Carmela Troncoso, Marios Isaakidis, George Danezis, and Harry Halpin

Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments

Abstract: Decentralized systems are a subset of distributed systems where multiple authorities control different components and no authority is fully trusted by all. This implies that any component in a decentralized system is potentially adversarial. We revise fifteen years of research on decentralization and privacy, and provide an overview of key systems, as well as key insights for designers of future systems. We show that decentralized designs can enhance privacy, integrity, and availability but also require careful trade-offs in terms of system complexity, properties provided, and degree of decentralization. These trade-offs need to be understood and navigated by designers. We argue that a combination of insights from cryptography, distributed systems, and mechanism design, aligned with the development of adequate incentives, are necessary to build scalable and successful privacy-preserving decentralized systems.

Keywords: Decentralization, Privacy, Peer-to-peer, Systematization of Knowledge.

DOI 10.1515/popets-2017-0056
Received 2017-02-28; revised 2017-06-01; accepted 2017-06-02.

1 Introduction: the Long Road from 2001 to 2016

The successful adoption of decentralized systems such as BitTorrent [24], Tor [57], and Bitcoin [112], and the revelations of mass surveillance against centralized cloud services [74], has contributed to the wide belief that decentralized architectures are beneficial to privacy. Yet, there does not exist a foundational treatment or even an established common definition of decentralization. In this paper we aim at defining decentralization and systematizing the ways in which a system can be decentralized, and, by presenting the key design decisions in decentralized systems, bring forth past lessons that can inform a new generation of decentralized privacy-enhancing technologies.

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: "It's like being back at the O'Reilly P2P conference in 1999," which signaled a peak of interest around decentralized architectures at the turn of the millennium [118]. The 'hype' around decentralization was followed in the early 2000s by research and deployment activity around decentralized systems.

To some extent, decentralization was originally a response to the threat of censorship. Perhaps the first rallying cry for decentralization was the Eternity Service [8]. Anderson created this system in response to the success of the Church of Scientology at closing down the anon.penet.fi remailer [77] "as a means of putting electronic documents beyond the censor's grasp." This motivation of censorship resistance is clear in more modern systems: Tor using a decentralized network of anonymous relays and a DHT-based hidden services naming infrastructure; Bitcoin emerging as a censorship-resistant way to transfer funds to organizations like Wikileaks after the centralized e-Gold [62] online currency had been shut down by the Department of Justice; or BitTorrent succeeding as a peer-to-peer (P2P) file sharing service using Mainline DHT [164] rather than having a central indexing service like Napster that could be subject to requests to keep track of file copying [6]. In each of these cases, decentralization arose as a response to the shutdown of a centralized authority, aiming to remove that single natural point of failure.

Despite the millennial fervour for decentralization, the 2000s witnessed the rise of massively distributed, *but not decentralized*, data centers and systems as the dominant technical paradigm embodied by the Cloud computing capabilities offered by Google, Facebook, Mi-

Carmela Troncoso: IMDEA Software Institute, E-mail: carmela.troncoso@imdea.org
Marios Isaakidis: University College London, E-mail: m.isaakidis@cs.ucl.ac.uk
George Danezis: University College London, E-mail: g.danezis@ucl.ac.uk
Harry Halpin: INRIA, E-mail: harry.halpin@inria.fr

crosoft, and others. Eventually, users were diverted away from software running locally on their machines, which essentially is a form of decentralization, towards cloud applications that enabled an unprecedented aggregation of user data by the providers. Snowden's revelations in 2013 on mass surveillance programs leveraging the centralized nature of these services gave credence to long-standing privacy concerns brought about by the rise and popularity of centralized services.

The desire to preserve privacy, liberty, and the autonomous control of infrastructure and services have led to a call to "re-decentralize" the Internet [128, 179]. As a result, in the 2010s we are observing an upsurge of alternatives to centralized infrastructures and services, although most alternatives to Cloud-based applications are still under development.

It is important for system designers to neither be nostalgic about past systems nor fatalistic about future ones. Today's networking and computing environments are vastly different from those in 2000: Smart-phones have placed a powerful computer in people's pockets; users are usually connected to the Internet over fast connections without time or bandwidth caps; clients, such as web browsers, are now mature end-used platforms with P2P communications enabled and cryptographic capabilities; and mobile code, in the form of Javascript, is ubiquitous.

Even though the design space for modern decentralized systems is less restricted than in the past, fundamental challenges remain. Our key objective is to support future work on decentralized privacy systems by systematizing the past 15 years of research, between O'Reilly's publication of "Peer-to-Peer: Harnessing the Power of Disruptive Technologies" [118] in 2001, and 2016. We aim at highlighting key findings in classic designs, and also the important problems faced by designers of past systems, so as to inform the choices made by engineers pursuing decentralization today.

2 Epistemology

Scope. There is a wide use of the term 'decentralized'. In this paper, we restrict ourselves to discussing systems that support privacy properties using decentralized architectures. We draw a distinction between *decentralized* and *distributed* architectures, as follows:

Distributed system: *A system with multiple components that have their behavior co-ordinated via message passing. These components are usually spatially separated and communicate using a network, and may be managed by a single root of trust or authority.* 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. Developments led by Google, ranging from BigTable [35] to MapReduce [49] are good examples of distributed systems.

Decentralized system: *A distributed system in which multiple authorities control different components and no single authority is fully trusted by all others.*

Following Baran [13], systems are conceived of as networks of interconnected components, where all the components of a system form a graph, where the nodes of the graph are the components and the edges the connections between them (see Fig. 1). Due to this analogy with graphs, the terms "decentralized network" and "decentralized system" tend to be used interchangeably. However, decentralized systems are not just network topologies, but systems that exist to fulfill some function or set of functions, otherwise called 'operations.' These operations are accomplished by passing messages between a sender and a receiver node, with other nodes serving as proxies to relay the message [91] (right graph in Fig. 1). On the contrary, in centralized systems messages and operations are orchestrated by a central trusted authority (depicted as an orange circle in the left graph in Fig. 1).

Centralized systems may be distributed, typically for efficiency or scaling, but not for privacy, and so the underlying components are fundamentally trusted. Only external entities are considered adversarial. Widely deployed systems such as Bitcoin, BitTorrent, and Tor are on the other hand decentralized. Contrary to generic distributed systems, in participating parties may choose their relationships of trust autonomously, including the case where there one may not trust any other components. This has profound implications in terms of security and privacy: no single entity that can act as a trusted computing base (TCB) [135] to enforce a global security or privacy policy. Any internal component of the system may be adversarial, in addition to external parties, requiring defences in depth.

In terms of security and privacy we adopt the following broad definitions, that we make more detailed at the corresponding section when the context requires clarification or preciseness.

Security: *We consider the security aspects of a system to be those that encompass traditional information se-*

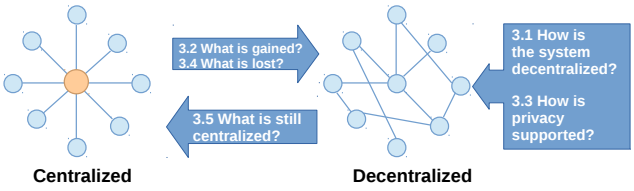

Fig. 1. From centralized to decentralized systems

curity properties. This include of course confidentiality, integrity, and authentication; but also less traditional ones such as availability, accountability, authorization, non-repudiation or non-equivocation.

Privacy: *We consider the privacy aspects of a system to be those related to the protection of users' related data (identities, actions, etc.). This protection is usually formalized in terms of privacy properties (anonymity, pseudonymity, unlinkability, unobservability) for which we follow the definitions by Pfitzmann and Hansen [121]. These definitions are extended in the privacy-oriented discussion in Section 3.3.*

Methods & Model. To systematize knowledge in decentralized privacy-preserving systems we performed a systematic literature review of all papers published in the top 4 computer security conferences (IEEE S&P, ACM CCS, Usenix Security, NDSS) as well as the specialized conferences (PETS, WPES and IEEE P2P) that are proposing or analyzing decentralized systems with privacy properties, from the years 2000 to 2016.

Our first analysis resulted in 165 papers (28 from IEEE S&P, 56 from ACM CCS, 18 from Usenix Security, 11 from NDSS, 11 from PETS, 10 from WPES, and 31 from IEEE P2P). Finally the paper contains only 90 references from these venues (13 from IEEE S&P, 32 from ACM CCS, 10 from Usenix Security, 11 from NDSS, 9 from PETS, 6 from WPES, and 9 from IEEE P2P), 19 are well-known deployed systems that do not have an associated peer-reviewed publication, and the rest come from an additional pool of 30 conferences and workshops (among them FOCI, WEIS, NSDI, SIGCOMM, SIGSAC, or CRYPTO). The selection was done on the basis of highlighting design decisions that reflect a key lesson worth of future reference.

Due to the vast amount of identified designs, by necessity we do not describe each system in detail, but instead show how each system exemplifies a property or design choice. We do, though, expand upon Tor, BitTorrent, and Bitcoin as they are are heavily deployed and have substantial academic analysis. As illustrated in Figure 1, we study the pool of selected designs with the intention to determine:

1. How is the system decentralized? (Section 3.1)
2. What advantages do we get from decentralizing? (Section 3.2)
3. How does decentralization support privacy? (Section 3.3)
4. What are the disadvantages of decentralizing? (Section 3.4)
5. What implicit centralized assumptions remain? (Section 3.5)
6. What can we learn from existing designs? (Section 3.6)

Insights.

- *The key difference between distributed systems and decentralized systems is one of authority and trust between components. Differences in architecture and use of security and privacy controls stem from it.*
- *Decentralized systems embody a complex set of relationships of trust between parties managing different aspects of the system. Untrusted insiders are common, and security controls must be deployed taking into account adversaries within the system.*
- *In distributed, but not decentralized, systems the existence of a single authority that provisions and manages all components that are trusted enables the use of simple security, many times based on dedicated trusted components that act as roots of trust.*
- *In decentralized systems no single authority can provision a root of trust or trusted computing base, making security mechanisms reliant on those (such as central access control or traditional public key infrastructures) inapplicable.*

3 Decentralization and Privacy

This section runs over the key questions we pose in the previous sections with regards to the current state of affairs in decentralized systems. Table 1 (page 417) provides a summary of the different design decisions and the properties achieved as a result.

