Accountability in Computing: Concepts and Mechanisms

Other titles in Foundations and Trends® in Privacy and Security

A Pragmatic Introduction to Secure Multi-Party Computation
David Evans, Vladimir Kolesnikov and Mike Rosulek
ISBN: 978-1-68083-508-3

Contextual Integrity through the Lens of Computer Science
Sebastian Benthall, Seda Gurses and Helen Nissenbaum
ISBN: 978-1-68083-384-3

Methods for Location Privacy: A comparative overview
Kostantinos Chatzikokolakis, Ehab ElSalamouny, Catuscia Palamidessi and Pazii Anna
ISBN: 978-1-68083-366-9

Principles and Implementation Techniques of Software-Based Fault Isolation
Gang Tan
ISBN: 978-1-68083-344-7

Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif
Bruno Blanchet
ISBN: 978-1-68083-206-8

Accountability in Computing: Concepts and Mechanisms

Joan Feigenbaum
Yale University
joan.feigenbaum@yale.edu

Aaron D. Jaggard
U.S. Naval Research Laboratory
aaron.jaggard@nrl.navy.mil

Rebecca N. Wright
Barnard College
rwright@barnard.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Privacy and Security

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

J. Feigenbaum, A. D. Jaggard and R. N. Wright. *Accountability in Computing: Concepts and Mechanisms*. Foundations and Trends® in Privacy and Security, vol. 2, no. 4, pp. 247–399, 2020.

ISBN: 978-1-68083-785-8

Foundations and Trends® in Privacy and Security
Volume 2, Issue 4, 2020

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Privacy and Security publishes survey and tutorial articles in the following topics:

- Access control
- Accountability
- Anonymity
- Application security
- Artifical intelligence methods in security and privacy
- Authentication
- Big data analytics and privacy
- Cloud security
- Cyber-physical systems security and privacy
- Distributed systems security and privacy
- Embedded systems security and privacy
- Forensics
- Hardware security
- Human factors in security and privacy
- Information flow
- Intrusion detection
- Malware
- Metrics
- Mobile security and privacy
- Language-based security and privacy
- Network security
- Privacy-preserving systems
- Protocol security
- Security and privacy policies
- Security architectures
- System security
- Web security and privacy

Information for Librarians

Foundations and Trends® in Privacy and Security, 2020, Volume 2, 4 issues. ISSN paper version 2474-1558. ISSN online version 2474-1566. Also available as a combined paper and online subscription.

Contents

Accountability in Computing: Concepts and Mechanisms

Joan Feigenbaum[1], Aaron D. Jaggard[2] and Rebecca N. Wright[3]

[1] *Yale University; joan.feigenbaum@yale.edu*
[2] *U.S. Naval Research Laboratory; aaron.jaggard@nrl.navy.mil*
[3] *Barnard College; rwright@barnard.edu*

ABSTRACT

Accountability is a widely studied but amorphous concept, used to mean different things across different disciplines and domains of application. Here, we survey work on accountability in computer science and other disciplines. We motivate our survey with a study of the myriad ways in which the term "accountability" has been used across disciplines and the concepts that play key roles in defining it. This leads us to identify a temporal spectrum onto which we may place different notions of accountability to facilitate their comparison. We then survey accountability mechanisms for different application domains in computer science and place them on our spectrum. Building on this broader survey, we review frameworks and languages for studying accountability in computer science. Finally, we offer conclusions, open questions, and future directions.

Joan Feigenbaum, Aaron D. Jaggard and Rebecca N. Wright (2020), "Accountability in Computing: Concepts and Mechanisms", Foundations and Trends® in Privacy and Security: Vol. 2, No. 4, pp 247–399. DOI: 10.1561/3300000002.

1

Introduction: The Problem of "Accountability"

1.1 Motivation

The best known and most widely deployed security technologies—*e.g.*, passwords, authentication protocols, firewalls, and access-control mechanisms—are *preventive* in nature. The idea is to stop unauthorized parties before they can download confidential data that they are not supposed to have access to, login to a proprietary network of an organization that they do not belong to, or take any other action that violates system policy. However, dramatically increased scale and complexity of Internet commerce, social networking, remote work, distance learning, and myriad other forms of social, economic, and intellectual engagement online with both strangers and friends has rendered preventive mechanisms inadequate. The result is growing interest in *accountability* mechanisms to complement preventive measures.

