# SoK: Log Based Transparency Enhancing Technologies

Alexander Hicks

**Abstract.** We outline the purpose, usefulness, and pitfalls of log based transparency enhancing technologies based on four mechanisms (logging, sanitization, release and query, and external) and threats to transparency to these mechanisms, which we illustrate with two case studies of large scale deployed systems, Certificate Transparency and cryptocurrencies.

## 1 Introduction

Transparency is often suggested as a way of identifying flaws in a system, enabling accountability, and making it more likely that flaws are rectified and their impacts mitigated. This does not, however, entail any specific meaning or way of implementing transparency, particularly in systems deployed in an environment that is adversarial to the accountability that transparency should enable. What information is revealed? In what form? By who? To whom? How? As a result, transparency does not always work as desired and is sometimes even counterproductive [167].

This paper considers transparency based on logging mechanisms. This involves technical considerations, such as logging, sanitizing, releasing and querying data, and non-technical external mechanisms that determine what can be done once transparency is in place.

**Outline of the paper** We motivate applying transparency for computer systems and give an overview of transparency and criticisms of transparency in Section 2, before outlining log based transparency enhancing technologies based on four essential mechanisms in section 3: logging, sanitization, release and query, and external mechanisms. Threats to transparency, mapped to these mechanisms, editorial control, and individual evidence, are discussed in Section 4 We consider the infrastructure that supports logging in Section 5 and the interaction between transparency and privacy in Section 6. To illustrate our discussion we provide in Section 7 two case studies of transparency systems, Certificate Transparency and cryptocurrencies. We then discuss related work in Section 8 before concluding in Section 9.

**Methodology** We have endeavoured to find publications relevant to log based transparency enhancing technologies at major security conferences like IEEE S&P, ACM CCS, NDSS, Usenix Security, PETS, and ACM FAccT, as well as in other conferences, workshops, and journals, including those in adjacent fields (e.g., HCI, STS) and other fields (e.g., Law, Philosophy, Business, and Economics) that provide a basis for thinking about transparency and computer systems. Work relating to transparency but not to log based transparency enhancing technologies (e.g., transparent machine learning) is out of scope and, therefore, not included.

## 2 A Short Overview of Transparency

If something is being done transparently then, in principle, it cannot be done badly without it being noticeable, creating an incentive to do things well if there is a high likelihood of being held to account, enabling accountability or other ethical principles (e.g., safety, welfare) [172] and mitigating information asymmetries.

There are many examples of transparency. Government open data practices [48, 49, 133, 148]. The sharing of data that allows scientific research to be reproduced. Freedom of information laws that make it possible to request information from public authorities. The GDPR [3] gives individuals the right to request a copy of their personal data that is held by a controller (Article 15) and requires that their data be "processed [...] in a transparent manner" (Article 5). Open source code, nutrition labels for datasets [69, 82, 142] and models [118], and privacy labels [91, 92].

### 2.1 Transparency matters for computer systems

No system is perfect. Designs can be flawed, implementations can have bugs and, even if we perfected the design of complex systems and (however unrealistic) formally verified them entirely, this would not prevent harms that occur because of a system that is misused or, operating as intended, applies harmful norms [79]. Information is routinely copied, aggregated, and analysed across networks operated by different parties, rendering strict enforcement mechanisms impractical compared to relying on accountability [182], especially when the use of data is context dependent; for example, in the case of an emergency that requires immediate access to medical data [61].

More generally, evaluating strict compliance with norms assumes reliable norms, despite many systems operating in grey areas [79]. As systems grow in complexity and scope of applications, the ability to evaluate systems is increasingly important, not only for auditors or regulators but also for users who may change how they interact with the system [56].

Evaluating systems is not new and system operators routinely do so internally, but this does not always work to reduce the harms of a faulty system. There can be issues with how the evaluation is done; for example, because of flawed mechanisms or metrics, and if an issue is detected the system operator may not have the incentive or the capacity to address it. On the other hand, when users are harmed they cannot necessarily detect or show faults in a system, despite having a greater incentive to do so.

Privilege over information about the system (e.g., known error logs) also means that system operators can manipulate disclosure procedures to their advantage [108]. This includes many types of systems, such as accounting systems (e.g., Horizon, linked to one of the biggest miscarriage of justice in the UK [88]), breathalysers (See Bellovin et al. [30]), and decision making systems that result in unfair harmful outcomes [22, 26].

Transparency enhancing technologies should address such issues at scale. For example, the IPCO, which audits law enforcement requests for telecommunications data in the UK, perform local inspections to produce their audit [85]. A transparency overlay on top of the data request system could allow for larger more efficient audits.

Transparency can also be seen as a tool for efficiency. Decentralized systems are often desired because they do not rely on a central party, but centralized systems can operate more efficiently. They can also make more sense logistically, for systems that either involve sensitive data that cannot be used in an encrypted form for operational reasons or simply to avoid the burden of coordinating many (sometimes unaligned) parties. A decentralized transparency overlay on top of a centralized system with a trustworthy interface between both could provide a useful compromise between the advantages of the centralized system and the lesser trust required by a decentralized system.

### 2.2 Criticisms of Transparency

**Lack of effectiveness** Transparency is assumed to enable accountability, better behaviour, and increased public trust. Criticism of this assumption is centred around the gap between the dissemination of information and its usefulness in enabling sanctions on a misbehaving party [66].

Etzioni has argued that there is little evidence supporting the view that transparency is an effective accountability mechanism [60]. Transparency is no alternative to regulation (it can only be complementary) if it only offloads the responsibility of demanding and analysing data to citizens without the resources to handle these tasks.

In the UK, the replacement of formal audits with requirements for English local authorities to publish datasets (with little contextual information) weakened accountability [62]. In countries without regulations that implement effective accountability, however, transparency can be effective at bypassing corrupt official audits [62].

The issue is that information does not, by itself, prevent or mitigate bad outcomes. For example, mandated disclosures such as nutrition labels do not prevent any nutritional harms that, in any case, are linked to many factors beyond the nutritional value of a food item. The same is likely to be true for data and privacy nutrition labels. A label stating that a dataset has flaws does not prevent anyone from using the dataset and producing a flawed model trained on that dataset.

Research on privacy labels has also shown that issues of judgement and misdirection can render transparency ineffective [6, 9]. Developers are not always well equipped to evaluate the labels they create, because privacy is not necessarily their expertise and they may not account for harms unknown to them [103]. If any harm originates from the use of a dataset or privacy-invasive system, a system operator will not be prevented from deploying such a system and may rely on labels as cover if the process that produces them can be influenced.

Yu and Robinson similarly argue about open government data that it does not imply any effect on how government works (other than publishing data) or create any accountability so any faulty process is likely to remain in place [189]. Thus, open data and transparency may be used as a trojan horse for other political goals [101].

If transparency by itself does not entail accountability, it follows that it also does not create trust. Despite greater access to information, for example, via government transparency and freedom of information, trust has not increased [106, 132, 135, 187]. If transparency only reveals systemic faults, why trust the system?

**Restricted transparency** Transparency is restricted when requesting information that is theoretically available requires specific expertise or resources (e.g., via freedom of information requests), or when bulk releases obfuscate important information [163]. Even if a party is honest, the release of information implied by transparency does not necessarily imply the effective communication and understanding of that information [132] or that the information that is released is not chosen purposefully to serve a chosen narrative [8].

These criticisms extend to transparency for black box computational systems [16, 184]. Burrell distinguishes three forms of opacity in this context, opacity as intentional corporate or state secrecy, opacity as technical illiteracy, and opacity as the way algorithms operate at the scale of application [37].

Rights such as data subject access requests also have practical issues [23]. Knowing the inputs, rules, and outcomes of a complex system may not be enough to understand its processes. Thus, auditing decisions that result from algorithms can still pose a significant challenge [119]. Even open source systems are not necessarily more or less secure than closed systems [18, 153] because there are many steps in between code being released and issues being fixed.

**Tension with privacy and confidentiality** Restrictions on transparency can also occur because of the potential privacy harms to users or confidentiality issues for businesses. This is particularly important for systems that process sensitive data, despite the fact that greater transparency about the sharing and processing of such data is desirable. For example, freedom of information requests in the UK may be refused if they involve the release of personal information that would contravene data protection principles [1, Chapter 36, Part II, Section 40]. Similarly, Uber invoked privacy concerns to impede a challenge by Uber drivers seeking to obtain information about the system that they were subject to [146]. Unless compelled to, companies are often reluctant to disclose anything that can fall under commercial confidentiality.

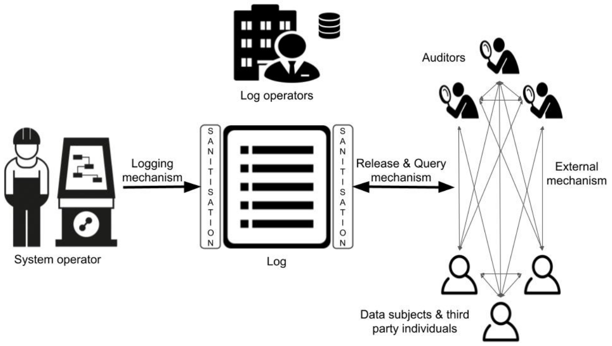

**Fig. 1.** Summary of essential mechanisms and their place in a transparency process.

## 3 Essential Mechanisms

This section introduces transparency based on four mechanisms, as illustrated in Figure 1.

Logging links the system to the log. Sanitization takes place either between the logging mechanism recording information and committing it to the log (e.g., to protect commercially confidential information that even trusted auditors may not see) or before the release and query mechanism (e.g., to allow for both privacy-preserving releases of information and access to raw data depending on the party information is released to, and enforce access control to information). The release and query mechanism relates the log to the users of transparency (auditors, data subjects, and other individuals) who then relate to each other and take action through external mechanisms.