3.1 How Is Decentralization Achieved?

We review key architectural decisions: how to orchestrate the infrastructure of the network, how to route messages, and how to distribute trust between nodes.

3.1.1 Infrastructure

A first key choice concerns the distribution of tasks needed for maintaining a service within the system. The provisioning of infrastructure impacts the design in terms of trust and message routing.

User-based Infrastructure. Some decentralized system consist solely of nodes that are users and there is no additional infrastructure. They rely solely on users to collectively contribute resources (bandwidth, storage) in order to provide a service. The advantage of this design is that by nature it does not require a third-party centralized authority. This user-based design can support services such as hosting of encrypted data, e.g. in Freenet [41] and Cachet [117]. A disadvantage is that user-based infrastructure may lead to poor performance due to evolving into sparsely connected topologies, and to "churn" caused by peers constantly joining and leaving the network.

User-independent Infrastructure. Here, the functions of the decentralized system are realized by nodes that are not users. A set of third-parties that are not necessarily trusted may provide all or part of the functionality to users. This design pattern underlies classic open federated protocols such as SMTP [123] and XMPP [9] based on a client-server model. The advantages of user-independent infrastructure include increased availability of the service, a reduced attack surface, and immunity to user churn. Servers do not necessarily threaten user privacy. The Eternity Service [8], as realized in systems like Tahoe-LAFS [139], combined encryption with the use of several servers controlled by different non-collaborating authorities for the private storage and replication of files. Other examples of systems that rely on user-independent infrastructure include DP5 [27] and Riposte [42] in terms of Private Information Retrieval [39] or anonymous communication systems like mix networks [36] or DC-nets [37].

Hybrid Systems. Functions may be shared between users and nodes run by third-parties. An example is Tor, where relays are mainly run by volunteers but Directory Authorities are operated by a closed 'known' group of servers. In terms of privacy and security, new elements such as distributed ledgers decentralize traditionally centralized cryptographic protocols in these hybrid systems. For example, computations can be locally and securely recorded to the blockchain with the support of multi-party computation protocols [189], even without a trusted third party [10, 189], or using a small number of stable entities to ensure reliability and low-latency, as in the Sharemind MPC system [26].

3.1.2 Network Topology

When considering a decentralized system, there are two distinct topologies. The first, *network topology* describes the connections between nodes used to route traffic; and the second, *authority topology* describes the power relations between the nodes. Thus, the network routing structure does not necessarily have to mirror how authority is decentralized in a system, although it often does. That can greatly affect the security and privacy properties of the system [53]. It must be noted that components of traditional network routing is done in a hierarchical manner, including spanning tree protocols such as in BGP [130] in the current Internet as well as 'next generation' designs like SCION [186].

Mesh. Mesh topologies are unstructured. Nodes can route messages to every other node they are connected with. One advantage is that mesh networks function in settings with no stable connections to other nodes to guarantee service in the presence of massive churn and changing connectivity, such as in mobile ad-hoc networking and file sharing in early versions of Gnutella [73]. A particularly popular communication means in mesh topologies [112] are *gossip protocols*. In gossiping, as opposed to flooding, a random subset of the nodes in the network are chosen to receive the messages. These nodes then continue to broadcast the message via another independently selected random subset of the network to relay messages. The reliability of message delivery under load is questionable and information propagation experiences delays. Historically mesh networking does not preserve user privacy of their users, but recent secure messaging systems such as Briar [28] use this topology to remain functional during Internet blackouts.

Distributed Hash Tables (DHT). DHTs are network topologies where each node maintains a small routing table of its neighbours, and messages are passed greedily to known nodes that are 'closer' to the intended recipient. Although efficient and decentralized, DHTs do not by themselves provide strong security, privacy and anonymity properties. While decentralized, DHTs are not secure and privacy-preserving by default: Tran et al. [153] show that low latency anonymity systems based on DHTs such as Salsa [113] are vulnerable to having large amounts of traffic captured by adversaries control-

ling a fraction of the relays. DHT nodes may, however, be grouped into byzantine quorums to defeat adversaries that control a minority of nodes [180].

Super-nodes. Super-nodes are nodes that are endowed with more, and contribute more, resources to the system. This may be in terms of computation power, storage, or network connectivity, stability and up time. In terms of routing, such super-nodes may be used to mediate operations requiring higher network throughput. They can be arranged in structured topologies, designed to leverage them; or they may emerge naturally in unstructured topologies, as a result of some nodes committing more resources. Most P2P systems such as BitTorrent eventually rely on super-nodes [50]. These super-nodes have serious implications on availability and integrity, as they may become targets for attack, and privacy, as they mediate, and are in a privileged position to observe, a larger fraction of activities.

Stratified. Some of the more complex decentralized systems use a stratified design where nodes have specialized roles in terms of routing, or other functions. A paradigmatic example is the Tor network. Tor users autonomously form circuits from an open-ended set of Tor relays, in layers of entry guards, middle nodes and exit nodes. A high-integrity global list of these relays is maintained through consensus by a closed group of specialized Directory Authorities. Simultaneously, Tor hidden services are resolved through a Hidden Service Directory maintained by a simple DHT topology. We note that, on some level, Tor has also evolved to use super-nodes on its topology and the distribution of traffic sent through Tor relays is far from uniform [84]. Cascades, are a particular case of Stratified topologies in anonymous communications, in which paths are pre-defined. The advantages and disadvantages of such choice as opposed to free routes has been discussed in [52].

3.1.3 Authority

We now consider the relation among nodes in terms of authority and describe mechanisms to mitigate the potentially effects of power disparity that could potentially harm the security and privacy of users.

Ad-hoc: Nodes Interact Directly. In ad-hoc there is no relationship of authority among nodes. Nodes directly interact with each other without the participation of other nodes, and they do so for the benefit of the involved parties only. In terms of routing, ad-hoc requires a mesh topology where nodes do not carry traffic for other nodes. However, note that mesh topologies do not always have a ad-hoc (lack of) authority relations, such as routing based on gossip. An example of this type of system would be point-to-point communication in Briar [28]. For purposes of privacy, direct interaction bypasses possibly compromised nodes, but not network adversaries. As for confidentiality, communications can be encrypted between the two nodes, and can be extended to group communication using group key agreement protocols [138].

P2P: Nodes Assist Other Nodes. P2P designs have no central authority. Unlike ad-hoc interaction, nodes provide services and resources to other nodes, such as routing messages or storing blocks of data. Nodes have equal authority and so each node may equally compel any other node, although services and resources are usually provided according to their capacity. In other words, P2P systems self-organize and all nodes are responsible for carrying out operations for all other nodes, rather than having any pre-configured special position of authority. Since nodes are not motivated by authority to help each other, mechanisms should instead be in place to provide 'incentives' for collaborative behaviour.

There are clear advantages for the security and privacy properties in P2P systems. Information about peers is not centralized and interaction typically remains local to a few nodes, so it is difficult for an adversary to obtain a global view of the system. Yet, relying on peers for functionality poses an additional threat to privacy, since requests may be served by adversarial nodes. These nodes can passively collect information on other nodes or they may actively disrupt the integrity of operations by forging messages or replay attacks that are hard to detect. Furthermore, since P2P systems are usually open, without any admissions control, adversaries may purposely inject a large number of Sybil nodes, to increase their chances of a successful attack [59]. P2P systems are not a silver bullet for decentralization: there is no clear and definite solution to Sybil attacks in P2P networks, although such an attack can be mitigated using reputation [43] or trust [83].

Social-based: Nodes Assist Friends. These designs take advantage of pre-existing decentralized relationships, such as "friendship". In terms of applicability of security mechanisms this approach maintains most advantages of a P2P system. It is less vulnerable to Sybil attacks as adversarial nodes can be excluded from participating in the network or may be easier to detect [47], as it is harder to infiltrate a social network

than a network. The downside is that, without cover traffic, a global passive adversary can discover the underlying social graph by monitoring network communications and violate privacy properties such as unobservability and unlinkability. This in turn may lead to user deanonymization [114], and techniques such as perturbation of the underlying graph may not be robust enough to prevent this [107].

A number of systems implement social-based communication to resist Sybil attacks. For instance Drac [44] and Pisces [108] use social-networks to support routing of messages. X-Vine [105] is a mechanism that, applied to distributed hash tables, helps resisting denial of service via Sybil attacks at the cost of higher latency. Tribler [124] uses social-based trust relations to improve performance that exploits similarity to improve performance, content discovery, and downloading in file sharing; or Nasir et al.'s socially-aware DHT [116], which reduce latency and improve the reliability of the communication.

Federated: Providers Assist Users. In federated designs, users are associated to *provider* nodes, which they trust and that act as authorities. Each provider is responsible only for its own users but collaborates with other providers in order to provide a service. No single provider has authority over other providers, and thus there is a "federation" of providers. Federated authorities typically use user-independent infrastructure and act as a super-node in terms of routing. This combination of design choices leads typically to high availability as long as the provider is accessible and not compromised, but the provider is a central point of attack to violate security properties and the provider itself can violate the privacy of nodes. The primary weakness of federated systems is the assumption that federated service providers largely act honestly. Some techniques can relax strong trust assumptions in the provider. End-to-end encryption can maintain confidentiality [145] using providers. Computation can be obscured using secret sharing [133] or differential privacy-based solutions [3].

Accountability: Transparency Assists Users. Transparency can be used to make an authority accountable in order to establish trust. It promotes integrity of operations by monitoring the correct behavior of nodes, e.g. a transparent log of a provider's operations in a federated system audited by users or other providers acting in lieu of their associated users. The nature of this auditor's authority is very different from the aforementioned previous types of authority relations and critically relies on the non-collusion of the auditor and the audited authority, e.g., Bitcoin consensus over its blockchain using proof-of-work. Other alternatives, such as Certificate Transparency [92], rely on a set of services and auditors to keep track of X.509 certificates and quickly detect potentially rogue or hacked certificate authorities. Similarly, electronic election protocols [75] achieve robustness through proofs of correct shuffling of votes, e.g., Helios [1]. Yet naïve designs of audit logs may violate the privacy of decentralized nodes by learning too much information.