Despite widespread agreement that "accountability" is needed if online life is to flourish, the term has no universally accepted definition. However, the concept has been studied extensively, both in computer science and in other disciplines. Our purpose in this monograph is to survey and contextualize these investigations, to identify key ideas, and to suggest interest directions for future research.

An essential premise for the study of accountability in computer science is that it is a natural approach to the design and implementation of security and privacy in computer systems. After all, it is a combination of preventive security measures and after-the-fact accountability with which rules have always been enforced in the offline world. We offer three examples of real-world scenarios in which after-the-fact accountability mechanisms complement preventive security in essential ways.

Digital copyright: US copyright law is a clear example of a set of policies that cannot be enforced in a purely preventive manner if they are to achieve their goals. One bedrock copyright principle is that the creator of a copyright work has certain exclusive rights, including the right to control copying and distribution of his work and the right to authorize or refuse to authorize the creation of derivative works (such as sequels or movie versions of books). However, the law also specifies exceptions to these exclusive rights in the form of fair-use provisions. Under the fair-use doctrine, a researcher, for example, may make a small number of copies of a scientific journal paper for use by the members of his lab without obtaining the author's permission, but he may not (without the author's permission) share the article with everyone at his university or some other wide audience. A properly attributed excerpt from a newspaper story may, without the author's permission, be copied and distributed widely without infringing copyright or committing plagiarism. The notion of fair use promotes socially desirable activities, such as education and criticism, and is regarded by many as an essential pillar of cultural production.

In the analog world of books, magazines, newspapers, and academic journals, there is no attempt at preventive enforcement of copyright law. It is technologically feasible to violate the law by making and distributing many unauthorized copies of a book, but anyone who does so runs the risk of being caught and sued for copyright infringement (*i.e.*, being "held accountable" for an illegal actions), and in any case one incurs the nonnegligible cost of copying and distribution. The fact that copyright enforcement is based on detection rather than prevention supports fair use. To determine whether, and if so how, one wants to use a document and whether such a use requires the author's permission,

one must be able to read (and, in particular, to have access to) the document; preventive copyright enforcement might restrict access to those who could justify that access *a priori* and those who are willing and able to pay for access.

Digital creation and distribution of books, songs, movies, *etc.* has motivated attempts at preventive copyright enforcement. Digital-Rights-Management (DRM) systems are justified in part by the negligible cost of copying and distributing digital works. Unfortunately, some DRM systems impose severe limits (or even prohibition) on uncompensated use of the works they manage. Crafting these limits in a manner that is consistent with the goals of copyright law is certainly hard and may be impossible; if the limits are very strict, they threaten fair use, but, if they are too permissive, the works might be too easily copied and distributed and the creators' rights vitiated. We believe that access and accountability together form a better approach to digital copyright than draconian forms of preventive DRM. Allowing users to access digital copyright works, just as they access analog works when browsing in physical stores and libraries, is consistent with enforcement of creators' rights provided that they are held accountable for subsequent use of those works in accordance with copyright law.

Break-glass scenarios: In some emergency situations, there is a clear need to augment or complement traditional, preventive access controls and usage policies. They are often called *break-glass scenarios*—a reference to the fact that one often must break a glass cover in order to pull a fire alarm.

For example, a physician in one medical practice may not, as a routine matter, have access to the patient records of another medical practice. If that physician encounters a patient of the other practice who needs emergency treatment, she could present her medical ID and a description of the emergency to the other practice and be granted temporary access to the patient's records. The information she provides to the emergency-care team should be logged, and all emergency-access logs should be audited periodically. If a doctor is discovered to have used her medical ID in this manner when not in a true emergency, the legal and professional penalties would be significant. The combination

of secure logs and substantial penalities should deter abuse of the emergency-access system and thus support patients' privacy—in other words, physicians would be "held accountable" for proper emergency use of patients' records.

Credit-card authorization: Retail use of credit cards is an excellent example of how accountability and authorization can prevail without strong authentication or even a conventional notion of identification. Thus, it supports our contention that "accountability" is not simply a matter of identifying all participants in a system, keeping track of all of their actions, and punishing actors who break the rules.

