

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



Documents

Document Sharing Hub

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

© 2020 J. Feigenbaum, A. D. Jaggard and R. N. Wright

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The 'services' for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Foundations and Trends® in Privacy and Security

Volume 2, Issue 4, 2020

Editorial Board

Editors-in-Chief

Anupam Datta

Carnegie Mellon University, USA

Jeannette Wing

Columbia University, USA

Editors

Martín Abadi

*Google and University of California,
Santa Cruz*

Michael Backes

Saarland University

Dan Boneh

Stanford University, USA

Véronique Cortier

LORIA, CNRS, France

Lorrie Cranor

Carnegie Mellon University

Cédric Fournet

Microsoft Research

Virgil Gligor

Carnegie Mellon University

Jean-Pierre Hubaux

EPFL

Deirdre Mulligan

University of California, Berkeley

Andrew Myers

Cornell University

Helen Nissenbaum

New York University

Michael Reiter

University of North Carolina

Shankar Sastry

University of California, Berkeley

Dawn Song

University of California, Berkeley

Daniel Weitzner

Massachusetts Institute of Technology



Documents

Document Sharing Hub

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
- Artificial 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

1	Introduction: The Problem of “Accountability”	2
1.1	Motivation	2
1.2	Why “accountability” is hard to pin down	6
1.3	Remarks on vocabulary	11
1.4	“Accountability” implicates many areas of computer science	12
1.5	Overview of contributions	16
2	Perspectives, Definitions, and Concepts Across Disciplines	18
2.1	Time, information, and action	19
2.2	Definitions of “accountability”	23
2.3	Accountability-related concepts and terms	45
3	Accountability Mechanisms and Domains across Disciplines	53
3.1	Evidence without focus on external parties	54
3.2	Evidence to present to external parties	62
3.3	Judgment or blame	72
3.4	Punishment	78
3.5	Summary of systems and mechanisms in computer science	88
3.6	Accountability mechanisms in other disciplines	93
4	Reasoning About Accountability	98
4.1	Tools for reasoning about accountability	99

4.2	Proofs about evidence in protocols	107
4.3	Accountability as a subject of study	109
5	Conclusions	119
5.1	Summary of key ideas	120
5.2	Key papers	124
5.3	Future work	125
	References	127
	Index	146

Accountability in Computing: Concepts and Mechanisms

Joan Feigenbaum¹, Aaron D. Jaggard² and Rebecca N. Wright³

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: 10.1561/33000000002.

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 penalties 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 participants 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-communication protocols; it guarantees that participants who follow the rules can remain anonymous but that those who deviate from the rules 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.

2

Perspectives, Definitions, and Concepts Across Disciplines

In this chapter, we identify broad themes that arise from considering accountability in computer science and other disciplines. We then synthesize some new perspectives on accountability that reflect this interdisciplinary point of view.

First, we argue that it is helpful to analyze accountability mechanisms in terms of the *time* at which they act, the *information* that they use, and the nature of the *action(s)* that they perform. The temporal spectrum that we identify separates different stages that are often conflated. It highlights *punishment* as the end goal and illuminates the fact that punishment is often implicitly assumed in mechanisms that operate at other times on this spectrum.

Second, we survey definitions of accountability. Some of them are formally stated in the literature, while others are implicit. We categorize definitions according to the various foci that emerge from our review: detection, evidence, identification or association of actions with the principals who perform them, answerability, blame, and punishment or deterrence.

Third, we discuss accountability-related concepts and terminology. Much of this discussion focuses on the relationships among principals,

their identities within a system, and their actions. Other work that we review considers the relationships of accountability to the concepts of transparency and causality.

2.1 Time, information, and action

As we survey approaches, we evaluate how they address three broad aspects of accountability: time, information, and action. We typically consider accountability with respect to a policy violation. The “time” aspect considers when the accountability mechanism is invoked relative to the time of the violation. The “information” aspect concerns what is known and by whom, and the “action” aspect concerns what is done and by whom. This analysis refines the one that we did with Xiao (Feigenbaum *et al.*, 2012).

