mathematician [60]. In order to engineer this new kind of collective intelligence and autopoeitic self-organization that makes the "whole greater than the sum of the parts," there needs to be trust between all the humans involved in the collectively intelligence as well as trust in the underlying Internet-driven architecture.

[22]The vocabulary is stored on the NEXTLEAP wiki at `https://github.com/nextleap-project/nextleap/wiki/SharedVocabulary`, so it may be continuously updated.

In a healthy society, humans are not actually born as individuals with their own autonomy of thought, but go through a process of individuation where they mature and develop their own thought through the family, wider social structures such as education, and their relationship with technology [115]. This is also a social process where the individuation of one individual is linked to the individuation of another and their collective techno-social environment.[23] There is increased evidence that the effect of the Internet on individuation has even neural effects on our cognitive capabilities, such as attention and language use [46]. It has even been theorized by Stiegler that the very foundations of our metaphysics that allow our ontological and epistemological grasp of the world - the Kantian *a priori* that exist before any concepts - are essentially conditioned by technology [217]. Given that technology co-evolves across the biological, cognitive, and social levels of human individuation, Stiegler calls for a new kind of "digital studies" to understand how all sorts of technical extensions of man, from McLuhan-esque extensions of perception to extended minds, can be studied holistically via philosophy [218].

The danger of censorship and surveillance is that they turn the emergent collective intelligence of the Internet against the very humans that constitute it, destroying the trust we have in our new cognitive infrastructure. Insofar as we are dependent on the Internet for our capabilities this is both the most intimate of violations and most destructive, as our own "extended mind" can be turned against us, with our exteriorized memories in terms of messages, photos, social media, and website visits weaponized against us by unseen agents for purposes that may not be our own. In this regard, the "adversary" of cryptography and secure systems thinking is real. Many adversaries are inherently local, such as the Egyptian government's local but extreme "internet shutdown" in 2011 in order to stop the Tahrir revolution. The Snowden revelations also revealed that the NSA was indeed a global adversary, capable of capturing huge amounts of internet traffic and then using that signals intelligence to maintain the geo-political power of the United States government. There is thus the danger that humanity, due to our dependence on the Internet, falls under the centralized authority of the most powerful agencies of surveillance. If the Internet is turned into a method of control, either via behavioral advertising or more nefarious geopolitical behavioral manipulation, then we have entered into what Deleuze terms the "society of control" where "individuals have become 'dividuals,' and masses, samples, data, markets, or banks" as part of a larger collective intelligence whose purpose is no longer the realization of free individuals but "the progressive and dispersed installation of a new system of domination," a process we are seeing at the present moment in the "the crisis of the institutions" ranging from representative democracy to the United Nations Charter of Human Rights [71]. Decentralization may be our best answer to this threat. What is at stake in the NEXTLEAP project is that new decentralized architectures can give us access to capabilities based on trust that cannot be hijacked by a malicious adversary, and so combat this emergent data-driven society of control.

Snowden himself pointed to the answer in dialogue with Tim Berners-Lee, the inventor of the Web: "We need to encode our vales not just in writing but in the structure of the Internet, and it's something that I invite everyone around the world to join and participate in."[24] Values are inherently built, either by accident or intentionally, into technological protocols. Take for example the value of *episteomological equality* embedded in network neutrality: Of course it is not true that everyone knows the same things, but that with unfiltered access to the Internet, everyone should at least have the possibility of accessing the same knowledge when they need it [147]. Although the relationships between values, protocols, and concepts is still in formation, it is clear that by embedding fundamental - and possibly new kinds of - rights into the protocols of the Internet that the capabilities that the Internet provides can be guaranteed for future generations. If these protocols are designed in a decentralized, privacy-enhanced, and secure manner, then these rights should apply universally regardless of the particular repression of certain nation-states and other centralized actors.

Decentralization based on rights would revive the Enlightenment in a form suitable for the digital era, a new kind of "Digital Enlightenment" that preserves trust in the underlying Internet infrastructure for both the defeat of ignorance via access to knowledge and the expansion of autonomy via decentralized infrastructure. In this manner, radical complexity - the negentropy characteristic of life itself - can be empowered on the Internet. By virtue of creating these protocols to support decentralization, we can escape the dangers of a universal

[23] Taking from Simondon, Stiegler calls this process *collective transindividuation* [115]

[24] As stated by Snowden during his TED talk: `http://www.ted.com/talks/edward_snowden_here_s_how_we_take_back_the_internet/transcript`

form of Western reason embedded in a centralized "Silicon Valley" monoculture, but instead promote a radically heterogeneous diversity of collective intelligences with radically different (and non-Western) epistemologies, ontologies, and metaphysics that co-exist globally.

7.12 Open Philosophical Questions for Decentralization

There is still very much work to be done in order to unite the socio-technical work of NEXTLEAP with wider philosophical discourse in order to formulate a science of decentralization. In this regard, this work will continue throughout the lifetime of the project, taking shape both internally via the production of a book-length work to investigate these themes, as well as in dialogue with larger society across Europe and beyond in order to formulate a new kind of *material* constitution that can be embedded on the level of protocols. This will require an in-depth exploration of the assumptions of the fields of distributed systems, cryptography, mechanism design, sociology, STS, and other fields, all the time while putting forward the new cognitively-extended and technically-embedded individual as having rights that must be designed into the protocols themselves in order to maintain epistemic equality, political autonomy, and collective intelligence. In this regard, the below questions that should be posed in the course of the design by NEXTLEAP of new protocols:

- What are the new forms of collective intelligence that are engendered by decentralized systems, and how do these rely on assumptions of trust as embedded both in the technical design and social frameworks?
- How do technical systems produce new kinds of individuals, from a biological (neural) level to a social level, and how can these individuals have new kinds of rights over their technically-embedded capacities, and does decentralization help enforce these rights?
- How can decentralized systems preserve and increase both autonomy and knowledge from the scale of individuals to new kinds of global collective intelligence, while preventing mass surveillance from hijacking new forms of collective intelligence and using them to destroy the autonomy of individuals and groups?

There is reason to be hopeful. Beyond NEXTLEAP, the larger research community around cryptography, shocked by the Snowden revelations, has also been mobilizing its resources to take the problems of real users and surveillance seriously in their protocol design. Recently, the prominent cryptographer Philip Rogaway put forward a manifesto in which he put forward the task of realizing a *cryptographic commons* based on "popular services in a secure, distributed, and decentralized way, powered by free software and free/open hardware. We need to build systems beyond the reach of super-sized companies and spy agencies"[193]. By mobilizing the intellectual and critical resources of philosophy that can point to different futures outside of the current landscape of centralized services, we can help build not only a new philosophy of the Internet but a new social movement that demands such a decentralized cryptographic commons as essential to their fundamental rights in the digital era.

8 Conclusions

As has been shown, the interdisciplinary study of decentralization is still a largely unknown field. Although a more distributed Internet was shown by the Oxford Internet Institute as one of the possible future scenarios of the Internet (as exemplified by trends ranging from peer-to-peer systems to e-democracy), it currently appears that the future of the Internet is more towards "Big Brother." [25], as shown in Figure 6.[26] While Web Science hopes to understand the "macro-level" effects of changes in "micro-level" protocols[27], the Web itself is rapidly centralizing between a few Cloud-based platforms such as Facebook, Apple, and Google. However, the future is unwritten: With the rise of Bitcoin and Tor (often misunderstood as the"Dark Web"), there is a clear popular interest in a decentralized alternative, and the number of grass-roots programmers working to create decentralized systems is growing rapidly. Although Internet Science has drawn on network science to understand networks

[25] See the Oxford Internet Institute Study: "Towards a Future Internet: Interrelations between Technological, Social and Economic aspects " at http://cordis.europa.eu/fp7/ict/fire/docs/tafi-final-report_en.pdf.

[26] The picture is from http://www.slideshare.net/TechSoupEurope/fabrizio-sestini-collective-awareness-platforms