While decentralized accountability can have clear advantages regarding integrity, there are difficulties in maintaining privacy in any distributed log. This disadvantage can nevertheless be reduced as shown by Zerocash [18], which uses zero-knowledge proofs in order to maintain unlinkability in auditing relationships; or CONIKS [101], that shows that auditing the consistency of a name-key binding through time enables verification of user public keys by the end users collectively and by other providers, while concealing the identities and the number of users at each provider using Verified Random Functions.

Insights.

- *Decentralization encompasses a large space of designs from decentralized ad-hoc mesh to federated super-node networks, not just peer-to-peer. These offer a variety of privacy and systems (e.g., availability, or reliability) properties. Developer instincts may often be incorrect in terms of their trade off.*
- *Despite being separate parts of the design, the network topology in decentralized systems often mirrors the authorities' trust relationships. However, a strict mapping between authority, infrastructure and networking topology is not necessary, and may come at the cost of harming privacy or availability.*
- *Centralization in terms of federated and super-nodes leads to better availability and system performance. However, it introduces single points of failure that impact availability and privacy. P2P models are by design more resilient to unstable routing and compromises, but entail higher engineering complexity.*
- *All networking topologies suffer under node churn, and pure P2P topologies must effectively address this effectively to be applicable at all.*
- *Decentralization does not imply the absence of any infrastructure. However, the infrastructure itself needs to be decentralized by being provided by a plurality of authorities. Such infrastructure may enhance performance by offering super-nodes or dedicated high-availability operations.*

- *De-facto super nodes may emerge naturally in decentralized designs, as a result of different node capabilities, and efficiency in centralizing certain operations. If this occurs outside the context of careful design, those super nodes become a single point of failure, and may lead to de facto re-centralization.*
- *Lack of relationships of authority imply that nodes must be willing to provide services to each other on a different basis. Designers of decentralized systems must carefully engineer such incentives, to ensure that natural (non adversarial) selfishness does not lead to dysfunction. Monetary incentives, reputation, and reciprocity can be the basis of such incentives – but off the shelf such mechanisms are often central points of failure.*

3.2 The Advantages of Decentralization

In this section we discuss a number of perceived intrinsic architectural advantages to decentralized designs that make them appealing compared to their centralized counterparts.

3.2.1 Flexible Trust Models

An intrinsic advantage of decentralized architectures relates to the existence of multiple independent authorities. These create a distributed trusted computing base that ensures that a subset of rogue nodes, at least up to a certain threshold, cannot compromise the overall security properties of the whole system.

Distributed Trust. Decentralized systems leverage multiple independent authorities into a security assumption: for example, all forms of threshold cryptography [141] assure that if some fraction of participants are honest, some security property can be guaranteed. This principle can also be applied to secure multi-party computation, distributed key generation, public randomness and threshold-based decryption, and signing. One such privacy system is Vanish [72] that guarantees deletion after a pre-set expiry date. It illustrates how a multi-authority system implements properties otherwise impossible, or implausible, to when implemented by a single entity. However, the system was in practice defeated by a Sybil attack that the security properties of its DHT did not take into account [172]. Reliance on multiple authorities to regain a degree of privacy has also been proposed for commercial cloud storage in case some providers are dishonest [146].

No Natural Central Authority. In some settings there exists 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 [171] and SDSI [64], and 'trust management', such as Keynote [25]. In such systems a set of distributed principals make claims about users and each other, and those claims need to be assembled and used to resolve access control decisions. Bauer et al. [15] show that the task of resolving access control decisions in a decentralized setting is faster than doing so centrally.

Leveraging Existing 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. Systems may use the underlying social trust structure to build overlay privacy-friendly social network services, as surveyed by Paul et al. [120]. As an example, the Frientegrity system [68] 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 [19], an open-source project that takes a different approach: users connect to a provider they trust – that gains full visibility of their activity – and delegate the access control on the content they share with their social circles to that provider.

3.2.2 Distributed Allocation of Resources Assists with Ease of Deployment

A central premise of P2P networks is that nodes contribute spare resources, and doing away with a central authority that is forced to bear the full costs (such as Google's server costs). This reduces costs and helps ease deployment by spreading these demands amongst multiple parties. Costs are lowered as spare capacity in the existing infrastructure is used, e.g., underutilized resources given by users such as the early SETI@home project [7] and the use of users' storage in Freenet [41].

In terms of availability, decentralized architectures exhibit fewer correlated failures by virtue of being distributed. As an example the Cachet system [117] uses a pool of untrusted peers as a storage back end of a decentralized Online Social Network.

3.2.3 Resilience Against Formidable Adversaries

Location Diversity. Decentralization provides properties that are inherently difficult to centralize, such as the network location diversity needed for Tor bridges [56] to bypass censorship both on the network and legal levels. A number of designs take advantage of this, like Publius [163], in order to resist censorship, although censorship resistance itself is a separate field with many centralized, as well as decentralized, solutions.

Survivability. Decentralized architectures can be designed to survive *catastrophic* attempts to take them down or inflict crippling damage, in a way that centralized systems cannot resist [176]. This property has been used to build highly robust botnets using a peer-to-peer architecture [134]. Although these bot-nets are decentralized on the technical level, they of course maintain central but covert command and control (C&C). Those botnets have demonstrably 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 wider 'Darknet' survivability is provided by Zhou et al. [97].

Separation of Development from Operations. Decentralized architectures clearly separate the authorities that provide public code – and that have no access to operational data and secrets – and those that run the code. Users and nodes, deploying software, can audit any such open source code for integrity, and chose whether to deploy it. The core development team maintains the code, that is publicly visible and auditable, but upgrading is up to independent relay operators. This model is followed by both Tor and Bitcoin. 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. Similarly in Ethereum, the exploitation of a vulnerability in the DAO smart-contract, led to the core developers proposing a "hard fork", and this fork was voluntarily adopted by the majority of the Ethereum mining node operators.

Publicly Verifiable Integrity. Due to the availability of multiple independent authorities, decentralized systems can implement accountability mechanisms to publicly verify integrity. Adversaries are disincentivised to compromise nodes, by ensuring attacks have an observable effect so that cheating can ideally be discovered before it has a negative effect. Verifiable logs can be used to help enable privacy as ensuring that actions are transparent enables users to know what happened with their data, as when Pulls et al. [125] 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 in e-voting: threshold cryptography is used for parameter and ballot generation, with privacy enforced via threshold decryption.

Insights.

- *Real-world relationships of trust and authority are personal, complex and localized, and rarely hierarchical or all-or-nothing. Decentralized systems offer flexible trust models that can leverage those relationships to support security and privacy properties.*
- *When it comes to high-availability and survivability against powerful adversaries – particularly with legal authority – decentralized designs are not just best, but sometimes the only available option. Designs that allow operations to continue despite some authorities being adversarial or not available, are necessary to support these properties.*
- *Decentralization's fundamental advantage in terms of security stems from an attacker having to compromise a set of independent authorities in order to disrupt or weaken the security properties of a system. Decentralized systems that do not offer this property may be more fragile than centralized equivalents.*
- *Decentralized designs decouple development from operations and have a multistakeholder governance model, where node operators influence the entire system based on the software configuration they choose to deploy.*
- *Decentralized systems can leverage public accountability to detect and exclude compromised or misbehaving authorities. Such accountability architectures may be used instead of more complex or expensive prevention techniques, but need to ensure that auditing will be effective and eventually acted upon.*
- *Leveraging spare resources of nodes allows decentralized system to scale, and ease deployment. However, this by itself opens the door to high-churn and cannot be a substitute for robust incentives to participate as the system scales or nodes are asked to take on real costs.*

3.3 How Does Decentralization Support Privacy?

In this section we survey the privacy properties obtained through mechanisms that are inherent to decentralized architectures. We limit ourselves to the analysis of technical properties that may be obtained in decentralized systems. We acknowledge that decentralized systems may offer both greater user privacy and autonomous control of the infrastructure. As such they are a possible technological solution to the legally-binding, but often technologically unenforced, demands from data protection laws [67, 136], that often are addressed involving a central authority, the data controller [54]. How decentralized systems relate to the law and business models is out of the scope of this paper.

Confidentiality from Third Parties. Some designs employ a decentralized architecture on the grounds that the lack of centralized components, which have full access to user data and can surveil their actions, would be beneficial to confidentiality and unobservability. Such systems may use threshold encryption [141] in order to trade off information confidentiality and information availability, such as the PASIS [176] 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 [148] or Plutus [87].

Confidentiality from Peers. In P2P architectures, nodes must interact with other nodes, but they want their communications or actions to remain confidential. For example, nodes need to perform a joint computation, but 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 [60] is such a system, in which further zero-knowledge proofs are integrated to protect computations against malicious users. More recent, blockchain-backed systems, such as Enigma [189] rely more heavily on transparency to achieve this goal. In terms of message-passing, systems that pass end-to-end encrypted messages across untrusted federated servers achieve peer confidentiality.

Anonymity. Due to the distribution of resources in decentralized networks, it is expensive for one entity to observe all actions in the network and track all activities from a user. Many [70, 78, 100, 105, 113], leverage this approach to provide anonymous communication, although the precise properties provided in terms of anonymity differ. Some decentralized systems fail to provide full anonymity but instead provide pseudonymity which is weaker [121], e.g. it allows multiple anonymous actions to be linked, providing weaker privacy, but enabling functionality such as detecting returning users and reducing the complexity of the system. For example, in Bitcoin every transaction is linked to a pseudonym and stored in the blockchain. This allows to trace money flows and avoid double-spending; but on the downside if a pseudonym is ever deanonymized (e.g. [21]), all actions from the person would be revealed. A number of decentralized systems, ranging from mix-nets [36, 45], to DC-nets [37], to Tor [57], provide some degree of anonymity.

Deniability. Deniability enables a subject to safely and believably deny having originated an action, so as to shield her from responsibility associated to performing such action. The fact that actions cannot be linked back to a user (i.e. "unlinkability" [121]), equips users with freedom to perform actions without fear of retaliation. For instance, in Freenet [41] requests are hard to link to their originator, thus users can freely search for information without revealing their preferences.