2.1.1 The time aspect and mechanism goals

Different accountability mechanisms are focused on different times relative to a policy violation; often this reflects the fact that different mechanisms have different goals. For example, the formal framework of Küsters, Truderung, and Vogt (Küsters *et al.*, 2010) explicitly models and focuses on judgments or verdicts, *i.e.*, declarations that a system participant is guilty of committing a violation. By contrast, the formal framework that we presented in (Feigenbaum *et al.*, 2011) focuses on punishment, which, in other frameworks, typically follows a declaration of guilt. Within our punishment-focused framework, however, there need not be an explicit judgment that identifies an individual principal as guilty; so the punishment focus is indeed distinct from the judgment focus.

Motivated by this type of distinction, we have identified (Feigenbaum *et al.*, 2012; Feigenbaum *et al.*, 2014) a discrete spectrum of times at which an accountability mechanism might play a role. Although we categorize these points in terms of their goals or effects and not in terms of strict temporal relationships, there is a natural temporal ordering that applies.

Prevention: (At least partially) concerned with preventing violations; plays a role before a violation occurs

Violation: Plays a role at the time a violation occurs

Detection: Facilitates, enables, *etc.*, detection of a violation either at the time the violation occurs or afterward

Evidence: Helps gather or preserve evidence about a violation that may be used against the accused violator. Gathering and preservation may be (but are not required to be) connected to detection. In some settings, the evidence is intended for use by the system operator; in others, it is intended for presentation to a third party, *e.g.*, a judge in a court of law.

Judgment or blame: Renders a verdict about an actor's guilt or blameworthiness with respect to a violation. For example, the judgment might be a verdict in a court of law or a determination by a system administrator that a user violated system policy.

Punishment: Punishes a violator in some way. For example, a user who violated system policy may be banned from the system for a period of time.

A single accountability mechanism might be involved at multiple points on this spectrum.

We believe that punishment is the crucial element of accountability and have explored this idea formally in earlier work. One reason for this belief is that accountability definitions that focus on other points on our spectrum, *e.g.*, evidence or blame, often assume, implicitly or explicitly, that there is a punishing mechanism in their environments. While a definition might then say that an entity is “accountable” if there is evidence connecting it to its misdeeds or if there is a mechanism that will blame it for a policy violation that it commits, the assumption that there is a punishing mechanism means that evidence or blame alone is not really enough to achieve the intended effect. In some sense, the earlier points in our temporal spectrum are all building toward punishment.

We discuss the centrality of punishment and some other perspectives on accountability further in Sec. 2.2.8.

2.1.2 Information

One question about accountability is the extent to which it implicates privacy. We are concerned with the information learned about the violation, the violator, and actors who do not violate any policy. Relevant questions include:

Participant identity: Does the mechanism require system participants to have identities? If so, how broadly is an identity known (*e.g.*, is it only learned by a trusted third party, is it learned by a limited set of participants, or is it potentially learned by all participants)?

Violations: Are violations disclosed? If so, how broadly (with the same set of possible answers as for identity)? How soon after the violation is this information learned?

Violators: Is the violator identified as such? If so, how broadly is this identification made (with the same set of possible answers as for identity and violations)?

2.1.3 Action

We address the following questions about actions, both in general operation and in detection and punishment of policy violations.

Centralization (no violation): Is the accountability mechanism (as it operates in the absence of a detected violation) centralized or decentralized?

Centralization (violation): Does the mechanism respond to a violation (in the gathering of evidence, judgment, and punishment) in a centralized or decentralized way?

Punishment: If violators are punished, is punishment “automatic” or “mediated,” as those terms are used in (Feigenbaum *et al.*, 2011)? If punishment is mediated, is the mediator a participant who might play other roles in the system, or is it a specialized entity?

Relies on violator participation: To what extent does the functioning of the accountability mechanism rely upon continued participation by, or access to, the violator? For example, is the violator only punished if he continues to interact with the system?

2.1.4 Other approaches to categorizing accountability

Outside of computer science, Bovens (2007), using his sense of accountability given in Def. 2.9 below, categorizes types of accountability based on four orthogonal dimensions: the nature of the forum to which an actor owes an account (“to whom is account to be rendered?”); the nature of the actor rendering account to the forum (“who should render account?”); the nature of the conduct (“about what is the account to be rendered?”); and the nature of the obligation that leads the actor to render an account to a forum (“why the actor feels compelled to render account”). Bovens illustrates each dimension with multiple distinct examples, *e.g.*, political, legal, administrative, professional, and social accountabilities that are distinguished by the nature of the forum to which an account is rendered.