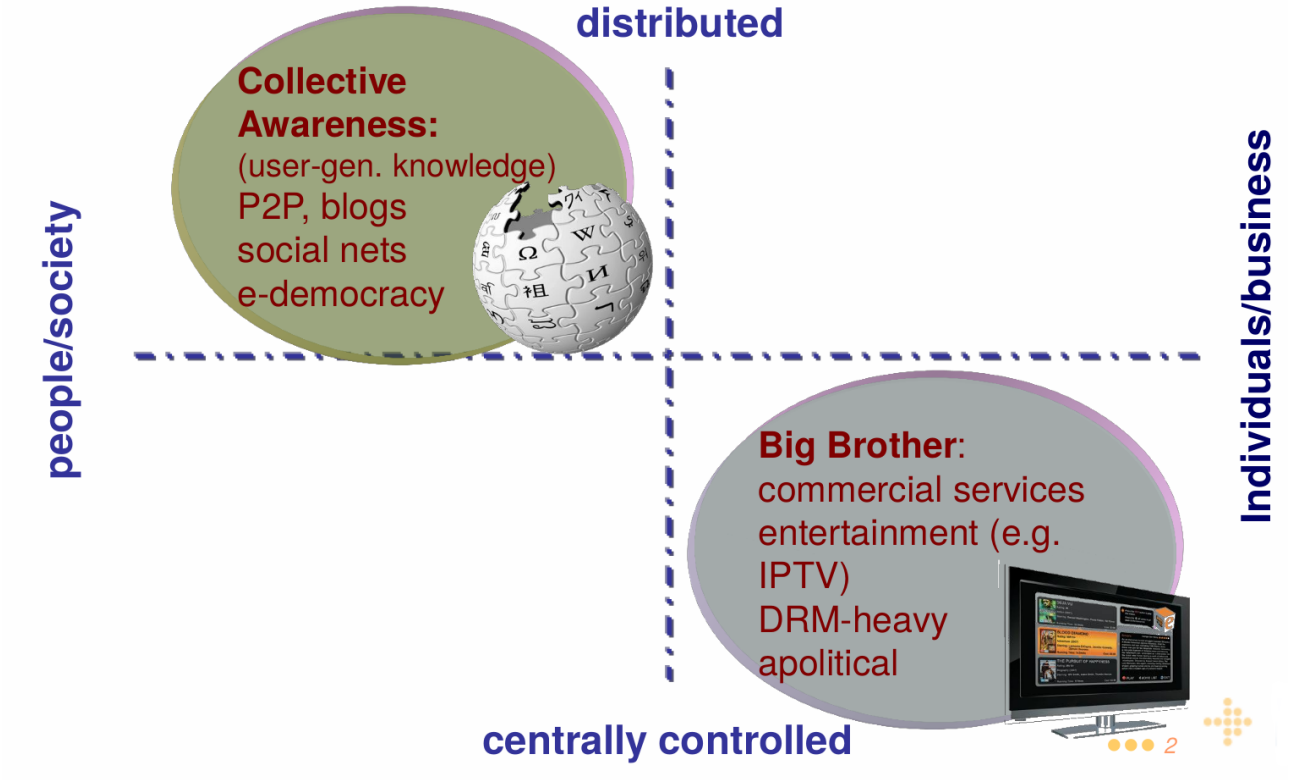

Figure 6: Future Scenarios for the development of the Internet

and completed various analyses of important technical issues ranging from net neutrality to the commons, there needs to be clear principles in order to both understand how decentralization powered the rise of the Internet and to create a more decentralized Internet in the future [229].

Yet the principles behind these decentralized systems are far from a science and we are currently not able to make most projects scale to offer genuine alternatives to centralized and privacy-invasive services. Furthermore, the current examples of decentralized systems suffer from their own faults. For example, Bitcoin maintains a high-integrity distributed ledger via mining, but at the cost of the privacy of all its users: The record of Bitcoin transactions is far more complete than the records of any bank, and so opens itself up to disturbing possibilities for new kinds of totalitarianism, such as a centrally-surveilled "social credit system" to replace cash under the pretense of eliminating illegal money laundering.[27]. Also, the mining process itself wastes electrical cycles[28] and transactions are far too inefficient to be used to replace current centrally-banked credit cards [64]. Likewise, Tor is the best privacy-enhancing technology available, but it does lead to performance costs. Also, as Tor maintains a list of all entry and exit nodes, it is increasingly possible to censor Tor in repressive environments. Lastly, due to the use of onion-routing in order to provide low latency web traffic, Tor does not offer perfect protection against a powerful global passive adversary, such as the NSA, that can monitor Tor entry and exit nodes [72].

There is much work to be done in designing practical systems, but our study showed that there is a vast space of possible decentralized systems, and only a few possible parts of the design space have been explored. With an understanding of distributed systems engineering, cryptographic protocols, and social science, one should be able to build a new generation of decentralized systems that maintains high privacy, availability, and integrity simultaneously. The key to building this new kind of science of decentralization is to understand the *actual* use of the systems, by investigating both their developers and users via techniques from social science and STS. By understanding the human needs, incentives, and values, we can help make sure that valuable resources in terms of technology are not wasted endlessly on academic systems, but can move forward the state of the art

[27] This system is already under development in China: `https://en.wikipedia.org/wiki/Social_Credit_System`

[28] Although currently not as many as one would think, as the cost of Bitcoin mining is currently estimated to be less than a single coal-burning power-plant: `https://bitcoinschool.gr/slides/session4.pdf`

so that we can build decentralized systems that solve real problems for the social good. As explored by Stiegler, given that more and more of our memory and even cognition is now deeply intertwined with the Internet, what is at stake in decentralization is far more than the future of the Internet, but also the fate of how we conceive of ourselves as humans, with rights and responsibilities, towards both ourselves and the larger social fabric.

Although based on the Internet, the science of decentralization could come "down to earth" to have vast impact outside of the Internet in helping decentralize all manner of existing centralized infrastructures from the electricity grid to government institutions. The essential problems of incentives for co-operation, dealing with adversarial behavior, and maintaining functionality in periods of crisis are all crucial not just for the Internet, but for wider collective self-organization. Although we have yet to uncover the general principles of decentralization and answers to the questioned outlined so far, decentralizing the Internet may be the first, but far from the last step.

9 Interdisciplinary Vocabulary

This interdisciplinary vocabulary is just the "starting point" for the research, a "common baseline" that lets researchers from different disciplines understand each other in order to create a unified Internet Science of decentralization. As we continue the NEXTLEAP project, we expect to change our shared understanding of the voabulary terms, and a common up-to-date version will be kept online[29] for further development.

access control A security control ensuring that only authorised parties may perform actions, such as reading or writing resources.

accountability The property by which a misbehaving entity may be detected, and held to account or punished for its actions.

active disruption An attack that involves actively injecting malformed or other malicious information to violate the security properties of a system.

adversarial behaviour Behaviour, either active or passive, that aims to violate the security properties of the system.

adversary An entity that aims to violate the security properties or interrupt the operation of the system.

analog A system that processes input signals as continuously variable quantities.

anonymity The property by which an entity or action cannot be linked to a long-term name or identifier.

anonymous channel A channel that ensures the sender or receiver of messages, or the initiator or server of the communication remains anonymous.

anonymous messaging A messaging system that offers senders or receivers of messages anonymity.

anonymous publishing A system allowing publishers of resources to remain anonymous, or readers of such material to remain anonymous.

application lifecycle The full set of activities from design through to development, testing, deployment, configuration, maintenance and decommissioning of software.

application platform A software system offering facilities for writing higher level application software. For example, an operating system, a browser or a generic web-server.

architecture The manner in which different software design elements are combined and connected to engineer a larger system.

attack An activity that aims to violate the security properties or availability of a system.

attack surface All the components that the adversary may access and influence, that could lead to successfully attacking the system.

auditor An entity that reviews the actions of another entity to ensure it has performed its operations correctly.

autopoiesis A system that maintains its own structure (Maturana).

authority An entity that may take actions independently of, and un-coerced by, other entities.

availability A security property that ensures the system can provide functionality despite the actions of an adversary.

[29] https://github.com/nextleap-project/nextleap/wiki/SharedVocabulary

backbone The part of the wide-area network that connects disparate networks to provide long range communications.

blacklist The activity of registering misbehaving identities and ensuring they are barred from using the system in the future.

blockchain A high-integrity append-only datastructure on which Bitcoin is based.

broadcast Sending of a message to all other parties in the network.

byzantine fault tolerance The ability of a distributed system to maintain consistency despite adversarial entities.

capabilities the potential for a being to act in the world. Also called "capacities" or "affordances."

censorship resistance The security property that guarantees that material may be published and accessed despite the actions of an adversary, behaving as a censor, attempting to block or alter it.

central clock A common reference for time or ordering that may assist in building consistent distributed systems.

centralised A system that relies on a single authority or single component to offer its properties.

centralised directory A central service offering a list and mapping between names and their properties, such as addresses and keys.

certificate authority A trusted entity that certifies the mapping between names and public keys, in the form of a certificate, to facilitate authentic secure communications.

churn The phenomenon in decentralized networks by which nodes constantly come on- and off-line.

circumvention mechanism A security mechanism allowing communication across attempts to block it, for example by a national firewall.

claim A signed statement by an authority attesting that an entity has an attribute.

client The software agent used by a user.

client-server architecture The common Internet service architecture by which a user client connects to a service provider.

cloud computing A distributed, but not decentralized, service architecture based on running Internet services in large data centers.

code The language in which software is written.

coercion An attack by which an adversary forces an otherwise honest party to collude into violating some security assumption.

cognition A process that refers to memory, language, or attention

collective intelligence The intelligence exhibited by a system composed of multiple distinct entities, where the system is autopoietic. (Halpin)

command and control The mechanism by which a system or network is controlled, usually centrally.

component failure A single technical component or entity behaving arbitrarily, but usually not maliciously.

compromised An entity that is entirely under the observation and control of the adversary.

confidentiality The family of security properties relating to keeping information secret from adversaries.

cooperation Every entity is expected to follow the same rules in the system depending on its roles. In the early internet this was feasible since anyone working on the Internet shared the same motivation: to maximize efficiency and optimize the system technologically to build a reliable, efficient, and powerful network, although it may not be the case today or in the future.

cooperative An entity that takes actions that benefit the system as a whole, as opposed to operating in a selfish manner.

corrupt insider An entity with some legitimate authority within the system, that is under the control of the adversary.

cover traffic Network traffic that is used as part of a security mechanism to obscure the meta-data of genuine traffic.

covertness The security property involving obscuring that a user's actions are taking place.