If Jane, a customer, attempts to pay for something in a store with her neighbor Mary's credit card, and Mary's card is far enough below its spending limit to accommodate the purchase (and has not been reported stolen), then Jane is unlikely to encounter any objections by the merchant. So has the credit-card authorization system functioned properly? Detailed examination of the process is instructive.

First, note that the merchant is the resource controller in this example. It is he who seeks assurance that, after a customer leaves his store with an item, he will be paid for that item. The system that he uses to obtain this assurance has both a technological component and a legal component: He can swipe the offered card, enter the price of the item and, after a few seconds, receive official approval or rejection of the purchase; if the purchase is approved, then the card issuer is legally obligated to pay him, regardless of any dispute that may subsequently arise between the issuer and the cardholder. (Presumably, if such a dispute had already arisen, the purchase would not be approved.)

Thus, whether it is the cardholder (Mary in our example) or someone else (*e.g.*, her neighbor Jane) who presents the card does not matter to the merchant—the resource that he controls is his store inventory, and the payment stream that he cares about is the one from the card issuer to him. Control over the card as a resource and concern about payment by the cardholder is the concern of the card issuer, not the merchant.

Note that the correctness conditions for which parties might be held accountable are not necessarily directly aligned.

Merchant: As noted above, the merchant desires assurance that, if an item leaves the store, he will receive its price.

Customer: The customer does not want to become obligated to pay any more than the price of the items she takes from the store.

Card issuer: The credit-card issuer desires assurance that, if he becomes obligated to the merchant for some amount, then some customer becomes obligated to the card issuer for at least that amount.

This example exposes the existence of legitimate confusion and some of the questions that are essential to ask when we look at an accountability problem. Who is accountable, and for what? To whom (if anyone) are they accountable? We elaborate on this legitimate confusion in the next section.

1.2 Why "accountability" is hard to pin down

Why is the concept of accountability is so elusive? Why has the term been defined in so many different ways (some of them mutually contradictory), particularly by computer scientists, and why are some uses of the term considered counter-intuitive or misleading? We ask these questions in order to help clarify the scope of our survey, not because we expect to resolve all of the terminological confusion surrounding accountability. Following Koppell (2005), we "do not suggest a new, all-encompassing definition of the word. There are enough already!"

We believe (and provide evidence in this section) that there is no agreement on whether or not accountable systems require certain basic system properties, *e.g.*, persistent identities of system participants, public identification of alleged policy violators and formal adjudication of allegations, or quantifiable punishment of those proven to be violators.

As explained in Sec. 1.1, computer scientists have traditionally approached information security through prevention: Before taking any security-sensitive action, an entity is expected prove that it is authorized to do so. In online life, which is characterized by enormous scale and complexity, the purely preventive approach to security has proven to be insufficient. Several researchers, including Weitzner, Abelson, Berners-Lee, Feigenbaum, Hendler, and Sussman (Weitzner *et al.*, 2008), Lampson (2009), and Datta (2014), have suggested that the preventive approach be augmented by an accountability approach. Our goal in

writing this survey is to focus attention on the broadest possible class of information systems that take an accountability approach—roughly speaking, on systems in which policy violations are punished. Following Weitzner *et al.*, we call these *accountable systems*. Traditional preventive measures are not precluded in accountable systems. However, when such a system cannot simply prevent all policy violations by using passwords, authentication protocols, and other classic security mechanisms, it *ensures that users who violate system policy incur negative consequences*. Both conceptually and pragmatically, this is a natural approach to the design and implementation of policy-governed information systems; after all, it is a combination of before-the-fact prevention and after-the-fact punishment with which laws and policies have always been enforced in the offline world.

1.2.1 Responsibility, adjudication, and sanctions

Lampson (2005a) put forth a simple but apparently quite general formulation of the term in the context of information systems: "Accountability is the ability to hold an entity, such as a person or organization, responsible for its actions." This formulation is similar to the one in earlier work in political science by Grant and Keohane (2005), who say that accountability "implies that some actors have the right to hold other actors to a set of standards, to judge whether they have fulfilled their responsibilities in light of these standards, and to impose sanctions if they determine that these responsibilities have not been met." However, the differences between these two formulations are typical of the difficulties one encounters in this area.