Also outside of computer science, Lindberg (2013, pp. 212–217) synthesizes a categorization that contrasts with our approach above. His view of accountability, highlighted in Sec. 2.2.4, is that it captures a principal’s delegating authority to an agent and then the agent’s providing information for, and justification of, its decisions. Lindberg identifies three dimensions of accountability relationships: the *source*, *i.e.*, whether the principal is “internal or external to the one being held accountable[;]” the *degree of control*, which might be high or low; and the *direction* of control, which might be upwards (as with, *e.g.*, elected representatives), downwards (as with managers requesting information from their subordinates), or horizontal (as among professional peers). In addition to these examples, Lindberg provides examples for each of the 12 points in this space.

These categorizations are closely tied to the definitions of “accountability” used by Bovens and Lindberg. The examples illustrating them are potentially informative but not directly related to the approach that we take here.

2.2 Definitions of “accountability”

In surveying the literature, we identify numerous definitions, both explicit and implicit, of “accountability.” Many definitions in the literature are essentially equivalent, and we do not present all of them. Rather, we categorize definitions based on their respective foci. Within each focus, we identify fundamentally distinct definitions and highlight some exemplars.

The first set of definitions focuses on detection of violations. The second set focuses on evidence of some sort, *e.g.*, the collection or presentation of evidence about participation or misbehavior. The third set is similar, but it focuses more tightly on identification of actors or the explicit association of actors with their actions. The fourth set addresses answerability for actions or explanation of actions to external parties; in some cases, it involves providing evidence, but we distinguish answerability from notions of evidence by focusing on frameworks in which the actors themselves provide the evidence (*e.g.*, that they acted properly). The fifth set of definitions focuses on blame for policy violations or misbehavior, and the sixth set focuses on punishment for such violations or misbehavior.

2.2.1 Detection

We first consider definitions in which the goal is to ensure that participants can detect policy violations. Although we classify them as “focused on detection,” they are closely related to the production and collection of evidence.

In the temporal spectrum of Sec. 2.1.1, “detection” refers to evidence collected by a participant, and “evidence” refers to evidence that is intended to convince a third party.

In the context of accountable storage, Yumerefendi and Chase (2007, p. 2) present the following definitions:

Definition 2.1. A system is *accountable* if it provides a means to detect and expose misbehavior by its participants.

A system is *strongly accountable* if it provides a means for each participant to determine for itself [whether] others are behaving correctly,

without trusting assertions of misbehavior by another participant who may itself be compromised.

In addition to the detection aspect highlighted here, they frame their approach as allowing participants to “demonstrate that their actions are *semantically* correct, as well as properly authorized” (Yumerefendi and Chase, 2007, p. 28). Thus they take the complementary approaches of allowing participants to detect misbehavior by others and providing participants with evidence to demonstrate (to other participants) that they have behaved correctly.

In arguing that “information accountability” should become a primary means through which society addresses the question of appropriate use of sensitive information, Weitzner *et al.* (2008) offer this general definition:

Definition 2.2. *Information accountability* means that

- The *use* of information should be transparent so that it is possible to determine whether a particular use is *appropriate under a given [information policy]*, and
- The system enables individuals and institutions to be held accountable for [policy violations].

This definition treats transparency as the means of allowing determination of violations and (implicitly) punishment.¹ By contrast, the Yumerefendi–Chase definitions do not specify the means by which detection must happen.

In the strong Yumerefendi–Chase and the Weitzner *et al.* definitions, there is a requirement that it be possible “to determine” whether a violation has taken place. Both strongly suggest that evidence of some sort is required for this to be done and, implicitly, that it must be computationally feasible to make the required determinations.

¹Orthogonal to our discussion here, this definition is also of interest because it shifts the focus from the disclosure of information to the use of information.

2.2.2 Evidence

Typically, definitions that focus on evidence highlight the need to convince a third party of something. In some but not all evidence-focused definitions, principals who have complied with system policy can convince the third party that they did not commit violations.

In defining a proof logic for accountability, Kailar (1996) views accountability as provable association:

Definition 2.3. Accountability is the property whereby the association of a unique originator with an object or action can be proved to a third party (*i.e.*, a party who is different from the originator and the prover).