### 3.1 Logging mechanism

Transparency requires information to be recorded and traceable [95]; for example, in the form of a chronological list of events or actions that have taken place, a record of the data used by the system to operate, or even a complete record of any byte in a current or past state [50].

For the purpose of transparency, logging mechanisms have coalesced under the notions of authenticated data structures [116, 166] and transparency overlays [40], typically in the form of Merkle Trees or blockchains designed to ensure that a log is verifiably append-only, can queried, and is consistent. Such logs satisfy *consistency* (a potentially dishonest log server cannot get away with presenting inconsistent versions of the log to different parties), *non-frameability*, (parties cannot blame the log server for misbehaviour if it has behaved honestly), and *accountability* (there exists evidence that can be used to implicate log servers that promised to include events but then did not) [40].

**Merkle Trees** Merkle Trees are binary trees based on a hash function $h$ such that each node $i$ takes the value $h_i = h(h_{left(i)}|h_{right(i)})$ based on its left and right children. Given that $h$ is collision-resistant, tamper resistance is guaranteed as modifying any node will result in a different root hash.

The leaves in Merkle trees can be arranged either in chronological order to form a history tree [45] that grows from left to right (making it easy to see new additions to the log) or lexicographically as in a prefix tree (making it easy to look-up values in a log e.g., see CONIKS [113]). When new values are added to the log, this results in a new root hash which, signed by the log server alongside the previous root hash, creates a chain of signed tree roots.

A prefix tree and a history tree can be combined to form a verifiable log-backed map [7,73,74,149]. (The prefix tree can alternatively be a hash treap [138, 143].) The prefix tree in a verifiable log-backed map, which can be in the form of a *sparse* Merkle tree pre-populated with all possible hashes (e.g., $2^{256}$ leaves to match all possible SHA-256 outputs) [46,99], serves as a map (i.e., key-value store), while the history tree is used as a log that records all signed root hashes for the map, ensuring that clients can verify that the map they are shown has also been shown to others that have audited the log. This combination of both types of Merkle trees allows for a wider range of efficient proofs than either type of Merkle tree could support on its own (i.e., append-only for the history tree, look-ups for the prefix tree) [7]. Users, however, still need to collectively check that both Merkle trees track the same keys and values.

A third Merkle tree can be added to construct an unequivocable log derived map [17], in which the first tree is a history tree log of operations, which are batched into a prefix tree that allows efficient lookups of operations, and the third tree records the root hashes of the second tree.

More recent work by Hu et al. [84] also combines history and prefix trees by proposing a history tree in which the internal nodes store the root hashes of prefix trees. At any given epoch, the root hash of the history tree summarizes the state of all prefix trees at that epoch, making it easier to monitor new changes, while the internal prefix trees make it easy to look up key values in the current epoch. Because the history and prefix trees are part of the same tree, checking that both trees track the same keys and values is easier.

Reijsbergen et al. [145] also combines several types of Merkle trees, this time a prefix tree in which all the leaves are the root of a Merkle sum tree in which nodes contain homomorphic commitments to the sum of the values of their child nodes, down to the *value* of each leaf. The prefix tree structure enables efficient lookups whilst the sum tree makes it possible to support a wider range of queries (sums, counts, averages, min/max, and quantiles) with integrity guarantees.

**Alternatives to Merkle trees** Append-only dictionaries based on bilinear accumulators [169] logarithmic-sized append-only proofs and polylogarithmic-sized lookup proofs are an alternative to Merkle trees, but high append times and memory usage mean this approach is not yet practical.

**Blockchains** Beginning with Bitcoin [126], blockchains have been used to provide a transparent record of transactions over a network. As Ethereum [186] and later projects have shown, it is possible to execute arbitrary programs (smart contracts) and record these executions, allowing applications to run transparently on top of a blockchain or to use an existing blockchain to store evidence in a tamper-resistant way [67, 72, 80, 127, 137].

Blocks in a blockchain store data (including the state of a smart contract) in Merkle trees so transparency applications that run on top of Merkle trees can use a blockchain's consensus protocol to replace the need for gossiping between clients that is required in a Merkle tree based system to guard against equivocation [34, 170].

Blockchains can be permissionless or permissioned. In a permissionless setting, it is possible to use an existing public blockchain (e.g., Ethereum). This removes the need to maintain the log (miners will do so) and the effort to write a smart contract for it may be less than deploying an entire system like Certificate Transparency (at a price determined by the underlying cryptocurrency). On the other hand, relevant events may not appear in an accurate chronological order because their inclusion will depend on miners who will primarily care about including the transactions that maximize their revenue rather than the needs of a single transparency application.

In a permissioned setting, known pre-determined parties will have to maintain the log because there is no underlying cryptocurrency, but the system could be set up to include new events to the chain as they arrive rather than at the wishes of a miner optimizing its revenue. Because all parties are known and the blockchain is more likely to be application specific than a general-purpose blockchain, this setting is also much closer to deploying a Merkle tree based system like Certificate Transparency, with the benefit (or cost) of having a consensus protocol.

### 3.2 Sanitization mechanism

The sanitization mechanism determines how logged information is processed, in plaintext (i.e., *unsanitized*), through a privacy-preserving form of data release (e.g., by adding noise or generating a synthetic data [80]), in an encrypted form to be decrypted by specific parties (e.g., trusted auditors being given access to raw data, individuals accessing individual evidence [80]), or using cryptographic techniques such as zero-knowledge proofs to assert relevant properties of the logged information without revealing the underlying data [67, 72, 137].

Access to unsanitized information may be required if no sanitization mechanism exists that is compatible with the desired transparency. For example, there may be no way to satisfy reasonable differential privacy bounds without adding excessive noise, to produce cryptographic proofs that assert the necessary properties of the logged information. In such cases, it may be necessary to permit access to unsanitized data by designated auditors, while the public is given sanitized data that can be used to verify an audit published by trusted auditors.

Identifiers (and other metadata) that allow users to verify their individual data may also need sanitization. For example, CONIKS uses a verifiable random function to produce a user identifier for the log that does not reveal the identity of the user to others [113], and more recent work has introduced append-only zero-knowledge sets that minimize the leakage from queries [39, 107].

### 3.3 Release and query mechanism

Once data is logged, it must also be possible to release it or perform queries on it. The logs we have introduced above support basic queries, but more can be done. Given a database, it is possible to store the hash of the database on a log, enabling users to verify that the database they are querying is the same as the one indicated by the log if they can download the entire database, but this does not guarantee the integrity of a query on that database.

Work on single client authenticated databases [191, 192]; that is, outsourced databases that guarantee the integrity of queries and updates to the database, has led to work combining authenticated databases with a log such as a blockchain on which a smart contract is running [139]. The log ensures consistency and allows clients to verify that the database they are querying (without needing to go through the blockchain) is the database that has been recorded on the log, allowing for a broader set of queries than what is natively supported by the log itself.

Specialized formal languages, similar to TILT [75] (developed for the GDPR transparency requirements), could also be developed to produce application-specific transparency APIs that return human-readable answers to queries.

As discussed in the case of sanitization mechanisms, data may appear in different forms to different parties. One way of doing this is simply to encrypt data under the relevant parties' public keys so that only they can decrypt the raw data, but another possibility is for the release and query mechanism to implement access control that determines who can query the log. A blockchain based system can implement access control via a smart contract that could also log queries but, for a Merkle tree based system, the access control mechanism would have to be built on top of the log.

### 3.4 External mechanisms

Transparency must work to enable action based on what it reveals. For example, by revealing system faults, it can enable aggrieved parties to take legal action, governance decisions about a platform or network [94], and the removal of parties from a network if they cause a fault [136]. This entails supporting processes such as public discussions about the system to which transparency is applied and, for practical accountability purposes, legal processes that resolve disputes about a system, or automated processes that similarly make it possible to contest theoutputs of a system. This is a key difference between tools that evaluate the compliance of a system with preset norms (e.g., the correct execution of a program) and transparency that can allow the norms enforced by a program to be contested [79].

This starts with users being able to check information that is relevant to them or being notified about such information. Notification tools [21, 63, 64, 121–124] can keep the user in the loop, without needing them to perform queries, when their explicit consent for an action is not required, but do not allow a user to contest any action that is taken.

For an action to be contested, there must first be evidence of that action, but systems often fail to produce such evidence [120]. Transparency should address this, and protocols can also play a part in spreading evidence and reaching a conclusion about evidence. The evidence must then be useful and *admissible*. For an automated process, this means it must fit the requirements of the program that evaluates it. For a non-automated process, such as a legal process, admissibility involves the data itself and also the authentication of the data, its integrity, the network over which the data is exchanged, and how it is then stored [109]. In both cases, this requires the form of the evidence and the process in which it will be used to be taken into account before it is produced.

In some cases, it may not always be clear when considering a single event, why the system failed [81]. This can require a broader discussion about the system and both individual evidence and aggregate evidence (e.g., error rates) to be considered. To act on information also requires the ability to understand that information, which can be made easier via explanations [144], context [48], and labels [82,91], although disclosure practices are not always well designed [131].

## 4 Threats Mapped to Essential Mechanisms

### 4.1 Logging

The logging mechanism relates to the system operator of the system, from which information is recorded, and the log operators that maintain the log. Assuming that the log is cryptographically secure then the threat is the ability of a malicious system operator (or whichever party is responsible for logging information) to compromise what is logged.