For example, note that Grant and Keohane speak of the *right* of some actors to behave in certain ways in order to hold others responsible, while Lampson speaks of the *ability* to hold others responsible. This difference may reflect the disciplines within which the research was conducted. Rights are a central focus in political science, and it is perfectly natural to assign particular rights to entities in a political system even if those entities do not have the ability to exercise said rights. In computer science, the focus is on the capabilities and limitations of various entities in a system and on the interactions among entities, and

it is not clear why one would be interested in an entity's having the right to do something if it did not have the technical ability to do it.

Note further that Grant and Keohane assume a system in which there are at least two actors, say A and B, one of whom (say B) is accountable to the other; moreover, A has the right both to judge whether B has fulfilled his responsibilities and to impose sanctions on him if he has not. Lampson does not assume that judgment and sanctions are performed by the same entity. Indeed, he does not say anything about judgment or sanctions: Accountability in his formulation is a system property; whether people and organizations in the system must be explicitly judged and, if so, whether the entity that judges them is the same as the one that holds them responsible is not specified.

1.2.2 Automatic *vs.* mediated punishment

In earlier work, we have formalized accountability so as to include accountable systems in which there are no explicit adjudication procedures (Feigenbaum *et al.*, 2011). What is required in an accountable system is that entities that violate system policies are punished, by which we mean that a violator's *utility* is lowered as a result of the violation. Our formal framework treats in a unified manner systems in which particiants are punished *automatically* and those in which punishment is *mediated* by a judge. Participants are punished automatically when the very act of violating a system policy causes their utility to decline.

Automatic punishment of this form need not expose the identity of a violator or the nature of his violation to the rest of the system participants; in fact, the violator himself need not be aware that he has violated a system policy. The key feature of this unified framework is that, because a violator's utility is decreased as a result of his violating a system policy, participants' actions are tied to consequences. Thus, we have advocated shifting focus from the *procedures* used by accountable systems (where it is in, *e.g.*, the Grant and Keohane formulation) to the *meaning* of accountability and, specifically, what accountability mechanisms must provide if they are to be a useful complement to preventive mechanisms.

The idea of a mechanism that imposes consequences for policy violations automatically, without exposing the violation to system participants (or even to the violator), may seem odd in a computer-security context, but it is actually standard in several fields, including economics. In a "truthful" (or "strategyproof") sealed-bid auction, a bidder's utility is (provably) maximized by his honestly bidding the maximum price that he is willing to pay for the item. A bidder who violates the "bid your true value for the item" policy may wind up losing utility because he wins the auction but pays more than he actually thinks the item is worth or because he loses the auction to someone else who did not value the item as highly as he did. The auction mechanism imposes consequences upon policy violators automatically in the process of choosing a winner and setting a price.

The main objection to our classifying this standard economic notion of *incentive compatibility* and other automatic-punishment mechanisms as forms of accountability is that they do not necessitate "calling the violator to account" or "making him account for himself" that popular usage of the term connotes and that some theorists of accountability, *e.g.*, Mulgan (2000), have identified as a core component. In that view of accountability, there must be *social exchange* between an accountable entity and the entity calling for the account, and there is social value in a *public accounting* that makes clear to all entities in the system the nature of the violation and the consequences that attach to it. The objection is not that automatic punishment without public accounting has no value but rather that it is not properly regarded as an "accountability" mechanism; some have suggested that *deterrence* is a better term for the system property that such mechanisms provide.

1.2.3 Identity, anonymity, and pseudonymity

The Grant–Keohane conception of accountability presented in Sec. 1.2.1 seems to assume that participants in accountable systems have persistent identities and, in particular, that they are identifiable by those who have the right to hold them accountable. But online interaction is sometimes anonymous or at least pseudonymous, and this characteristic of online life is highly valued by many. Indeed, the intuition that

support for "accountability" in online life necessarily implies opposition to anonymity and pseudonymity has caused many cyber-rights advocates to be suspicious of the quest for accountability. Because accountability and anonymous and pseudonymous interactions are all worthwhile goals, we ask whether they must be in tension. As usual, that depends on what one means by "accountability."