Other definitions also focus on associating actions with principals, but this one explicitly requires provability to third parties. Some later work on logics for reasoning about accountability in protocols roughly follows the approach of Kailar. Kudo (1998) extends Kailar’s logic to a “temporal accountability logic” and uses it to analyze a protocol for submitting something to a server where the submission must be made within a certain time window. Kungpisdan and Permpoontanalarp (2002) stress the ability to hide information from the third party, saying that accountability “involves the ability of a party, called a prover[,] to convince another party, called a verifier, that a statement is true *without revealing any secret information to the verifier.*”

In describing an “accountable” timestamping service, Buldas, Lipmaa, and Schoenmakers (Buldas *et al.*, 2000b) focus on evidence that can be used to convince an external party that misbehavior has occurred (implicitly defining what it means to hold a participant accountable) and also on evidence that allows an honest participant to prove her innocence. In particular, they say that a timestamping service as a whole is “accountable” if (Buldas *et al.*, 2000b, p. 296)

whenever there is at least one uncorrupted party (say, one honest client interested in legal value of a particular timestamp) during the creation of stamped information:

Fraud detection The service makes the trusted third parties accountable for their actions by enabling a principal

to detect and later prove to a judge any frauds affecting the relative ordering between timestamps.

Anti-framing If a party has honestly followed the protocol but is still accused in forgeries, she can [disprove] any false accusations.

Bella and Paulson (2006) have taken accountability to mean the delivery to a principal of evidence, which can then be presented to a judge, of another principal's participation in a protocol. In general, their formal approach involves proving (1) the validity of the evidence and (2) some sort of fairness, *i.e.*, that one protocol participant gets evidence if and only if the other one does. Their general approach requires interpretation in the context of an individual protocol; we discuss their particular results in Sec. 4.2.1.

The PeerReview mechanism of Haeberlen, Kouznetsov, and Druschel (Haeberlen *et al.*, 2007) uses the following notion of accountable systems.

Definition 2.4. [A]n *accountable system* maintains a tamper-evident record that provides non-repudiable evidence of all nodes' actions.

Although this definition is stated in terms of evidence, it does not explicitly discuss the utility of that evidence in convincing third parties of claims. At the same time, it also differs in focus from a definition of Yumerefendi and Chase (2005) that it draws upon and that we review in Def. 2.5.

PeerReview, which we discuss in Sec. 3.2.2, has had significant influence on other accountability mechanisms. It is worth noting that Haeberlen *et al.* see the benefits of accountability as including deterrence, through the threat of punishment; fault detection, enabling “timely recovery of faulty nodes[;]” and providing evidence both to assign blame and to allow innocent principals to demonstrate their innocence.

Haeberlen (2010) argues for the importance of accountability in cloud settings and identifies factors that make it difficult to apply accountability mechanisms, such as PeerReview, to the cloud. His definition of “accountability” includes both association of actions with principals and evidence that can be used to convince a third party of a

claim; because of the latter feature, we include it here. In particular, (Haeberlen, 2010, p. 53) says that

a distributed system is accountable if a) faults can be reliably detected, and b) each fault can be undeniably linked to at least one faulty node. More specifically, we consider systems that have the following features:

Identities: Each action (such as the transmission of a message) is undeniably linked to the node that performed it;

Secure record: The system maintains a record of past actions such that nodes cannot secretly omit, falsify, or tamper with its entries;

Auditing: The record can be inspected for signs of faults; and

Evidence: When an auditor detects a fault, it can obtain evidence of the fault that can be verified independently by a third party.

The comprehensive, tamper-evident record and the ability to audit it narrows the class of systems that would be considered accountable.

Haeberlen also postulates an **Audit** primitive that a customer could invoke to determine whether a cloud provider has fulfilled an agreement. **Audit** should either return an indication that the service did satisfy the agreement or evidence that it did not. Of particular interest here are his three goals for such a primitive (Haeberlen, 2010, p. 55):

Completeness: If the agreement is violated, **Audit** will report this eventually, and it will produce evidence of the violation;

Accuracy: If the agreement is not violated, **Audit** will not report a violation; and

Verifiability: Any evidence of an alleged violation can be checked independently by a third party, even if the third party trusts neither the customer nor the provider.

He argues that the accuracy goal must be met in order to interest cloud providers (who would not want to be blamed falsely) but that the others could be strengthened or weakened. We view all three as conceptually important properties. An accountable system might or might not have them, but they should be considered explicitly in designing and analyzing accountable systems.