### 4.2 sanitization

With respect to sanitization, threats can come from either the system operator (before logging) or from data releases and queries (after logging). A sanitization step taking place before the information is committed to the log could be abused by the system operator to hide information without compromising the logging mechanism. Once information is logged, threats are posed by parties attempting to learn private information about others.

The sanitization mechanism could also be used by log operators, if sanitization is done at the interface between the log and users of transparency, or auditors, if they are given access to raw information that they sanitize for public release, to compromise the information that is released. This can be achieved either by producing sanitized information that does not relate to the original information (e.g., releasing false statistics) or relying on an honest use of a sanitization mechanism that obfuscates information (e.g., by adding noise).

**Table 1.** Threats based on editorial control (EC) and individual evidence (IE).

| **Mechanism** | **Threat** | **Affected transparency property** | **Threat actor(s)** |
|---|---|---|---|
| Logging | Compromised logging mechanism (EC, IE) | Integrity | System operator |
| | Compromised log server (EC, IE) | Integrity, Availability | Log operator |
| | Collusion between system operator and log operators (EC, IE) | Integrity, Availability | System operators, log operators |
| sanitization | Loss of privacy for data subjects | Respect of privacy and confidentiality | Users of transparency |
| | Control over logging (EC. IE) | Availability | System operator |
| | Control over release and query responses (EC, IE) | Integrity | Log operators, auditors |
| Release & Query | Access to raw data or individual evidence | Respect of privacy and confidentiality | Users of transparency |
| | Restricted releases (EC, IE) | Availability, interpretability | Log operators, auditors |
| | Constraints on queries (EC, IE) | Availability, interpretability | Log operators |
| External mechanisms | Misinformation & disinformation | Interpretability | Auditors, data subjects, third parties |
| | Lying about individual evidence (IE) | Trustworthiness | Data subjects |
| | Discrediting individual evidence (IE) | Actionability | Third party individuals |

### 4.3 Release and query

The form of the information made available by release and query mechanisms depends on the sanitization mechanism, so the threats that are specific to release and query mechanisms will target the access control it implements and the integrity of the information (sanitized or unsanitized) that is released. Given that information should broadly be released to everyone except for individual evidence (available only to data subjects) and unsanitized information (available only to trusted auditors), the threat is another party posing as an individual or trusted auditor to gain access to their privileged information. The right to access of the GDPR has been abused for this purpose [51] and to infer information about the organization answering the query [157].

If information is released without the need for queries, threats could be posed by having only a partial release of information, or a different release of information to different users. When queries are involved, the threats are that the query mechanism could constrain acceptable queries to queries that are not practically useful. It could even do so for a priori valid reasons such as limiting the privacy loss associated with queries, as in a differential private query model once the privacy budget is used up. A limited query mechanism could also serve to require an impractically large number of queries to obtain any useful information.

### 4.4 External

External mechanisms (not necessarily technical mechanisms) represent the interactions between users of transparency and the actions that they can take based on it. The threat in this case is misinformation and disinformation by anyone giving inaccurate information. This can be seen as an attack on the integrity of the information made available through transparency, which can be mitigated by ensuring that the same information (barring individual evidence) is available to all. In the specific case of individual evidence, it should be ensured that an individual cannot lie about their individual evidence, but also that they can show that any individual evidence they disclose is correct.

### 4.5 Editorial control and individual evidence

Looking at attempts to implement transparency around the world, Taylor and Kelsey found that the two general threats to transparency were editorial control i.e., the ability to control what is made transparent, and individual evidence i.e., the ability to suppress the ability of a person to find information that relates to themselves [167]. We relate this to the mechanism-specific threats we have outlined above in Table 1. Both editorial control and lack of available individual evidence can occur through the system operator (logging mechanism), and the log operators and auditors (sanitization and release and query mechanism), and affect the external mechanisms.

## 5 Transparency Infrastructure

### 5.1 Requiring and maintaining transparency

Deploying transparency requires an infrastructure that supports the logs and the storage of any data required (on or off the log). Because logs and data may be used after the system (or its operator) it originates from stops operating, they must be stored independently from the system. Thus, although a centralized approach could be sensible on the basis that only the system operator has a business reason to store that information, it may be unreliable for transparency.

Relying on distributed storage, however, raises questions about how to distribute it. Parties such as NGOs monitoring government activities or public institutions monitoring some businesses may have a strong incentive to support transparency infrastructure that relates to issues that they investigate as it directly supports their goals.

This can also be the case in commercial settings. For example, Google is responsible for the design and deployment of Certificate Transparency and because Google Chrome is the dominant browser [185], it has a direct interest in keeping Certificate Transparency operational, requires that any certificate appears in at least two logs, and operates some of the logs itself. (Google previously required one of the two logs to be a log operated by Google [110].)

Unfortunately, this example does not generalize well. In most cases, the parties that design the transparency enhancing technology may not be those that operate it or may not have the resources or the incentives to operate it successfully. Proponents of blockchains and cryptocurrencies argue that they offer the possibility of designing decentralized systems that, via mechanism design, can ensure that participants have financial incentives to maintain the system. Services such as Filecoin [96] could also offer decentralized storage.

Users could drive businesses to provide greater transparency as, for example, they react when shown the extent to which they are tracked [180] and how moderation is applied [87]. However, they often have to rely on tools set up by system operators to provide transparency that is neither complete nor understandable [20,173].

Regulation could also impose a requirement to provide transparency that is enforced through regulators like Federal Trade Commission (in the US) or a data protection authority. The European GDPR notably includes several articles concerning transparency and the German Network Enforcement Act (NetzDG) requires transparency about how unlawful content is dealt with, resulting in fines for companies such as Meta. Designated auditors may also have the power to ask for the infrastructure needed to operate a transparency enhancing technology. For example, the IPCO in the UK is tasked with auditing how law enforcement accesses telecommunications data [85] and can require that public authorities and telecommunication operators provide any assistance required to carry out audits, which could include implementing IT infrastructure [2, Section 235(2)].

Companies differ in how they implement their compliance with regulations like NetzDG [178] and are likely to differ in implementing any transparency requirement. Standardization may, therefore, be required if there is any hope of achieving reliable transparency across different types of systems, and this should be done taking into account threat models and mechanisms to deal with these threat models, and still allow enough flexibility to adapt to, for example, case-specific sanitization needs.

### 5.2 Truth

A limitation of logs is that their security properties cannot ensure that any logged data or event is true. Dealing with this depends on how the logging mechanism can ensure that the recorded value matches that of the object of interest, and what the logging mechanism actually records.

In Bitcoin, miners reach consensus about the transactions that are executed on the network (a subset of submitted transactions), but not all transactions may be logged as an address' private key can also be exchanged offline. Likewise, Certificate Transparency is transparent with respect to the set of certificates accepted by log servers, not with respect to all certificates emitted by certificate authorities. Browsers can reject certificates that do not appear in Certificate Transparency logs, however, so log servers are incentivized to log all valid certificates.

The interface between the device that records information that is logged and the log is also important. A malicious recording device would be a clear weakness so a trusted hardware interface could be used. The security of trusted hardware components may, however, be centralized if all units are the same. If one unit is broken and, for example, the attestation key leaks [174], it renders other units worthless. This is a case of weakest-link security [176], in a scenario where the weakest link is an adversary with full physical access to their hardware.

An alternative is to rely on non-colluding parties to cross-verify information.

Problems can also occur when logging physical objects e.g., paper documents, because this requires mapping physical objects to digital objects that can be authenticated once logged. Mechanisms that provide cryptographic-like authentication of physical objects do however exist e.g., authenticating a paper document based on its physical characteristics [29,36,43,76,105,150,156,171,179], allowing it to be authenticated later if required.

# 6 Balancing Transparency With Privacy

Because privacy concerns can create restrictions on transparency, privacy enhancing technologies could in principle support transparency. (In turn, transparency can help users identify privacy risks [47, 78, 104, 175].) There are two types of information to consider, aggregate information and information related to individuals (i.e., individual evidence). Aggregate information can show how the system is functioning as a whole and whether it is, for example, (un)fair, (un)biased, or error-prone, and, therefore, whether the system should be modified, shut down, or make the choice of participating in the system, but for individuals already subject to the system, it must also be possible to determine how they are personally affected by the system as, for example, a biased system will not impact all users in the same way.

For aggregate information, the goal is to not leak information about an individual in the data used to produce aggregate information. This often involves differentially private mechanisms that can satisfy data protection requirements [44, 130], and zero-knowledge proofs that allow the execution of a process to be verified without revealing anything else [25, 70, 71].

For individual information, the goal is to ensure that information is revealed only to the relevant individual.

While differentially private mechanisms and zero-knowledge proofs appear necessary to balance transparency and privacy requirements, there are concerns tied to editorial control and individual evidence that we consider here.

## 6.1 Editorial control

Editorial control encompasses any way of influencing what is or is not recorded, the format in which it is recorded, what is shared with whom, and the terms under which information is shared.

Differential privacy does this by changing the information that is shared, for example through the addition of noise or by sharing a synthetic dataset rather than the original one. While differentially private mechanisms work to preserve as much utility as possible, this is a form of editorial control that can help an adversarial system operator. This is because the addition of noise disproportionately affects less represented groups in the data [24, 168], which could be used to mask bad outcomes on minority groups or to make low-frequency faults disappear.

Because differential privacy quantifies the sensitivity of queries and a privacy budget, which limits what and how much data subjects, third-party auditors, and third-party individuals can query. It can also allow an adversarial auditor (perhaps colluding with the system operator) to exhaust the privacy budget by performing high-sensitivity queries that do not reveal anything unwanted. This can be avoided by releasing synthetic data (although not a general solution [160]) that can be queried forever, rather than relying on differentially private queries on the original data.