In computer science, different researchers have taken a wide range of approaches to participants' identities in accountable systems. For example, several influential experimental works (*e.g.*, Andersen *et al.*, 2008; MIT Decentralized Information Group, 2009; MIT Decentralized Information Group, 2010; MIT Decentralized Information Group, 2011) require that system participants have persistent identities that are known to those who hold them accountable. Anonymous participation in such an accountable system is not possible. Other influential research (*e.g.*, Camenisch and Lysyanskaya, 2001; Camenisch *et al.*, 2007; Chaum, 1982; Corrigan-Gibbs and Ford, 2010) exemplifies a completely different (and incompatible) view of the role of identity in accountable systems; in these works, a participant is held accountable precisely in the sense that, under normal circumstances, he is anonymous, but his identity can be exposed if he violates the prescribed security policy or protocol.

We believe that neither of these approaches is sufficiently general for the plethora of online interactions in which a robust notion of accountability is desirable. Our notion of accountable systems in which punishment is automatic is fully consistent with anonymous participation. More generally, we have explored accountable systems in which participants may be bound to their system identities with varying degrees of strength as a condition of participation (Feigenbaum *et al.*, 2014).

1.2.4 Concepts *vs.* terminology

Adjudication and identification are just two of many concepts whose relationships to "accountability" are handled differently by different researchers in multiple disciplines. In the following chapters, we explore these relationships and their implications for the power and limitations of accountable systems. This exploration will provide an opportunity to

clear up, to some extent, the terminological confusion that has bedeviled the study of accountability.

However, as explained above, our primary focus is not on perfecting a taxonomy of security properties. Rather, our goal is a comprehensive exploration of techniques that can usefully complement traditional preventive security mechanisms in that they can enable punishment of policy violations in a variety of realistic scenarios. Ultimately, some of these techniques may be termed "detection," "deterrence," "incentive compatibility," *etc.*, but they are within scope of this investigation if they are not purely preventive and are potentially useful in the search for systems and applications that are more secure, more usable, and more compliant with agreed-upon policies.

We conclude this section with an eloquent cautionary note from Charles Raab (2012, p. 24): "[T]he short message is that 'accountability' is not a term to be trifled with, or used casually and rhetorically, or as a fashion accessory."

1.3 Remarks on vocabulary

Many volumes have been written about *secure systems*—roughly speaking, systems in which policy violations cannot occur, because preventive security measures stop the participants from committing them. In this volume, we consider *accountable systems*—roughly speaking, systems in which policy violations, when they occur, are punished. We use the word "system" to mean an application or network protocol (*e.g.*, an auction service or a multicast protocol) that has a "goal" (determining winners and prices or delivering content to all the subscribers, respectively). The system "policy" specifies how the participants are supposed to behave, and the accountability "mechanism" ensures—or at least facilitates—their punishment if they violate the policy. Note that accountable systems may also use preventive measures (*e.g.*, passwords or authorization protocols) to stop certain policy violations from occurring, but they operate under the assumption that some policy violations might occur and seek to impose consequences when they do.

Participants (also referred to as "principals" or "actors") in an accountable system are computational agents that may represent human

beings. Some are good actors, meaning that they always comply with the system policy, and some are bad actors (attackers, intruders, or other miscreants), meaning that they at least sometimes violate system policy. The purpose of preventive measures is to stop bad actors from accessing the system; when that is not possible, the purpose of accountability mechanisms is to impose, or enable the imposition of, consequences on bad actors. One important aspect in which accountable systems differ is in whether the participants have persistent "identities" that are known to other participants or, rather, may participate anonymously. Note that, because participants are computational agents, one flesh-and-blood human may be represented in the system by more than one agent and thus may have more than one identity.

In some accountable systems, all actors are equivalent, but in others they play different roles. For example, in keeping with the common-parlance meaning of the word "accountability," it may fall to one actor, in his role as a judge, to call to account another actor, who has been accused of a policy violation. In his attempt to defend himself, the accused may provide "evidence" of the actions that he has taken or "credentials" that prove that he is authorized by the system policy to have taken those actions.