CPU cycle The unit of computation on modern central processing units.

cryptographic proof A piece of information generated by a prover to convince a verifier of a statement.

cryptographic protocol A directive for a sequence of messages exchanged by two or more parties that are part of a cryptographic protection mechanism achieving specific security properties.

cryptography The mathematical discipline dealing with building techniques that protect secrecy and integrity.

darknet An overlay network that is somehow hidden from view and can be accessed using specialized software. In some social and popular parlance, this is confused with any "illegal" activity on the Internet, although the activity on darknets may be legal

data collection The set of operations that data is subject to in a system, including the visibility of the data for each user. See "privacy." In Europe, enforced by Data Protection rules.

decentralized A distributed system involving multiple entities with separate authorities. This kind of architecture may not only apply to technical systems but an entire class of phenomena ranging from the biological to the social systems, and how they are intertwined with technical architectures, including issues of governance, management, cooperation, not as separate, juxtaposed phenomena, but as they are embedded in the architecture itself.

denial-of-service An attack that attempts to degrade the availability properties of a system.

deployments The actual use of a software system by users, as compared to its specification, design or engineering.

dev-ops The discipline that combines the development of software with aspects of its operation such as deployment, configuration management and monitoring.

device independence The property of a service that allows its users to seamlessly use it from multiple different and new devices.

digital A system that processes digital signals generated by digital modulation. Electronic devices such as computers and mobile phones are digital systems.

digital studies The study of the digital in the widest sense, not just with a focus on the humanities as in "digital humanities."

differential privacy A security property of the system ensuring that decisions and information are not overly dependent on single user records, therefore protecting their privacy.

distributed A property of a technical system by which multiple hardware elements are combined through networking to build a larger system.

distributed hash table A peer-to-peer system that assigns peers fixed addressing identifiers in such a way that efficient routing is achieved.

distributed ledger A distributed system that provides a high-integrity ledger.

diversity A feature of a network containing elements with different capabilities.

ecological diversity A security mechanism using different software and hardware components to reduce correlated failures.

efficiency The effective use of resources towards achieving an engineering goal, without waste.

encrypted flow A bidirectional sequence of messages protected through cryptographic techniques.

encryption A cryptographic security mechanism that achieves confidentiality.

end-point The receiver of a message or a sender. Often neglected in security models, it may be the user's client or computer.

energy efficient A system that can operate under strict energy constraints.

entity A discrete part of a system that can be functionally separated.

entropy A measurement of randomeness. A system that is completely random has maximum entropy. Necessary for key generation in terms of encryption.

ephemeral key A cryptographic key that is only used for a short window of time, and securely deleted afterwards.

epistemology The study of what can be known. This is usually consider smaller than what exists, i.e. ontology.

everyday engineering Underlines the need to understand “how things are done” in daily engineering practice: the negotiation work and organizational politics subtending engineering, i.e. how the creation process by engineers exists in close relation to the social, and how design decisions are more often than not based, in addition to technical data, on other dynamics (Dominique Vinck)

extended mind The theory that cognition can be extended into the world and outside the barriers of an individual like the brain or skin. (Andy Clark)

federated A system that is composed of interconnected providers serving users.
fork A split in a software project or other common infrastructure, often led by a fork in the community itself.
freshness A property of keys or nonces, which ensures that they have not been replayed from past information.

global passive adversary An adversary that may observe all messages in the network.
gossip A routing protocol by which messages are passed on to neighbouring nodes without any directed routing.

governance The set of decision-making processes, the ensemble of procedures that frame the choices subtending the organizational design of systems, including technical, legal and value-sharing choices. This includes how governance of the system is created/maintained and how the system copes with crisis.
group key agreement A cryptographic protocol that leads participants to sharing a secret key.
group secure communications A cryptographic protocol that allows participants to exchange protected messages.
group signature An unforgeable signature that does not divulge who, out of a defined set, was the signer.
hardening Engineering a system to resist certain classes of attacks.
heterogeneity see 'diversity'.
high-availability A property of the system that ensures minimal down time.
incentive A reason for an entity to behave in a certain, usually desirable, fashion.
inconsistency The state of a system in which a contradiction exists in the information considered authoritative by one or more nodes.
indexing Algorithms for processing data to allow for efficient search.
individuation The process by which a being becomes an individual with capacities (Simondon)
information dispersion code A technique allowing information to be split into smaller fragments and reconstructed through a subset of them.
infrastructure A system that is used by others to provide one or more services necessary for higher level applications.
integrity The property by which a system state is not affected by the adversary.
IP address space The space of names for machines interconnected through the Internet.
load balancing The process by which incoming requests are distributed across different machines to avoid any of them being overloaded.
key A number used in encryption and decryption. If private, it should be kept secret and should be randomly generated from a high entropy source.
locality The practice of keeping information or processing close to each other or the users.
location-based service A service the customises its outputs by the location of the user.
low-latency A property of systems with human-unnoticeable delays when sending a message to its recipient.
malicious insider An entity that has some legitimate authority in the system, but is also controlled by the adversary.
mass surveillance An attack that involves the mass and indiscriminate collection and possibly processing of data.
mechanism design The economic discipline that creates systems in which honest parties have incentives to behave truthfully and cooperatively.
mesh network A network in which nodes are connected to each other physically to allow for wide-area routing.
meta-data All data about a communication that are not its content.
metaphysics The fundamental assumptions around time and space that shape possible ontologies.
middle box A network element that transparently processes flows of traffic.
mix system A security mechanism that offers communication anonymity.
mobile A network user that physically moves.
mobile code Software that is delivered dynamically across the network.
national firewall A network element, usually placed around the inter-networks of a national state, that allows it to control and block access to parts of the outside network.

negentropy The process that characterizes life as it struggles against the energy dissipation and disorganization that results (SchrÃűdinger). The concept can be generalized to describe anything that tends to create the difference, choice or new, in a system developing in the direction of self-preservation or an improvement (Stiegler)

node A peer or entity in the network.

node enumeration An attack by which the attacker learns all other participants in the system.

non-colluding An entity that does not collaborate with others to violate security properties.

onion routing A security mechanism delivering communication anonymity for interactive streams of traffic.

ontology The study of being, i.e. "what exists."

organology The study of all artifices (tools, machines, prosthetics, recording and communication devices) and their interrelation.

open system A system that anyone may join.

out-of-band communication A message that is transmitted outside the system considered.

outsourced computation A computation that is performed on behalf of the user by a remote service.

overlay network A network that uses another network for basic communications.

passive collection An attack technique involving only collection of information.

peer see ‘node’

peer discovery A mechanism by which peers may discover other peers of interest to them.

pharmacology From the Greek word meaning both poison and medicine, means something that is simultaneously both positive and negative) and so must inform the politics and ethics of care within a larger historical context (Stiegler).

peer-to-peer A network in which all nodes are equal and may perform all functions.

phenomenology Subjective experience that cannot be measured easily by science, such as the feeling of "being there."

platform insecurity The issue that end user computing devices may be vulnerable to attacks.

plausible deniability The security property that ensures users of a system can deny allegations of having specific knowledge or having acted in a certain way.

poisoning An attack by which the adversary injects false information about a system state, for example into honest parties’ routing tables.

principal see entity

privacy The possibility for each user to know and master which operations involving his/her data is collected by third parties, and the balances of power and control that take shape as a result.

privacy system A system that supports one or more privacy properties.

private information retrieval A security mechanism that allows for querying records from a database without disclosing which record to anyone.

provider An entity within a possibly federated system that serves users.

proxy A network relay, possibly obscuring who is talking with whom.

pseudonymity The security property of associating another name to users that is stable over time for a system, yet conceals their real identity.

real-time A system that guarantees that certain properties will hold by a certain deadline.

reference monitor The security component that is entrusted to decide and enforce access control.

reputation The deeds of an entity that make it more or less trustworthy to others.

resilience The property of operating despite failures and attack.

revocable The ability to uncover the identity of an otherwise anonymous party.

rights Equal access to capabilities given by an institutional framework

root of trust The entity that is entrusted by all others.

routing The process by which messages are routed in a wide-area network to their ultimate destination.

routing decision The process by which a router decides where to send a message that is being routed.

routing table The information necessary to make routing decisions.

scalability The property of a system to handle more load as more machines are devoted to the task.

secure deletion The security property that ensures deleted information may not be recovered.

secure multi-party computation A security mechanism that allows for a computation to be executed privately over multiple entities.

security policy The statement of the properties that must hold in the secure system despite the attempts of a motivated strategic adversary to subvert them.

selfish A node that chooses between valid options to maximize their return with no regard for the welfare of the network.

sensor network A mesh network of sensor nodes.

server A machine that runs a service and makes it available to users / clients.

service A computer software on a remote system that users may use.

share A piece of information that along with others may be used to reconstruct a secret.

smart contract A contract that is encoded in a computer language and triggers automatically when certain conditions are fulfilled.

social graph A graph of users and the relations between them.

social link A connection between two users that denotes a relationship of some kind.

software update A modification to software that fixes certain bugs or attacks, or adds new features.

structured peer-to-peer see 'distributed hash table'.

super-node A peer that is entrusted with performing a wider function than other peers or has many more connections.

sybil attack An attack by which an adversary tries to build multiple identities they control.

systemic failure A failure that is due to the fundamental way in which the system was put together.

telemetry Data sent back by an application with analytics of its actual behavior.

threshold cryptography Cryptographic techniques involving multiple parties, and that can tolerate a fraction of parties being corrupt.