Zero-knowledge proofs can also act as editorial control because such proof systems are designed for a context where there is a prover (the system operator) wanting to convince a verifier (users or auditors) of a statement they choose (i.e., with editorial control), rather than to answer queries. Revealing nothing but the truth of a statement removes context from an answer to a query and requiring that any query be expressed as a provably true or false statement within the constraints of a formal language restricts the range of possible queries, and prevents open ended queries. Querying can also be made inefficient this way e.g., iterating over queries of the type "is the number of data points with attribute $\alpha$ greater than $x$", as queries must be designed without access to data.

Moreover, detecting a flawed implementation of a zero-knowledge proof system that allows counterfeit proofs to be produced is hard. Such flaws have occurred by accident [117, 165] and there is a precedent for cryptosystems with plausibly intentional flaws [31]. A malicious system operator could attempt to introduce an intentionally flawed zero-knowledge proof system that would allow them to appear compliant with any desired norm.

### 6.2 Individual evidence

Individual evidence is necessary because aggregate information can reveal issues (e.g., bias or bugs) but fail to show their impact on individuals (e.g., whether one was affected). This requires knowledge of the system's outcome for that individual, which usually will be known if the outcome had an effect (although this may not always be the case e.g., for confidential processes) and some form of counterfactual for what the outcome could have been otherwise.

For example, the covid-19 pandemic caused secondary education exams in the UK to be cancelled. Grades were awarded based on an algorithm such that the distribution of grades matched historical distributions for each school. Students who performed outside their school's historical norm could then be awarded lower or higher grades than expected for the sake of preserving the past distribution. Individual evidence in the form of teacher predicted grades, however, made it possible to determine how students were affected (e.g., a student with a high predicted grade being awarded a low grade) and the algorithmic marking scheme was replaced with predicted grades [27].

Individual evidence can also be useful when there is a dispute about whether an individual is at fault or has been a victim of a fault in a system. Human and system faults can both be low frequency events so conclusively determining which one is more likely than the other can be impractical and neither can be used to invalidate the other [81]. In such cases, individual evidence (e.g., a record of user actions and processes executed by the system in response) could make it easier to determine whether the error was human or due to a bug in the system.

The role of privacy enhancing technologies, however, is to prevent linking an individual to an input or output of the subject system's process. Differential privacy guarantees that an individual's effect on any output is bounded so that it cannot be determined their data was used to obtain that output. Zero-knowledge proofs reveal nothing but the truth of an assertion about some data or computation. If individual evidence exists, however, a zero-knowledge proof could, for example, be used to show an individual that their individual evidence was used in the computation of aggregate statistics. Without this, a system operator or auditor could use any inputs they choose to obtain valid zero-knowledge proofs for whatever they want. This means that the use of these privacy enhancing technologies to allow the release of aggregate information requires that additional mechanisms be used for individuals to obtain the individual evidence necessary to contextualize the aggregate information and the effect the system has had on them.

## 7 Case Studies

### 7.1 Certificate Transparency

SSL certificates allow a browser to verify the owner of a website. Certificates are issued by certificate authorities who can be the source of security incidents [42, 53]; for example, the DigiNotar hack [114] led to hundreds of rogue certificates being issued with DigiNotar's signing key and DigiNotar certificates being rejected by popular browsers [10, 115, 128].

Certificate Transparency [98, 164] was developed to address this type of incident, making certificate issuance transparent to reveal cases where this happens. Certificates are submitted to logs and browsers will only accept certificates that come with a signed certificate timestamp from log servers, so a malicious certificate authority cannot compromise the usefulness of the logging mechanism by not submitting certificates to logs, and collusion between a certificate authority and a log server is handled by requiring multiple signed certificate timestamps from different logs.

Significant infrastructural backing from Google, Mozilla, and Cloudflare, and free services such as Let's Encrypt [5] has led to Certificate Transparency being widely deployed. The percentage of main-frame HTTPS page loads and HTTPS connections with at least two valid signed certificate timestamps reached above 60% in 2018 for Chrome users [161].

Certificate Transparency involves little sanitization and, as such, there are privacy concerns e.g., when a browser queries a proof of inclusion in a log, it reveals the website that the user is browsing. Thus, most clients do not directly request proofs of inclusion, although solutions based on fuzzy ranges, private set intersection, and private set membership protocols have been proposed [110].

Reporting that a certificate has not been included in a log also reveals a user's browsing activity for that website. Zero-knowledge proofs could allow the browser to prove to a browser vendor (e.g., Google) that it knows a signed certificate timestamp signed by a log server (without revealing it) despite the log omitting this certificate, therefore showing that the log does not have integrity [55]. This approach has downsides, however, as it would require changes to log implementations and APIs, and obfuscate details in investigations of log misbehaviour [162].

Logs can also be used to identify websites with expired certificates that are more likely to have software that is vulnerable to CVEs [141], monitored to identify new DNS names i.e., service endpoints, that may be vulnerable to an attack, rather than inefficiently scanning the IP space [152], and mined to detect subdomains, as well as other sensitive information (e.g., names, business relationships, and unreleased products) [147]. The volume of logged certificates poses scalability issues as it makes monitoring certificates increasingly difficult [102].

External mechanisms play an important role in Certificate Transparency. Certificates must be revoked as time passes or in the event of an incident. In such a case, a human decision must be made based on the information available and the ability to act, and power is concentrated in browser vendors (e.g., Google, Mozilla, Microsoft, Apple, Brave) which are the only parties who can act at scale if Certificate Transparency reveals a malicious or compromised certificate authority by blocklisting it. Expert users can also inspect logs, but represent a tiny minority of users.

Gossip protocols should play a role in enabling clients to exchange messages containing warnings or inconsistencies between signed tree heads of logs [41], but gossiping is not widespread [68]. To work around this, gossiping could be replaced with a protocol that is not external. One way of doing this is to use a blockchain's consensus protocol for consistency [34, 170] but this can be expensive because of transaction costs and has slow finality if relying on a slow blockchain (e.g., Bitcoin or Ethereum). Another way is to rely on witnesses (e.g., the different Certificate Transparency log servers) to collectively sign a checkpoint of a log, producing some form of consensus that the log has been verified up until the checkpoint [111], but liveness issues could occur if there are too few witnesses.

### 7.2 Blockchain based cryptocurrencies

Cryptocurrencies aim to enable decentralized peer-to-peer transactions between users without centralized institutions such as banks, Paypal, and VisaNet [126]. This requires solving the problem of currency minting and double-spending such that no user can unilaterally determine the number of tokens they control or spend the same tokens multiple times. This is achieved by relying on a blockchain to log blocks of transactions mined (i.e., validated) by miners expending a scarce resource such as computational work (e.g., proof-of-work, proof-of-storage) or stake in the currency (proof-of-stake) for the right to mine blocks. The state of the blockchain is public and agreed upon by the nodes in the network through a consensus protocol, allowing anyone to track any asset on the network.

A difference between the Certificates Transparency model and the blockchain model is that miners in permissionless blockchain systems are not known and, therefore, cannot be held responsible for faults and are not trusted to provide consistent views of the blockchain [40]. This can be dealt with through penalties and *slashing* mechanisms that exist in proof-of-stake cryptocurrencies (e.g., Ethereum [57]) to directly fine or remove miners that misbehave because being elected to be a block proposer or validator requires staking funds.

Nonetheless, although it is possible to see what is going on with blockchain explorers (e.g., https://www.blockchain.com/explorer) that display the latest block information, users must download, store, and verify the entire blockchain to assure themselves they have the correct information.

As blockchains record an increasing number of transactions they become larger and more expensive to download, store, and verify. For example, the Bitcoin and Ethereum blockchains now amount to hundreds of gigabytes of data, making it difficult for most users to operate a node that independently verifies the state of the blockchain. As a result, users often run light clients that verify only block headers and the transactions inside blocks, decreasing security.

Transparency in this setting, whether at the stage of validating blocks or later auditing past transactions, is useless if it is not used to verify the system's consistency and ensure that only valid transactions are processed, so this is a problem that relates to the transparency of the system.

One approach to solving this issue is based on succinct blockchains that reduce the computational costs of verifying the blockchain [35, 93]. Recursive succinct arguments of knowledge can be produced in time proportional to the number of transactions added since the previous block and verified in constant time [35]. This allows blockchains to effectively be compressed from hundreds of gigabytes (the size of a blockchain after a few years) to a 22 kilobyte proof that verifies transactions and consensus rules, which can be verified in milliseconds.

Another approach is based on fraud proofs, which involve full nodes producing proofs of invalid transactions that light clients can efficiently verify to narrow the security gap between full nodes and light clients [13, 190]. Fraud proofs also play a role in enabling scaling solutions such as optimistic rollups on Ethereum [58], which process transactions off the main chain (reducing congestion and transaction fees) and then post compressed transaction data on the main chain. This provides enough transparency to verify the validity of transactions and produce fraud proofs for any invalid transactions. (Zero-knowledge rollups rely instead on proofs of validity to prevent invalid transactions [59].)

Another commonality with Certificate Transparency is that blockchains do not necessarily offer much in terms of sanitisation mechanisms. Bitcoin and Ethereum, for example, do not offer any privacy because users can be identified by studying the public transaction flows recorded on the blockchain [112] and trace coins linked to unwanted activity [11, 19], a practice that has been commercialized by companies such as Chainalysis, TRM, and Elliptic.

More recent systems have attempted to provide greater privacy [14] through the use of zero-knowledge proofs (e.g., Zcash [151]), ring signatures (e.g., Monero [15]), coin mixing services (e.g., Tornado cash [140], sanctioned by the US Treasure since August 2022 [4]), and network level mixing (e.g., Nym [52]). Not all attempts have been successful in achieving their privacy goals because of low adoption, design flaws, and the inherent availability of auxiliary information available via blockchain analysis that can be exploited [32, 33, 83, 89, 125, 188].