Plausible deniability is crucial in facilitating anonymous and censorship-resistant publishing, and may be implemented using cryptographic techniques allowing of 'repudiation'. This was the motivation behind the original Eternity service [8] and well-known designs such as Publius [163]or Tangler [162].

Covertness. Some systems protect even the act of participation of nodes in the decentralized network from outside observers ("unobservability"[121] if the items of interest is the existence of users). In addition to more well-known work like Tor pluggable transports [122], the Membership Concealing Overlay Network (MCON) [157] leverages this to provide strong forms of covertness. All nodes in MCON only have links with trusted friends, and a complex overlay network is jointly created that allows all nodes to communicate indirectly with all nodes. As any node only connects to other locally trusted peers, the system defends against attempts to enumerate all users by malicious nodes.

Insights.

– *The key bet of decentralized systems in terms of privacy is that a local adversary may not observe all communications, data, or actions. However, global adversaries are increasingly realistic. Thus decen-*

tralized systems that rely solely on dispersion of information to provide confidentiality are fragile.
- *Decentralization can harm privacy: Distributing trust and resource contribution to multiple authorities may provide adversarial nodes with extended visibility of user data and network traffic. Thus, naive decentralization designs may in fact create more, not fewer, attack points to breach privacy.*
- *Decentralization alone cannot balance the needs for privacy, integrity and availability. It is only combined with the use of advanced cryptography that decentralized architectures obtain those properties. In particular, the reliance on others to perform actions, may naturally expose personal information to other nodes without the use of cryptography. However, naive encryption alone may not be sufficient to support the integrity of operations that are more complex than end-to-end messaging.*
- *Decentralized networks can provide privacy properties like anonymity and even covertness. Yet, most real-world decentralized systems do not use the advanced cryptography and traffic analysis resistance necessary for that purpose as it increases design, implementation, operations and coordination costs.*

3.4 The Disadvantages of Decentralization

Sadly, there is no free lunch in decentralization. While decentralizing has many advantages, there is no guarantee that the properties and features of centralized systems are maintained in the process. This section summarizes problems emerging when decentralizing designs. A further critique of decentralized systems, focusing on personal data, is provided by Narayanan [115].

3.4.1 Increased Attack Surface

Decentralizing systems across different nodes inherently augments the number of points (attack vectors) that an adversary could use to launch an attack or to observe the users' traffic.

Internal Adversaries. In centralized systems, system components can be monitored and evaluated by a trusted entity to detect malicious insiders. In a decentralized system it is easier to insert a malicious node undetected. A number of such attacks have been documented against decentralized systems: the predecessor attack [174, 175] uncovers communication partners in many anonymous communication schemes [37, 57, 129, 150], or the Sybil attack which can be used to bias reputation scores [59] or corrupt the information exchanged in collaborative decentralized systems [82]. Furthermore, when messages are relayed through other nodes, e.g., to gain anonymity, their content is exposed to potential adversaries, as in Crowds [129] for Web transactions or in Yacy [177] for searching information.

Traffic Analysis. Decentralization inherently implies that information will traverse a network. Even in the presence of encryption, metadata is available to external adversaries. For instance, in anonymous communications networks it has been repeatedly shown that both passive local [103] or (partially) global [84, 111], as well as active adversaries [167], can reduce or break anonymity by looking at traffic patterns.

Inconsistent Views. Decentralization typically implies that nodes have a partial, thus non-consistent, view of the network which can have an impact on integrity. These non-consistent views allow adversaries to "cheat" without being detected. For instance, in Bitcoin adversaries can perform double spending by forcing non-consistency through fast operations [89], or eclipse attacks [76] in which the adversary gains control over all connections of a target node thus isolating her from the rest of the network. Furthermore, the lack of global information results in users not necessarily making the optimal choices with respect to optimizing their privacy, as studied both in the context of anonymous communications [55] and location privacy [71].

3.4.2 Cumbersome Management

An obvious problem of decentralization is that no entity has a global vision of the system, and there is no central authority to direct nodes in making optimal decisions with regard to software updates, routing, or solving consensus. This makes the availability of a decentralized network more difficult to maintain, a factor significant enough to contribute in the failure of a system, as pointed out by the Mojo Nation developers [168]. It is very common that nodes in a decentralized system have hugely varying capabilities (bandwidth, computation power, etc.) [69, 160], making super-nodes attractive targets [102]. Finally, decentralized systems need to overcome the shortcomings of underlying technologies (such as NAT [98]), that favor the client-server paradigm over peer-to-peer networking.

Defense Difficulties. The lack of central management hinders the establishment of effective protection mechanisms. For instance, the non-consistent view of the network not only enables attacks, but also hampers the use of collaborative approaches to detect incorrect information [88]. Similarly, it becomes extremely difficult to prevent Sybil attacks, and defenses must either leverage local information, for example defenses based on social networks [47, 181], or collaborative approaches that combine information from several nodes [119].

Routing Difficulties. A straightforward consequence of the lack of centralized control is an increased complexity in routing. Nodes do not have an overview of the network and its capabilities [149] and consequently cannot globally optimize routing decisions [183], falling back to inefficient flooding or gossiping methods in mesh topologies. This is made harder by highly diverse nodes [69], the existence of churn [11] and the reliance on possibly malicious nodes [166]. Solutions to these problems include using complex routing algorithms to enable secure and private discovery of nodes [100, 104, 108], or avoiding the use of a centralized directory via next-generation DHTs. The lack of centralized routing information in decentralized topologies also impacts performance as it hinders the selection of optimal routes or load balancing. We find two approaches to alleviate this problem: using local estimations to improve performance [4, 5, 152], or providing means for users to make better decisions about routing individually [144]. The latter is known to be prone to attacks [78, 110].

3.4.3 Lack of Reputation

Decentralization is also an obstacle to the implementation of accountability and reputation mechanisms. The negative effect is amplified when privacy and anonymity mechanisms are in place, as it becomes even more difficult to identify misbehaving nodes such as Sybils [79]. An effect of this lack of reputation is that nodes have no incentive to behave correctly and can misbehave to obtain advantages within the system (e.g., better performance). This problem has been identified in many settings such as P2P file sharing [184], multicast communication [182], or reputation [79]. In particular, the presence of churn, which make nodes short-lived and difficult to track over time, makes the establishment of reputation to guarantee veracity a very challenging problem [127], even more if privacy has to be preserved [137].

Poor Incentives. Without reputation, reciprocity and retaliation it is hard 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 having witnesses to report on malicious nodes in a privacy-preserving manner [187]. However, the most popular approach is the use of (anonymous) payments that incentivize good and collaborative behavior that benefits all users in the network [17, 38, 90]. In contrast, one example of negative reinforcement is the tit-for-tat strategy to encourage users to share blocks to incentivize sharing, as in BitTorrent.

Insights.

- *Decentralized designs may prevent conventional attacks but also introduce new ones. Unless they are carefully designed, they may expose personal information to more, rather than fewer parties; and the need to perform joint computation across many authorities introduces threats to integrity.*
- *Decentralized systems are particularly susceptible to traffic analysis, compared with centralized designs, since their distributed operations are mediated through networks and adversarial nodes that may use meta-data to compromise privacy.*
- *Decentralized systems by nature require complex management of routing, naming and consistent state – due to the lack of a central coordinator. Conventional defences against network attacks, like denial of service, require centralization and cannot be straightforwardly applied.*
- *Sybil attacks are the great unsolved problem of decentralized systems that allow open and dynamic participation. Solutions based on social networks rely on fragile social assumptions; admission control through identification or payment re-introduce centralization. Proof-of-work defences increase the cost of participation.*

3.5 What Is Still Centralized in Decentralized Designs?

Even when systems claim to be decentralized, usually there are "hidden" centralized assumptions and parts of the design that need to be centralized to operate correctly. These are often implicit.

3.5.1 Centralization of Network Information & Computations

In any decentralized system routing packets across the network is a challenge for both operational and privacy reasons. Typically routing can be divided in two main task. The first task is how to find candidate nodes to relay traffic, and second task is how to select among these nodes. While as detailed in Sect. 3.1.2, there are many decentralized algorithms to choose the route, actually finding candidate nodes is difficult, as highlighted in Sect. 3.4.

Centralized Directories. A common solution for the first problem is to assume that there exists a centralized directory that knows all network members. The most prominent example is the Domain Name System (DNS) that resolves easy-to-remember domain names to associated IP addresses in order 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 [109] or availability [158], and thus several alternatives are being proposed [161] and deployed [58]. Another example are Tor Directory authorities [57] that provide Tor clients with the full list of onion routers. These directories solve the discovery problem but have become a bottleneck for the scalability of the system [100]. How to decentralize these authorities in an efficient, privacy-preserving manner is an active area of research. Solutions are based on having multiple copies of the publicly verifiable directory kept consistent via consensus protocol and distributed via gossiping, although it risks covertness; or to use friend-of-a-friend discovery and routing [100, 106].

Path Selection. Once routing alternatives are known the question remains: Which route to choose? Thus typically, a centralized server is considered that can "rank" routing options to allow for path optimization with respect to adversaries [2, 12, 61, 86], performance [143, 144, 159], or with respect to users' reputation [165]. Such a centralized ranking approach has been shown to be vulnerable to attacks [14, 22]. Typically DHTs are the possible solution, although only a few have the necessary security and privacy properties for use in decentralized systems [46].

Distributed Computations.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 [34, 65, 188] 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 [126], anonymous surveys [80], or private statistics [63] in which, even though nodes perform decentralized computations, interaction with a central authority is needed to produce the final result.

3.5.2 Trust Establishment

A challenge when decentralizing networks is to ensure that nodes can be trusted to perform the actions they are assigned or can authenticate themselves as the intended receiver of a message. Often, to avoid dealing with this problem, a common implicit centralized assumption is that a set of trusted servers is assumed to exist, such as in Dissent [173] or the Directory Authorities in Tor.

Decentralized trust establishment is still an open problem, though some of the excitement around mining in Bitcoin is precisely due to their attempt to avoid this problem and so build a 'trustless' decentralized system.