tit-for-tat A strategy by which users reflect each other's positive actions and punish deviation.

toolchain A set of tools that facilitate the process of software creation.

traces Marks left in the world that can be detected. Often the term "digital traces" is used for data left by users.

traffic analysis The disciplines of extracting information out of communications meta-data.

transindividuation How the process of individuation can be effected by the larger society and technical artefacts.

transparent log A security system that guarantees all parties observe the same high-integrity data.

trust The construction of shared meanings among the actors concerned by the use of a specific system – shared meanings on which they rely for subsequent operations on and by means of the technology.

trusted Technically, a component that, if controlled by the adversary, may violate the security properties of the system. In a general sense, a component whose behavior is predictable or expected according to shared meanings.

trusted party An entity that is trusted.

unobservability The security property ensuring that adversaries cannot determine whether an action has, or has not, taken place.

untrusted entity An entity, potentially centralized, that offers a service to others but is however not trusted, i.e. could fail without affecting the security properties of the system.

values in design The core hypothesis that architecture and design features may be systematically related to political, social, ethical values, such as security, privacy, and freedom. The goal of a VID approach is to identify, define and analyze these relationships, and in parallel, point out the ways in which law and policy normative systems interact with material technologies. This entails looking at values "from the ground up" - observing how they become embodied in artefacts. (Helen Nissenbaum)

verified protocol A protocol that has a proof or other formal argument of security associated with it.

x.509 certificate A format in which certificate authorities package their claims about name to key bindings.

zero-knowledge proof A cryptographic proof that makes assertions on secret values without revealing them.

References

[1] B. Adida. Helios: Web-based open-audit voting. In *17th USENIX Security Symposium*, pages 335–348, 2008.

[2] P. Agre. Peer-to-peer and the promise of internet equality. *Communications of the ACM*, 46(2):39–42, 2003.

[3] F. Aidouni, M. Latapy, and C. Magnien. Ten weeks in the life of an edonkey server. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1–5, 2009.

[4] P. Aigrain. Another narrative. addressing research challenges and other open issues session. In *PARADISO Conference, 7-9 September 2011*, Brussels, 2011.

[5] M. Akhoondi, C. Yu, and H. V. Madhyastha. LASTor: A low-latency as-aware tor client. In *IEEE Symposium on Security and Privacy (S&P)*, pages 476–490. IEEE Computer Society, 2012.

[6] D. Alhadidi, N. Mohammed, B. C. M. Fung, and M. Debbabi. Secure distributed framework for achieving ε-differential privacy. In S. Fischer-Hübner and M. K. Wright, editors, *12th Privacy Enhancing Technologies Symposium (PETS)*, volume 7384 of *LNCS*, pages 120–139. Springer, 2012.

[7] M. AlSabah, K. S. Bauer, and I. Goldberg. Enhancing Tor's performance using real-time traffic classification. In T. Yu, G. Danezis, and V. D. Gligor, editors, *19th ACM Conference on Computer and Communications Security (CCS'12)*, pages 73–84. ACM, 2012.

[8] M. AlSabah, K. S. Bauer, I. Goldberg, D. Grunwald, D. McCoy, S. Savage, and G. M. Voelker. Defenestrator: Throwing out windows in tor. In S. Fischer-Hübner and N. Hopper, editors, *11th Privacy Enhancing Technologies Symposium (PETS)*, volume 6794 of *LNCS*, pages 134–154. Springer, 2011.

[9] R. Anderson. The Eternity service. In *Pragocrypt*, 1996.

[10] P. S. Andre. IETF RFC 6120 Extensible Messaging and Presence Protocol (xmpp): Core. `https://www.ietf.org/rfc/rfc6120.txt`, 2011. Last accessed: June 30th 2016.

[11] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek. Secure multiparty computations on bitcoin. In *IEEE Symposium on Security and Privacy (S&P)*, pages 443–458. IEEE Computer Society, 2014.

[12] M. S. Artigas and P. G. López. On routing in distributed hash tables: Is reputation a shelter from malicious behavior and churn? In H. Schulzrinne, K. Aberer, and A. Datta, editors, *9th International Conference on Peer-to-Peer Computing (P2P)*, pages 31–40. IEEE, 2009.

[13] O. Auber. Le net, un bien commun: Quel projet politique pour le reseau? `http://hyperrepublique.blogs.com/public/2007/03/quel_projet_pol.html`. Last accessed: June 30th 2016.

[14] N. Auray. Information communities and open governance: Boundaries, statuses and conflicts. In E. Brousseau, M. Marzouki, and C. Meadel, editors, *Governance, Regulations and Powers on the Internet*. Cambridge University Press, Cambridge, 2012.

[15] F. Baccelli. Internet : modeliser le trafic pour mieux le gerer. `http://interstices.info/jcms/c_12842/internet-modeliser-le-trafic-pour-mieux-le-gerer`. Last accessed: June 30th 2016.

[16] M. Backes, A. Kate, S. Meiser, and E. Mohammadi. (nothing else) MATor(s): Monitoring the anonymity of tor's path selection. In G. Ahn, M. Yung, and N. Li, editors, *21st ACM Conference on Computer and Communications Security*, pages 513–524. ACM, 2014.

[17] B. Barak, R. Canetti, Y. Lindell, R. Pass, and T. Rabin. Secure computation without authentication. In *Annual International Cryptology Conference*, pages 361–377. Springer, 2005.

[18] P. Baran et al. On distributed communications. *Volumes I-XI, RAND Corporation Research Documents, August*, pages 637–648, 1964.

[19] K. S. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. C. Sicker. Low-resource routing attacks against tor. In P. Ning and T. Yu, editors, *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 11–20. ACM, 2007.

[20] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In *IEEE Symposium on Security and Privacy (S&P)*, pages 81–95. IEEE Computer Society, 2005.

[21] M. Bauwens. P2p and human evolution: Placing peer to peer theory in an integral framework. http://www.integralworld.net/bauwens2.html. Last accessed: June 30th 2016.

[22] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *29th Annual International Cryptology Conference Advances in Cryptology (CRYPTO)*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009.

[23] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, A. Lysyanskaya, and E. Rachlin. Making P2P accountable without losing privacy. In P. Ning and T. Yu, editors, *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 31–40. ACM, 2007.

[24] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE Symposium on Security and Privacy (S&P)*, pages 459–474. IEEE Computer Society, 2014.

[25] Y. Benkler. Sharing nicely: On shareable goods and the emergence of sharing as a modality of economic production. *The Yale Law Journal*, 114 (2):273–358, 2004.

[26] A. Bergkvist, D. Burnett, C. Jennings, and A. Arayanan. W3c webRTC 1.0: Real-time communication between browsers. https://www.w3.org/TR/webrtc/, 2016. Last accessed: June 30th 2016.

[27] T. Berners-Lee, W. Hall, J. Hendler, and D. J. Weitzner. Creating a science of the web. *Science*, 313(5788):769–771, 2006.

[28] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang. The growth of diaspora-a decentralized online social network in the wild. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 13–18. IEEE, 2012.

[29] A. Biryukov, D. Khovratovich, and I. Pustogarov. Deanonymisation of clients in bitcoin P2P network. In G. Ahn, M. Yung, and N. Li, editors, *21st ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 15–29. ACM, 2014.

[30] A. Biryukov, I. Pustogarov, and R. Weinmann. Trawling for tor hidden services: Detection, measurement, deanonymization. In *IEEE Symposium on Security and Privacy, (S&P)*, pages 80–94. IEEE Computer Society, 2013.

[31] J. Biskup and U. Flegel. Threshold-based identity recovery for privacy enhanced applications. In D. Gritzalis, S. Jajodia, and P. Samarati, editors, *7th ACM Conference on Computer and Communications Security*, pages 71–79. ACM, 2000.

[32] M. Blaze, J. Feigenbaum, and A. D. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In B. Christianson, B. Crispo, W. S. Harbison, and M. Roe, editors, *6th International Workshop on Security Protocols*, volume 1550 of *LNCS*, pages 59–63. Springer, 1998.

[33] D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008*, pages 192–206. Springer, 2008.

[34] N. Borisov, G. Danezis, and I. Goldberg. DP5: A private presence service. *PoPETs*, 2015(2):4–24, 2015.

[35] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz. Denial of service or denial of security? In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 92–102. ACM, 2007.

[36] D. Boyd. Friendster and publicly articulated social networks. In *Conference on Human Factors and Computing Systems, April 24-29, 2004*, Vienna, 2004. Association for Computing Machinery.

[37] S. Braman. Designing for instability: Internet architecture and constant change. In *Media In Transition 7 (MIT7) Unstable Platforms: the Promise and Peril of Transition, May 13-15, 2011*, Cambridge, MA, 2011.

[38] S. Buchegger, D. Schiöberg, L. Vu, and A. Datta. Peerson: P2P social networking: early experiences and insights. In T. Stein and M. Cha, editors, *2nd ACM EuroSys Workshop on Social Network Systems (SNS)*, pages 46–52. ACM, 2009.

[39] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. IETF RFC 4880 open pgp message format. https://www.ietf.org/rfc/rfc4880.txt, 2007. Last accessed: June 30th 2016.

[40] M. Callon. Society in the making: The study of technology as a tool for sociological analysis. In W. Bijker, T. Hughes, and T. Pinch, editors, *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*, pages 83–103. The MIT Press, Cambridge, MA and London, 1987.