Balancing privacy goals with the goal of stopping tainted funds (e.g., stolen funds) from being laundered through, for example, mixing services has also been shown to be possible. One solution is to provably avoid mixing one's funds with those originating from addresses linked to tainted funds. Thus, the transparency that allows addresses linked to stolen funds to be identified would allow others to use privacy services without facilitating the laundering of stolen funds [28, 38]. Another possible solution is collaborative deanonymization [90], which would allow users to contribute information that helps identify a source of coins processed by a mixing service, enabling transparency that can be determined by users themselves rather than system designers.

External mechanisms also play an important role in the governance of blockchains. The blockchain publicly reveals information about miner behaviour, hacks, trace stolen funds, and so on, which has led to important debates; for example, whether the 2016 DAO hack on Ethereum should be reversed with a hard fork (splitting Ethereum and Ethereum Classic) [94], or whether the size of Bitcoin blocks should be increased (leading to Bitcoin Cash and Bitcoin SV).

Social influence also plays a role in such discussions as public figures (e.g., Vitalik Buterin for Ethereum) and influential companies (e.g., Blockstream employed many Bitcoin Core developers) can sway public opinion. In principle, anyone can suggest improvements and fork a blockchain to implement their suggested improvements and publicly showcase them. Thus, although miners have the power to enforce changes as they run the software and validate transactions, and the few developers with write access to the software repositories have privilege over the code, transparency enables some redistribution of power as discussions can be based on entirely public information.

## 8 Related Work

A number of past surveys related to transparency enhancing technologies exist. Murmann and Fischer-Hübner [123] focus on the usability of transparency enhancing technologies. Hedbom [77], Janic et al. [86], and Zimmermann [193] focus on transparency tools that can be used to help users control or verify their privacy online. Spagnuelo et al. [158, 159] look at transparency enhancing technologies in the context of providing and complying with the transparency required by the GDPR.

In contrast to these papers, our focus is not specifically on existing tools (although we survey some and consider two use cases), but more generally on how to design and build transparency enhancing technologies based on cryptographic logs under realistic threat models that consider issues of editorial control and access to individual evidence.

## 9 Conclusion

There are many use cases for log based transparency beyond Certificate Transparency and cryptocurrencies. Key transparency [39, 107, 113] is seeing increasing adoption following implementations by Google, Meta, and Apple [54, 73, 100]. Many other applications have also been proposed, including binary transparency [12, 129], decentralized authorization [17], identifying wage gaps [97], financial markets [65], legal processes [67, 72, 137], data sharing [80] and usage [154], data mining [183], inference [181], advertising [177], and open government data [134, 155].

As we have discussed, there are many challenges related to the infrastructure that would enable transparency, and balancing transparency with privacy and confidentiality concerns, that make transparency hard to deploy, particularly in cases that handle sensitive data and that may require user involvement. The two case studies we have provided, Certificate Transparency and cryptocurrencies, show how many of these challenges arise in practice for each essential mechanism and, in some cases, how they can be addressed. Several additional challenges must also be resolved for transparency enhancing technologies to be practically useful in supporting users and processes such as legal disputes, in which they will engage based on what transparency reveals, and regulations that require transparency. The design of transparency enhancing technologies should, therefore, ensure that any technological attempt to enable greater transparency focuses on making transparency not a goal in itself but a tool that serves a broader aim in the system in which it is put in place.

## References

1. Freedom of information act 2000 (2000), https://www.legislation.gov.uk/ukpga/2000/36/contents
2. Investigatory Powers Act (2016)
3. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (2016), https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679&qid=1652290481627
4. U.S. Treasury sanctions notorious virtual currency mixer Tornado Cash (2022), https://home.treasury.gov/news/press-releases/jy0916
5. Aas, J., Barnes, R., Case, B., Durumeric, Z., Eckersley, P., Flores-López, A., Halderman, J.A., Hoffman-Andrews, J., Kasten, J., Rescorla, E., et al.: Let's encrypt: an automated certificate authority to encrypt the entire web. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2473–2487 (2019)
6. Acquisti, A., Adjerid, I., Brandimarte, L.: Gone in 15 seconds: The limits of privacy transparency and control. IEEE Security & Privacy **11**(4), 72–74 (2013)
7. Adam Eijdenberg, B.L., Cutter, A.: Trillian – verifiable data structures (2017), https://github.com/google/trillian/blob/master/docs/VerifiableDataStructures.pdf
8. Adelberg, A.H.: Narrative disclosures contained in financial reports: means of communication or manipulation? Accounting and Business Research **9**(35), 179–190 (1979)

9. Adjerid, I., Acquisti, A., Brandimarte, L., Loewenstein, G.: Sleights of privacy: Framing, disclosures, and the limits of transparency. In: Proceedings of the ninth symposium on usable privacy and security. pp. 1–11 (2013)
10. Adkins, H.: An update on attempted man-in-the-middle attacks (August 2011), https://security.googleblog.com/2011/08/update-on-attempted-man-in-middle.html
11. Ahmed, M., Shumailov, I., Anderson, R.: Tendrils of crime: Visualizing the diffusion of stolen bitcoins. In: International Workshop on Graphical Models for Security. pp. 1–12. Springer (2018)
12. Al-Bassam, M., Meiklejohn, S.: Contour: A practical system for binary transparency. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 94–110. Springer (2018)
13. Al-Bassam, M., Sonnino, A., Buterin, V., Khoffi, I.: Fraud and data availability proofs: Detecting invalid blocks in light clients. In: International Conference on Financial Cryptography and Data Security. pp. 279–298. Springer (2021)
14. Almashaqbeh, G., Solomon, R.: Sok: Privacy-preserving computing in the blockchain era. Cryptology ePrint Archive (2021)
15. Alonso, K.M., et al.: Zero to monero (2020)
16. Ananny, M., Crawford, K.: Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. new media & society **20**(3), 973–989 (2018)
17. Andersen, M.P., Kumar, S., AbdelBaky, M., Fierro, G., Kolb, J., Kim, H.S., Culler, D.E., Popa, R.A.: Wave: A decentralized authorization framework with transitive delegation. In: Proceedings of the 28th USENIX Security Symposium. Univ. of California, Berkeley, CA (United States) (2019)
18. Anderson, R.: Open and closed systems are equivalent (that is, in an ideal world) (2005)
19. Anderson, R., Shumailov, I., Ahmed, M., Rietmann, A.: Bitcoin redux (2019)
20. Andreou, A., Venkatadri, G., Goga, O., Gummadi, K., Loiseau, P., Mislove, A.: Investigating ad transparency mechanisms in social media: A case study of facebook's explanations. In: NDSS 2018-Network and Distributed System Security Symposium. pp. 1–15 (2018)
21. Angulo, J., Fischer-Hübner, S., Pulls, T., Wästlund, E.: Usable transparency with the data track: a tool for visualizing data disclosures. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. pp. 1803–1808 (2015)
22. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. ProPublica **23**, 77–91 (2016)
23. Ausloos, J., Dewitte, P.: Shattering one-way mirrors. data subject access rights in practice. Data Subject Access Rights in Practice (January 20, 2018). International Data Privacy Law **8**(1), 4–28 (2018)
24. Bagdasaryan, E., Poursaeed, O., Shmatikov, V.: Differential privacy has disparate impact on model accuracy. Advances in neural information processing systems **32** (2019)
25. Bamberger, K.A., Canetti, R., Goldwasser, S., Wexler, R., Zimmerman, E.J.: Verification dilemmas in law and the promise of zero-knowledge proofs. Berkeley Technology Law Journal **37**(1) (2022)
26. Barocas, S., Selbst, A.D.: Big data's disparate impact. Calif. L. Rev. **104**, 671 (2016)

