SoK: Log Based Transparency Enhancing Technologies

Alexander Hicks
University College London

***Abstract*—This paper systematizes log based Transparency Enhancing Technologies. Based on work on transparency from multiple disciplines we outline the purpose, usefulness, and pitfalls of transparency.**

We describe the mechanisms that allow log based transparency enhancing technologies to be implemented, in particular logging mechanisms, sanitisation mechanisms and the trade-offs with privacy, data release and query mechanisms, and how transparency relates to the external mechanisms that enable contesting a system and holding system operators accountable.

We illustrate this with two examples, Certificate Transparency and cryptocurrencies, and show the role that transparency plays in their function as well as the issues these systems face in delivering transparency.

1. Introduction

As systems perform operations and assist decisions that can have an important impact on a person's life, 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.

Transparency, however, is a complex property to require from a system. It does not 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 [183].

In this chapter, we consider achieving transparency based on logging mechanisms. This involves technical considerations, such as logging, sanitising, releasing and querying data, as well as non-technical external mechanisms that determine what can be done once transparency is in place. Our aim is to provide a systematization that brings the relevant aspect of each mechanism into one view of log based transparency enhancing technologies.

Outline of the chapter. We motivate applying transparency to 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.

In Section 4 we discuss threats to transparency mapped to the essential mechanisms outlined in the previous section and editorial control and individual evidence.

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.

Methodology. Transparency is a broad topic that many fields have independently studied, not all of which can be covered here. For work on transparency from other fields, we have, therefore, focused on work from Law, Philosophy, Business, and Economics, which provide a basis for thinking about transparency and computer systems.

Because of our focus on log based transparency enhancing technologies and the security of the mechanisms involved in such systems, we have endeavoured to find relevant papers from the information security literature by going through publications at major conferences like IEEE S&P, ACM CCS, NDSS, Usenix Security, PETS, and ACM FAccT, as well as searching for papers from other smaller conferences, workshops, and journals, including those in adjacent fields (e.g., HCI, STS). Work that relates to transparency but not directly to log based transparency enhancing technologies (e.g., work on transparent machine learning) is out of scope and, therefore, not included.

2. A Short Overview of Transparency

Transparency can be defined as "the quality of being done in an open way without secrets" [44]. Applied to an organization, it can mean that the organization is "open, public; having the property that theories and practices are publicly visible, thereby reducing the chance of corruption" [205].

These definitions express the basic intuition that if something is being done transparently then it cannot be done badly without it being noticeable. As Brandeis put it, "sunlight is said to be the best of disinfectants" [41].

This should create an incentive to ensure that things are done well if there is a high likelihood of being held to account, making transparency an enabler of accountability or other ethical principles (e.g., safety, welfare) [188].

Transparency such as open data practices promoted by both governments [56], [143] and academics [55], [159], lead to the public release of data that is used to determine

policy. Open data practices are also used in scientific research to allow results to be reproduced, further research to be conducted, and new algorithms to be benchmarked.

In a more bottom-up manner, freedom of information laws have enabled the media, NGOs, and the public, to make requests for information that can be used to hold public authorities to account. Other regulations, such as the GDPR [3] give individuals the right to request a copy of their personal data that is held by a controller (Article 15), require that personal data should be "processed [...] in a transparent manner in relation to individuals" (Article 5), and as general data processing principles that "it should be transparent to natural persons that personal data concerning them are collected, used, consulted or otherwise processed and to what extent the personal data are or will be processed" and "the principle of transparency requires that any information and communication relating to the processing of those personal data be easily accessible and easy to understand, and that clear and plain language be used" (Recital 39).

Access to data also helps mitigate information asymmetries. The work of Akerlof on information asymmetries in markets [12] has led to security economics re-framing many security issues (e.g., software security) as problems of information asymmetry [19], [21], which can be dealt with by requiring data standards and disclosures.

This ties into Saltzer and Schroeder's open design principle [161]. Most security mechanisms (e.g., cryptographic algorithms) are open, enabling users with the technical knowledge required to assess a system's code or specification to determine whether they want to rely on the system. Beyond the specification and code of a system, nutrition labels for datasets [77], [90], [152] and models [128], and privacy labels [101], [102], have been proposed.

2.1. Transparency matters for computer systems

Security mechanisms are designed to allow certain properties of systems (or the data they operate on) to hold; for example, integrity, confidentiality, or availability, but no system is perfect. However, designs can be flawed, implementations can suffer from software bugs or faulty hardware, and systems can be misused. Security Economics tells us that we should not expect perfect security in practice, even when technical mechanisms appear to be sufficient in principle [19], [21].

Even if we perfected the design of systems, designing and implementing complex systems that are entirely formally verified is currently unrealistic and would not prevent harms that occur because of a system that, operating as intended, applies harmful norms [87]. Information is routinely copied, aggregated, and analysed across networks operated by different parties, rendering strict enforcement mechanisms impractical compared to relying on accountability [201]. Notions of appropriate use may depend on the data itself as well as the context – an emergency that requires immediate access to medical data would render any strict security mechanism preventing this useless [69].

More generally, evaluating strict compliance with norms assumes that there are reliable norms, despite many systems operating in grey areas [87]. As systems grow in size, complexity, and scope of applications that impact people's lives, 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 [64].

Evaluating systems is not new, and system operators routinely do so internally but this does not always work to reduce the harm that a faulty system can cause. There can be issues with how the evaluation is done; for example, because of flawed mechanisms or metrics. Even if a system operator detects faults in the system it operates, it still has to address these faults and may not do so if it does not have the incentive or the capacity (technical or economic) to do so.

Systems are not inherently inscrutable [105], but those to whom harm is caused cannot necessarily detect or show that the system is at fault, despite being those that have a greater incentive to do so. Access control mechanisms that regulate rights over a system tend to favour those who design or commission these access control mechanisms (e.g., system operators), rather than those subject to the system who have no ability to access useful information via the system itself.

Privilege over information about the system, such as known error logs, means that system operators can manipulate disclosure procedures to their advantage [119]. 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 [98]), breathalysers (See Bellovin et al. [32]), and newer data processing systems that result in unfair and harmful outcomes [25], [28].

Transparency enhancing technologies offer a way to not only provide trustworthy transparency through the use of security mechanisms but also to scale transparency. For example, the IPCO, which audits law enforcement requests for telecommunications data in the UK, perform local inspections of a limited amount of offices to produce their audit [95]. Transparency enhancing technologies could allow for larger and more efficient audits of many practices.

Moreover, while transparency can have negative effects on people if they respond to transparency with hiding behaviour, impacting their performance [35], the opposite could be true for computational systems with secure transparency mechanisms because the performance of such systems is determined by the code and infrastructure it runs on, not on whether or not it is being observed. Given two systems that perform similarly, if transparency is cheap enough to implement and expensive enough to cheat once implemented (e.g., breaking the logging mechanism's cryptographic properties), the honest transparent system will be cheaper to operate than the one that tries to cheat transparency, which should make it more competitive. (That is unless the system is so broken in the first place that whatever is revealed by transparency condemns the system.)

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 are typically

more efficient to operate 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 enhancing technology, overlaid on top of a centralized system with a trustworthy interface between both, can provide a useful compromise between the inherent efficiency and logistical advantages of the centralized system and the lesser trust required by a decentralized system.

2.2. Forms Of Transparency

Transparency can take numerous forms based on the direction in which information flows, the type of information that flows, and when it flows.

Directions of transparency are reminiscent of basic access control models (e.g., Bell-LaPadula [30] and BIBA [36]), which determine in which direction (upwards or downwards) information can be read or written. Unlike many access control mechanisms, however, transparency requires that information leaves the system and be accessible by users with no privilege over the system, and restricts the write access of privileged users over this information.

Concerning the type of information, there can be information about inputs to a system, processes executed within the system, and outputs of the system, where different levels of transparency (or data granularity) matter. For example, when revealing the inputs to a system, the ordering of inputs can also be important as the ordering of data used to train a model can affect its performance [171].

Timing determines when information is made available. It is uncommon to have real-time transparency when humans are involved as knowingly being surveilled can affect behaviour [34]. A computer cannot be aware that its actions are being logged but a human user of the computer will be, so this can still be a concern in some cases. Even for entirely computational systems, transparency may only be useful if there is enough information to obtain an aggregate view of the system's performance but systems such as cryptocurrencies offer a live transparent view of the system.

2.3. Criticisms of Transparency

Lack of effectiveness. The assumption that underpins much of the belief in transparency is that it will lead to 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 [74].

Etzioni has argued that there is little evidence that supports the view that transparency is an effective accountability mechanism [68]. The argument is that transparency is no alternative to regulation (it can only be complementary) because regulations cannot be replaced by offloading the responsibility of demanding and analysing data to citizens without the time or other resources to handle these tasks.

This is backed up by Ferry and Eckersley, who found that, in the UK, the replacement of formal audits with requirements for English local authorities to publish datasets (with little contextual information) weakened accountability [70]. In countries without regulations that implement effective accountability, however, transparency can be effective at bypassing corrupt official audit processes [70].