[41] M. Callon and B. Latour, editors. *La science telle qu'elle se fait*. La Decouverte, Paris, 1990.

[42] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *13th ACM Conference on Computer and Communications Security (CCS)*, pages 201–210. ACM, 2006.

[43] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *International Conference on the Theory and Application of Cryptographic Techniques Advances in Cryptology (EUROCRYPT)*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

[44] J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. In *2007 IEEE Symposium on Security and Privacy (S&P)*, pages 101–115. IEEE Computer Society, 2007.

[45] D. Cardon. Le design de la visibilite. un essai de cartographie du web 2.0. *Reseaux*, 6:93–137, 2008.

[46] N. Carr. *The shallows: What the Internet is doing to our brains*. WW Norton & Company, 2010.

[47] M. Castells. Toward a sociology of the networked society. *Contemporary Sociology*, 29 (5):693–699, 2000.

[48] M. Castro, P. Druschel, A. J. Ganesh, A. I. T. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. In D. E. Culler and P. Druschel, editors, *5th Symposium on Operating System Design and Implementation (OSDI)*. USENIX Association, 2002.

[49] H. Chan and A. Perrig. Efficient security primitives derived from a secure aggregation algorithm. In P. Ning, P. F. Syverson, and S. Jha, editors, *15th ACM Conference on Computer and Communications Security (CCS)*, pages 521–534. ACM, 2008.

[50] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[51] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[52] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology (EUROCRYPT)*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.

[53] Y. Chen, R. Sion, and B. Carbunar. Xpay: practical anonymous payments for tor routing and other networked services. In E. Al-Shaer and S. Paraboschi, editors, *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 41–50. ACM, 2009.

[54] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[55] J. Claessens, C. Díaz, C. Goemans, J. Dumortier, B. Preneel, and J. Vandewalle. Revocable anonymous access to the internet? *Internet Research*, 13(4):242–258, 2003.

[56] A. Clark and D. Chalmers. The extended mind. *Analysis*, 58(1):7–19, 1998.

[57] S. Clark, T. Goodspeed, P. Metzger, Z. Wasserman, K. Xu, and M. Blaze. Why (special agent) Johnny (still) can't encrypt: A security analysis of the APCO project 25 two-way radio system. In *20th USENIX Security Symposium*. USENIX Association, 2011.

[58] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In H. Federrath, editor, *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66. Springer, 2000.

[59] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: An anonymous messaging system handling millions of users. In *2015 IEEE Symposium on Security and Privacy (S&P)*, pages 321–338. IEEE Computer Society, 2015.

[60] J. Cranshaw and A. Kittur. The polymath project: lessons from a successful online collaboration in mathematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011.

[61] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In V. Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS)*, pages 207–216. ACM, 2002.

[62] G. Danezis, C. Díaz, C. Troncoso, and B. Laurie. Drac: An architecture for anonymous low-volume communications. In M. J. Atallah and N. J. Hopper, editors, *10th Privacy Enhancing Technologies Symposium (PETS)*, pages 202–219. Springer, 2010.

[63] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 2–15. IEEE, 2003.

[64] G. Danezis and S. Meiklejohn. Centrally banked cryptocurrencies. *arXiv preprint arXiv:1505.06895*, 2015.

[65] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2009.

[66] G. Danezis and P. F. Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. In N. Borisov and I. Goldberg, editors, *8th Privacy Enhancing Technologies Symposium (PETS)*, volume 5134 of *LNCS*, pages 151–166. Springer, 2008.

[67] N. Daswani and H. Garcia-Molina. Query-flood dos attacks in Gnutella. In V. Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS)*, pages 181–192. ACM, 2002.

[68] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[69] C. Decker, R. Eidenbenz, and R. Wattenhofer. Exploring and improving BitTorrent topologies. In *13th IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2013.

[70] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013, Trento, Italy, September 9-11, 2013, Proceedings*, pages 1–10, 2013.

[71] G. Deleuze. Society of control. *L'autre journal*, 1, 1990.

[72] C. Díaz, S. J. Murdoch, and C. Troncoso. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In M. J. Atallah and N. J. Hopper, editors, *10th Privacy Enhancing Technologies Symposium (PETS)*, volume 6205 of *LNCS*, pages 184–201. Springer, 2010.

[73] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun. Andana: Anonymous named data networking application. In *19th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2012.

[74] R. Dingledine and N. Mathewson. Anonymity loves company: Usability and the network effect. In *5th Annual Workshop on the Economics of Information Security (WEIS)*, 2006.

[75] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In M. Blaze, editor, *13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[76] Dot-BIT project: A decentralized, open dns system based on the bitcoin technology. `https://bit.namecoin.info/`. Last accessed: June 30th 2016.

[77] J. R. Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[78] J. R. Douceur. The sybil attack. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, *1st International Worksop on Peer-to-Peer Systems (IPTPS)*, volume 2429 of *LNCS*, pages 251–260. Springer, 2002.

[79] Y. Duan, N. Youdao, J. Canny, and J. Z. Zhan. P4P: practical large-scale privacy-preserving distributed computation robust against malicious users. In *19th USENIX Security Symposium*, pages 207–222. USENIX Association, 2010.

[80] M. Dulong de Rosnay. *Les golems du numerique. Droit d'auteur et Lex Electronica*. Presses des mines, Paris, 2016.

[81] M. Dulong de Rosnay and F. Musiani. Towards a (de)centralization-based typology of peer production. *TripleC : communication, capitalism & critique*, 14 (1):189–207, 2016.

[82] M. Edman and P. F. Syverson. AS-awareness in tor path selection. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security (CCS)*, pages 380–389, 2009.

[83] T. Elahi, G. Danezis, and I. Goldberg. PrivEx: Private collection of traffic statistics for anonymous communication networks. In G. Ahn, M. Yung, and N. Li, editors, *21st ACM Conference on Computer and Communications Security*, pages 1068–1079. ACM, 2014.

[84] N. Elkin-Koren. It's all about control: Rethinking copyright in the new information landscape. In N. Elkin-Koren and N. W. Netanel, editors, *The Commodification of Information*, pages 415–431. Kluwer Law International, The Hague, Netherlands, 2002.

[85] N. Elkin-Koren. Making technology visible: Liability of internet service providers for peer-to-peer traffic. *New York University Journal of Legislation & Public Policy*, 9 (15):15–76, 2006.

[86] C. M. Ellison. Establishing identity without certification authorities. In *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography - Volume 6*, SSYM'96, pages 7–7. USENIX Association, 1996.

[87] D. C. Engelbart. Augmenting human intellect: A conceptual framework. summary report afosr-3223 under contract af 49 (638)-1024, sri project 3578 for air force office of scientific research. *Stanford Research Institute. Retrieved March*, 1:2007, 1962.

[88] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In V. Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS)*, pages 41–47. ACM, 2002.

[89] I. Eyal. The miner's dilemma. In *IEEE Symposium on Security and Privacy (S&P)*, pages 89–103. IEEE Computer Society, 2015.

[90] A. J. Feldman, A. Blankstein, M. J. Freedman, and E. W. Felten. Social networking with frientegrity: Privacy and integrity with an untrusted provider. In *Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012*, pages 647–662, 2012.

[91] M. Feldotto, C. Scheideler, and K. Graffi. Hskip+: A self-stabilizing overlay network for nodes with heterogeneous bandwidths. In *14th IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2014.

[92] C. Fournet, F. Le Fessant, L. Maranget, and A. Schmitt. Jocaml: A language for concurrent distributed and mobile programming. In *Advanced Functional Programming*, pages 129–158. Springer, 2003.

[93] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 193–206. ACM, 2002.

[94] J. Freudiger, M. H. Manshaei, J. Hubaux, and D. C. Parkes. On non-cooperative location privacy: a game-theoretic analysis. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security (CCS)*, pages 324–337. ACM, 2009.

[95] M. Fuller, editor. *Software Studies: A Lexicon*. The MIT Press, Cambridge, MA, 2008.

[96] C. Garman, M. Green, and I. Miers. Decentralized anonymous credentials. In *21st Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2014.

[97] U. Gasser and S. Ernst. European Union Copyright Directive best practice guide: Implementing the EU copyright directive in the digital age. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=952561`. Last accessed: June 30th 2016.

[98] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy. Vanish: Increasing data privacy with self-destructing data. In *18th USENIX Security Symposium*, pages 299–316, 2009.

[99] Gnutella: File sharing and distribution network. `http://rfc-gnutella.sourceforge.net/`. Last accessed: June 30th 2016.

[100] I. Goldberg. Improving the robustness of private information retrieval. In *IEEE Symposium on Security and Privacy (S&P)*, pages 131–148. IEEE Computer Society, 2007.

[101] I. Goldberg, B. Ustaoglu, M. V. Gundy, and H. Chen. Multi-party off-the-record messaging. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security*, pages 358–368. ACM, 2009.

[102] D. A. Gritzalis. *Secure electronic voting*, volume 7. Springer Science & Business Media, 2012.

[103] D. Hales. Emergent group-level selection in a peer-to-peer network. *Complexus*, 3:108–118, 2006.

[104] D. Hales, S. Artecon, A. Marcozzi, and I. Chao. Towards a group selection design pattern. In F. Meyer, editor, *The European Integrated Project Dynamically Evolving, Large Scale Information Systems (DELIS) Proceedings of the final workshop*. 2008.