Authentication. In general certificate infrastructures are not decentralized, e.g., PKI. Therefore, some decentralized systems rely on centralized certification authorities to authenticate nodes that can be used for secure routing [33, 147], user authentication [29], or to enrol users in the system in the context of anonymous credentials [16, 30, 31], a privacy-preserving alternative for authentication without requiring user identification. Such centralized authorities are simpler for deployability or usability, but become a single point of failure as pointed out by Lesueur et al. in [95]. They also introduce an imbalance of power unnatural for decentralized environments since they allow a single entity to revoke peers' authentication credentials. Many decentralized designs do not address authentication (e.g. [117, 142], see [120] for more details), although work from TAOS [170] and SDSI [132] onwards has been working in this direction [20]. Authentication is useful to prevent Sybil attacks, and work on decentralized and privacy-preserving authentication via threshold cryptography is one promising solution [99], as is the use of zero-knowledge systems for anonymous credentials [16].

Authorization. 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 from SDSI [132] to more recent systems [94, 96, 169]. OAuth was designed to be federated in terms of authorization, but in practice only a few large providers use this standard [140]. So if an adversary compromises a user's single authentication method such as a password, it can compromise them across multiple decentralized systems. Work descending from SDSI [132] to limited-time authorization via pseudonyms and blind signatures present one way forward to decentralize authorization [99].

Abuse Prevention. As mentioned in Sect. 3.4 accountability is a challenge in decentralized systems. 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 [154, 155], or to store blacklists [85, 156]. Similarly, identity escrow [23] or revocable anonymous communication solutions [40], 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. Typically, these are built from pre-existing trusted social networks, and only recently have reputation systems such as AnonRep (based on homomorphic encryption and verified shuffles) allowed reputation to be done in a privacy-preserving and decentralized manner [185].

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. [17, 32, 178], 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 [17], or on authorities that record transactions to help taxation on the operations run in the system, even if the transactions are anonymized [151]. Decentralized crypto-currencies can help ameliorate this problem.

Trusted Developer Community. All decentralized systems work by virtue of having the nodes communicate via the same protocol. Thus, the actual software can be a centralized point of failure if the protocol is flawed. If the protocol is standardized or otherwise uniformly specified, the implementation of the protocol itself may be a failure. Furthermore, the developers themselves could be compromised. his danger is augmented by the software monoculture prevalent in deployed systems, that results in a bug in a popular platform capable of compromising a large set of authorities. One solution is to apply the technique of forcing public transparency and auditing of the integrity of the development process. Open-source development, done in public repositories, is increasingly required. Integrity is ensured via deterministic builds [131] so that everybody can verify the genuine binary, and the authority to run new versions of the software remains in the hands of the operators. This approach is already followed by Tor and increasingly by Bitcoin, where the choice to deploy particular open-source code is up to miners.

Insights.

- *Many decentralized systems implicitly rely on centralized components to hold network information for efficient routing or for establishing trust and defending against Sybil attacks.*
- *Essential user-facing infrastructure, from authentication to authorization is centralized even in decentralized systems. Developing alternatives seems to be an open problem, with no clear established design. For payments, Bitcoin has recently provided a decentralized solution, but it suffers from a number of scalability, privacy, and financial volatility problems.*
- *The developer community of a system is usually an implicit centralized authority, making social attacks on the developer community itself one of the largest dangers to any decentralized system.*

3.6 Systematization of Existing Designs

Table 1 presents a systematic analysis of decentralized designs, clustered based on their principal goal. The columns infrastructure, network topology, authority relations, privacy properties, follow closely the definitions of the previous subsections. We applied some level of simplification to complex systems with multiple components or multiple use-cases. The systematization focuses on parts of the system relevant for its main use-case as used in prototype or deployment.

Insights.

- *Many systems that provide good coverage of privacy properties and decentralization (usually via DHTs) have not been widely deployed*

Table 1. Selected decentralized privacy systems evaluated on how they achieve decentralization, the privacy properties they provide, and implicit centralized assumptions.

System	Infrastructure	Network Topology	Authority	3rd party-Confidentiality	Peer Confidentiality	Anonymity	Deniability	Unobservability	Centralized Directories	Central Trust Establishment
User Anonymity										
Tor‡ [57]	Hybrid	Stratified	P2P	✓	✓	✓			✓	✓
Mixnets‡ [36, 45]	User-independent	Super-Node	P2P	✓	✓	✓			✓	✓
I2P‡ [81]	User-based	DHT	P2P	✓	✓	✓				
Crowds§ [129]	User-based	Mesh	P2P	✓		✓	✓			
MCON§ [157]	User-Based	Mesh	Social	✓	✓	✓	✓	✓		✓
File Sharing/Censorship Resistance										
BitTorrent‡ [24]	User-based	Super-Node	P2P							
Freenet‡ [41]	User-based	DHT	P2P	✓	✓	✓	✓	✓		
Gnutella‡ [73]	User-based	Super-Node	P2P							
Publius† [163]	User-independent	Mesh	Federated			✓			✓	✓
Eternity§ [8]	User-independent	Super-Node	Federated	✓	✓		✓		✓	✓
Tribbler‡ [124]	User-based	DHT	Social	✓		✓		✓		
Vanish† [72]	User-based	DHT	P2P	✓	✓					
Tangler§ [162]	User-independent	Super-Node	Federated	✓	✓		✓		✓	
Tahoe-LAFS‡ [139]	User-independent	Stratified	Federated	✓	✓				✓	✓
Cryptocurrencies										
Bitcoin‡ [112]	User-based	Super-Node	Ad-hoc±							
Zerocash‡ [18]	User-based	Super-Node	Ad-hoc±	✓	✓	✓				
MojoNation‡ [168]	User-based	Mesh	P2P	✓	✓					✓
Ethereum‡ [66]	User-based	Super-Node	Ad-hoc±							
Secure Messaging										
SMTP+PGP‡ [123]	User-independent	Stratified	Federated	✓	✓				✓	✓
XMPP+OTR‡ [9]	User-independent	Stratified	Federated	✓	✓		✓		✓	✓
Briar† [28]	User-based	Mesh	Ad-hoc	✓				✓		
DP5† [27]	User-independent	Stratified	Federated	✓	✓	✓			✓	✓
Riposte§ [42]	User-independent	Stratified	Federated	✓	✓	✓	✓	✓	✓	✓
Dissent/Buddies§ [173]	User-independent	Stratified	Federated	✓	✓	✓			✓	✓
Drac§ [44]	User-based	Mesh	Social	✓	✓	✓	✓			
ShadowWalker§ [104]	User-based	DHT	P2P	✓	✓	✓				
Social Applications										
Diaspora‡ [51]	User-based	Stratified	Federated						✓	✓
X-Vine§ [105]	User-based	DHT	Social	✓	✓	✓				
Auditable Systems										
CONIKS† [101]	User-independent	Stratified	Federated±			✓			✓	✓
Enigma† [189]	User-based	Super-Node	Federated±	✓	✓					
Certificate Transparency‡ [92]	User-independent	Stratified	Federated±						✓	✓
Helios‡ [1]	User Independent	Super-Node	Federated±	✓	✓	✓			✓	✓

✓= provides property, = does not provide property; § = academic proposal, † = prototype implemented, ‡ = deployed; ± = publicly auditable

- *Widely deployed systems either are user-independent federated systems or user-based DHT-based systems, both without advanced privacy properties.*
- *Hybrid and stratified systems such as Tor provide provide advanced privacy properties at the cost of centralized assumptions.*
- *The space of ad-hoc, mesh, and covert designs is under-explored.*

4 Future Research Lines

4.1 Address Decentralization's Shortcomings

To build the next generation of decentralized systems, good will, slogans, and demands are not enough. What is needed is a clear research plan. A number of designs we review consider decentralization as a goal and virtue in itself and do too little to address the inherent challenge of maintaining privacy properties and deployment with high availability. In particular we studied in Section 3.4 a number of those challenges: 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 are serious and real threats, and not acknowledging them and confronting them head on leads to weak systems that cannot credibly compete with centralized solutions. This is demonstrated by the failure of Ethereum to promptly address the DAO vulnerability [48]. Indeed, decentralization in the style of early BitTorrent simply ends up being an inefficient way to do redundancy and availability without a centralized authority — and with no credible privacy properties. Likewise, Bitcoin and Ethereum provides this style of decentralization with the addition of integrity but their simplistic accountability designs harms privacy. Therefore, more research is required looking at systems such as Tor and Bitcoin as platforms rather than purely as channels, including understanding their interfaces, performance, quality of service guarantees and the privacy properties as a whole system in order to deliver better privacy properties.

Availability without centralization is a key promise of decentralized systems, but often fails when the system grows. The most important engineering challenge of those reviewed is that decentralized systems often do not scale and are inefficient in comparison to centralized systems. In practice, in a world with limited resources and investment, inefficient decentralization leads to a failure of decentralization. This problematic dynamic is 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 network of miners grows the less it scales, as all miners need to detect and verify new blocks and transactions. Even worse, Ethereum smart contracts are executed on each node in the network. In both Bitcoin and Ethereum, as the number of nodes grows, the system gets slower. Due to this unfortunate design flaw, Bitcoin and Ethereum will face serious issues when scaling without major design changes that accountability as such does not address. We can be assured the current generation of attempts to "re-decentralize" the Internet will fail without more research on how to scale efficiently.

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 prevent Sybil attacks and establish privacy preserving reputation to curtail abuse; 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 mere proofs of concepts.

Even with motivated users, human fallibility must be addressed realistically. Decentralization advocates desire of users to return to a 'lost golden age' of self-hosting services, as in the 're-decentralize' project [128]. However, the popularity of services like Facebook and Gmail shows that most do not have the time or skills to host decentralized nodes unless a powerful incentive exists such as file-sharing. Worse, users may not be qualified at protecting their own systems, when even most skilled professional administrators cannot. Building successful decentralized systems that do not betray the security and privacy of their users is hard, and entails much more than tacking a blockchain or P2P network to a pre-existing problem, but also has to take into account platform security and ease of user operations.