Of note, Haeberlen identifies particular aspects of privacy that he considers relevant to cloud accountability. He suggests that the cloud provider could keep a separate log for each of its customers in order to protect customer privacy. He also suggests that an **Audit** primitive might provide different information depending on who invokes it: A customer would learn less information, protecting the provider's internal workings, while the provider would learn more information to help troubleshoot its systems.

Although Haeberlen does not develop a particular system in this work, he does identify tools that might be used to provide accountability for cloud systems. These include tamper-evident logs, replay using virtual machines, and trustworthy timestamping.

2.2.3 Identification and association

The identification of the principal that performed a particular action and the general association of a principal with all of its actions have been prominent in computer-science definitions of “accountability” over many years. Definitions in this class vary with respect to the nature of “identification”: Is it a nominal identity, *e.g.*, a username, that is associated with action(s) or a real-world individual who might have taken on multiple identities (“Sybils”)?

In an early survey, the National Research Council (1991, p. 78) associates accountability policies with “knowing who has had access to information or resources.” Lampson (2004) uses this association as the definition of “accountability” in an early treatment of security in terms of “locks” and “punishment.” In describing after-the-fact defensive strategies, he also highlights recovery (or “undo[ing] the damage”), *i.e.*, restoring a system from backups. Although recovery is separate from accountability, it is worth noting that preventive and deterrent measures are not the only ones available.

Lampson also highlights auditing as a key component of practical security (along with authentication and authorization). In the context of a “guard” deciding whether or not a principal can access an object in a system, he says that

[i]f the guard records the proof in a reasonably tamper-resistant log, an auditor can review it later to establish accountability or to determine whether the system granted some unintended access and why. Since detection and punishment are the primary instruments of practical security, this is extremely important (Lampson, 2004, p. 45).

Note that Lampson’s view is consistent with our notions of detection, evidence, judgment, and punishment. We agree with him that auditing can play an important role in accountability but do not view it as inherently essential.

Early work of Ko, Frincke, Goan, Heberlein, Levitt, Mukherjee, and Wee (Ko *et al.*, 1993) looked at the problem of attackers’ moving through a network, performing actions that might seem innocuous on a per-host basis but that would be identifiable as an attack when viewed at network scale. They identified “accountability” as a goal and took it to mean the ability “to associate all activities of multiple instances of the same individual to the same [...] identifier[.]”

Yumerefendi and Chase (2005) present the following view of accountability that has, directly and indirectly, influenced much subsequent work.

Definition 2.5. [A]n accountable system associates states and actions with identities, and provides primitives for actors to validate the states and actions of their peers, such that cheating or misbehavior become detectable, provable, and undeniable by the perpetrator.

They contrast this view with some earlier uses of “accountable,” noting that “the goal is to assign responsibility for states and actions, and not just to track and audit access control decisions.” Yumerefendi and Chase also present a high-level vision for accountable systems that fits with the approach subsequently implemented by Haeberlen *et al.* (2007) in PeerReview.

The Accountable Internet Protocol (AIP) of Andersen, Balakrishnan, Feamster, Koponen, Moon, and Shenker (Andersen *et al.*, 2008) takes accountability to be the “associatiat[ion of] an action with the responsible entity[.]” AIP’s creators identify the detection and prevention of source-address spoofing and the stopping of unwanted traffic as applications of such an association. Taking this as a general definition requires explication of “responsible,” which may be more complex in other settings.

The BACKREF system of Backes, Clark, Kate, Simeonovski, and Druschel (Backes *et al.*, 2014) is designed to provide anonymous-communication networks with *repudiation* (allowing exit nodes to demonstrate that they did not originate a communication stream) and *traceability* (allowing the identification of the source of a stream when relays cooperate). This work does not provide a single definition of “accountability,” but identification implicitly plays a major role in the way the concept is treated. Backes *et al.* note that other work on adding “accountability” to anonymous-communication networks has included “allowing misbehaving users to be selectively traced, exit nodes to deny originating traffic [they forward], misbehaving users to be banned, and misbehaving participants to be discovered.”