[105] H. Halpin. Does the web extend the mind? In *Proceedings of the 5th annual ACM web science conference*, pages 139–147. ACM, 2013.

[106] H. Halpin and F. Bria. Crowdmapping digital social innovation with linked data. In *European Semantic Web Conference*, pages 606–620. Springer, 2015.

[107] H. Halpin and A. Monnin. *Philosophical Engineering: Toward a Philosophy of the Web*. John Wiley & Sons, 2013.

[108] D. Hardt. IETF RFC 6749 the OAuth 2.0 Authorization Framework. `https://www.ietf.org/rfc/rfc6749.txt`, August 1982. Last accessed: June 30th 2016.

[109] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In J. Jung and T. Holz, editors, *24th USENIX Security Symposium*, pages 129–144. USENIX Association, 2015.

[110] K. Hellekson and K. Busse, editors. *Fan Fiction and Fan Communities in the Age of the Internet*. McFarland, Jefferson, NC, 2006.

[111] M. Herrmann and C. Grothoff. Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using I2P. In S. Fischer-Hübner and N. Hopper, editors, *Privacy Enhancing Technologies (PETS)*, volume 6794 of *LNCS*, pages 155–174. Springer, 2011.

[112] K. J. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1), 2009.

[113] S. Hohenberger, S. Myers, R. Pass, and A. Shelat. ANONIZE: A large-scale anonymous survey system. In *IEEE Symposium on Security and Privacy (S&P)*, pages 375–389. IEEE Computer Society, 2014.

[114] A. Houmansadr, C. Brubaker, and V. Shmatikov. The parrot is dead: Observing unobservable network communications. In *IEEE Symposium on Security and Privacy (S&P)*, pages 65–79. IEEE Computer Society, 2013.

[115] Y. Hui and H. Halpin. Collective individuation: the future of the social web. *The Unlike Us Reader*, pages 103–116, 2013.

[116] I2P: The invisible internet project. `https://geti2p.net/en/`. Last accessed: June 30th 2016.

[117] R. Janakiraman, M. Waldvogel, and Q. Zhang. Indra: A peer-to-peer approach to network intrusion detection and prevention. In *12th IEEE International Workshops on Enabling Technologies (WETICE)*, pages 226–231. IEEE Computer Society, 2003.

[118] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann. The sniper attack: Anonymously deanonymizing and disabling the tor network. In *21st Annual Network and Distributed System Security Symposium(NDSS)*. The Internet Society, 2014.

[119] A. Johnson, P. F. Syverson, R. Dingledine, and N. Mathewson. Trust-based anonymous communication: adversary models and routing algorithms. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *18th ACM Conference on Computer and Communications Security (CCS)*, pages 175–186. ACM, 2011.

[120] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson. Users get routed: Traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 337–348, New York, NY, USA, 2013. ACM.

[121] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. F. Syverson. Users get routed: traffic correlation on tor by realistic adversaries. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *20th ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 337–348. ACM, 2013.

[122] P. C. Johnson, A. Kapadia, P. P. Tsang, and S. W. Smith. Nymble: Anonymous ip-address blocking. In N. Borisov and P. Golle, editors, *7th Privacy Enhancing Technologies Symposium (PETS)*, volume 4776 of *LNCS*, pages 113–133. Springer, 2007.

[123] J. Juen, A. Johnson, A. Das, N. Borisov, and M. Caesar. Defending tor from network adversaries: A case study of network path prediction. *PoPETs*, 2015(2):171–187, 2015.

[124] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure file sharing on untrusted storage. In J. Chase, editor, *Conference on File and Storage Technologies*. USENIX, 2003.

[125] A. Kapadia and N. Triandopoulos. Halo: High-assurance locate for distributed hash tables. In *Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2008.

[126] G. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. In T. Yu, G. Danezis, and V. D. Gligor, editors, *19th ACM Conference on Computer and Communications Security (CCS)*, pages 906–917. ACM, 2012.

[127] M. Kirschenbaum. Virtuality and VRML: Software studies after Manovich. `http://www.electronicbookreview.com/thread/technocapitalism/morememory`. Last accessed: June 30th 2016.

[128] S. Köpsell, R. Wendolsky, and H. Federrath. Revocable anonymity. In G. Müller, editor, *Emerging Trends in Information and Communication Security (ETRICS)*, volume 3995 of *LNCS*, pages 206–220. Springer, 2006.

[129] R. Kumaresan and I. Bentov. How to use bitcoin to incentivize correct computations. In G. Ahn, M. Yung, and N. Li, editors, *21st ACM SIGSAC Conference on Computer and Communications Security(CCS)*, pages 30–41. ACM, 2014.

[130] R. Küsters, T. Truderung, and A. Vogt. Accountability: definition and relationship to verifiability. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *17th ACM Conference on Computer and Communications Security (CCS)*, pages 526–535. ACM, 2010.

[131] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[132] L. Lamport. Brief announcement: leaderless byzantine paxos. In *International Symposium on Distributed Computing*, pages 141–142. Springer, 2011.

[133] B. Laurie. Certificate transparency. *Queue*, 12(8):10, 2014.

[134] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[135] S. Le Blond, A. Legout, F. Lefessant, W. Dabbous, and M. A. Kaafar. Spying the world from your laptop. *LEET'10*, 2010.

[136] F. Le Fessant. Un point de vue technique sur la loi internet et creation. `http://fabrice.lefessant.net/lefessant-hadopi-2009.pdf`. Last accessed: June 30 2016.

[137] F. Le Fessant. *Peer-to-peer: comprendre et utiliser*. Eyrolles, Paris, 2006.

[138] M. Lepinski and S. Kent. IETF RFC 6480 an infrastructure to support secure internet routing. `https://www.ietf.org/rfc/rfc6480.txt`, 2012. Last accessed: June 30th 2016.

[139] C. Lesniewski-Laas, B. Ford, J. Strauss, R. Morris, and M. F. Kaashoek. Alpaca: extensible authorization for distributed services. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 432–444. ACM, 2007.

[140] L. Lessig. *The Future of Ideas*. Vintage Books, New York, 2002.

[141] F. Lesueur, L. Mé, and V. V. T. Tong. An efficient distributed PKI for structured P2P networks. In H. Schulzrinne, K. Aberer, and A. Datta, editors, *9th International Conference on Peer-to-Peer Computing*, pages 1–10. IEEE, 2009.

[142] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, 2003.

[143] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *IEEE Symposium on Security and Privacy (S&P)*, pages 112–126. IEEE, 2013.

[144] P. Lincoln, P. A. Porras, and V. Shmatikov. Privacy-preserving sharing and correlation of security alerts. In M. Blaze, editor, *13th USENIX Security Symposium*, pages 239–254. USENIX, 2004.

[145] Y. Liu and J. Pan. The impact of NAT on BitTorrent-like P2P systems. In H. Schulzrinne, K. Aberer, and A. Datta, editors, *9th International Conference on Peer-to-Peer Computing (P2P)*, pages 242–251. IEEE, 2009.

[146] J. V. Lock. A new image: Online communities to facilitate teacher professional development. *Journal of Technology and Teacher Education*, 14 (4):663–678, 2006.

[147] M. P. Lynch. *The Internet of Us: Knowing More and Understanding Less in the Age of Big Data*. WW Norton & Company, 2016.

[148] L. Manovich. *The Language of New Media*. The MIT Press, Cambridge, MA, 2001.

[149] A. Marcozzi and D. Hales. Emergent social rationality in a peer-to-peer system. *Advances in Complex Systems*, 11 (4):581–595, 2006.

[150] H. Marcuse. *One dimensional man: The ideology of industrial society*. Sphere Books, 1968.

[151] M. C. Marino. Critical code studies. `http://www.electronicbookreview.com/thread/electropoetics/codology`. Last accessed: June 30 2016.

[152] S. Matic, P. Kotzias, and J. Caballero. CARONTE: detecting location leaks for deanonymizing tor hidden services. In I. Ray, N. Li, and C. Kruegel, editors, *22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1455–1466. ACM, 2015.

[153] H. Maturana and F. J. Varela. Autopoiesis and cognition, d. *Reidel, Dordrecht, Holland*, 1980.

[154] D. McCoy, J. A. Morales, and K. Levchenko. Proximax: Measurement-driven proxy dissemination (short paper). In *Financial Cryptography and Data Security - 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28 - March 4, 2011, Revised Selected Papers*, pages 260–267, 2011.

[155] J. McLachlan, A. Tran, N. Hopper, and Y. Kim. Scalable onion routing with Torsk. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security (CCS)*, pages 590–599. ACM, 2009.

[156] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman. CONIKS: bringing key transparency to end users. In J. Jung and T. Holz, editors, *24th USENIX Security Symposium*, pages 383–398. USENIX Association, 2015.

[157] L. Melis, G. Danezis, and E. D. Cristofaro. Efficient private statistics with succinct sketches. In *23rd Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2016.

[158] B. Mitra, F. Peruani, S. Ghose, and N. Ganguly. Analyzing the vulnerability of superpeer networks against attack. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 225–234. ACM, 2007.

[159] P. Mittal and N. Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In P. Ning, P. F. Syverson, and S. Jha, editors, *15th ACM Conference on Computer and Communications Security (CCS)*, pages 267–278. ACM, 2008.