The issue is that information being transmitted about a bad outcome does not prevent it. Moreover, it does not prevent future bad outcomes either as it does not, by itself, mitigate their possibility. A practical example of this is mandated disclosures such as nutrition labels, which 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 with proposals 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 the effectiveness of privacy labels has also shown that issues of judgement and misdirection could render transparency ineffective [6], [9]. Developers themselves 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 that are unknown to them [113]. If any harm is perceived as originating from the use of a problematic dataset or privacy-invasive system, a system operator will not be prevented from deploying such a system and may also rely on nutritional labels as cover if the process that produces these labels can be influenced.

Yu and Robinson have a similar view on open government technology and data, arguing that while it may allow the public to contribute in new ways, it does not create any government accountability [209]. Open government initiatives generally do not imply any effect on how government works (other than publishing data) so any faulty process is likely to remain in place. Thus, open data and transparency may be used as a trojan horse for other political goals [111].

If transparency by itself does not entail accountability, it follows that it also does not necessarily create trust. Despite greater access to information, for example in the case of government transparency and freedom of information, trust has not increased [117], [142], [145], [207]. If transparency only reveals systemic faults, why trust such a system?

Restricted transparency. Obtaining information that is theoretically available, for example through Freedom of Information requests, can also be an issue that requires people to develop specific expertise. In other cases, the release of bulks of information may also obfuscate important information [179]. 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 [142] or that the information that is released is not chosen purposefully to serve a chosen narrative [8].

These criticisms extend to algorithmic transparency for black box computational systems [17], [203]. Burrell distinguishes three forms of opacity in the context of algorithmic systems, opacity as intentional corporate or state secrecy,

opacity as technical illiteracy, and opacity as the way algorithms operate at the scale of application [43].

Rights such as data subject access requests may also not work well in practice [26]. This highlights the gap between transparency and other properties (e.g., fairness and explainability) of a computational system. Knowing the inputs, rules, and outcomes of a complex system may not be enough to understand its processes. Thus, while auditing is necessary and possible, auditing decisions that result from algorithms can still pose a significant challenge [129].

Even systems that are open source are not necessarily more or less secure than closed systems [20], [167] because there are many steps in between code being released in open source form and bugs in the code being identified and fixed, such as having the necessary resources and processes to fix bugs. Again, this highlights the gap between the availability of information and actions taken based on that information – in this case, auditing for and fixing vulnerabilities.

Tension with privacy and confidentiality. Another criticism of transparency is that it can cause harm privacy or negatively affect businesses that rely on confidential components in their systems. This is particularly important for systems that process sensitive data, despite the fact that greater transparency about the sharing and processing of sensitive data may be desirable.

The potential privacy harms brought on by the release of information are also used to restrict transparency. Freedom of information requests may be refused if they involve the release of personal information that would contravene data protection principles [1, Chapter 36, Part II, Section 40].

Similar situations occur when it comes to challenging systems. For example, Uber invoked privacy concerns to impede a challenge by Uber drivers seeking to obtain information about the system that they were subject to [157]. More generally, unless compelled to, companies are often extremely reluctant to disclose anything that they can argue falls under commercial confidentiality.

3. Essential Mechanisms

This section introduces transparency enhancing technologies based on logging mechanisms, sanitization mechanisms that process the data into a format suitable for release, release and query mechanisms, and external mechanisms to make use of transparency. Figure 1 illustrates where each mechanism takes place and the parties it relates to.

Logging involves the system operator of the subject system and log, which is maintained by log operators.

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).

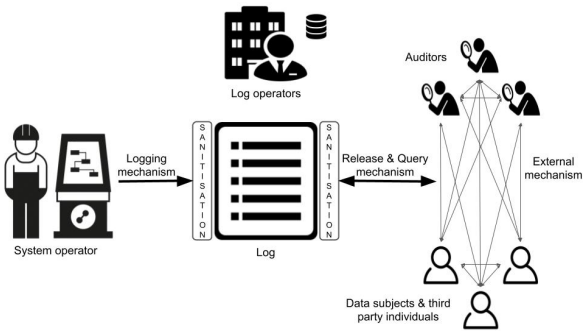

Figure 1. Summary of essential mechanisms for transparency enhancing technologies (logging, sanitization, release and query, external) and their place in a transparency process.

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 [106], 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 [57].

Secure logging mechanisms have been of interest to cryptographers for a long time [31], [47], [91], [116], [154], [165], [166], [198]. For the purpose of transparency, they have coalesced under authenticated data structures [126], [182] and transparency overlays [46], which are designed to broadly ensure that the log is verifiably append-only, can be used to lookup information, and is consistent in the sense that it shows the same information to everyone and does not equivocate. This is typically achieved with Merkle trees or blockchains, although more recent work has also explored the use of append-only dictionaries.

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. This makes it possible to verify the integrity of any data encoded as a leaf in the tree.

A history tree, following the work of Crosby and Wallach [52], grows from left to right and is used by systems like Certificate Transparency [109]. This allows for logarithmic-sized proofs that the log is append-only as new values (e.g., the hashes of new certificates in Certificate Transparency) are added to the log by a log server. This addition results in a new Merkle tree and root hash, which is signed by the log server. Because the tree grows from left to right, it is then possible to efficiently verify that the new Merkle tree includes everything that was included in the

old one, showing that it is append-only. Looking up specific certificates, however, requires linear-sized proofs.

As shown by Chase and Meiklejohn [46] the Certificate Transparency log satisfies *consistency*, meaning a potentially dishonest log server cannot get away with presenting inconsistent versions of the log to different parties, *non-frameability*, meaning that parties cannot blame the log server for misbehaviour if it has behaved honestly, and *accountability*, meaning that there exists evidence that can be used to implicate log servers that promised to include events but then did not.

A prefix tree, as used by CONIKS [124] to allow users reliant on a PKI (e.g., for communication apps) to verify the consistency of the public keys of other users, has leaf nodes ordered in lexicographic order. This makes it efficient to look up values in the tree, although showing that the log is append-only now requires linear-sized proofs. For example, a client can register name-to-key bindings in the Merkle tree's leaf nodes, which other clients can then lookup on behalf of other users. To verify a name-to-key binding in the tree, a client checks the signed tree root (STR), which includes the root hash and a hash of the previous STR (successive STRs form a chain), and the inclusion of that name-to-key binding with the path from the root to the leaf node for that name-to-key binding.

Non-inclusion of a name-to-key binding can also be checked by verifying that given an index (i.e., a name), there is no key data mapped to it.

To prevent incidents, clients monitor their user's key bindings do not change unexpectedly and verify that the PKI's identity providers are presenting consistent versions of their key directories to all participants by checking that a provider has correctly signed the STR and that the hash of the previous STR matches what was previously seen.

Combining Merkle trees. A prefix tree and a history tree can be combined to form a verifiable log-backed map [7], [81], [82], [160]. (The prefix tree can alternatively be a hash treap [148], [153].)

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) [53], [110], 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 un-equivocable log derived map [18], 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. [94] 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. [156] 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.

Append-only dictionaries. Append-only dictionaries based on bilinear accumulators [185] have been proposed as an alternative to Merkle trees, enabling logarithmic-sized append-only proofs and polylogarithmic-sized lookup proofs, although high append times and memory usage, meaning this approach is not yet practical.

Blockchains. Blockchains provide a decentralized and tamper-resistant way of updating and maintaining a global state. Transactions that update the state are logged on the blockchain, making it possible to replay all transactions and to verify that something has happened if it is included in the blockchain, as well as when it was included.

Beginning with Bitcoin [136], blockchains have been used by cryptocurrencies to provide a transparent record of transactions over a network. As Ethereum [206] and later projects have shown, it is possible to rely on blockchains to execute arbitrary programs (smart contracts) and record these executions on the blockchain. This allows a wide range of applications to run transparently on top of a blockchain or to use an existing blockchain to store evidence in a tamper-resistant way [75], [80], [88], [137], [147].

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 be adapted to a blockchain so that its consensus protocol replaces the need for gossiping between clients that is required in a Merkle tree based system to guard against equivocation [39], [186].

Blockchains can be permissionless or permissioned. For logging purposes, the effect of choosing one or the other is that in a permissionless setting, it is possible to use an existing public blockchain, such as Ethereum, in which case the blockchain will be maintained regardless of your use case because many other applications rely on it, as well of the value of the underlying cryptocurrency. Thus, any incentives to maintain (or not) a reliable log are taken care of (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.

The effort required to use an existing public blockchain and write a smart contract for it may also be much less than deploying an entire system like Certificate Transparency, allowing for more applications of transparency.

In a permissioned setting, known pre-determined parties will have to ensure that the log is maintained but, because there is no need for an underlying cryptocurrency, the system could be set up to include new events to the chain as they arrive rather than at the wishes of an uninterested miner. In this case, 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 information recorded on a log will often be sensitive, in the sense that it affects the privacy of an individual or that it reveals confidential information about the system it is pulled from. For this reason, sanitising the information that is logged will be necessary but must be done in a way that does not compromise the desired transparency.

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 [88]), in an encrypted form to be decrypted by specific parties (e.g., designated auditors being given access to raw data, individuals accessing individual evidence [88]), or using cryptographic techniques such as zero-knowledge proofs to assert relevant properties of the logged information without revealing the underlying data [75], [80], [147].

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 differentially private bounds without adding excessive noise, to produce zero-knowledge proofs that assert the necessary properties of the logged information, or simply to rely only on cryptographic proofs about data. In such cases, it may be necessary to permit access to unsanitized data by designated auditors, while the public is given access only to sanitized data that can be used to verify the results of an audit published by the designated auditors.