27. BBC: A-levels and GCSEs: U-turn as teacher estimates to be used for exam results (August 2020), https://www.bbc.co.uk/news/uk-53810655
28. Beal, J., Fisch, B.: Derecho: Privacy pools with proof-carrying disclosures. Cryptology ePrint Archive (2023)
29. van Beijnum, F., van Putten, E., Van der Molen, K., Mosk, A.: Recognition of paper samples by correlation of their speckle patterns. arXiv preprint physics/0610089 (2006)
30. Bellovin, S.M., Blaze, M., Landau, S., Owsley, B.: Seeking the source: Criminal defendants' constitutional right to source code. Ohio St. Tech. LJ **17**, 1 (2021)
31. Bernstein, D.J., Lange, T., Niederhagen, R.: Dual ec: A standardized back door. In: The New Codebreakers, pp. 256–281. Springer (2016)
32. Biryukov, A., Feher, D.: Privacy and linkability of mining in zcash. In: 2019 IEEE Conference on Communications and Network Security (CNS). pp. 118–123. IEEE (2019)
33. Biryukov, A., Feher, D., Vitto, G.: Privacy aspects and subliminal channels in zcash. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1813–1830 (2019)
34. Bonneau, J.: Ethiks: Using ethereum to audit a coniks key transparency log. In: International Conference on Financial Cryptography and Data Security. pp. 95–105. Springer (2016)
35. Bonneau, J., Meckler, I., Rao, V., Shapiro, E.: Mina: Decentralized cryptocurrency at scale. New York Univ. O (1) Labs, New York, NY, USA, Whitepaper pp. 1–47 (2020)
36. Buchanan, J.D., Cowburn, R.P., Jausovec, A.V., Petit, D., Seem, P., Xiong, G., Atkinson, D., Fenton, K., Allwood, D.A., Bryan, M.T.: 'fingerprinting'documents and packaging. Nature **436**(7050), 475–475 (2005)
37. Burrell, J.: How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data & Society **3**(1), 2053951715622512 (2016)
38. Buterin, V., Illum, J., Nadler, M., Schär, F., Soleimani, A.: Blockchain privacy and regulatory compliance: Towards a practical equilibrium. Available at SSRN (2023)
39. Chase, M., Deshpande, A., Ghosh, E., Malvai, H.: Seemless: Secure end-to-end encrypted messaging with less trust. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 1639–1656 (2019)
40. Chase, M., Meiklejohn, S.: Transparency overlays and applications. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 168–179. ACM (2016)
41. Chuat, L., Szalachowski, P., Perrig, A., Laurie, B., Messeri, E.: Efficient gossip protocols for verifying the consistency of certificate logs. In: 2015 IEEE Conference on Communications and Network Security (CNS). pp. 415–423. IEEE (2015)
42. Clark, J., Van Oorschot, P.C.: Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements. In: 2013 IEEE Symposium on Security and Privacy. pp. 511–525. IEEE (2013)
43. Clarkson, W., Weyrich, T., Finkelstein, A., Heninger, N., Halderman, J.A., Felten, E.W.: Fingerprinting blank paper using commodity scanners. In: 2009 30th IEEE Symposium on Security and Privacy. pp. 301–314. IEEE (2009)
44. Cohen, A., Nissim, K.: Towards formalizing the gdpr's notion of singling out. Proceedings of the National Academy of Sciences **117**(15), 8344–8352 (2020)
45. Crosby, S.A., Wallach, D.S.: Efficient data structures for tamper-evident logging. In: USENIX Security Symposium. pp. 317–334 (2009)

46. Dahlberg, R., Pulls, T., Peeters, R.: Efficient sparse merkle trees. In: Nordic Conference on Secure IT Systems. pp. 199–215. Springer (2016)
47. De, S.J., Le Métayer, D.: Privacy risk analysis to enable informed privacy settings. In: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 95–102. IEEE (2018)
48. Degbelo, A., Kauppinen, T.: Increasing transparency through web maps. In: Companion Proceedings of the The Web Conference 2018. pp. 899–904 (2018)
49. Department for Digital, Culture, Media & Sport: Uk open government national action plan (July 2019), https://www.data.gov/
50. Devecsery, D., Chow, M., Dou, X., Flinn, J., Chen, P.M.: Eidetic systems. In: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14). pp. 525–540 (2014)
51. Di Martino, M., Robyns, P., Weyts, W., Quax, P., Lamotte, W., Andries, K.: Personal information leakage by abusing the {GDPR}'right of access'. In: Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019) (2019)
52. Diaz, C., Halpin, H., Kiayias, A.: The nym network (2021)
53. Durumeric, Z., Kasten, J., Bailey, M., Halderman, J.A.: Analysis of the https certificate ecosystem. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 291–304 (2013)
54. Engineering, A.S., Architecture: Advancing imessage security: imessage contact key verification (2023), https://security.apple.com/blog/imessage-contact-key-verification/
55. Eskandarian, S., Messeri, E., Bonneau, J., Boneh, D.: Certificate transparency with privacy. Proceedings on Privacy Enhancing Technologies **2017**(4), 329–344 (2017)
56. Eslami, M., Vaccaro, K., Lee, M.K., Elazari Bar On, A., Gilbert, E., Karahalios, K.: User attitudes towards algorithmic opacity and transparency in online reviewing platforms. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–14. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300724, https://doi.org/10.1145/3290605.3300724
57. Ethereum: Proof-of-stake rewards and penalties (2022), https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/
58. Ethereum: Optimistic rollups (2023), https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/
59. Ethereum: Zero-knowledge rollups (2023), https://ethereum.org/en/developers/docs/scaling/zk-rollups/
60. Etzioni, A.: Is transparency the best disinfectant? Journal of Political Philosophy **18**(4), 389–404 (2010). https://doi.org/10.1111/j.1467-9760.2010.00366.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9760.2010.00366.x
61. Feigenbaum, J., Hendler, J.A., Jaggard, A.D., Weitzner, D.J., Wright, R.N.: Accountability and deterrence in online life. In: Proceedings of the 3rd International Web Science Conference. pp. 1–7 (2011)
62. Ferry, L., Eckersley, P.: Accountability and transparency: a nuanced response to etzioni. Public Administration Review **75**(1), 11 (2015)
63. Fischer-Hübner, S., Angulo, J., Karegar, F., Pulls, T.: Transparency, privacy and trust–technology for tracking and controlling my data disclosures: Does this work? In: IFIP International Conference on Trust Management. pp. 3–14. Springer (2016)

64. Fischer-Hübner, S., Angulo, J., Pulls, T.: How can cloud users be supported in deciding on, tracking and controlling how their data are used? In: IFIP PrimeLife International Summer School on Privacy and Identity Management for Life. pp. 77–92. Springer (2013)
65. Flood, M.D., Katz, J., Ong, S.J., Smith, A.: Cryptography and the economics of supervisory information: Balancing transparency and confidentiality (2013)
66. Fox, J.: The uncertain relationship between transparency and accountability. Development in practice **17**(4-5), 663–671 (2007)
67. Frankle, J., Park, S., Shaar, D., Goldwasser, S., Weitzner, D.: Practical accountability of secret processes. In: 27th {USENIX} Security Symposium ({USENIX} Security 18). pp. 657–674 (2018)
68. Gasser, O., Hof, B., Helm, M., Korczynski, M., Holz, R., Carle, G.: In log we trust: revealing poor security practices with certificate transparency logs and internet measurements. In: International Conference on Passive and Active Network Measurement. pp. 173–185. Springer (2018)
69. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Iii, H.D., Crawford, K.: Datasheets for datasets. Communications of the ACM **64**(12), 86–92 (2021)
70. Goldreich, O., Micali, S., Wigderson, A.: How to prove all np statements in zero-knowledge and a methodology of cryptographic protocol design. In: Conference on the Theory and Application of Cryptographic Techniques. pp. 171–185. Springer (1986)
71. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on computing **18**(1), 186–208 (1989)
72. Goldwasser, S., Park, S.: Public accountability vs. secret laws: Can they coexist? a cryptographic proposal. In: Proceedings of the 2017 on Workshop on Privacy in the Electronic Society. pp. 99–110 (2017)
73. Google: Key transparency (2017), https://github.com/google/keytransparency/
74. Google: Trillian (2017), https://github.com/google/trillian
75. Grünewald, E., Pallas, F.: Tilt: A gdpr-aligned transparency information language and toolkit for practical privacy engineering. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 636–646 (2021)
76. Guarnera, F., Allegra, D., Giudice, O., Stanco, F., Battiato, S.: A new study on wood fibers textures: documents authentication through lbp fingerprint. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 4594–4598. IEEE (2019)
77. Hedbom, H.: A survey on transparency tools for enhancing privacy. In: IFIP Summer School on the Future of Identity in the Information Society. pp. 67–82. Springer (2008)
78. Henze, M., Kerpen, D., Hiller, J., Eggert, M., Hellmanns, D., Mühmer, E., Renuli, O., Maier, H., Stüble, C., Häußling, R., et al.: Towards transparent information on individual cloud service usage. In: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp. 366–370. IEEE (2016)
79. Hicks, A.: Transparency, compliance, and contestability when code is(n't) law. In: Proceedings of the 2022 New Security Paradigms Workshop. p. 130–142. NSPW '22, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3584318.3584854, https://doi.org/10.1145/3584318.3584854
80. Hicks, A., Mavroudis, V., Al-Bassam, M., Meiklejohn, S., Murdoch, S.J.: Vams: Verifiable auditing of access to confidential data. arXiv preprint arXiv:1805.04772 (2018)

81. Hicks, A., Murdoch, S.J.: Transparency enhancing technologies to make security protocols work for humans. In: Cambridge International Workshop on Security Protocols. pp. 3–10. Springer (2019)
82. Holland, S., Hosny, A., Newman, S., Joseph, J., Chmielinski, K.: The dataset nutrition label: A framework to drive higher data quality standards. arXiv preprint arXiv:1805.03677 (2018)
83. Hong, Y., Kwon, H., Lee, J., Hur, J.: A practical de-mixing algorithm for bitcoin mixing services. In: Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts. pp. 15–20 (2018)
84. Hu, Y., Hooshmand, K., Kalidhindi, H., Yang, S.J., Popa, R.A.: Merklê 2: A low-latency transparency log system. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 285–303. IEEE (2021)
85. Investigatory Powers Commitioner's Office: Annual report 2018 (2020), https://ipco.org.uk/
86. Janic, M., Wijbenga, J.P., Veugen, T.: Transparency enhancing tools (tets): an overview. In: 2013 Third Workshop on Socio-Technical Aspects in Security and Trust. pp. 18–25. IEEE (2013)
87. Jhaver, S., Bruckman, A., Gilbert, E.: Does transparency in moderation really matter? user behavior after content removal explanations on reddit. Proceedings of the ACM on Human-Computer Interaction **3**(CSCW), 1–27 (2019)
88. Jolly, J.: Uk government sets aside up to £233m to cover post office payouts (2021), https://www.theguardian.com/business/2021/jul/25/uk-government-sets-aside-up-to-233m-to-cover-post-office-payouts
89. Kappos, G., Yousaf, H., Maller, M., Meiklejohn, S.: An empirical analysis of anonymity in zcash. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 463–477 (2018)
90. Keller, P., Florian, M., Böhme, R.: Collaborative deanonymization. In: International Conference on Financial Cryptography and Data Security. pp. 39–46. Springer (2021)
91. Kelley, P.G., Bresee, J., Cranor, L.F., Reeder, R.W.: A" nutrition label" for privacy. In: Proceedings of the 5th Symposium on Usable Privacy and Security. pp. 1–12 (2009)
92. Kelley, P.G., Cesca, L., Bresee, J., Cranor, L.F.: Standardizing privacy notices: an online study of the nutrition label approach. In: Proceedings of the SIGCHI Conference on Human factors in Computing Systems. pp. 1573–1582 (2010)
93. Kiayias, A., Miller, A., Zindros, D.: Non-interactive proofs of proof-of-work. In: International Conference on Financial Cryptography and Data Security. pp. 505–522. Springer (2020)
94. Kiffer, L., Levin, D., Mislove, A.: Stick a fork in it: Analyzing the ethereum network partition. In: Proceedings of the 16th ACM Workshop on Hot Topics in Networks. pp. 94–100 (2017)
95. Kroll, J.A.: Outlining traceability: A principle for operationalizing accountability in computing systems. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 758–771 (2021)
96. Labs, P.: Filecoin: A decentralized storage network (2017), https://filecoin.io/filecoin.pdf
97. Lapets, A., Jansen, F., Albab, K.D., Issa, R., Qin, L., Varia, M., Bestavros, A.: Accessible privacy-preserving web-based data analysis for assessing and addressing economic inequalities. In: Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies. pp. 1–5 (2018)