4.2 Develop Design and Evaluation Strategies

Systems that claim to be decentralized today simply often use the adjective in an informal manner, resulting in decentralized "snake oil", as is the case for some blockchain-based start-ups. Unlike formal security definitions, information-theoretic definitions of anonymity, and differential privacy, there are no coherent quantitative metrics to characterize decentralization. Aside from having a common definition of the privacy and security properties, decentralization engineering also requires the development of design strategies that measure both decentralization and its effect on the properties systematized earlier. More often than not, properties are neglected, rarely mentioned or evaluated, including the impact of decentralization and availability. Section 3.1, for instance, illustrates the variety of options in this design space.

Beyond the impact of decentralization on availability, a key missing piece is a systematic means for evaluating the privacy and security properties provided by a given decentralization system. As we evidence, decentralization can support privacy in many ways (Section 3.3), as well as supporting other properties too (Section 3.2). We observe that systems are often designed with one particular privacy goal in mind, which is frequently redefined to suit the design, and system designers tend to resort to ad-hoc evaluation. A particular case in which a lack of systematic evaluation has great impact in terms of understanding the protection provided by decentralized system is the case of compound systems (i.e, systems that combine different schemes to try to improve overall protection); or the case where systems are deployed in environments with different characteristics than those assumed in their design. In decentralized systems, it is not granted that the protection of the whole is greater or equal than the sum of the parts. In fact, the inverse may hold: combining different decentralized systems with different assumptions may violate the properties each system guarantees by itself. For example, while a user may assume using BitTorrent over Tor provides anonymity for file-sharing, in fact the reverse holds: Tor provides no anonymity to UDP-based systems like BitTorrent, and users can even be deanonymized by virtue of running BitTorrent [93]. In other words, systems to not exist in a vacuum. Their analysis and evaluation needs to account for interactions with their environment or other systems.

A similar trend is observed in terms of measuring the severity of disadvantages introduced by decentralization. Though, as we show in Section 3.4, many weaknesses arise from decentralizing, few works evaluate their implications, or do so in a design specific way that is difficult to extrapolate to other systems. As a result it is extremely difficult to compare systems and find promising new directions. This slows the development of robust decentralized systems by obscuring good design decisions. For example, in many systems there is a trade-off between privacy and availability.

Further work is also required to radically simplify the deployment and management of "real-world" 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. This gap negatively affects developer's productivity and makes the engineering and maintenance of decentralized systems very expensive. Building toolchains that support easy management – 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 *reproducible builds* [131] to address security concerns that may be generalized.

Key Research Questions for Decentralization.

- *Are there generalized techniques to provide privacy and integrity properties for decentralized systems without damaging availability?*
- *Can we develop systematic techniques to evaluate decentralized systems both in isolation and when they are deployed in different environments?*
- *How can human users be incentivised to work in a decentralized manner?*
- *How do real-world deployment of decentralization lead to scalability challenges that change the desired properties and defeat decentralization?*
- *Can we develop a mathematical metric to define degrees of decentralization?*

In the next section we will provide provisional answers to these questions to guide future research. These answers will be based on the observations built in previous sections.

5 Conclusions: Towards Full Decentralization

Availability, Privacy, and Integrity. Our analysis points to some fundamental trade-off between availability, privacy, and integrity in decentralized systems: A good design for one is an unsafe design pattern for another. Systems use a wide variety of infrastructure, network topology, and authority relation choices (as systematized in Table 1). Three widely deployed decentralized systems demonstrate a different set of design goals. Bitcoin comes with high-integrity at the cost of a public ledger with little privacy. Tor routers 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 to its users against powerful adversaries.

We believe it is not pre-ordained that there is a trade-off between privacy, availability, and integrity in decentralized systems by virtue of using advanced cryptographic techniques. Unlike Bitcoin, Zerocash[18] combines both privacy and integrity using zero-knowledge proofs. Likewise, many academic systems, such as Drac[44], tackle traffic analysis to defend privacy in a P2P network. Simply put, advanced techniques for providing everything from dummy traffic for anonymity to succinct zero-knowledge proofs are not yet part of the toolbox for many decentralized system engineers.

Interdisciplinarity. Reviewing the literature reveals that to build good secure privacy-preserving decentralized systems, one needs:

- Expertise in building *distributed systems*, as decentralized systems are by definition distributed.
- Knowledge of modern *cryptography*, as complex cryptographic protocols are necessary to achieve simultaneously privacy, integrity and availability.
- An understanding of *mechanism design, game theory and sociology* to motivate cooperation amongst possibly selfish actors.

The focus on social incentive structures is usually left out, and thus most decentralized systems do not gain real-world wide deployment. In general, the involvement of nodes in decentralized systems varies and this is usually mirrored in the power allowed to authorities, as well as in inter-node relationships that reflect social behavior. Some designs assume centralized components, for better availability and performance. Others push for sheer decentralization, in pursuit of resilience to censorship and network outages. Are these design choices often social or political rather than technical? Most designs, though, fall somewhere in the middle and generally impose cryptographic techniques and rely on real-world dynamics in order to defend against adversarial nodes. Certainly, the way decentralization is achieved affects the privacy of the users and thus their behavior. It falls upon decentralized system designers to achieve satisfactory performance and deployability, while taking into account not just the technical but the necessary social structure of the system.

Real-world Scalability. From our study of the literature, we have shown that a number of key functions of decentralized systems often fall-back to centralized models in practice for scalability, even when unnecessary. First, network directories, key management, and naming often remain centralized. Thus, the there is a need to design of collective high-integrity and re-usable infrastructures to support directories, node discovery, and key exchange. These mechanisms need to scale up and remain decentralized, while not being open to corruption or inconsistencies.

Second, reputation and abuse control often require either centralized entities, or building on pre-existing social networks in user-based infrastructure. 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.

Third, it is important to make credible assumptions about the platform security and computing environment of end-users or other devices. It is too facile to heavily rely on end-user systems keeping secret keys and data, and ignore that they are often compromised. Achieving perfect end-point security is an ambitious goal in and of itself – and so needed but beyond the strict remit of building secure decentralized systems. Decentralized 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 or silently under those conditions.

Defining Decentralization. In general, decentralized systems are networks. Yet as shown by the difference between network topologies for routing and the relationships of authority, a decentralized network is not simply a single network, but multiple kinds of networks

connected on different levels of abstraction. Worse, the overly simplified models of decentralization presented in many papers and research prototypes do not take into account the changes produced by real-world usage into account. As shown by BitTorrent, simple decentralized networks tend to evolve from P2P into super-node systems. In general, as a system scales there is a tendency towards distribution, but not decentralization, in order to maintain efficiency. Using network science, one can show simple models such as random graphs with basic mechanism design such as preferential attachment scale into small-world systems over time, and these systems often simply transform into a federated client-server architecture or a simple centralized distributed system. In order to maintain decentralization as an emergent property, it appears that advanced hybrid and stratified system, e.g. Tor, are necessary to "unnaturally" maintain decentralization and the relevant privacy properties. Yet, the Tor network has many centralized technical (complete network information by directory authorities) and social assumptions (control by a core group of developers). The key point of a real measure of decentralization should be to take these more stratified designs into account. An ideal decentralized system would remove all centralized assumptions while maintaining the needed security and privacy properties.

The ultimate bet of decentralized systems is still open: is being vulnerable to a (possibly random) subset of decentralized authorities better than being vulnerable to a single centralized authority? Decentralization seems to be the result of a breakdown in trust in centralized institutions, but we do not yet understand how to build decentralized social institutions to support decentralized technical systems despite the promises of Bitcoin to produce algorithmic monetary policy, or the promise of Ethereum to support modern civilization with scripts with dubious security properties. Decentralization is a hard problem, but the fact that it is technically amendable to advanced techniques from distributed systems and cryptography should indicate that the social questions at the heart of decentralization are not unsolvable.

Acknowledgements. The authors would like to thank the reviewers for insightful comments that helped improving the paper, in particular Prateek Mittal for acting as shepherd. This work is supported by the EU H2020 project NEXTLEAP (GA 688722).

References

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

[2] M. Akhoondi, C. Yu, and H. V. Madhyastha. LASTor: A low-latency as-aware tor client. In *IEEE Symposium on Security and Privacy*, 2012.

[3] D. Alhadidi, N. Mohammed, B. C. M. Fung, and M. Debbabi. Secure distributed framework for achieving ϵ-differential privacy. In *12th Privacy Enhancing Technologies Symposium*, 2012.

[4] M. AlSabah, K. S. Bauer, and I. Goldberg. Enhancing Tor's performance using real-time traffic classification. In *19th ACM Conference on Computer and Communications Security*, 2012.

[5] M. AlSabah, K. S. Bauer, I. Goldberg, D. Grunwald, D. McCoy, S. Savage, and G. M. Voelker. DefenestraTor: Throwing Out Windows in Tor. In *11th Privacy Enhancing Technologies Symposium*, 2011.

[6] A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004. https://law.resource.org/pub/us/case/reporter/F3/239/239.F3d.1004.00-16403.00-16401.html, 2001. Last accessed: September 27, 2017.

[7] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, 2002.

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

[9] P. S. Andre. IETF RFC 6120 Extensible Messaging and Presence Protocol (xmpp): Core. https://www.ietf.org/rfc/rfc6120.txt, 2011. Last accessed: September 27, 2017.

[10] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek. Secure multiparty computations on Bitcoin. In *IEEE Symposium on Security and Privacy*, 2014.

[11] M. S. Artigas and P. G. López. On routing in Distributed Hash Tables: Is reputation a shelter from malicious behavior and churn? In *9th IEEE Conference on Peer-to-Peer Computing*, pages 31–40, 2009.

[12] M. Backes, A. Kate, S. Meiser, and E. Mohammadi. (nothing else) MATor(s): Monitoring the anonymity of Tor's path selection. In *21st ACM Conference on Computer and Communications Security*, 2014.

[13] P. Baran et al. On distributed communications. *Volumes I-XI, RAND Corporation Research Documents, August*, 1964.

[14] K. S. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. C. Sicker. Low-resource routing attacks against Tor. In *ACM Workshop on Privacy in the Electronic Society*, 2007.

[15] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In *IEEE Symposium on Security and Privacy*, 2005.

[16] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *29th International Cryptology Conference Advances in Cryptology*, 2009.