Beyond the data itself, identifiers (and other metadata) that allow users to verify their individual data may also need sanitization. CONIKS, for example, uses a verifiable random function to produce a user identifier for the log that does not reveal the identity of the user to others [124], and more recent work has introduced append-only zero-knowledge sets that minimize the leakage from queries [45], [118].

3.3. Release and query mechanism

Once data is logged, it must also be possible to release the data or perform queries on it. As shown by Reijsbergen et al. [156], it is possible to implement (Merkle tree) logs in such a way that they natively support broader queries than simple lookups, but more can also 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 [211], [212]; 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 [149]. 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 [83] (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. For example, only some designated auditors may be able to access raw data. 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. Depending on the type of log, this may be more or less simple. For example, a blockchain based system can implement access control via a smart contract. This could also be set up to log queries if necessary. For a Merkle tree based system the access control mechanism would have to be built on top of the logs.

3.4. External mechanisms

Transparency cannot be expected to be effective by itself, it must work to enable action based on what it reveals. For example, if transparency produces evidence that a system has malfunctioned, it can allow aggrieved parties to take legal action, governance decisions about a platform or network [104], and the removal of parties from a network if they cause a fault [146]. 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 more automated processes that similarly make it possible to contest actions taken by the 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 enhancing technologies that can allow the norms enforced by a program to be contested [87].

This process starts with users being able to check information that is relevant to them or being notified about such information. Notification tools [24], [71], [72], [131]–[134] are a useful way to keep the user in the loop, without needing them to perform queries, when their explicit consent for an action is not required, but this does not necessarily allow a user to contest any action that is taken.

For an action to be contested, there must first be evidence of that action. Often a program is assumed to have been correctly executed unless there is evidence of the contrary, but systems often fail to produce such evidence [130]. Transparency should address this, and gossip and consensus protocols can also play a part in spreading evidence and reaching a conclusion about evidence. What is then important is that the evidence be useful.

For an automated process, the proof must fit the requirements of the program that will evaluate it. For a non-automated process, such as a legal process, evidence being useful means that it should be *admissible* in the relevant jurisdiction. 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 [120].

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 for it to be useful. In the non-automated case in particular, evidence is not sufficient to contest a system by itself (unlike automated processes) and the outcome of the dispute process can vary much more, up to contesting the existence and norms of the system.

In such cases, it may not always be clear when considering a single event, why the system failed [89]. This can require a broader discussion about the system and both the individual evidence and aggregate evidence (e.g., error rates) about the system to be considered to see which is more likely. To act on information also requires the ability to understand that information, which can be made easier via explanations [155], context [55], and labels [90], [101]. This is particularly important, but also challenging, because disclosure practices are not always well designed [141].

4. Transparency and Security

Although many transparency enhancing technologies have come from security and cryptography research (e.g., cryptographic logs) and, therefore, have involved a security-focused approach, this is not always the case. Moreover, even for cryptographic mechanisms, threats are typically expressed in terms of the cryptographic properties of the mechanisms, particularly when these mechanisms are introduced as abstract primitives, useful for applications outside of transparency, rather than as part of a system focused on transparency, which is our approach here.

4.1. Assets and beneficiaries of transparency

The inputs, processes, and outputs, of systems are assets for the parties that own and operate them. The value of these assets can depend on their confidentiality. Datasets, a codebase, a machine learning model and its outputs, can all contribute to a competitive advantage, and their confidentiality can also help avoid liability for flaws in the system, or give the illusion of technical sophistication.

Transparency can benefit system operators if it increases public trust. This can be true regardless of whether or not the system is good by any measure because an organization operating a flawed system may engineer a form of transparency that does not reveal these flaws by, for example, limiting transparency to only reveal favourable information.

Because transparency does not necessarily increase trust, however, operators of reliable systems may feel they have little to gain and operators of unreliable systems may have little to lose. That is unless transparency is deployed in such a way that, for example, it harms those who operate unreliable systems by enabling consequences.

For the public, transparency should be a valuable asset, revealing useful information about a system over which they have no control and allowing them to take action by choosing whether to use the system, contest it, and hold the system operator to account for any faults. Privacy concerns over the public release of sensitive data that pertains to them may, however, be an important drawback.

Thus, transparency can be both beneficial and a drawback for system operators and the public, and, importantly, the ways in which the public may benefit from transparency may be a drawback for the system operator. When this is the case, it should be ensured that blame avoidance strategies (e.g., avoidance of record keeping, gaming performance metrics) are not put into place [93].

4.2. Threats based on essential mechanisms

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 logging mechanism is based on sound cryptography (e.g., a secure hash function, public key encryption scheme, and digital signature scheme) then what remains as a threat is the ability of a malicious system operator (or whichever party is responsible for logging information) that attempts to compromise what makes it to the log in the first place.

sanitization. As sanitization can take place before or after information is logged, threats can come from either the system operator (before logging) or from data releases and queries (after logging).

A system operator could try to compromise a sanitization mechanism just as they would the logging mechanism itself. A sanitization step taking place before the information is committed to the log would be intended to work towards the confidentiality of commercially sensitive information about the system or to respect the privacy of users who relate to logged data. This could be abused by the system operator to hide other information without having to compromise the logging mechanism.

For a sanitization mechanism that takes place after information is logged, threats are posed by parties attempting to

learn private information about others from the information they have access to.

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 wrong statistics) or relying on an honest use of a sanitization mechanism that obfuscates some information as part of its use (e.g., by adding noise).

Release and query. The form of the information made available by release and query mechanisms will depend on the sanitization mechanism, so the threats that are specific to release and query mechanisms will be those that 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 that any other party may try to pose as an individual or trusted auditor to gain access to their privileged information. The right to access under the GDPR has been abused for this purpose [58], as well as to infer information about the organization answering the query [172].

If information is simply 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.

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 and the threat actor can be any user of transparency giving (mistakenly or intentionally) 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 use that to show that any individual evidence they disclose is correct.

Editorial control and individual evidence. Examining different attempts to implement transparency around the world, Taylor and Kelsey found that the two general threats to transparency were editorial control, the ability to control what is made transparent, and individual evidence, the ability to suppress the ability of a person to find information that relates to themselves through transparency [183].

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), resulting in effects on the external mechanisms.

5. Transparency Infrastructure

5.1. Requiring and maintaining transparency

Deploying transparency requires an infrastructure that supports the operation of logs and the storage of any data required, including data that may not be stored on the log. Because logs (and any other 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 not be reliable 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. Google, for example, is responsible for the design and deployment of Certificate Transparency. Because Google Chrome is the dominant browser [204], 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 [121].)

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 a direct incentive to ensure its success or the resources both in terms of influence on the ecosystem and technical resources (e.g., in the case of NGOs) to guarantee it. Proponents of blockchains and cryptocurrencies argue that they offer the possibility of designing decentralized systems that, via mechanism design, can ensure that participants in the system have incentives – typically financial – that are aligned with maintaining the system. Blockchain can then serve as logs, requiring only a smart contract to deploy, and services such as Filecoin [107] could also offer decentralized storage when it is necessary to store more than logs.

Users themselves could drive businesses to provide greater transparency as they do react to, for example, being shown the extent to which they are tracked [199] and how moderation is applied [97]. However, they often have to rely on tools set up by system operators that do not

TABLE 1. THREATS FOR TRANSPARENCY ENHANCING TECHNOLOGIES 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

provide complete transparency, or transparency that users can understand [23], [189]. As we already noted in Section 4, system operators may not be incentivized to provide effective transparency, leading to a market for lemons.

Regulation could also play a part by imposing a statutory requirement to provide transparency could be through enforcement action of a regulator such as the Federal Trade Commission or a data protection authority. The European GDPR, which effectively applies globally to any service that has users who are citizens of the EU, notably includes several articles concerning transparency.

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 access telecommunications data (a yearly report is published [95]) 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)].

Some regulations, like the German Network Enforcement Act (NetzDG), do include require transparency requirements about, for example, how unlawful content is dealt with and have resulted in fines for companies such as Meta. Companies differ in how they implement their compliance with this regulation [196] and are likely to differ in implementing any other kind of 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.

In particular, because regulators are not the people affected by flawed systems and can typically only levy fines on system operators who treat these as a cost of business, transparency that provides information to regulators is unlikely to offer much progress. Transparency that is user-facing, and can inform users in a way that allows them to take action on the basis of that information may be more effective.

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 on which public keys own each bitcoin. A user may want to send bitcoins to another user but if the transaction is dropped by the network the transaction fee was too low, then the transaction is never executed or recorded. Thus, the Bitcoin network is transparent about how the miners view the network, not about every action of the users in the network.

Moreover, not all real-world transactions are logged because Bitcoin private keys may be exchanged offline with no mapping between keys and identities to restrict this.

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 as some may not be logged. Browsers can reject certificates that do not appear in Certificate Transparency logs, however, which ensures that log servers that are operated by, for example, Google, have the incentive to log all valid certificates sent to them by certificate authorities.

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 then, for example, the attestation key could leak [191], rendering all other units worthless. This is a case of weakest-link security that depends on the party with the lowest benefit-cost ratio in securing their unit [194], in a scenario where that party may be adversarial and have full physical access to their hardware.

Alternatively, it may be possible to rely on non-colluding parties to cross-verify information.