98. Laurie, B.: Certificate transparency. Communications of the ACM **57**(10), 40–46 (2014)
99. Laurie, B., Kasper, E.: Revocation transparency. Google Research, September **33** (2012)
100. Lawlor, S., Lewi, K.: Deploying key transparency at whatsapp (2023), https://engineering.fb.com/2023/04/13/security/whatsapp-key-transparency/
101. Levy, K.E., Johns, D.M.: When open data is a trojan horse: The weaponization of transparency in science and governance. Big Data & Society **3**(1), 2053951715621568 (2016). https://doi.org/10.1177/2053951715621568, https://doi.org/10.1177/2053951715621568
102. Li, B., Lin, J., Li, F., Wang, Q., Li, Q., Jing, J., Wang, C.: Certificate transparency in the wild: Exploring the reliability of monitors. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2505–2520 (2019)
103. Li, T., Reiman, K., Agarwal, Y., Cranor, L.F., Hong, J.I.: Understanding challenges for developers to create accurate privacy nutrition labels. In: CHI Conference on Human Factors in Computing Systems. pp. 1–24 (2022)
104. Li, Y., Chen, F., Li, T.J.J., Guo, Y., Huang, G., Fredrikson, M., Agarwal, Y., Hong, J.I.: Privacystreams: Enabling transparency in personal data processing for mobile apps. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1**(3), 1–26 (2017)
105. Li, Z., Rathore, A.S., Song, C., Wei, S., Wang, Y., Xu, W.: Printracker: Fingerprinting 3d printers using commodity scanners. In: Proceedings of the 2018 ACM sigsac conference on computer and communications security. pp. 1306–1323 (2018)
106. Mabillard, V., Pasquier, M.: Transparency and trust in government (2007–2014): A comparative study. NISPAcee Journal of Public Administration and Policy **9**(2), 69–92 (2016)
107. Malvai, H., Kokoris-Kogias, L., Sonnino, A., Ghosh, E., Oztürk, E., Lewi, K., Lawlor, S.: Parakeet: Practical key transparency for end-to-end encrypted messaging. Cryptology ePrint Archive (2023)
108. Marshall, P., Christie, J., Ladkin, B., Littlewood, B., Mason, S., Newby, M., Rogers, J., Thimbleby, H., Thomas, M.: Recommendations for the probity of computer evidence. Digital Evidence and Electronic Signature Law Review **18** (2020)
109. Mason, S., Seng, D.: Electronic Evidence and Electronic Signatures. University of London Press (2021)
110. Meiklejohn, S., DeBlasio, J., O'Brien, D., Thompson, C., Yeo, K., Stark, E.: Sok: Sct auditing in certificate transparency. arXiv preprint arXiv:2203.01661 (2022)
111. Meiklejohn, S., Kalinnikov, P., Lin, C.S., Hutchinson, M., Belvin, G., Raykova, M., Cutter, A.: Think global, act local: Gossip and client audits in verifiable data structures. arXiv preprint arXiv:2011.04551 (2020)
112. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 127–140 (2013)
113. Melara, M.S., Blankstein, A., Bonneau, J., Felten, E.W., Freedman, M.J.: {CONIKS}: Bringing key transparency to end users. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 383–398 (2015)
114. Van der Meulen, N.: Diginotar: Dissecting the first dutch digital disaster. Journal of Strategic Security **6**(2), 46–58 (2013)

115. Microsoft: Fraudulent digital certificates could allow spoofing (August 2011), https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2011/2607712?redirectedfrom=MSDN
116. Miller, A., Hicks, M., Katz, J., Shi, E.: Authenticated data structures, generically. ACM SIGPLAN Notices **49**(1), 411–423 (2014)
117. Miller, J.: Coordinated disclosure of vulnerabilities affecting girault, bulletproofs, and plonk (2022), https://blog.trailofbits.com/2022/04/13/part-1-coordinated-disclosure-of-vulnerabilities-affecting-girault-bulletproofs-and-plonk/
118. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T.: Model cards for model reporting. In: Proceedings of the conference on fairness, accountability, and transparency. pp. 220–229 (2019)
119. Mittelstadt, B.: Automation, algorithms, and politics| auditing for transparency in content personalization systems. International Journal of Communication **10**, 12 (2016)
120. Murdoch, S.J., Anderson, R.: Security protocols and evidence: Where many payment systems fail. In: International Conference on Financial Cryptography and Data Security. pp. 21–32. Springer (2014)
121. Murmann, P.: Usable transparency for enhancing privacy in mobile health apps. In: Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct. pp. 440–442 (2018)
122. Murmann, P.: Eliciting design guidelines for privacy notifications in mhealth environments. International Journal of Mobile Human Computer Interaction (IJMHCI) **11**(4), 66–83 (2019)
123. Murmann, P., Fischer-Hübner, S.: Tools for achieving usable ex post transparency: a survey. IEEE Access **5**, 22965–22991 (2017)
124. Murmann, P., Reinhardt, D., Fischer-Hübner, S.: To be, or not to be notified: eliciting privacy notification preferences for online mhealth services. In: ICT Systems Security and Privacy Protection: 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings 34. pp. 209–222. Springer (2019)
125. Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Hogan, K., Hennessey, J., Miller, A., Narayanan, A., Christin, N.: An empirical analysis of traceability in the monero blockchain. Proceedings on Privacy Enhancing Technologies **2018**(3), 143–163 (2018). https://doi.org/doi:10.1515/popets-2018-0025, https://doi.org/10.1515/popets-2018-0025
126. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
127. Nguyen, H.L., Ignat, C.L., Perrin, O.: Trusternity: auditing transparent log server with blockchain. In: Companion Proceedings of the The Web Conference 2018. pp. 79–80 (2018)
128. Nightingale, J.: Fraudulent *.google.com certificate (August 2011), https://blog.mozilla.org/security/2011/08/29/fraudulent-google-com-certificate/
129. Nikitin, K., Kokoris-Kogias, E., Jovanovic, P., Gailly, N., Gasser, L., Khoffi, I., Cappos, J., Ford, B.: {CHAINIAC}: Proactive software-update transparency via collectively signed skipchains and verified builds. In: 26th {USENIX} Security Symposium ({USENIX} Security 17). pp. 1271–1287 (2017)
130. Nissim, K., Bembenek, A., Wood, A., Bun, M., Gaboardi, M., Gasser, U., O'Brien, D.R., Steinke, T., Vadhan, S.: Bridging the gap between computer science and legal approaches to privacy. Harv. JL & Tech. **31**, 687 (2017)
131. Norval, C., Cornelius, K., Cobbe, J., Singh, J.: Disclosure by design: Designing information disclosures to support meaningful transparency and accountability.

In: 2022 ACM Conference on Fairness, Accountability, and Transparency. pp. 679–690 (2022)