[17] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, A. Lysyanskaya, and E. Rachlin. Making P2P accountable without losing privacy. In *ACM Workshop on Privacy in the*

Electronic Society, 2007.

[18] 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*, 2014.

[19] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang. The growth of diaspora-a decentralized online social network in the wild. In *IEEE Conference on Computer Communications Workshops*, 2012.

[20] A. Birgisson, J. G. Politz, Úlfar Erlingsson, A. Taly, M. Vrable, and M. Lentczner. Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud. In *Network and Distributed System Security Symposium*, 2014.

[21] A. Biryukov, D. Khovratovich, and I. Pustogarov. Deanonymisation of clients in bitcoin P2P network. In *21st ACM Conference on Computer and Communications Security*, 2014.

[22] A. Biryukov, I. Pustogarov, and R. Weinmann. Trawling for Tor Hidden Services: Detection, measurement, deanonymization. In *IEEE Symposium on Security and Privacy,*, 2013.

[23] J. Biskup and U. Flegel. Threshold-based identity recovery for privacy enhanced applications. In *7th ACM Conference on Computer and Communications Security*, 2000.

[24] BitTorrent. http://www.bittorrent.org/. Last accessed: September 27, 2017.

[25] M. Blaze, J. Feigenbaum, and A. D. Keromytis. Keynote: Trust Management for Public-Key Infrastructures (position paper). In *6th International Workshop on Security Protocols*, 1998.

[26] D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A framework for fast privacy-preserving computations. In *13th European Symposium on Research in Computer Security*. 2008.

[27] N. Borisov, G. Danezis, and I. Goldberg. DP5: A private presence service. *Proceedings on Privacy Enhancing Technologies*, 2015(2):4–24, 2015.

[28] The Briar Project. https://briarproject.org. Last accessed: September 27, 2017.

[29] S. Buchegger, D. Schiöberg, L. Vu, and A. Datta. PeerSoN: P2P social networking: early experiences and insights. In *2nd ACM EuroSys Workshop on Social Network Systems*, 2009.

[30] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *13th ACM Conference on Computer and Communications Security*, 2006.

[31] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *International Conference on the Theory and Application of Cryptographic Techniques Advances in Cryptology*, 2001.

[32] J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. In *2007 IEEE Symposium on Security and Privacy*, 2007.

[33] 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 *5th USENIX Symposium on Operating System Design and Implementation*, 2002.

[34] H. Chan and A. Perrig. Efficient security primitives derived from a secure aggregation algorithm. In *15th ACM Conference on Computer and Communications Security*, 2008.

[35] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 2008.

[36] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 1981.

[37] D. Chaum. The Dining Cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1988.

[38] Y. Chen, R. Sion, and B. Carbunar. XPay: practical anonymous payments for tor routing and other networked services. In *ACM Workshop on Privacy in the Electronic Society*, 2009.

[39] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *J. ACM*, 1998.

[40] J. Claessens, C. Díaz, C. Goemans, J. Dumortier, B. Preneel, and J. Vandewalle. Revocable anonymous access to the Internet? *Internet Research*, 2003.

[41] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[42] 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*, 2015.

[43] 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 *9th ACM Conference on Computer and Communications Security*, 2002.

[44] G. Danezis, C. Díaz, C. Troncoso, and B. Laurie. Drac: An architecture for anonymous low-volume communications. In *10th Privacy Enhancing Technologies Symposium*, 2010.

[45] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *IEEE Symposium on Security and Privacy*, 2003.

[46] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. Anderson. Sybil-resistant dht routing. In *European Symposium On Research In Computer Security*, pages 305–318. Springer, 2005.

[47] G. Danezis and P. Mittal. SybilInfer: Detecting sybil nodes using social networks. In *Network and Distributed System Security Symposium*, 2009.

[48] Critical update re: Dao vulnerability. https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/. Last accessed: September 27, 2017.

[49] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 2008.

[50] C. Decker, R. Eidenbenz, and R. Wattenhofer. Exploring and improving BitTorrent topologies. In *13th IEEE International Conference on Peer-to-Peer Computing*, 2013.

[51] diaspora*: The online social world where you are in control. https://diasporafoundation.org/. Last accessed: September 27, 2017.

[52] C. Díaz, G. Danezis, C. Grothoff, A. Pfitzmann, and P. F. Syverson. Panel Discussion - Mix Cascades Versus Peer-to-Peer: Is One Concept Superior? In *Privacy Enhancing Technologies*, pages 242–242, 2004.

[53] C. Díaz, S. J. Murdoch, and C. Troncoso. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In *10th Privacy Enhancing Technologies Symposium*, 2010.

[54] C. Diaz, O. Tene, and S. Gurses. Hero or villain: The data controller in privacy law and technologies. *Ohio St. LJ*, 74:923–963, 2013.

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

[56] R. Dingledine and N. Mathewson. Design of a blocking-resistant anonymity system. *The Tor Project, Tech. Rep*, 1, 2006.

[57] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In *13th USENIX Security Symposium*, 2004.

[58] Dot-Bit: Secure Decentralized DNS. https://bit.namecoin.info/. Last accessed: September 27, 2017.

[59] J. R. Douceur. The sybil attack. In *1st International Worksop on Peer-to-Peer Systems*, 2002.

[60] 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*, 2010.

[61] M. Edman and P. F. Syverson. AS-awareness in tor path selection. In *16th ACM Conference on Computer and Communications Security*, 2009.

[62] e-gold. http://e-gold.com/. Last accessed: September 27, 2017.

[63] T. Elahi, G. Danezis, and I. Goldberg. PrivEx: Private collection of traffic statistics for anonymous communication networks. In *21st ACM Conference on Computer and Communications Security*, 2014.

[64] C. M. Ellison. Establishing identity without certification authorities. In *6th USENIX Security Symposium*, 1996.

[65] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *9th ACM Conference on Computer and Communications Security*, 2002.

[66] Ethereum Project. https://www.ethereum.org/. Last accessed: September 27, 2017.

[67] European Data Protection Supervisor. Opinion on privacy in the digital age (march 2010): "Privacy by Design" as a key tool to ensure citizen's trust in ICTS, 2010.

[68] A. J. Feldman, A. Blankstein, M. J. Freedman, and E. W. Felten. Social networking with Frientegrity: Privacy and integrity with an untrusted provider. In *21th USENIX Security Symposium*, 2012.

[69] 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*, 2014.

[70] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *9th ACM conference on Computer and communications security*, 2002.

[71] J. Freudiger, M. H. Manshaei, J. Hubaux, and D. C. Parkes. On non-cooperative location privacy: a game-theoretic analysis. In *16th ACM Conference on Computer and Communications Security*, 2009.

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

[73] Gnutella: File sharing and distribution network. http://rfc-gnutella.sourceforge.net/. Last accessed: September 27, 2017.

[74] G. Greenwald. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan, 2014.

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

[76] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on Bitcoin's peer-to-peer network. In *24th USENIX Security Symposium*, 2015.

[77] S. Helmers. A brief history of anon.penet.fi: the legendary anonymous remailer. *CMC Magazine*, 1997.

[78] M. Herrmann and C. Grothoff. Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using I2P. In *Privacy Enhancing Technologies*, 2011.

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

[80] S. Hohenberger, S. Myers, R. Pass, and A. Shelat. ANONIZE: A large-scale anonymous survey system. In *IEEE Symposium on Security and Privacy*, 2014.

[81] I2P: The invisible internet project. https://geti2p.net/en/. Last accessed: September 27, 2017.

[82] 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*, 2003.

[83] A. Johnson, P. F. Syverson, R. Dingledine, and N. Mathewson. Trust-based anonymous communication: adversary models and routing algorithms. In *18th ACM Conference on Computer and Communications Security*, 2011.

[84] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. F. Syverson. Users get routed: traffic correlation on Tor by realistic adversaries. In *20th ACM SIGSAC Conference on Computer and Communications Security*, 2013.

[85] P. C. Johnson, A. Kapadia, P. P. Tsang, and S. W. Smith. Nymble: Anonymous IP-address blocking. In *7th Privacy Enhancing Technologies Symposium*, 2007.

[86] 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.

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

[88] A. Kapadia and N. Triandopoulos. Halo: High-assurance locate for distributed hash tables. In *Network and Distributed System Security Symposium*, 2008.

[89] G. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in Bitcoin. In *19th ACM Conference on Computer and Communications Security*, 2012.

[90] R. Kumaresan and I. Bentov. How to use Bitcoin to incentivize correct computations. In *21st ACM SIGSAC Conference on Computer and Communications Security*, 2014.

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

[92] B. Laurie. Certificate transparency. *Queue*, 2014.

[93] S. Le Blond, P. Manils, A. Chaabane, M. A. Kaafar, A. Legout, C. Castellucia, and W. Dabbous. Poster: Deanonymizing BitTorrent users on Tor. In *7th USENIX Symposium on Network Design and Implementation (NSDI'10)*, 2010.

[94] C. Lesniewski-Laas, B. Ford, J. Strauss, R. Morris, and M. F. Kaashoek. Alpaca: extensible authorization for distributed services. In *14th ACM Conference on Computer and Communications Security*, 2007.

[95] F. Lesueur, L. Mé, and V. V. T. Tong. An efficient distributed PKI for structured P2P networks. In *9th International Conference on Peer-to-Peer Computing*, 2009.

[96] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 2003.

[97] 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*, 2013.

[98] Y. Liu and J. Pan. The impact of NAT on BitTorrent-like P2P systems. In *9th International Conference on Peer-to-Peer Computing*, 2009.

[99] J. Maheswaran, D. I. Wolinsky, and B. Ford. Crypto-book: an architecture for privacy preserving online identities. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 14. ACM, 2013.

[100] J. McLachlan, A. Tran, N. Hopper, and Y. Kim. Scalable onion routing with Torsk. In *16th ACM Conference on Computer and Communications Security*, 2009.

[101] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman. CONIKS: bringing key transparency to end users. In *24th USENIX Security Symposium*, 2015.

[102] B. Mitra, F. Peruani, S. Ghose, and N. Ganguly. Analyzing the vulnerability of superpeer networks against attack. In *14th ACM Conference on Computer and Communications Security*, 2007.