Problems may also occur if there is no ground truth for the logged data. For example, wage transparency could identify wage gaps but if the party that logs salaries is the

business itself, the logging mechanism (or any computation used to identify a wage gap [108]) can execute correctly regardless of the data (and the resulting analysis) being true if individuals cannot verify their inclusion in the computation.

Problems can also occur when dealing with physical objects, because this requires a secure way of mapping physical objects to digital objects that can be authenticated once logged. Mechanisms that provide cryptographic-like mechanisms to authenticate certain physical objects do exist, however. There is a body of work that studies how paper documents could be authenticated based on their physical characteristics [42], [50], [84], [115], [162], [170], [187], [190], [197]. This would allow the document to be logged with its fingerprint, allowing it to be authenticated later if required.

6. Balancing Transparency With Privacy

Because privacy concerns can create legitimate restrictions on transparency, privacy enhancing technologies that preserve privacy while retaining the utility of information can enable transparency. (In turn, transparency can help users identify privacy risks [54], [86], [114], [193].)

There are two types of information to consider, aggregate information related to a population and information related to individuals. Aggregate information makes it possible to determine how the system is functioning as a whole and whether it is, for example, (un)fair, (un)biased, or error-prone. By itself, this can be enough to reach a conclusion about the system such as whether the system should be modified, shut down, or to make the choice of participating in the system. For individuals, it is also important to be able 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.

In the case of aggregate information, the privacy requirement is that the aggregate information should not leak information about an individual, including the inclusion of an individual's data in the data that was used to produce aggregate information. This often involves differentially private mechanisms that determine the kind of perturbed data that can satisfy data protection requirements [51], [140], and zero-knowledge proofs, which allow the execution of a process to be verified without revealing anything else about the process [27], [78], [79].

For individual information, controlling access to information also matters since revealing information only causes a loss of privacy if it is revealed to someone other than the individual it relates to.

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 not only the ability to prevent access to information (e.g., information being logged by the transparency enhancing technology) but also any way of influencing what is or is not recorded, the format in which it is recorded, what is shared with who, 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 nonetheless a form of editorial control that can work in favour of an adversarial system operator. This is because the addition of noise disproportionately affects less represented groups in the data. For example, the adoption of differential privacy for the U.S. Census could effectively erase smaller towns from census data [184]. More generally, differential privacy could be used, under the cover of it being a required privacy enhancement, as a way of masking bad outcomes on minority groups, or to make low-frequency faults disappear.

Another way in which differential privacy can lead to editorial control is by limiting the number or type of queries that can be made as part of the query mechanism of the transparency enhancing technology. Differential privacy assigns a privacy budget that dictates how many queries can be made (based on their sensitivity), placing a limit on what and how much data subjects, third-party auditors, and third-party individuals can do through a query mechanism. It could also allow an adversarial auditor (perhaps colluding with the system operator wanting to work against transparency) to exhaust the privacy budget by performing high-sensitivity queries that do not reveal anything unwanted.

However, this can be avoided by relying on a release mechanism that generates synthetic data (although not a general solution [176]) that can be queried ad infinitum, rather than relying on a query mechanism that serves differential private answers to queries on the database of original data.

Zero-knowledge proofs can also act as a form of editorial control. A zero-knowledge proof reveals nothing but the truth of a statement, which can remove context from a query. Requiring that any query by an auditor be expressed as a provably true or false statement within the constraints of a formal language may also restrict the range of possible queries, and prevent necessarily vague queries.

Querying for provable statements can also be made inefficient this way as queries must be designed without access to data. This could mean iterating over queries of the type "is the number of data points with attribute α greater than x". The result of this is that practically speaking, it is only possible for auditors and individuals to verify statements that are given to them by those who control the information that is queried, rather than being able to perform their own investigation.

Moreover, detecting a flawed implementation of a zero-knowledge proof system that allows counterfeit proofs to be produced can be hard. Flaws in zero-knowledge proof systems have only happened by accident so far [127], [181], but there is a precedent for cryptosystems that could plausibly be exploitable by design [33]. 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 desirable for the simple reason that a general overview of a system may reveal issues with the system (e.g., it is biased against certain attributes or has bugs) but fail to show their impact on individuals (e.g., if one was discriminated against or affected by a bug). This requires not only knowledge of the system's outcome for that individual, which usually will be known for the outcome to have any effect although this may not always be the case (e.g., for confidential processes) but also some form of ground truth for what the outcome could have been, which in general may be harder to obtain.

For example, the covid-19 pandemic caused secondary education exams in the UK to be cancelled in 2020 and grades to be awarded based on an algorithm using results of past students as input. The population outcome was normal by design – the distribution of grades matched historical distributions for each school – but it meant that students who performed outside the historical norm could be awarded lower or higher grades than expected for the sake of preserving the historical grade distribution. Individual evidence in the form of teacher predicted-grades, however, made it possible to easily identify how students had been affected (e.g., a student with a high teacher-predicted grade being awarded a low grade) and the algorithmic marking scheme was quickly replaced with teacher-predicted grades [29].

Individual evidence can also be useful when there is a dispute about whether an individual has made an error when using a system or has been a victim of a bug. Human errors and bugs can happen at reasonably low frequencies so conclusively determining whether one is more likely than the other can be impractical, and neither the presence of bugs in the system nor the possibility of a human error can be used to invalidate the other [89]. Individual evidence that makes it possible to identify the error in an event log and a record of actions by the individual could make it much more efficient to determine whether the error was human or due to a bug in the system.

The role of privacy enhancing technologies, however, is often to make it impossible to link an individual to an input or output of the subject system's process.

Differential privacy guarantees that an individual does not have too much of an effect on outputs so that it cannot be determined their data was used to obtain that output without an additional mechanism that deals with this.

Zero-knowledge proofs remove the relation between the output of the computation it verifies and its inputs. If individual evidence exists, however, a zero-knowledge proof could be used to show an individual that their individual evidence was used in the computation. Without this, an adversarial system operator or auditor could simply use inputs that they choose or generate 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 are an essential part of web security, allowing a user's browser to verify the owner of a website. Certificates are issued and signed by trusted third parties, certificate authorities, who can be the source of security incidents [49], [61] An example of this is the DigiNotar hack [192], which led to hundreds of rogue certificates being issued with DigiNotar's signing key and DigiNotar certificates being rejected by most browsers [10], [125], [138].

Certificate Transparency [109], [180] was developed to address this type of incident. Acknowledging that it is not possible to prevent rogue certificates from being issued, Certificate Transparency works by making certificate issuance transparent and working against malicious certificate issuance by helping reveal cases where this happens. This is achieved by using logs based on Merkle history trees that ensure the list of logged certificates is a secure append-only transparency overlay [46], [60].

Certificate authorities submit certificates to the logs themselves and browsers will only accept certificates that come with a signed certificate timestamp from log servers, so a malicious certificate authority cannot compromise the efficacy of the logging mechanism by not submitting certificates that they issue to logs and collusion between a certificate authority and a log server is mitigated by requiring multiple signed certificate timestamps from different logs.

Certificate Transparency is widely deployed, with the percentage of main-frame HTTPS page loads and HTTPS connections with at least two valid signed certificate timestamps reaching above 60% as of 2018 for Chrome users [177]. There is significant infrastructural backing from organizations like Google, Mozilla, and Cloudflare, and free services such as Let's Encrypt [5].

There is no sanitization mechanism involved in Certificate Transparency, although some interactions involve privacy concerns for users. For example, when their browser queries a proof of inclusion in a log, it reveals the website that the user is browsing. As a result, 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 [121].

Reporting that a certificate has not been included in a log also reveals a user's browsing activity for that website. This can be mitigated by using zero-knowledge proofs to 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 [63]. This approach has downsides, however, as it would require changes to log implementations and APIs, and obfuscate details in investigations of log misbehaviour [178], showing the tension between transparency and privacy goals.

Other issues exist with the certificates themselves and logs, which can be used to identify potentially vulnerable websites because websites with expired certificates tend to more outdated software that may be vulnerable to CVEs [151]. The volume of information available through Certificate Transparency also makes it possible to monitor logs to identify new DNS names (i.e., service endpoints) that may be vulnerable to an attack, rather than inefficiently scanning the IP space [164]. Logs can also be mined to detect subdomains, as well as other sensitive information including names, usernames, email addresses, business relationships, and unreleased products [158].

The volume of logged certificates poses scalability issues as well. Monitors, who fetch and try to spot suspicious certificates, cannot guarantee that fetching certificates returns a complete set of certificates, meaning that fraudulent certificates may be logged but not spotted [112].

External mechanisms play an important role in Certificate Transparency. Certificates must be revoked as time passes or in the event of an incident (e.g., DigiNotar). In such a case, a human decision must be made based on the information available and the potential to act on that information. The latter means that power is concentrated in browser vendors (e.g., Google, Mozilla, Microsoft, Apple, Brave) which are the only parties who can act on certificate transparency revealing a malicious or compromised certificate authority by blocklisting it. Expert users can in principle 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 [48], but gossiping is not widespread [76]. There are several ways to work around this, replacing gossiping as a type of external mechanism with a protocol that is integral to the transparency overlay.

The first way is to use a blockchain and rely on its consensus protocol for consistency [39], [186], but this can be expensive because of transaction costs and has slow finality if relying on a slow blockchain (e.g., Bitcoin or Ethereum).