132. O Neill, O.: Transparency and the ethics of communication. In: Proceedings-British Academy. vol. 1, pp. 75–90. Oxford University Press (2006)
133. Obama, B.: Transparency and open government. Memorandum for the heads of executive departments and agencies (2009)
134. O'Hara, K.: Transparent government, not transparent citizens: a report on privacy and transparency for the cabinet office (2011)
135. O'neill, O.: A question of trust: The BBC Reith Lectures 2002. Cambridge University Press (2002)
136. O'Brien, D., Sleevi, R., Whalley, A.: Chrome's plan to distrust symantec certificates (2017), https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html
137. Panwar, G., Vishwanathan, R., Misra, S., Bos, A.: Sampl: Scalable auditability of monitoring processes using public ledgers. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2249–2266 (2019)
138. Peeters, R., Pulls, T.: Insynd: Improved privacy-preserving transparency logging. In: European Symposium on Research in Computer Security. pp. 121–139. Springer (2016)
139. Peng, Y., Du, M., Li, F., Cheng, R., Song, D.: Falcondb: Blockchain-based collaborative database. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. pp. 637–652 (2020)
140. Pertsev, A., Semenov, R., Storm, R.: Tornado cash privacy solution version 1.4 (2019)
141. Pletinckx, S., Nguyen, T.D., Fiebig, T., Kruegel, C., Vigna, G.: Certifiably vulnerable: Using certificate transparency logs for target reconnaissance. In: 2023 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE (2023)
142. Project, T.D.N.: The data nutrition project (2021), https://datanutrition.org/
143. Pulls, T., Peeters, R.: Balloon: A forward-secure append-only persistent authenticated data structure. In: European Symposium on Research in Computer Security. pp. 622–641. Springer (2015)
144. Rader, E., Cotter, K., Cho, J.: Explanations as mechanisms for supporting algorithmic transparency. In: Proceedings of the 2018 CHI conference on human factors in computing systems. pp. 1–13 (2018)
145. Reijsbergen, D., Maw, A., Yang, Z., Dinh, T.T.A., Zhou, J.: Tap: Transparent and privacy-preserving data services. arXiv preprint arXiv:2210.11702 (2022)
146. Reuters: Uber drivers consider appeal in dutch case over data access (2021), https://www.reuters.com/article/netherlands-uber-lawsuit/uber-wins-dutch-court-case-over-data-brought-by-british-drivers-idINL1N2LA1LH
147. Roberts, R., Levin, D.: When certificate transparency is too transparent: Analyzing information leakage in https domain names. In: Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society. pp. 87–92 (2019)
148. Robinson, D., Yu, H., Zeller, W.P., Felten, E.W.: Government data and the invisible hand. Yale JL & Tech. **11**, 159 (2008)
149. Ryan, M.D.: Enhanced certificate transparency and end-to-end encrypted mail. In: NDSS. pp. 1–14 (2014)
150. Samsul, W., Uranus, H.P., Birowosuto, M.: Recognizing document's originality by laser surface authentication. In: 2010 second international conference on advances in computing, control, and telecommunication technologies. pp. 37–40. IEEE (2010)

151. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE symposium on security and privacy. pp. 459–474. IEEE (2014)
152. Scheitle, Q., Gasser, O., Nolte, T., Amann, J., Brent, L., Carle, G., Holz, R., Schmidt, T.C., Wählisch, M.: The rise of certificate transparency and its implications on the internet ecosystem. In: Proceedings of the Internet Measurement Conference 2018. pp. 343–349 (2018)
153. Schryen, G.: Is open source security a myth? Communications of the ACM **54**(5), 130–140 (2011)
154. Seneviratne, O., Kagal, L.: Enabling privacy through transparency. In: 2014 Twelfth Annual International Conference on Privacy, Security and Trust. pp. 121–128. IEEE (2014)
155. Shadbolt, N., O'Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H., Hall, W., et al.: Linked open government data: Lessons from data. gov. uk. IEEE Intelligent Systems **27**(3), 16–24 (2012)
156. Sharma, A., Subramanian, L., Brewer, E.A.: Paperspeckle: microscopic fingerprinting of paper. In: Proceedings of the 18th ACM conference on Computer and communications security. pp. 99–110 (2011)
157. Singh, J., Cobbe, J.: The security implications of data subject rights. IEEE Security & Privacy **17**(6), 21–30 (2019)
158. Spagnuelo, D., Ferreira, A., Lenzini, G.: Accomplishing transparency within the general data protection regulation. In: ICISSP. pp. 114–125 (2019)
159. Spagnuelo, D., Ferreira, A., Lenzini, G.: Transparency enhancing tools and the gdpr: Do they match? In: International Conference on Information Systems Security and Privacy. pp. 162–185. Springer (2019)
160. Stadler, T., Oprisanu, B., Troncoso, C.: Synthetic data–anonymisation groundhog day. arXiv preprint arXiv:2011.07018 (2021)
161. Stark, E., Sleevi, R., Muminovic, R., O'Brien, D., Messeri, E., Felt, A.P., McMillion, B., Tabriz, P.: Does certificate transparency break the web? measuring adoption and error rate. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 211–226 (2019)
162. Stark, E., DeBlasio, J., O'Brien, D.: Certificate transparency in google chrome: Past, present, and future. IEEE Security & Privacy **19**(6), 112–118 (2021)
163. Stohl, C., Stohl, M., Leonardi, P.M.: Digital age| managing opacity: Information visibility and the paradox of transparency in the digital age. International Journal of Communication **10**, 15 (2016)
164. Stradling, R.: Certificate transparency version 2.0 draft-ietf-trans-rfc6962-bis-39 (2021)
165. Swihart, J., Winston, B., Bowe, S.: Zcash counterfeiting vulnerability successfully remediated (2019), https://electriccoin.co/blog/zcash-counterfeiting-vulnerability-successfully-remediated/
166. Tamassia, R.: Authenticated data structures. In: European symposium on algorithms. pp. 2–5. Springer (2003)
167. Taylor, R., Kelsey, T.: Transparency and the open society: Practical lessons for effective policy. Policy Press (2016)
168. the New York Times: Changes to the census could make small towns disappear (february 2020), https://www.nytimes.com/interactive/2020/02/06/opinion/census-algorithm-privacy.html
169. Tomescu, A., Bhupatiraju, V., Papadopoulos, D., Papamanthou, C., Triandopoulos, N., Devadas, S.: Transparency logs via append-only authenticated dictionar-

ies. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1299–1316 (2019)
170. Tomescu, A., Devadas, S.: Catena: Efficient non-equivocation via bitcoin. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 393–409. IEEE (2017)
171. Toreini, E., Shahandashti, S.F., Hao, F.: Texture to the rescue: Practical paper fingerprinting based on texture patterns. ACM Transactions on Privacy and Security (TOPS) **20**(3), 1–29 (2017)
172. Turilli, M., Floridi, L.: The ethics of information transparency. Ethics and Information Technology **11**(2), 105–112 (2009)
173. Urban, T., Degeling, M., Holz, T., Pohlmann, N.: " your hashed ip address: Ubuntu." perspectives on transparency tools for online advertising. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 702–717 (2019)
174. Van Bulck, J., Minkin, M., Weisse, O., Genkin, D., Kasikci, B., Piessens, F., Silberstein, M., Wenisch, T.F., Yarom, Y., Strackx, R.: Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient {Out-of-Order} execution. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 991–1008 (2018)
175. Van Kleek, M., Liccardi, I., Binns, R., Zhao, J., Weitzner, D.J., Shadbolt, N.: Better the devil you know: Exposing the data sharing practices of smartphone apps. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. pp. 5208–5220 (2017)
176. Varian, H.: System reliability and free riding. In: Economics of information security, pp. 1–15. Springer (2004)
177. Venkatadri, G., Mislove, A., Gummadi, K.P.: Treads: transparency-enhancing ads. In: Proceedings of the 17th ACM Workshop on Hot Topics in Networks. pp. 169–175 (2018)
178. Wagner, B., Rozgonyi, K., Sekwenz, M.T., Cobbe, J., Singh, J.: Regulating transparency? facebook, twitter and the german network enforcement act. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 261–271 (2020)
179. Wang, S., Toreini, E., Hao, F.: Anti-counterfeiting for polymer banknotes based on polymer substrate fingerprinting. IEEE Transactions on Information Forensics and Security **16**, 2823–2835 (2021). https://doi.org/10.1109/TIFS.2021.3067440
180. Weinshel, B., Wei, M., Mondal, M., Choi, E., Shan, S., Dolin, C., Mazurek, M.L., Ur, B.: Oh, the places you've been! user reactions to longitudinal transparency about third-party web tracking and inferencing. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 149–166 (2019)
181. Weitzner, D., Abelson, H., Berners-Lee, T., Hanson, C., Hendler, J., Kagal, L., McGuinness, D., Sussman, G., Waterman, K.K.: Transparent accountable inferencing for privacy risk management. In: AAAI Spring Symposium on The Semantic Web meets eGovernment. AAAI Press, Stanford University (2006)
182. Weitzner, D.J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., Sussman, G.J.: Information accountability. Communications of the ACM **51**(6), 82–87 (2008)
183. Weitzner, D.J., Abelson, H., Berners-Lee, T., Hanson, C., Hendler, J., Kagal, L., McGuinness, D.L., Sussman, G.J., Waterman, K.K.: Transparent accountable data mining: New strategies for privacy protection (2006)
184. Weller, A.: Transparency: motivations and challenges. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 23–40. Springer (2019)

185. Wikipedia: Usage share of web browsers (2022), https://en.wikipedia.org/wiki/Usage_share_of_web_browsers
186. Wood, G., et al.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper **151**(2014), 1–32 (2014)
187. Worthy, B.: More open but not more trusted? the effect of the freedom of information act 2000 on the united kingdom central government. Governance **23**(4), 561–582 (2010)
188. Wu, M., McTighe, W., Wang, K., Seres, I.A., Bax, N., Puebla, M., Mendez, M., Carrone, F., De Mattey, T., Demaestri, H.O., et al.: Tutela: An open-source tool for assessing user-privacy on ethereum and tornado cash. arXiv preprint arXiv:2201.06811 (2022)
189. Yu, H., Robinson, D.G.: The new ambiguity of open government. UCLA L. Rev. Discourse **59**, 178 (2011)
190. Yu, M., Sahraei, S., Li, S., Avestimehr, S., Kannan, S., Viswanath, P.: Coded merkle tree: Solving data availability attacks in blockchains. In: International Conference on Financial Cryptography and Data Security. pp. 114–134. Springer (2020)
191. Zhang, Y., Genkin, D., Katz, J., Papadopoulos, D., Papamanthou, C.: vsql: Verifying arbitrary sql queries over dynamic outsourced databases. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 863–880. IEEE (2017)
192. Zhang, Y., Katz, J., Papamanthou, C.: Integridb: Verifiable sql for outsourced databases. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1480–1491 (2015)
193. Zimmermann, C.: A categorization of transparency-enhancing technologies. arXiv preprint arXiv:1507.04914 (2015)