[160] P. Mittal and N. Borisov. Shadowwalker: peer-to-peer anonymous communication using redundant structured topologies. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security (CCS)*, pages 161–172. ACM, 2009.

[161] P. Mittal, M. Caesar, and N. Borisov. X-vine: Secure and pseudonymous routing in dhts using social networks. In *19th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2012.

[162] P. Mittal, F. G. Olumofin, C. Troncoso, N. Borisov, and I. Goldberg. Pir-tor: Scalable anonymous communication using private information retrieval. In *20th USENIX Security Symposium*. USENIX Association, 2011.

[163] P. Mittal, M. K. Wright, and N. Borisov. Pisces: Anonymous communication using social networks. In *20th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2013.

[164] F. Monrose and S. Krishnan. DNS prefetching and its privacy implications: When good things go bad. In *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX Association, 2010.

[165] S. J. Murdoch and R. N. M. Watson. Metrics for security and performance in low-latency anonymity systems. In N. Borisov and I. Goldberg, editors, *8th Privacy Enhancing Technologies Symposium*, volume 5134 of *LNCS*, pages 115–132. Springer, 2008.

[166] S. J. Murdoch and P. Zielinski. Sampled traffic analysis by internet-exchange-level adversaries. In N. Borisov and P. Golle, editors, *7th International Symposium on Privacy Enhancing Technologies*, volume 4776, pages 167–183. Springer, 2007.

[167] F. Musiani. *Nains sans geants. Architectures decentralisees et services Internet*. Presses des Mines, Paris, 2015.

[168] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[169] A. Nambiar and M. K. Wright. Salsa: a structured approach to large-scale anonymity. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *13th ACM Conference on Computer and Communications Security (CCS*, pages 17–26. ACM, 2006.

[170] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy (S&P)*, pages 173–187. IEEE Computer Society, 2009.

[171] A. Narayanan, V. Toubiana, S. Barocas, H. Nissenbaum, and D. Boneh. A critical look at decentralized personal data architectures. *arXiv preprint arXiv:1202.4503*, 2012.

[172] M. A. U. Nasir, S. Girdzijauskas, and N. Kourtellis. Socially-aware distributed hash tables for decentralized online social networks. In Y. Liu, M. P. Barcellos, J. R. Lorch, and A. Wang, editors, *IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE, 2015.

[173] L. Neumann and S. L. Star. Making infrastructure: the dream of a common language. In *Proceedings of the PDC '96*, pages 231–240, Palo Alto, CA, 2001. Computer Professionals for Social Responsibility.

[174] T. Ngan, R. Dingledine, and D. S. Wallach. Building incentives into tor. In R. Sion, editor, *14th Financial Cryptography and Data Security Conference (FC)*, volume 6052 of *LNCS*, pages 238–256. Springer, 2010.

[175] S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, and A. Kapadia. Cachet: a decentralized architecture for privacy preserving social networking with caching. In *Conference on emerging Networking Experiments and Technologies, CoNEXT*, pages 337–348, 2012.

[176] F. G. Olumofin, P. K. Tysowski, I. Goldberg, and U. Hengartner. Achieving efficient query privacy for location based services. In M. J. Atallah and N. J. Hopper, editors, *10th Privacy Enhancing Technologies Symposium (PETS)*, volume 6205 of *LNCS*, pages 93–110. Springer, 2010.

[177] A. Oram. *Peer-to-Peer: Harnessing the power of disruptive technologies*. O'Reilly, 2001.

[178] B. Parno, A. Perrig, and V. D. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy (S&P)*, pages 49–63. IEEE Computer Society, 2005.

[179] T. Paul, A. Famulari, and T. Strufe. A survey on decentralized online social networks. *Computer Networks*, 75:437–452, 2014.

[180] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology. Technical report, 2005.

[181] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland. Time-critical social mobilization. *Science*, 334(6055):509–512, 2011.

[182] S. Popoveniuc and R. Carback. Clearvote: an end-to-end voting system that distributes privacy between printers. In E. Al-Shaer and K. B. Frikken, editors, *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 119–122. ACM, 2010.

[183] J. Postel. IETF RFC 821 Simple Mail Transfer Protocol. `https://www.ietf.org/rfc/rfc821.txt`, 1982. Last accessed: June 30th 2016.

[184] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. J. T. Reinders, M. van Steen, and H. J. Sips. Tribler: A social-based peer-to-peer system. In E. G. Sirer and B. Y. Zhao, editors, *5th International workshop on Peer-To-Peer Systems (IPTPS)*, 2006.

[185] T. Pulls, R. Peeters, and K. Wouters. Distributed privacy-preserving transparency logging. In A. Sadeghi and S. Foresti, editors, *12th ACM Workshop on Privacy in the Electronic Society,*, pages 83–94. ACM, 2013.

[186] M. A. Rajab, F. Monrose, and A. Terzis. On the effectiveness of distributed worm monitoring. In P. McDaniel, editor, *14th USENIX Security Symposium,*. USENIX Association, 2005.

[187] M. Raya, M. H. Manshaei, M. Félegyházi, and J. Hubaux. Revocation games in ephemeral networks. In P. Ning, P. F. Syverson, and S. Jha, editors, *15th ACM Conference on Computer and Communications Security (CCS)*, pages 199–210. ACM, 2008.

[188] J. Reagle. *Good Faith Collaboration: The Culture of Wikipedia*. The MIT Press, Cambridge, MA, 2010.

[189] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998.

[190] Y. Rekhter, T. Li, and H. S. IETF RFC a border gateway protocol 4 (bgp-4). `https://www.ietf.org/rfc/rfc4271.txt`, 2006. Last accessed: June 30th 2016.

[191] D. Ribes and C. Lee. Sociotechnical studies of cyberinfrastructure and e-research: Current themes and future trajectories. *Computer Supported Cooperative Work*, 19:581–595, 2010.

[192] T. Ristenpart, G. Maganis, A. Krishnamurthy, and T. Kohno. Privacy-preserving location tracking of lost or stolen devices: Cryptographic techniques and replacing trusted third parties with dhts. In P. C. van Oorschot, editor, *17th USENIX Security Symposium*, pages 275–290. USENIX Association, 2008.

[193] P. Rogaway. The moral character of cryptographic work. *URI: http://web. cs. ucdavis. edu/˜ rogaway/papers/moral. pdf*, 2015.

[194] P. Rogaway and M. Bellare. Robust computational secret sharing and a unified account of classical secret-sharing goals. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 172–184. ACM, 2007.

[195] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich, and H. Bos. Sok: P2PWNED - modeling and evaluating the resilience of peer-to-peer botnets. In *2013 IEEE Symposium on Security and Privacy (S&P)*, pages 97–111, 2013.

[196] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran. AMOEBA: robust location privacy scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.

[197] V. Schafer, H. L. Crosnier, and F. Musiani. *La neutralite de l'Internet, un enjeu de communication*. CNRS Editions/Les Essentiels d'Hermes, Paris, 2011.

[198] S. Schiffner, A. Pashalidis, and E. Tischhauser. On the limits of privacy in reputation systems. In Y. Chen and J. Vaidya, editors, *10th ACM workshop on Privacy in the electronic society (WPES)*, pages 33–42. ACM, 2011.

[199] B. Schmidt, R. Sasse, C. Cremers, and D. A. Basin. Automated verification of group key agreement protocols. In *2014 IEEE Symposium on Security and Privacy (S&P)*, pages 179–194. IEEE Computer Society, 2014.

[200] D. Schoder and K. Fischbach. Peer-to-peer prospects. *Communications of the ACM*, 46 (3):27–29, 2003.

[201] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of the IEEE 2001 International Conference on Peer-to-Peer Computing (P2P2001) August 27-29, 2001*, pages 101–102, LinkÃűping, Sweden, 2001.

[202] A. Sen. *Development as freedom*. Oxford Paperbacks, 2001.

[203] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam. PrPl: a decentralized social networking infrastructure. In *1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, page 8. ACM, 2010.

[204] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[205] R. Sharma and A. Datta. Supernova: Super-peers based architecture for decentralized online social networks. In K. K. Ramakrishnan, R. Shorey, and D. F. Towsley, editors, *4th International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–10. IEEE, 2012.

[206] M. Sherr, M. Blaze, and B. T. Loo. Scalable link-based relay selection for anonymous routing. In I. Goldberg and M. J. Atallah, editors, *9th Privacy Enhancing Technologies Symposium (PETS)*, volume 5672 of *LNCS*, pages 73–93. Springer, 2009.

[207] C. Shirky, K. Truelove, R. Dornfest, L. Gonze, and D. Dougherty, editors. *2001 P2P networking overview*. O'Reilly, Sebastopol, CA, 2001.

[208] K. Singh, S. Bhola, and W. Lee. xbook: Redesigning privacy control in social networking platforms. In *18th USENIX Security Symposium, Montreal, Canada, August 10-14, 2009, Proceedings*, pages 249–266, 2009.

[209] R. Snader and N. Borisov. A tune-up for tor: Improving security and performance in the tor network. In *15th Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2008.

[210] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman. Security analysis of the estonian internet voting system. In G. Ahn, M. Yung, and N. Li, editors, *22nd ACM Conference on Computer and Communications Security*, pages 703–715. ACM, 2014.

[211] S. L. Star. The ethnography of infrastructure. *American Behavioral Scientist*, 43 (3):377–391, 1999.