The second way is to rely on witnesses (e.g., the different Certificate Transparency log servers) could collectively sign a checkpoint of a log, producing some form of consensus that the log has been verified up until the checkpoint [122], but this could suffer from liveness issues if there are too few witnesses.

7.2. Blockchain based cryptocurrencies

Cryptocurrencies, such as Bitcoin [136], Ethereum [206], and many others, aim to enable decentralized peer-to-peer transactions between users that do not rely on any centralized institutions such as banks, Paypal, and VisaNet [136].

This requires solving the problem of currency minting and double-spending such that no single user can unilaterally determine the amount of tokens they control, or spend the same tokens multiple times. This is achieved by relying on a blockchain, which records blocks of transactions (that refer to the previous block in the chain), which are 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.

Chase and Meiklejohn [46] considered the Bitcoin blockchain as one of their two case studies (the other being Certificate Transparency) in their formalization of transparency overlays. The important difference between the two systems that emerged 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. This can be dealt with through penalties and *slashing* mechanisms that exist in proof-of-stake cryptocurrencies, such as Ethereum [65], to directly fine or remove from the network block validators 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 [40], [103]. Recursive succinct arguments of knowledge can be produced in time proportional only to the number of transactions added since the previous block and verified in constant time [40]. To verify the blockchain, 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 secu-

rity gap between full nodes and light clients [14], [210]. Fraud proofs also play a role in enabling scaling solutions such as optimistic rollups on Ethereum [66], which process transactions off the main chain (reducing congestion and transaction fees) and then post only compressed transaction data on the main chain. The transparency obtained from the transaction data posted on the main chain makes it possible to verify the validity of transactions and produce fraud proofs for any invalid transactions. (Zero-knowledge rollups, the alternative to optimistic rollups, rely instead on proofs of validity to prevent invalid transactions [67].)

Another commonality with Certificate Transparency is that blockchains do not necessarily offer much in terms of sanitization mechanisms, and there is no right level of privacy that is agreed upon, between full transparency that compromises basic privacy expectations and fully obfuscated transactions that rely on the blockchain as an integrity check rather than a transparency mechanism.

Early systems, such as Bitcoin and Ethereum, do not offer any privacy because, although they are pseudonymous, it is easy enough to identify unique users by studying the public transaction flows recorded on the blockchain [123] and trace coins that have been used as part of some unwanted activity [11], [22], a practice that has been commercialized by companies such as Chainalysis, TRM, and Elliptic.

More recent systems have attempted to provide greater privacy [15] through the use of zero-knowledge proofs (e.g., Zcash [163]), ring signatures (e.g., Monero [16]), coin mixing services (e.g., Tornado cash [150], sanctioned by the US Treasure since August 2022 [4]), and network level mixing (e.g., Nym [59]). 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 [37], [38], [92], [99], [135], [208].

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 possible solution is to produce a zero-knowledge proof that the funds one has put through the mixing service did not come from any address that is publicly associated with tainted funds. In this case, the transparency that allows the addresses containing stolen funds to be identified would allow other addresses to use privacy services without the risk of facilitating the laundering of stolen funds [173].

Another possible solution is collaborative deanonymization [100], 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 blockchains and their governance. The blockchain can show miner behaviour such as front-running [62], evidence of hacks, trace stolen funds, and so on. This has led to important debates about, for example, whether the 2016 DAO hack on Ethereum should be reversed with a hard fork (leading to the split between Ethereum and Ethereum Classic) [104], 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 [133] focus on the usability of transparency enhancing technologies. Hedbom [85], Janic et al. [96], and Zimmermann [213] focus on transparency tools that can be used to help users control or verify their privacy online. Spagnuelo et al. [174], [175] 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

This chapter provides a systematization of log based transparency enhancing technologies, identifying the requirements and essential mechanisms of transparency enhancing technologies, and showing how threat models relate to issues of editorial control and individual evidence.

There are many use cases for transparency: Certificate and Key Transparency [39], [45], [81], [109], [118], [124], [137], [180], cryptocurrencies [16], [136], [163], [206], binary transparency [13], [139], decentralized authorization [18], and socially driven applications such as transparency about wage gaps [108], financial markets [73], legal processes [75], [80], [147], data sharing [88] and usage [168], data mining [202], inference [200], advertising [195], and open government data [144], [169]. Many of these rely (or could as they adapt their threat models) on logs and sanitization mechanisms as we have described.

There are clear challenges to tackle, relating to the infrastructure that would enable transparency, and balancing transparency with privacy and confidentiality concerns. 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.

As we have discussed, there are many possible use cases and approaches that can be taken in designing and deploying transparency enhancing technologies. Based on the history of transparency, effectiveness is not guaranteed. The design of transparency enhancing technologies should, therefore, ensure that any technological attempt to enable greater transparency focus 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.

[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.

[4] U.S. Treasury sanctions notorious virtual currency mixer Tornado Cash, 2022.

[5] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, 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*, pages 2473–2487, 2019.

[6] Alessandro Acquisti, Idris Adjerid, and Laura Brandimarte. Gone in 15 seconds: The limits of privacy transparency and control. *IEEE Security & Privacy*, 11(4):72–74, 2013.

[7] Ben Laurie Adam Eijdenberg and Al Cutter. Trillian – verifiable data structures, 2017.

[8] Arthur Harris Adelberg. Narrative disclosures contained in financial reports: means of communication or manipulation? *Accounting and Business Research*, 9(35):179–190, 1979.

[9] Idris Adjerid, Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. Sleights of privacy: Framing, disclosures, and the limits of transparency. In *Proceedings of the ninth symposium on usable privacy and security*, pages 1–11, 2013.

[10] Heather Adkins. An update on attempted man-in-the-middle attacks, August 2011.

[11] Mansoor Ahmed, Ilia Shumailov, and Ross Anderson. Tendrils of crime: Visualizing the diffusion of stolen bitcoins. In *International Workshop on Graphical Models for Security*, pages 1–12. Springer, 2018.

[12] George A Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in economics*, pages 235–251. Elsevier, 1978.

[13] Mustafa Al-Bassam and Sarah Meiklejohn. Contour: A practical system for binary transparency. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 94–110. Springer, 2018.

[14] Mustafa Al-Bassam, Alberto Sonnino, Vitalik Buterin, and Ismail Khoffi. Fraud and data availability proofs: Detecting invalid blocks in light clients. In *International Conference on Financial Cryptography and Data Security*, pages 279–298. Springer, 2021.

[15] Ghada Almashaqbeh and Ravital Solomon. Sok: Privacy-preserving computing in the blockchain era. *Cryptology ePrint Archive*, 2021.

[16] Kurt M Alonso et al. Zero to monero, 2020.

[17] Mike Ananny and Kate Crawford. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *new media & society*, 20(3):973–989, 2018.

[18] Michael P Andersen, Sam Kumar, Moustafa AbdelBaky, Gabe Fierro, John Kolb, Hyung-Sin Kim, David E Culler, and Raluca Ada Popa. Wave: A decentralized authorization framework with transitive delegation. In *Proceedings of the 28th USENIX Security Symposium*. Univ. of California, Berkeley, CA (United States), 2019.

[19] Ross Anderson. Why information security is hard-an economic perspective. In *Seventeenth Annual Computer Security Applications Conference*, pages 358–365. IEEE, 2001.

[20] Ross Anderson. Open and closed systems are equivalent (that is, in an ideal world), 2005.

[21] Ross Anderson and Tyler Moore. The economics of information security. *science*, 314(5799):610–613, 2006.

[22] Ross Anderson, Ilia Shumailov, Mansoor Ahmed, and Alessandro Rietmann. Bitcoin redux. 2019.

[23] Athanasios Andreou, Giridhari Venkatadri, Oana Goga, Krishna Gummadi, Patrick Loiseau, and Alan Mislove. Investigating ad transparency mechanisms in social media: A case study of facebook's explanations. In *NDSS 2018-Network and Distributed System Security Symposium*, pages 1–15, 2018.

[24] Julio Angulo, Simone Fischer-Hübner, Tobias Pulls, and Erik Wästlund. 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*, pages 1803–1808, 2015.

[25] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. *And it's biased against blacks. ProPublica*, 23:77–91, 2016.

[26] Jef Ausloos and Pierre Dewitte. 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.

[27] Kenneth A Bamberger, Ran Canetti, Shafi Goldwasser, Rebecca Wexler, and Evan J Zimmerman. Verification dilemmas in law and the promise of zero-knowledge proofs. *Berkeley Technology Law Journal*, 37(1), 2022.

[28] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[29] BBC. A-levels and GCSEs: U-turn as teacher estimates to be used for exam results, August 2020.

[30] D Elliott Bell and Leonard J LaPadula. Secure computer systems: Mathematical foundations. Technical report, MITRE CORP BEDFORD MA, 1973.

[31] Mihir Bellare and Bennet Yee. Forward integrity for secure audit logs. Technical report, Citeseer, 1997.

[32] Steven M Bellovin, Matt Blaze, Susan Landau, and Brian Owsley. Seeking the source: Criminal defendants' constitutional right to source code. *Ohio St. Tech. LJ*, 17:1, 2021.

[33] Daniel J Bernstein, Tanja Lange, and Ruben Niederhagen. Dual ec: A standardized back door. In *The New Codebreakers*, pages 256–281. Springer, 2016.