Ishai, Ostrovsky, and Zikas (Ishai *et al.*, 2014) define a notion of *identifiable abort* for secure, multiparty computation. For an arbitrary functionality $\mathcal{F}$, they define a corresponding functionality $[\mathcal{F}]_{\perp}^{\text{ID}}$ that

behaves as $\mathcal{F}$ with the following modification: upon receiving from the simulator a special command (abort, p_i) , where p_i [...] is a corrupted party [...], $[\mathcal{F}]_{\perp}^{\text{ID}}$ sets the output of all (honest) parties to (abort, p_i) . (Ishai *et al.*, 2014, p. 373)

In particular, when the protocol fails, all of the honest parties *agree on the identity* of a party that is indeed corrupt. Ishai *et al.* present results on multiparty-computation protocols in both information-theoretic and computational settings.²

²As noted by, *e.g.*, Ishai *et al.* (2014), while the classic work of Goldreich, Micali, and Wigderson (Goldreich *et al.*, 1987) predates the notion of “identifiable abort,” the GMW protocol does indeed provide identifiable abort.

In providing an efficient approach to identifiable abort, Baum, Orsini, and Scholl (Baum *et al.*, 2016) consider how to ensure that the identity of the corrupt party that is identified to the honest participants can also be proved to outside parties. The ability to do this is not inherently part of identifiable abort. They also provide a concise review of the different lines of work on secure, multiparty computation with identifiable abort.

The works reviewed in this subsection illustrate what is in our view a fundamental gap in the computer-science literature on accountability. They present a broad, technically interesting variety of mechanisms for identifying entities *that have violated system policy and therefore should be held accountable for those violations*, but they do not give technical definitions of “held accountable.” We view subsequent punishment as essential to the notion of holding violators accountable.

2.2.4 Answerability

The notion of accountability as *answerability*—*i.e.*, a requirement to give an answer for, or account of, actions and decisions—appears primarily in disciplines other than computer science. A comprehensive treatment of all such definitions is outside the scope of our survey, but we highlight some perspectives that take this viewpoint. While we argue for a view that focuses on punishment, we note that many of the answerability-focused perspectives implicitly assume a punishment mechanism, the existence of which implies that answerability produces deterrence.

Closely related to answerability is the notion of transparency. Instead of after-the-fact answering for actions and decisions, transparency opens up the decision-making process itself to inspection before, during, or after decisions are made. This variation on answerability has also been seen mainly in work outside of computer science.

First, some common English-language definitions point to this view of accountability. The Oxford English Dictionary defines *accountability* and *accountable* as follows:

Definition 2.6 (Accountability). The quality of being accountable; liability to account for and answer for one’s conduct, performance of duties,

etc. (in modern use often with regard to parliamentary, corporate, or financial liability to the public, shareholders, etc.); responsibility.³

Definition 2.7 (Accountable). Chiefly of persons (in later use also organizations, etc.): liable to be called to account or to answer for responsibilities and conduct; required or expected to justify one's actions, decisions, etc.; answerable, responsible.⁴

In the context of summarizing and arguing against the expansion of “accountability” in public administration (see Sec. 3.6.2), Mulgan (2000) argues for the following.

Definition 2.8 (Mulgan's “core sense of accountability”). One sense of ‘accountability’, on which all are agreed, is that associated with the process of being called ‘to account’ to some authority for one's actions[.]

He argues that key components of the core sense are the fact that it is “external,” that “it involves social interaction and exchange,” and that “it implies rights of authority[.]” He also notes that sanctions may be implicit in authority but that the possibility of sanctions is less clearly required by “giving an account” than it is by “calling to account.”

In related work on measuring accountability attributes in the cloud, Nuñez, Fernandez-Gago, Pearson, and Felici (Nuñez *et al.*, 2013) relate accountability to

policy enforcement, risk management, incident management and remediation. Accountability is a high-level concept that entails all these practices. In general, accountability deals with being able to demonstrate that the accounts provided by an organisation (to regulators, auditors, data subjects or other service providers) are adequate and appropriate for the context, and implementing mechanisms for responding to the situation (including sanctions and remediation) if this is not the case.

They suggest “non-functional properties” of systems—“not directly related to functionality, but to a quality or behavioral attribute of a

³<http://www.oed.com/view/Entry/1197>, accessed August 11, 2017.

⁴<http://www.oed.com/view/Entry/1198>, accessed August 11, 2017.

system”—that are related to accountability. These include “transparency, verifiability, observability, liability, responsibility, attributability, and remediation.”