[212] S. L. Star. Infrastructure and ethnographic practice: Working on the fringes. *Scandinavian Journal of Information Systems*, 14 (2):107–122, 2002.

[213] S. L. Star and G. C. Bowker. How to infrastructure. In L. A. Lievrouw, editor, *Handbook of New Media*, pages 151–162. Sage, London, 2002.

[214] S. L. Star and K. Ruhleder. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*, 7:111–133, 1996.

[215] E. Stefanov and E. Shi. Multi-cloud oblivious storage. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 247–258. ACM, 2013.

[216] B. Stiegler. *Technics and time: The fault of epimetheus*, volume 1. Stanford University Press, 1998.

[217] B. Stiegler. *Technics and time, 3: Cinematic time and the question of malaise*. Stanford University Press, 2010.

[218] B. Stiegler. Relational ecology and the digital pharmakon. *Culture Machine*, 13:1–19, 2012.

[219] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, pages 149–160, 2001.

[220] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti. POTSHARDS: secure long-term storage without encryption. In J. Chase and S. Seshan, editors, *USENIX Annual Technical Conference*, pages 142–156. USENIX, 2007.

[221] R. Süselbeck, G. Schiele, P. Komarnicki, and C. Becker. Efficient bandwidth estimation for peer-to-peer systems. In T. Asami and T. Higashino, editors, *IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 10–19. IEEE, 2011.

[222] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security & Privacy*, pages 44–54. IEEE Computer Society, 1997.

[223] D. Szajda, B. G. Lawson, and J. Owen. Hardening functions for large scale distributed computations. In *IEEE Symposium on Security and Privacy (S&P)*, pages 216–224. IEEE Computer Society, 2003.

[224] Tahoe-LAFS: The least-authority file store. `https://tahoe-lafs.org/trac/tahoe-lafs`. Last accessed: June 30th 2016.

[225] Taler: Taxable anonymous libre electronic reserve. `https://taler.net/`. Last accessed: June 30th 2016.

[226] A. S. Tanenbaum and M. Van Steen. *Distributed systems*. Prentice-Hall, 2007.

[227] C. Tang and I. Goldberg. An improved algorithm for tor circuit scheduling. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *17th ACM Conference on Computer and Communications Security (CCS)*, pages 329–339. ACM, 2010.

[228] I. Taylor and A. Harrison. *From P2P to Web Services and Grids: Evolving Distributed Communities. Second and Expanded Edition*. Springer-Verlag, London, 2009.

[229] T. Tiropanis, W. Hall, J. Crowcroft, N. Contractor, and L. Tassiulas. Network science, web science, and internet science. *Communications of the ACM*, 58(8):76–82, 2015.

[230] A. Tran, N. Hopper, and Y. Kim. Hashing it out in public: common failure modes of DHT-based anonymity schemes. In E. Al-Shaer and S. Paraboschi, editors, *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 71–80. ACM, 2009.

[231] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. Blacklistable anonymous credentials: blocking misbehaving users without ttps. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 72–81. ACM, 2007.

[232] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. PEREA: towards practical ttp-free revocation in anonymous authentication. In P. Ning, P. F. Syverson, and S. Jha, editors, *15th ACM Conference on Computer and Communications Security (CCS)*, pages 333–344. ACM, 2008.

[233] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith. Nymble: Blocking misbehaving users in anonymizing networks. *IEEE Trans. Dependable Sec. Comput.*, 8(2):256–269, 2011.

[234] B. van Schewick. *Internet Architecture and Innovation*. The MIT Press, Cambridge, MA, 2010.

[235] E. Y. Vasserman, R. Jansen, J. Tyra, N. Hopper, and Y. Kim. Membership-concealing overlay networks. In *16th ACM Conference on Computer and Communications Security (CCS)*, pages 390–399, 2009.

[236] J. Verkamp and M. Gupta. Inferring mechanics of web censorship around the world. In R. Dingledine and J. Wright, editors, *2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI)*. USENIX Association, 2012.

[237] D. Verma. *Legitimate Applications of Peer-to-Peer Networks*. John Wiley & Sons, Hoboken, NJ, 2004.

[238] C. Wacek, H. Tan, K. S. Bauer, and M. Sherr. An empirical evaluation of relay selection in tor. In *20th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2013.

[239] M. Wachs, F. Oehlmann, and C. Grothoff. Automatic transport selection and resource allocation for resilient communication in decentralised networks. In *14th IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 1–5. IEEE, 2014.

[240] M. Wachs, M. Schanzenbach, and C. Grothoff. A censorship-resistant, privacy-enhancing and fully decentralized name system. In D. Gritzalis, A. Kiayias, and I. G. Askoxylakis, editors, *13th International Conference on Cryptology and Network Security (CANS)*, volume 8813 of *LNCS*, pages 127–142. Springer, 2014.

[241] M. Waldman and D. Mazières. Tangler: a censorship-resistant publishing system based on document entanglements. In M. K. Reiter and P. Samarati, editors, *8th ACM Conference on Computer and Communications Security (CCS)*, pages 126–135. ACM, 2001.

[242] M. Waldman, A. D. Rubin, and L. F. Cranor. Publius: A robust, tamper-evident, censorship-resistant, and source-anonymous web publishing system. In S. M. Bellovin and G. Rose, editors, *9th USENIX Security Symposium*. USENIX Association, 2000.

[243] L. Wang and J. Kangasharju. Measuring large-scale distributed systems: case of BitTorrent mainline DHT. In *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013*, pages 1–10. IEEE, 2013.

[244] Q. Wang, Z. Lin, N. Borisov, and N. Hopper. rbridge: User reputation based tor bridge distribution with privacy preservation. In *20th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2013.

[245] Q. Wang, P. Mittal, and N. Borisov. In search of an anonymous and secure lookup: attacks on structured peer-to-peer anonymous communication systems. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *17th ACM Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2010.

[246] X. Wang, S. Chen, and S. Jajodia. Tracking anonymous peer-to-peer voip calls on the internet. In V. Atluri, C. A. Meadows, and A. Juels, editors, *12th ACM Conference on Computer and Communications Security (CCS)*, pages 81–91. ACM, 2005.

[247] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose. Phonotactic reconstruction of encrypted voip conversations: Hookt on fon-iks. In *32nd IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE Computer Society, 2011.

[248] M. Winslett, C. C. Zhang, and P. A. Bonatti. Peeraccess: a logic for distributed authorization. In V. Atluri, C. A. Meadows, and A. Juels, editors, *12th ACM Conference on Computer and Communications Security (CCS)*, pages 168–179. ACM, 2005.

[249] E. Wobber, M. Abadi, M. Burrows, and B. W. Lampson. Authentication in the taos operating system. In *14th ACM Symposium on Operating System Principles (SOSP)*, pages 256–269, 1993.

[250] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel. Defeating vanish with low-cost sybil attacks against large dhts. In *Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2010.

[251] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson. Dissent in numbers: Making strong anonymity scale. In *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 179–182, 2012.

[252] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2002.

[253] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur.*, 7(4):489–522, 2004.

[254] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *20th USENIX Security Symposium*, 2011.

[255] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, and P. K. Khosla. Survivable information storage systems. *Computer*, 33(8):61–68, 2000.

[256] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliççöte, and P. K. Khosla. Survivable information storage systems. *IEEE Computer*, 33(8):61–68, 2000.

[257] B. Yang and H. Garcia-Molina. Ppay: micropayments for peer-to-peer systems. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *10th ACM Conference on Computer and Communications (CCS)*, pages 300–310. ACM, 2003.

[258] M. Young, A. Kate, I. Goldberg, and M. Karsten. Practical robust communication in dhts tolerating a byzantine adversary. In *ICDCS*, volume 10, pages 2009–31, 2010.

[259] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against Sybil attacks. *IEEE/ACM Trans. Netw.*, 18(3):885–898, 2010.

[260] H. Yu, P. B. Gibbons, and C. Shi. Dcast: sustaining collaboration in overlay multicast despite rational collusion. In T. Yu, G. Danezis, and V. D. Gligor, editors, *19th ACM Conference on Computer and Communications Security (CCS)*, pages 567–580. ACM, 2012.

[261] D. J. Zage and C. Nita-Rotaru. On the accuracy of decentralized virtual coordinate systems in adversarial networks. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *14th ACM Conference on Computer and Communications Security (CCS)*, pages 214–224. ACM, 2007.

[262] E. Zhai, R. Chen, Z. Cai, L. Zhang, E. K. Lua, H. Sun, S. Qing, L. Tang, and Z. Chen. Sorcery: Could we make P2P content sharing systems robust to deceivers? In H. Schulzrinne, K. Aberer, and A. Datta, editors, *9th International Conference on Peer-to-Peer Computing (P2P)*, pages 11–20. IEEE, 2009.

[263] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. Scion: Scalability, control, and isolation on next-generation networks. In *2011 IEEE Symposium on Security and Privacy*, pages 212–227. IEEE, 2011.

[264] B. Zhu, S. Setia, and S. Jajodia. Providing witness anonymity in peer-to-peer systems. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *13th ACM Conference on Computer and Communications Security (CCS)*, pages 6–16. ACM, 2006.

[265] S. Zhu, S. Setia, and S. Jajodia. Leap: Efficient security mechanisms for large-scale distributed sensor networks. *TOSN*, 2(4):500–528, 2006.

[266] G. Zyskind, O. Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184. IEEE, 2015.