[34] Ethan Bernstein. The transparency trap. *Harvard Business Review*, 92(10):58–66, 2014.

[35] Ethan S Bernstein. The transparency paradox: A role for privacy in organizational learning and operational control. *Administrative Science Quarterly*, 57(2):181–216, 2012.

[36] Kenneth J Biba. Integrity considerations for secure computer systems. Technical report, MITRE CORP BEDFORD MA, 1977.

[37] Alex Biryukov and Daniel Feher. Privacy and linkability of mining in zcash. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 118–123. IEEE, 2019.

[38] Alex Biryukov, Daniel Feher, and Giuseppe Vitto. Privacy aspects and subliminal channels in zcash. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1813–1830, 2019.

[39] Joseph Bonneau. Ethiks: Using ethereum to audit a coniks key transparency log. In *International Conference on Financial Cryptography and Data Security*, pages 95–105. Springer, 2016.

[40] Joseph Bonneau, Izaak Meckler, Vanishree Rao, and Evan Shapiro. Mina: Decentralized cryptocurrency at scale. *New York Univ. O (1) Labs, New York, NY, USA, Whitepaper*, pages 1–47, 2020.

[41] Louis Dembitz Brandeis. *Other peoples money, and how the bankers use it / by Louis D. Brandeis.* Stokes, New York, 1914.

[42] James DR Buchanan, Russell P Cowburn, Ana-Vanessa Jausovec, Dorothee Petit, Peter Seem, Gang Xiong, Del Atkinson, Kate Fenton, Dan A Allwood, and Matthew T Bryan. 'fingerprinting'documents and packaging. *Nature*, 436(7050):475–475, 2005.

[43] Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1):2053951715622512, 2016.

[44] Cambridge Dictionary. Transparency.

[45] Melissa Chase, Apoorvaa Deshpande, Esha Ghosh, and Harjasleen Malvai. Seemless: Secure end-to-end encrypted messaging with less trust. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 1639–1656, 2019.

[46] Melissa Chase and Sarah Meiklejohn. Transparency overlays and applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 168–179. ACM, 2016.

[47] Cheun Ngen Chong, Zhonghong Peng, and Pieter H Hartel. Secure audit logging with tamper-resistant hardware. In *Security and Privacy in the Age of Uncertainty: IFIP TC11 18 th International Conference on Information Security (SEC2003) May 26–28, 2003, Athens, Greece 18*, pages 73–84. Springer, 2003.

[48] Laurent Chuat, Pawel Szalachowski, Adrian Perrig, Ben Laurie, and Eran Messeri. Efficient gossip protocols for verifying the consistency of certificate logs. In *2015 IEEE Conference on Communications and Network Security (CNS)*, pages 415–423. IEEE, 2015.

[49] Jeremy Clark and Paul C Van Oorschot. Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements. In *2013 IEEE Symposium on Security and Privacy*, pages 511–525. IEEE, 2013.

[50] William Clarkson, Tim Weyrich, Adam Finkelstein, Nadia Heninger, J Alex Halderman, and Edward W Felten. Fingerprinting blank paper using commodity scanners. In *2009 30th IEEE Symposium on Security and Privacy*, pages 301–314. IEEE, 2009.

[51] Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *Proceedings of the National Academy of Sciences*, 117(15):8344–8352, 2020.

[52] Scott A Crosby and Dan S Wallach. Efficient data structures for tamper-evident logging. In *USENIX Security Symposium*, pages 317–334, 2009.

[53] Rasmus Dahlberg, Tobias Pulls, and Roel Peeters. Efficient sparse merkle trees. In *Nordic Conference on Secure IT Systems*, pages 199–215. Springer, 2016.

[54] Sourya Joyee De and Daniel Le Métayer. Privacy risk analysis to enable informed privacy settings. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 95–102. IEEE, 2018.

[55] Auriol Degbelo and Tomi Kauppinen. Increasing transparency through web maps. In *Companion Proceedings of the The Web Conference 2018*, pages 899–904, 2018.

[56] Department for Digital, Culture, Media & Sport. Uk open government national action plan, July 2019.

[57] David Devecsery, Michael Chow, Xianzheng Dou, Jason Flinn, and Peter M Chen. Eidetic systems. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 525–540, 2014.

[58] Mariano Di Martino, Pieter Robyns, Winnie Weyts, Peter Quax, Wim Lamotte, and Ken Andries. Personal information leakage by abusing the {GDPR}'right of access'. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*, 2019.

[59] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. The nym network, 2021.

[60] Benjamin Dowling, Felix Günther, Udyani Herath, and Douglas Stebila. Secure logging schemes and certificate transparency. In *European Symposium on Research in Computer Security*, pages 140–158. Springer, 2016.

[61] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. Analysis of the https certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 291–304, 2013.

[62] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok: Transparent dishonesty: front-running attacks on blockchain. In *International Conference on Financial Cryptography and Data Security*, pages 170–189. Springer, 2019.

[63] Saba Eskandarian, Eran Messeri, Joseph Bonneau, and Dan Boneh. Certificate transparency with privacy. *Proceedings on Privacy Enhancing Technologies*, 2017(4):329–344, 2017.

[64] Motahhare Eslami, Kristen Vaccaro, Min Kyung Lee, Amit Elazari Bar On, Eric Gilbert, and Karrie Karahalios. User attitudes towards algorithmic opacity and transparency in online reviewing platforms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[65] Ethereum. Proof-of-stake rewards and penalties, 2022.

[66] Ethereum. Optimistic rollups, 2023.

[67] Ethereum. Zero-knowledge rollups, 2023.

[68] Amitai Etzioni. Is transparency the best disinfectant? *Journal of Political Philosophy*, 18(4):389–404, 2010.

[69] Joan Feigenbaum, James A Hendler, Aaron D Jaggard, Daniel J Weitzner, and Rebecca N Wright. Accountability and deterrence in online life. In *Proceedings of the 3rd International Web Science Conference*, pages 1–7, 2011.

[70] Laurence Ferry and Peter Eckersley. Accountability and transparency: a nuanced response to etzioni. *Public Administration Review*, 75(1):11, 2015.

[71] Simone Fischer-Hübner, Julio Angulo, Farzaneh Karegar, and Tobias Pulls. Transparency, privacy and trust–technology for tracking and controlling my data disclosures: Does this work? In *IFIP International Conference on Trust Management*, pages 3–14. Springer, 2016.

[72] Simone Fischer-Hübner, Julio Angulo, and Tobias Pulls. 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*, pages 77–92. Springer, 2013.

[73] Mark D Flood, Jonathan Katz, Stephen J Ong, and Adam Smith. Cryptography and the economics of supervisory information: Balancing transparency and confidentiality. 2013.

[74] Jonathan Fox. The uncertain relationship between transparency and accountability. *Development in practice*, 17(4-5):663–671, 2007.

[75] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel Weitzner. Practical accountability of secret processes. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 657–674, 2018.

[76] Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle. In log we trust: revealing poor security practices with certificate transparency logs and internet measurements. In *International Conference on Passive and Active Network Measurement*, pages 173–185. Springer, 2018.

[77] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[78] Oded Goldreich, Silvio Micali, and Avi Wigderson. 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*, pages 171–185. Springer, 1986.

[79] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[80] Shafi Goldwasser and Sunoo Park. Public accountability vs. secret laws: Can they coexist? a cryptographic proposal. In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pages 99–110, 2017.

[81] Google. Key transparency, 2017.

[82] Google. Trillian, 2017.

[83] Elias Grünewald and Frank Pallas. 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*, pages 636–646, 2021.

[84] Francesco Guarnera, Dario Allegra, Oliver Giudice, Filippo Stanco, and Sebastiano Battiato. A new study on wood fibers textures: documents authentication through lbp fingerprint. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4594–4598. IEEE, 2019.

[85] Hans Hedbom. A survey on transparency tools for enhancing privacy. In *IFIP Summer School on the Future of Identity in the Information Society*, pages 67–82. Springer, 2008.

[86] Martin Henze, Daniel Kerpen, Jens Hiller, Michael Eggert, David Hellmanns, Erik Mühmer, Oussama Renuli, Henning Maier, Christian Stüble, Roger Häußling, et al. Towards transparent information on individual cloud service usage. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 366–370. IEEE, 2016.

[87] Alexander Hicks. Transparency, compliance, and contestability when code is(n't) law. In *Proceedings of the 2022 New Security Paradigms Workshop*, NSPW '22, page 130–142, New York, NY, USA, 2023. Association for Computing Machinery.

[88] Alexander Hicks, Vasilios Mavroudis, Mustafa Al-Bassam, Sarah Meiklejohn, and Steven J Murdoch. Vams: Verifiable auditing of access to confidential data. *arXiv preprint arXiv:1805.04772*, 2018.

[89] Alexander Hicks and Steven J Murdoch. Transparency enhancing technologies to make security protocols work for humans. In *Cambridge International Workshop on Security Protocols*, pages 3–10. Springer, 2019.

[90] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv preprint arXiv:1805.03677*, 2018.

[91] Jason E Holt and Kent E Seamons. Logcrypt: forward security and public verification for secure audit logs. *Cryptology ePrint Archive*, 2005.

[92] Younggee Hong, Hyunsoo Kwon, Jihwan Lee, and Junbeom Hur. A practical de-mixing algorithm for bitcoin mixing services. In *Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts*, pages 15–20, 2018.