Note that we refer to a design as a "system" if actors participate in it for reasons other than "accountability," *e.g.*, to share updates with acquaintances or to compute a function, and we refer to it as a "mechanism" if participation is for the express purpose of providing accountability-related properties. In a small number of works that we consider, it may not be immediately clear which term applies to a particular design, because actions taken by the participants may serve *both* to accomplish goals such as sharing updates or computing functions *and* to provide accountability. We will identify the cases in which accountability is intertwined with system goals in this fashion.

1.4 "Accountability" implicates many areas of computer science

The study of accountability touches upon many topics in computer-science research, all of which have their own specialized vocabularies. Examples include:

Distributed computation: Accountable systems are, in general, distributed systems in the sense that there are multiple participants and that they use multiple computers. Similarly, accountability mechanisms are typically implemented as distributed algorithms that run on multiple computers.

It is important to distinguish between distributed computations that are *centralized* and those that are *decentralized*. In the centralized case, there is unified administrative control over the entire computation. Different processes may run on different machines, but they do not make independent decisions or respond to competing incentives; in other words, all of the participants are part of the same organization or *administrative domain*. In the decentralized case, different participants not only use different machines but can be organizationally or economically separate as well; they make strategic decisions independently of each other and may have competing interests.

Because most interesting distributed systems and networks are *asynchronous*, they present technical challenges for accountability mechanisms that rely on *tamper-evident* logging to preserve evidence. Definitions of accountability in asynchronous distributed systems draw on work in *fault detection*, focusing on guarantees that violations are *eventually* detected and that valid evidence cannot be created against policy-compliant participants.

Logic and language: Proofs and evidence are intrinsic to the goals of many accountability mechanisms. Participants may be called upon to prove that they fulfilled all of their responsibilities (as defined by the system policy), to prove that someone else violated the policy, to prove that evidence was acquired at a certain time and has not been tampered with, *etc.* Even mechanisms that do not demand fully rigorous mathematical proofs often rely on interactions among participants that benefit from formal reasoning. Therefore, researchers have proposed a number of proof logics and programming languages for specifying, implementing, and reasoning about accountability mechanisms. Many of these contributions draw on previous work on *modal logic* (particularly *temporal logic*), *belief logic*, and *causality*. Important distinctions among these logics and languages include whether or not proofs are fully

automated or require human intervention and whether they enable the identification of *actual causes* and the participants responsible for them or the weaker notion of a set of *all possible causes*.

Game theory: Key elements of several approaches to accountability include blame and punishment. How to "punish" participants for a policy violation obviously depends on the nature of the accountable system in which they are participating. In systems that feature an intrinsic notion of *utility*, *e.g.*, those in which participants accumulate points or exchange money and maintain "bank" accounts, a natural way to punish a violator is to reduce his utility. One challenge that arises in this approach is the need to ensure that the punishing action that reduces a participant's utility is *causally* linked to the decision that he committed a violation; a mechanism cannot be said to have held a violator accountable if he loses utility because of some unrelated "bad luck" that he experiences after the violation. Another challenge is the question of whether punishment should be *targeted* or *collective*; a system in which all participants' utilities are reduced significantly whenever a violation occurs may deter violations, but collective punishment does not satisfy most people's intuitive understanding of an "accountability" mechanism.

Game-theoretic models have also been used in the study of accountability mechanisms that rely on *auditing*. For example, in *audit games*, the standard security-game framework (in which a defender chooses how to invest in defense, and an attacker chooses which systems to target) is enhanced with a notion of costly punishment. Audit games can be used to develop an efficient algorithm that determines an approximately optimal strategy for auditing.

We discuss these and other connections between accountability and incentives in Sec. 4.3.2

Cryptography: Like logic and languages, cryptographic techniques are useful in accountability mechanisms that need to construct evidence or proofs. *Signatures*, *timestamping*, *encryption*, *hashing*, and *authentication codes* are examples of cryptographic operations that allow participants who acquire data that they need to use as evidence (of a

policy violation or of policy compliance) to ensure that it is preserved in a confidential, tamper-evident fashion and to tie it securely to appropriate meta-data, *e.g.*, the time it was discovered or the identity of the discoverer.