[103] P. Mittal and N. Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In *15th ACM Conference on Computer and Communications Security*, 2008.

[104] P. Mittal and N. Borisov. ShadowWalker: peer-to-peer anonymous communication using redundant structured topologies. In *16th ACM Conference on Computer and Communications Security*, 2009.

[105] P. Mittal, M. Caesar, and N. Borisov. X-Vine: Secure and pseudonymous routing in DHTs using social networks. In *19th Network and Distributed System Security Symposium*, 2012.

[106] 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*, 2011.

[107] P. Mittal, C. Papamanthou, and D. Song. Preserving link privacy in social network based systems. In *20th Network and Distributed System Security Symposium(NDSS)*. Internet Society, 2013.

[108] P. Mittal, M. K. Wright, and N. Borisov. Pisces: Anonymous communication using social networks. In *20th Network and Distributed System Security Symposium*, 2013.

[109] 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*, 2010.

[110] S. J. Murdoch and R. N. M. Watson. Metrics for security and performance in low-latency anonymity systems. In *8th Privacy Enhancing Technologies Symposium*, 2008.

[111] S. J. Murdoch and P. Zielinski. Sampled traffic analysis by Internet-exchange-level adversaries. In *7th International Symposium on Privacy Enhancing Technologies*, 2007.

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

[113] A. Nambiar and M. K. Wright. Salsa: a structured approach to large-scale anonymity. In *13th ACM Conference on Computer and Communications Security (CCS*, 2006.

[114] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy*, 2009.

[115] 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.

[116] M. A. U. Nasir, S. Girdzijauskas, and N. Kourtellis. Socially-aware distributed hash tables for decentralized online social networks. In *IEEE International Conference on Peer-to-Peer Computing*, 2015.

[117] 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*, 2012.

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

[119] B. Parno, A. Perrig, and V. D. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, 2005.

[120] T. Paul, A. Famulari, and T. Strufe. A survey on decentralized Online Social Networks. *Computer Networks*, 2014.

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

[122] Pluggable transports. https://obfuscation.github.io/. Last accessed: September 27, 2017.

[123] J. Postel. IETF RFC 821 Simple Mail Transfer Protocol. https://www.ietf.org/rfc/rfc821.txt, 1982. Last accessed: September 27, 2017.

[124] 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 *5th International workshop on Peer-To-Peer Systems (IPTPS)*, 2006.

[125] T. Pulls, R. Peeters, and K. Wouters. Distributed privacy-preserving transparency logging. In *12th ACM Workshop on Privacy in the Electronic Society,*, 2013.

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

[127] M. Raya, M. H. Manshaei, M. Félegyházi, and J. Hubaux. Revocation games in ephemeral networks. In *15th ACM Conference on Computer and Communications Security*, 2008.

[128] Redecentralize.org. http://redecentralize.org/. Last accessed: September 27, 2017.

[129] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1998.

[130] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). Technical report, 2005.

[131] Reproducible Builds - Provide a verifiable path from source code to binary. https://reproducible-builds.org/. Last accessed: September 27, 2017.

[132] R. L. Rivest and B. Lampson. Sdsi-a simple distributed security infrastructure. Crypto, 1996.

[133] P. Rogaway and M. Bellare. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *14th ACM Conference on Computer and Communications Security*, 2007.

[134] 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*, 2013.

[135] J. M. Rushby. *Design and verification of secure systems*, volume 15. ACM, 1981.

[136] P. Schaar. Privacy by design. *Identity in the Information Society*, 3(2):267–274, 2010.

[137] S. Schiffner, A. Pashalidis, and E. Tischhauser. On the limits of privacy in reputation systems. In *10th ACM workshop on Privacy in the electronic society*, 2011.

[138] 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*, 2014.

[139] M. Selimi and F. Freitag. Tahoe-LAFS distributed storage service in community network clouds. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, pages 17–24, 2014.

[140] 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*, 2010.

[141] A. Shamir. How to share a secret. *Commun. ACM*, 1979.

[142] R. Sharma and A. Datta. SuperNova: Super-peers based architecture for decentralized online social networks. In *4th International Conference on Communication Systems and Networks*, 2012.

[143] M. Sherr, M. Blaze, and B. T. Loo. Scalable link-based relay selection for anonymous routing. In *9th Privacy Enhancing Technologies Symposium*, 2009.

[144] 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*, 2008.

[145] E. Sparrow, H. Halpin, K. Kaneko, and R. Pollan. LEAP: A next-generation client VPN and encrypted email provider. In *International Conference on Cryptology and Network Security*, pages 176–191. Springer, 2016.

[146] E. Stefanov and E. Shi. Multi-cloud oblivious storage. In *ACM SIGSAC Conference on Computer and Communications Security*, 2013.

[147] 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*, 2001.

[148] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti. POTSHARDS: secure long-term storage without encryption. 2007.

[149] R. Süselbeck, G. Schiele, P. Komarnicki, and C. Becker. Efficient bandwidth estimation for peer-to-peer systems. In *IEEE International Conference on Peer-to-Peer Computing*, 2011.

[150] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security & Privacy*, 1997.

[151] Taler: Taxable anonymous libre electronic reserve. https://taler.net/. Last accessed: September 27, 2017.

[152] C. Tang and I. Goldberg. An improved algorithm for tor circuit scheduling. In *17th ACM Conference on Computer and Communications Security*, 2010.

[153] A. Tran, N. Hopper, and Y. Kim. Hashing it out in public: common failure modes of DHT-based anonymity schemes. In *ACM Workshop on Privacy in the Electronic Society*, 2009.

[154] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. Blacklistable anonymous credentials: blocking misbehaving users without ttps. In *14th ACM Conference on Computer and Communications Security*, 2007.

[155] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. PEREA: towards practical TTP-free revocation in anonymous authentication. In *15th ACM Conference on Computer and Communications Security*, 2008.

[156] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith. Nymble: Blocking misbehaving users in anonymizing networks. *IEEE Trans. Dependable Sec. Comput.*, 2011.

[157] 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*, 2009.

[158] J. Verkamp and M. Gupta. Inferring mechanics of web censorship around the world. In *2nd USENIX Workshop on Free and Open Communications on the Internet*, 2012.

[159] C. Wacek, H. Tan, K. S. Bauer, and M. Sherr. An empirical evaluation of relay selection in Tor. In *20th Network and Distributed System Security Symposium*, 2013.

[160] 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*, 2014.

[161] M. Wachs, M. Schanzenbach, and C. Grothoff. A censorship-resistant, privacy-enhancing and fully decentralized name system. In *13th International Conference on Cryptology and Network Security*, 2014.

[162] M. Waldman and D. Mazières. Tangler: a censorship-resistant publishing system based on document entanglements. In *8th ACM Conference on Computer and Communications Security*, 2001.

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

[164] 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*, 2013.

[165] Q. Wang, Z. Lin, N. Borisov, and N. Hopper. rBridge: User reputation based Tor bridge distribution with privacy preservation. In *20th Network and Distributed System Security Symposium*, 2013.

[166] 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 *17th ACM Conference on Computer and Communications Security*, 2010.

[167] X. Wang, S. Chen, and S. Jajodia. Tracking anonymous peer-to-peer VoIP calls on the Internet. In *12th ACM Conference on Computer and Communications Security*, 2005.

[168] B. Wilcox-O'Hearn. Experiences deploying a large-scale emergent network. In *International Workshop on Peer-to-Peer Systems*, pages 104–110. Springer, 2002.

[169] M. Winslett, C. C. Zhang, and P. A. Bonatti. PeerAccess: a logic for distributed authorization. In *12th ACM Conference on Computer and Communications Security*, 2005.

[170] E. Wobber, M. Abadi, M. Burrows, and B. Lampson. Authentication in the taos operating system. *ACM Transactions on Computer Systems (TOCS)*, 12(1):3–32, 1994.

[171] E. Wobber, M. Abadi, M. Burrows, and B. W. Lampson. Authentication in the Taos operating system. In *14th ACM Symposium on Operating System Principles*, 1993.

[172] 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*, 2010.

[173] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson. Dissent in numbers: Making strong anonymity scale. In *10th USENIX Symposium on Operating Systems Design and Implementation*, 2012.

[174] 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*, 2002.

[175] 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.*, 2004.

[176] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliççöte, and P. K. Khosla. Survivable information storage systems. *IEEE Computer*, 2000.

[177] YaCy: The Peer to Peer Search Engine. http://yacy.net/en/index.html. Last accessed: September 27, 2017.

[178] B. Yang and H. Garcia-Molina. PPay: micropayments for peer-to-peer systems. In *10th ACM Conference on Computer and Communications*, 2003.

[179] youbroketheinternet. http://youbroketheinternet.org/. Last accessed: September 27, 2017.

[180] M. Young, A. Kate, I. Goldberg, and M. Karsten. Practical robust communication in DHTs tolerating a Byzantine adversary. In *ICDCS*, 2010.

[181] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social network defense against Sybil attacks. *IEEE/ACM Trans. Netw.*, 2010.

[182] H. Yu, P. B. Gibbons, and C. Shi. DCast: sustaining collaboration in overlay multicast despite rational collusion. In *19th ACM Conference on Computer and Communications Security*, 2012.

[183] D. J. Zage and C. Nita-Rotaru. On the accuracy of decentralized virtual coordinate systems in adversarial networks. In *14th ACM Conference on Computer and Communications Security*, 2007.

[184] 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 *9th IEEE International Conference on Peer-to-Peer Computing*, 2009.

[185] E. Zhai, D. I. Wolinsky, R. Chen, E. Syta, C. Teng, and B. Ford. Anonrep: Towards tracking-resistant anonymous reputation. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 583–596. USENIX Association, 2016.

[186] 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 *IEEE Symposium on Security and Privacy*, 2011.

[187] B. Zhu, S. Setia, and S. Jajodia. Providing witness anonymity in peer-to-peer systems. In *13th ACM Conference on Computer and Communications Security*, 2006.

[188] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *TOSN*, 2006.

[189] G. Zyskind, O. Nathan, and A. Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *CoRR*, abs/1506.03471, 2015.