[93] Christopher Hood. What happens when transparency meets blame-avoidance? *Public Management Review*, 9(2):191–210, 2007.

[94] Yuncong Hu, Kian Hooshmand, Harika Kalidhindi, Seung Jin Yang, and Raluca Ada Popa. Merklê 2: A low-latency transparency log system. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 285–303. IEEE, 2021.

[95] Investigatory Powers Commitioner's Office. Annual report 2018, 2020.

[96] Milena Janic, Jan Pieter Wijbenga, and Thijs Veugen. Transparency enhancing tools (tets): an overview. In *2013 Third Workshop on Socio-Technical Aspects in Security and Trust*, pages 18–25. IEEE, 2013.

[97] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 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.

[98] Jasper Jolly. Uk government sets aside up to £233m to cover post office payouts, 2021.

[99] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. An empirical analysis of anonymity in zcash. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 463–477, 2018.

[100] Patrik Keller, Martin Florian, and Rainer Böhme. Collaborative deanonymization. In *International Conference on Financial Cryptography and Data Security*, pages 39–46. Springer, 2021.

[101] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A" nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[102] Patrick Gage Kelley, Lucian Cesca, Joanna Bresee, and Lorrie Faith Cranor. Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 1573–1582, 2010.

[103] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. Non-interactive proofs of proof-of-work. In *International Conference on Financial Cryptography and Data Security*, pages 505–522. Springer, 2020.

[104] Lucianna Kiffer, Dave Levin, and Alan Mislove. Stick a fork in it: Analyzing the ethereum network partition. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, pages 94–100, 2017.

[105] Joshua A Kroll. The fallacy of inscrutability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180084, 2018.

[106] Joshua A Kroll. Outlining traceability: A principle for operationalizing accountability in computing systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 758–771, 2021.

[107] Protocol Labs. Filecoin: A decentralized storage network, 2017.

[108] Andrei Lapets, Frederick Jansen, Kinan Dak Albab, Rawane Issa, Lucy Qin, Mayank Varia, and Azer Bestavros. 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*, pages 1–5, 2018.

[109] Ben Laurie. Certificate transparency. *Communications of the ACM*, 57(10):40–46, 2014.

[110] Ben Laurie and Emilia Kasper. Revocation transparency. *Google Research, September*, 33, 2012.

[111] Karen EC Levy and David Merritt Johns. When open data is a trojan horse: The weaponization of transparency in science and governance. *Big Data & Society*, 3(1):2053951715621568, 2016.

[112] Bingyu Li, Jingqiang Lin, Fengjun Li, Qiongxiao Wang, Qi Li, Jiwu Jing, and Congli Wang. Certificate transparency in the wild: Exploring the reliability of monitors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2505–2520, 2019.

[113] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I Hong. Understanding challenges for developers to create accurate privacy nutrition labels. In *CHI Conference on Human Factors in Computing Systems*, pages 1–24, 2022.

[114] Yuanchun Li, Fanglin Chen, Toby Jia-Jun Li, Yao Guo, Gang Huang, Matthew Fredrikson, Yuvraj Agarwal, and Jason I Hong. 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.

[115] Zhengxiong Li, Aditya Singh Rathore, Chen Song, Sheng Wei, Yanzhi Wang, and Wenyao Xu. Printracker: Fingerprinting 3d printers using commodity scanners. In *Proceedings of the 2018 ACM sigsac conference on computer and communications security*, pages 1306–1323, 2018.

[116] Di Ma and Gene Tsudik. A new approach to secure logging. *ACM Transactions on Storage (TOS)*, 5(1):1–21, 2009.

[117] Vincent Mabillard and Martial Pasquier. Transparency and trust in government (2007–2014): A comparative study. *NISPAcee Journal of Public Administration and Policy*, 9(2):69–92, 2016.

[118] Harjasleen Malvai, Lefteris Kokoris-Kogias, Alberto Sonnino, Esha Ghosh, Ercan Oztürk, Kevin Lewi, and Sean Lawlor. Parakeet: Practical key transparency for end-to-end encrypted messaging. *Cryptology ePrint Archive*, 2023.

[119] Paul Marshall, James Christie, B Ladkin, Bev Littlewood, Stephen Mason, Martin Newby, Jonathan Rogers, Harold Thimbleby, and M Thomas. Recommendations for the probity of computer evidence. *Digital Evidence and Electronic Signature Law Review*, 18, 2020.

[120] Stephen Mason and Daniel Seng. *Electronic Evidence and Electronic Signatures*. University of London Press, 2021.

[121] Sarah Meiklejohn, Joe DeBlasio, Devon O'Brien, Chris Thompson, Kevin Yeo, and Emily Stark. Sok: Sct auditing in certificate transparency. *arXiv preprint arXiv:2203.01661*, 2022.

[122] Sarah Meiklejohn, Pavel Kalinnikov, Cindy S Lin, Martin Hutchinson, Gary Belvin, Mariana Raykova, and Al Cutter. Think global, act local: Gossip and client audits in verifiable data structures. *arXiv preprint arXiv:2011.04551*, 2020.

[123] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[124] Marcela S Melara, Aaron Blankstein, Joseph Bonneau, Edward W Felten, and Michael J Freedman. {CONIKS}: Bringing key transparency to end users. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 383–398, 2015.

[125] Microsoft. Fraudulent digital certificates could allow spoofing, August 2011.

[126] Andrew Miller, Michael Hicks, Jonathan Katz, and Elaine Shi. Authenticated data structures, generically. *ACM SIGPLAN Notices*, 49(1):411–423, 2014.

[127] Jim Miller. Coordinated disclosure of vulnerabilities affecting girault, bulletproofs, and plonk, 2022.

[128] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[129] Brent Mittelstadt. Automation, algorithms, and politics— auditing for transparency in content personalization systems. *International Journal of Communication*, 10:12, 2016.

[130] Steven J Murdoch and Ross Anderson. Security protocols and evidence: Where many payment systems fail. In *International Conference on Financial Cryptography and Data Security*, pages 21–32. Springer, 2014.

[131] Patrick Murmann. 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*, pages 440–442, 2018.

[132] Patrick Murmann. Eliciting design guidelines for privacy notifications in mhealth environments. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 11(4):66–83, 2019.

[133] Patrick Murmann and Simone Fischer-Hübner. Tools for achieving usable ex post transparency: a survey. *IEEE Access*, 5:22965–22991, 2017.

[134] Patrick Murmann, Delphine Reinhardt, and Simone Fischer-Hübner. 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*, pages 209–222. Springer, 2019.

[135] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. An empirical analysis of traceability in the monero blockchain. *Proceedings on Privacy Enhancing Technologies*, 2018(3):143–163, 2018.

[136] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[137] Hoang-Long Nguyen, Claudia-Lavinia Ignat, and Olivier Perrin. Trusternity: auditing transparent log server with blockchain. In *Companion Proceedings of the The Web Conference 2018*, pages 79–80, 2018.

[138] Johnathan Nightingale. Fraudulent *.google.com certificate, August 2011.

[139] Kirill Nikitin, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Justin Cappos, and Bryan Ford. {CHAINIAC}: Proactive software-update transparency via collectively signed skipchains and verified builds. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1271–1287, 2017.

[140] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R O'Brien, Thomas Steinke, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. *Harv. JL & Tech.*, 31:687, 2017.

[141] Chris Norval, Kristin Cornelius, Jennifer Cobbe, and Jatinder Singh. Disclosure by design: Designing information disclosures to support meaningful transparency and accountability. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 679–690, 2022.

[142] Onora O Neill. Transparency and the ethics of communication. In *Proceedings-British Academy*, volume 1, pages 75–90. Oxford University Press, 2006.

[143] Barack Obama. Transparency and open government. *Memorandum for the heads of executive departments and agencies*, 2009.

[144] Kieron O'Hara. Transparent government, not transparent citizens: a report on privacy and transparency for the cabinet office. 2011.

[145] Onora O'neill. *A question of trust: The BBC Reith Lectures 2002*. Cambridge University Press, 2002.

[146] Devon O'Brien, Ryan Sleevi, and Andrew Whalley. Chrome's plan to distrust symantec certificates, 2017.

[147] Gaurav Panwar, Roopa Vishwanathan, Satyajayant Misra, and Austin Bos. Sampl: Scalable auditability of monitoring processes using public ledgers. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2249–2266, 2019.

[148] Roel Peeters and Tobias Pulls. Insynd: Improved privacy-preserving transparency logging. In *European Symposium on Research in Computer Security*, pages 121–139. Springer, 2016.

[149] Yanqing Peng, Min Du, Feifei Li, Raymond Cheng, and Dawn Song. Falcondb: Blockchain-based collaborative database. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 637–652, 2020.

[150] Alexey Pertsev, Roman Semenov, and Roman Storm. Tornado cash privacy solution version 1.4. 2019.

[151] Stijn Pletinckx, Thanh-Dat Nguyen, Tobias Fiebig, Christopher Kruegel, and Giovanni Vigna. Certifiably vulnerable: Using certificate transparency logs for target reconnaissance. In *2023 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023.

[152] The Data Nutrition Project. The data nutrition project, 2021.

[153] Tobias Pulls and Roel Peeters. Balloon: A forward-secure append-only persistent authenticated data structure. In *European Symposium on Research in Computer Security*, pages 622–641. Springer, 2015.