Although explicit punishment plays an important role in some accountable systems, there are others in which *identification* of a policy violator is, by itself, considered an accountability mechanism. Often, there is a tacit assumption that, once identified, a violator will be expelled from the system; expulsion may be regarded as a qualitative form of punishment, in contrast to the quantitative punishments used in utility-based accountable systems. *Accountable anonymity* plays a crucial role in several applications, including digital-cash and group-commuication protocols; it guarantees that participants who follow the rules can remain anonymous but that those who deviate from the roles will have their identities revealed (at least with nontrivial probability).

Privacy-preserving, aggregate reporting has been proposed as an appropriate accountability technique in law-enforcement, surveillance, and other scenarios in which a government agency must act in secrecy but is required to have proper authorization and to follow rules. For example, a law-enforcement agency may be required to make public the approximate number of wiretaps that it conducts each year but not to reveal the identities or locations of the subjects of those wiretaps or the ongoing investigations for which it conducted them. The cryptographic technique of *secure, multiparty computation (SMPC)* has a natural role to play in this type of reporting. Individual, authenticated officers can submit required information about their wiretapping activities in an encrypted (or other privacy-preserving) form, and an SMPC protocol can check that all activities are in compliance with the applicable laws and procedures and compute the total number of wiretaps without revealing any details about subjects, locations, or ongoing investigations.

Formal methods: As we explain in Sec. 1.2 and Chap. 2, there are many plausible definitions of "accountability," some of which are quite subtle. In some frameworks, accountability must be interpreted in the context of a specific accountable system. For example, if one's general notion of accountability focuses on an actor's acquiring evidence

that can convince a judge of another actor's participation, a natural interpretation in the context of a certified-email system is that the recipient gets the email and the sender gets evidence that the recipient received the email. Formal methods such as *proof logics* and *theorem provers* are useful in precisely specifying properties that capture such context-specific interpretations (see Sec. 4.2.1). They can also be used for *automatically* or *semi-automatically* proving that the system has that property.

Accountability mechanisms that use logging and auditing are good candidates for formal methods. It is quite natural to try to achieve accountability by logging every action taken or message sent by a system participant, preserving the information in a tamper-evident manner, and examining it for proof of a violation and identification of the violator after an accusation is made or an alarm is raised. Unfortunately, capturing literally all of the information may be infeasible or may result in logs that are too voluminous to be audited in the time available. Automatically or semi-automatically producible proofs that systems that log more selectively preserve enough evidence to prove the desired accountability properties are highly desirable.

Of course, the notion of accountability has also been studied extensively in disciplines other than computer science. We address some of the similarities and differences between computer scientists' ideas on the subject and others' in Sec. 3.6.

1.5 Overview of contributions

Chapter 2 surveys *accountability-related concepts*. We present categorizations in terms of time, information, and action in Sec. 2.1; in particular, we identify a temporal spectrum of accountability goals that will prove useful in understanding work in the area: prevention, violation, detection, evidence, judgment or blame, and punishment. We subsequently use these categorizations to analyze different accountability mechanisms. In Sec. 2.2, we survey different definitions of "accountability" that are both explicit and implicit in the literature and categorize them according to their focus. The remainder of Chap. 2 discusses other accountability-related concepts.

Chapter 3 surveys *accountability mechanisms*, both implemented and proposed. Of course, the type of "accountability" provided by different mechanisms varies. Describing all of the technical details of the proposed mechanisms would be lengthy and likely not particularly useful. Instead, we identify the major distinct approaches to accountability and selected proposals that exemplify these approaches and locate them on the temporal spectrum put forth in Chap. 2. We categorize mechanisms according to how they achieve the accountability properties they provide, largely paralleling the categorization of definitions in Sec. 2.2. We summarize in Sec. 3.5 the properties of these accountability mechanisms through the lens of the time/information/action framework of Sec. 2.1. Sec. 3.6 focuses on mechanisms that have been proposed and studied in disciplines other than computer science.

Chapter 4 surveys *languages and frameworks for the study of accountability.* They are more abstract than the technical accountability mechanisms considered in Chap. 3. Languages and frameworks provide ways to describe or reason about accountable systems and accountability-related properties. We also present technical results on accountability and identity in Chap. 4. Here, the focus is on research in which accountability itself is the subject, as opposed to work that seeks to achieve a particular type of accountability in a particular context.

Finally, based on consideration of the material in this survey, Chap. 5 summarizes key ideas in the accountability literature, identifies key papers, and suggests directions for future research.

Index