Charlesworth and Pearson (2013) and Pearson and Wainwright (2013) both discuss accountability broadly, placing it in a social and legal context. They take an interdisciplinary approach to accountability, bringing together both legal and technical perspectives, with a focus on cloud-computing services. Pearson and Wainwright address overall security, reliability, and privacy for cloud computing services, while Charlesworth and Pearson focus specifically on privacy of personally identifiable information (PII). From a definitional perspective, Charlesworth and Pearson incorporate legal responsibility, which we view as a form of answerability, noting:

[A]ccountability in this context [of Australian, U.S., and Canadian law] means placing a legal responsibility upon an organization that uses PII to ensure that contracted partners to whom it supplies the PII are compliant, wherever in the world they may be. Our accountability model reflects the basic premise of this approach, but extends it by suggesting ways in which organizations might take the “accountability” approach further in order to develop a reflexive, continually evolving, privacy process.

Pearson and Wainwright take a “co-design” approach, envisioning an accountability framework or ecosystem captured by a matrix. They place users, providers, and regulators on one axis and preventive, detective, and corrective mechanisms on the other. They also review various solutions developed in industry to address specific properties in the accountability matrix, namely risk assessment, data obfuscation, consent management, sticky policies, and monitoring for information use. Charlesworth and Pearson highlight privacy-related issues in cloud computing, including outsourcing, offshoring, virtualization, and autonomic technology. They discuss the extent to which regulation, such as the EU’s privacy directive (EU Directive 95/46/EC) and the UK Data Protection Act of 1998, can and cannot handle such issues, and they suggest that an accountability-based approach that incorporates

co-design of legal and technical mechanisms and procedures would be more suitable.

In discussing different conceptions of accountability in information privacy and data protection, Raab (2012) highlights its connection to “stewardship, by which is meant that one party entrusts another with resources and/or responsibilities.” He connects accountability in data protection to accountability in political sense. In particular, Raab notes (quoting Mulgan (2000)) that “[a]ccountability, in the sense of acquitting the responsibilities of stewardship through the giving of an account, is therefore somewhat similar to another meaning of accountability as ‘dialogue’, in which ‘officials ... answer, explain and justify, while those holding them to account engage in questioning, assessing and criticizing’”

We now note some additional perspectives from outside of computer science that fall under “answerability” and that have informed work in computer science. The differences among them highlight views that might be argued for or against when designing systems and mechanisms in computer science. However, we do not attempt to judge the relative merits of competing perspectives for use in other disciplines.

Writing about European policy, Bovens (2007) provides a self-described “narrow” sense of the concept of accountability in order to allow more concrete analysis than is allowed by broader uses of the term. For example, he notes:

Particularly in American scholarly and political discourse, ‘accountability’ often is used interchangeably with ‘good governance’ or virtuous behaviour. (Bovens, 2007, p. 449)

As a short summary of his view, Bovens describes accountability as “the obligation to explain and justify conduct.” Enriching this description, he makes the following definition:

Definition 2.9 (Bovens’s “narrow” sense of accountability). Accountability is a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgement, and the actor may face consequences. (Bovens, 2007, p. 450)

Notably, in Bovens’s view, the judgment that is passed is *on the conduct of the actor*.

Lindberg (2013) surveys conceptualizations of accountability across disciplines outside of computer science and categorizes types of accountability. He identifies in accounting, and political science, the “underlying principle of delegating some authority, evaluating performance, and applying sanctions” as a key aspect of accountability. In particular, Lindberg views accountability as distinct from responsiveness:

At a very fundamental level then, accountability is closely associated with authority though not necessarily political authority. A puppet acting as an extension of someone else’s will is not a legitimate object of accountability. That is why accountability is different from “responsiveness.” Only actors with some discretion to make authoritative decisions can be the objects of accountability relationships[.] (Lindberg, 2013, p. 208)

Of note for our survey, Lindberg takes a different view from Bovens and focuses on sanctioning agents for the information and justifications they provide, not the underlying actions.

[A]n important distinction should be made between the right to sanction [an agent] for failure to provide the information requested and justifications for decisions and actions taken, and the right to sanction agents or institutions [...] for the content or effects of such decisions and actions. At its core, accountability only necessitates the right to sanction [an agent] for failure to provide information and justify decisions. (Lindberg, 2013, p. 210)

As noted above, we do not take a position on which of these better comports with “accountability” as used in other disciplines. However, it is worth distinguishing between punishing or rewarding an agent for its underlying actions and for the information it provides about those actions.