[154] Tobias Pulls, Roel Peeters, and Karel Wouters. Distributed privacy-preserving transparency logging. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 83–94, 2013.

[155] Emilee Rader, Kelley Cotter, and Janghee Cho. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018.

[156] Daniel Reijsbergen, Aung Maw, Zheng Yang, Tien Tuan Anh Dinh, and Jianying Zhou. Tap: Transparent and privacy-preserving data services. *arXiv preprint arXiv:2210.11702*, 2022.

[157] Reuters. Uber drivers consider appeal in dutch case over data access, 2021.

[158] Richard Roberts and Dave Levin. 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*, pages 87–92, 2019.

[159] David Robinson, Harlan Yu, William P Zeller, and Edward W Felten. Government data and the invisible hand. *Yale JL & Tech.*, 11:159, 2008.

[160] Mark Dermot Ryan. Enhanced certificate transparency and end-to-end encrypted mail. In *NDSS*, pages 1–14, 2014.

[161] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[162] Wiwi Samsul, Henri P Uranus, and MD Birowosuto. Recognizing document's originality by laser surface authentication. In *2010 second international conference on advances in computing, control, and telecommunication technologies*, pages 37–40. IEEE, 2010.

[163] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.

[164] Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C Schmidt, and Matthias Wählisch. The rise of certificate transparency and its implications on the internet ecosystem. In *Proceedings of the Internet Measurement Conference 2018*, pages 343–349, 2018.

[165] Bruce Schneier and John Kelsey. Cryptographic support for secure logs on untrusted machines. In *USENIX Security Symposium*, volume 98, pages 53–62. San Antonio, TX, 1998.

[166] Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security (TISSEC)*, 2(2):159–176, 1999.

[167] Guido Schryen. Is open source security a myth? *Communications of the ACM*, 54(5):130–140, 2011.

[168] Oshani Seneviratne and Lalana Kagal. Enabling privacy through transparency. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pages 121–128. IEEE, 2014.

[169] Nigel Shadbolt, Kieron O'Hara, Tim Berners-Lee, Nicholas Gibbins, Hugh Glaser, Wendy Hall, et al. Linked open government data: Lessons from data. gov. uk. *IEEE Intelligent Systems*, 27(3):16–24, 2012.

[170] Ashlesh Sharma, Lakshminarayanan Subramanian, and Eric A Brewer. Paperspeckle: microscopic fingerprinting of paper. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 99–110, 2011.

[171] Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A Erdogdu, and Ross Anderson. Manipulating sgd with data ordering attacks. *arXiv preprint arXiv:2104.09667*, 2021.

[172] Jatinder Singh and Jennifer Cobbe. The security implications of data subject rights. *IEEE Security & Privacy*, 17(6):21–30, 2019.

[173] Ameen Soleimani. Privacy pools with opt-in or opt-out anonymity sets, 2022.

[174] Dayana Spagnuelo, Ana Ferreira, and Gabriele Lenzini. Accomplishing transparency within the general data protection regulation. In *ICISSP*, pages 114–125, 2019.

[175] Dayana Spagnuelo, Ana Ferreira, and Gabriele Lenzini. Transparency enhancing tools and the gdpr: Do they match? In *International Conference on Information Systems Security and Privacy*, pages 162–185. Springer, 2019.

[176] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data–anonymisation groundhog day. *arXiv preprint arXiv:2011.07018*, 2021.

[177] E. Stark, R. Sleevi, R. Muminovic, D. O'Brien, E. Messeri, A. P. Felt, B. McMillion, and P. Tabriz. Does certificate transparency break the web? measuring adoption and error rate. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 211–226, 2019.

[178] Emily Stark, Joe DeBlasio, and Devon O'Brien. Certificate transparency in google chrome: Past, present, and future. *IEEE Security & Privacy*, 19(6):112–118, 2021.

[179] Cynthia Stohl, Michael Stohl, and Paul M Leonardi. Digital age—managing opacity: Information visibility and the paradox of transparency in the digital age. *International Journal of Communication*, 10:15, 2016.

[180] R Stradling. Certificate transparency version 2.0 draft-ietf-trans-rfc6962-bis-39. 2021.

[181] Josh Swihart, Benjamin Winston, and Sean Bowe. Zcash counterfeiting vulnerability successfully remediated, 2019.

[182] Roberto Tamassia. Authenticated data structures. In *European symposium on algorithms*, pages 2–5. Springer, 2003.

[183] Roger Taylor and Tim Kelsey. *Transparency and the open society: Practical lessons for effective policy*. Policy Press, 2016.

[184] the New York Times. Changes to the census could make small towns disappear, february 2020.

[185] Alin Tomescu, Vivek Bhupatiraju, Dimitrios Papadopoulos, Charalampos Papamanthou, Nikos Triandopoulos, and Srinivas Devadas. Transparency logs via append-only authenticated dictionaries. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1299–1316, 2019.

[186] Alin Tomescu and Srinivas Devadas. Catena: Efficient non-equivocation via bitcoin. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 393–409. IEEE, 2017.

[187] Ehsan Toreini, Siamak F Shahandashti, and Feng Hao. Texture to the rescue: Practical paper fingerprinting based on texture patterns. *ACM Transactions on Privacy and Security (TOPS)*, 20(3):1–29, 2017.

[188] Matteo Turilli and Luciano Floridi. The ethics of information transparency. *Ethics and Information Technology*, 11(2):105–112, 2009.

[189] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. " your hashed ip address: Ubuntu." perspectives on transparency tools for online advertising. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 702–717, 2019.

[190] Frerik van Beijnum, EG van Putten, KL Van der Molen, and AP Mosk. Recognition of paper samples by correlation of their speckle patterns. *arXiv preprint physics/0610089*, 2006.

[191] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient {Out-of-Order} execution. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 991–1008, 2018.

[192] Nicole Van der Meulen. Diginotar: Dissecting the first dutch digital disaster. *Journal of Strategic Security*, 6(2):46–58, 2013.

[193] Max Van Kleek, Ilaria Liccardi, Reuben Binns, Jun Zhao, Daniel J Weitzner, and Nigel Shadbolt. 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*, pages 5208–5220, 2017.

[194] Hal Varian. System reliability and free riding. In *Economics of information security*, pages 1–15. Springer, 2004.

[195] Giridhari Venkatadri, Alan Mislove, and Krishna P Gummadi. Treads: transparency-enhancing ads. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pages 169–175, 2018.

[196] Ben Wagner, Krisztina Rozgonyi, Marie-Therese Sekwenz, Jennifer Cobbe, and Jatinder Singh. Regulating transparency? facebook, twitter and the german network enforcement act. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 261–271, 2020.

[197] S. Wang, E. Toreini, and F. Hao. Anti-counterfeiting for polymer banknotes based on polymer substrate fingerprinting. *IEEE Transactions on Information Forensics and Security*, 16:2823–2835, 2021.

[198] Brent R Waters, Dirk Balfanz, Glenn Durfee, and Diana K Smetters. Building an encrypted and searchable audit log. In *NDSS*, volume 4, pages 5–6, 2004.

[199] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L Mazurek, and Blase Ur. 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*, pages 149–166, 2019.

[200] Daniel Weitzner, Hal Abelson, Tim Berners-Lee, Christ Hanson, Jim Hendler, Lalana Kagal, D McGuinness, Gerry Sussman, and K Krasnow Waterman. Transparent accountable inferencing for privacy risk management. In *AAAI Spring Symposium on The Semantic Web meets eGovernment*. AAAI Press, Stanford University, 2006.

[201] Daniel J Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman. Information accountability. *Communications of the ACM*, 51(6):82–87, 2008.

[202] Daniel J Weitzner, Harold Abelson, Tim Berners-Lee, Chris Hanson, James Hendler, Lalana Kagal, Deborah L McGuinness, Gerald Jay Sussman, and K Krasnow Waterman. Transparent accountable data mining: New strategies for privacy protection. 2006.

[203] Adrian Weller. Transparency: motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40. Springer, 2019.

[204] Wikipedia. Usage share of web browsers, 2022.

[205] Wiktionary. Transparent.

[206] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[207] Ben Worthy. 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.

[208] Mike Wu, Will McTighe, Kaili Wang, Istvan A Seres, Nick Bax, Manuel Puebla, Mariano Mendez, Federico Carrone, Tomás De Mattey, Herman O Demaestri, et al. Tutela: An open-source tool for assessing user-privacy on ethereum and tornado cash. *arXiv preprint arXiv:2201.06811*, 2022.

[209] Harlan Yu and David G Robinson. The new ambiguity of open government. *UCLA L. Rev. Discourse*, 59:178, 2011.

[210] Mingchao Yu, Saeid Sahraei, Songze Li, Salman Avestimehr, Sreeram Kannan, and Pramod Viswanath. Coded merkle tree: Solving data availability attacks in blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 114–134. Springer, 2020.

[211] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. vsql: Verifying arbitrary sql queries over dynamic outsourced databases. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 863–880. IEEE, 2017.

[212] Yupeng Zhang, Jonathan Katz, and Charalampos Papamanthou. Integridb: Verifiable sql for outsourced databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1480–1491, 2015.

[213] Christian Zimmermann. A categorization of transparency-enhancing technologies. *arXiv preprint arXiv:1507.04914*, 2015.