Returning to computer science, Klonick (2018) provides an analysis of how and why online platforms (specifically Facebook, Twitter, and

YouTube) moderate content; she takes this analysis as a point of departure for a broader discussion of *online governance* and free speech (or the lack thereof) on private platforms, which she refers to as our “New Governors.” Although Klonick does not define “accountability,” she clearly uses the term to mean answerability. She describes the platforms as “private, self-regulating entities” that are inclined to support the free-speech expectations of their users because it is good for their bottom lines to do so and because free speech is a cherished value in the tech world; rules and procedures for moderating content are heavily influenced by US free-speech norms. Klonick identifies the main threats to democratic culture posed by private online governance as potential loss of fair opportunity to participate (given that platform operators can block users whose posts violate their “community standards”) and platforms’ “lack of direct accountability to [their] users.” Attempts to counter these threats, she argues, should start with technical changes to the architecture and content-moderation systems put in place by platforms. If a purely technical approach fails, she advocates regulation that emphasizes balance between the democratizing force of the internet (which requires fair opportunity to participate) and the innovative force of the New Governors.

Kroll (2015)—in his thesis focusing on “a *decision authority* making decisions using a set *policy*[.]”—defines “accountable” as follows. He notes that it requires not just explainability but also consistency with norms.

Definition 2.10 (Def. 2 of (Kroll, 2015)). We say that a process or entity is *accountable* for a decision if that process is consistent with the *legal, political, and social norms* that define its context, and this fact is publicly evident, either because such consistency can be readily determined using public information or because an entity empowered and trusted to define such consistency (*e.g.*, a court, a legislature, or other designated authority) can be shown to have the necessary information to determine this consistency.

He identifies four properties as necessary for a decision process to be accountable (Kroll, 2015, p. 57):

1. the authority must be committed to its policy in advance;
2. the result asserted by the authority must be the correct output of the authority’s committed policy when applied to an individual decision subject;
3. any randomness used in the decision policy must be generated fairly; and
4. if necessary, the authority can reveal its policy to an oversight body for examination later, without revealing decision subjects’ private data, and that body as well as each decision subject can verify that the revealed policy is the one that was used to make the decisions in question.

In Kroll’s view, accountability is “inherently political,” and it may thus be impossible to determine sufficient properties to ensure that a decision process is accountable. This perspective suggests that transparency may be more important than some other answerability-focused perspectives, but subsequent work by Kroll, Huey, Barocas, Felten, Reidenberg, Robinson, and Yu (Kroll *et al.*, 2017) identifies the limitations of an approach that leans too heavily on transparency; see Sec. 2.3.2.

2.2.5 Blame

Accountability approaches that focus on blame are in some ways similar to those that focus on associating principals with the actions they perform. Indeed, such associations can be useful in assigning blame or responsibility. However, we distinguish these two types of approaches. We regard blame or responsibility (terms we use interchangeably) as something that requires additional analysis beyond the mapping of actions to principals. The concept of blame captures the possibility that, although action a done by principal P violates a system policy, actions previously taken (or neglected) by principal Q mean that *no* action done by P could possibly satisfy system policy—nor could the policy be satisfied by the complete *lack* of action by P . Alternatively, Q could

have coerced P into doing a . Blaming Q for a system violation in such cases requires more than simply knowing that it was P who did a .

The 1991 National Research Council (NRC) report provides the following definitions that are relevant for historical context.

Definition 2.11.

Accountability The concept that individual subjects can be held responsible for actions that occur within a system.

Auditing The process of making and keeping the records necessary to support accountability.

(National Research Council, 1991, pp. 286–287)

By contrast, the report notes elsewhere that “individual accountability” has classically been part of management controls:

Individual accountability answers the question: Who is responsible for this statement or action? Its purpose is to keep track of what has happened, of who has had access to information and resources and what actions have been taken. In any real system there are many reasons why actual operation may not always reflect the original intentions of the owners: people make mistakes, the system has errors, the system is vulnerable to certain attacks, the broad policy was not translated correctly into detailed specifications, the owners changed their minds, and so on. When things go wrong, it is necessary to know what has happened, and who is the cause. This information is the basis for assessing damage, recovering lost information, evaluating vulnerabilities, and initiating compensating actions, such as legal prosecution, outside the computer system. (National Research Council, 1991, p. 57)

Finally, the NRC connects accountability policies to “knowing who has had access to information or resources” (National Research Council, 1991, p. 78); this is informed by the perspective of Lampson discussed in Sec. 2.2.3.

Irwin, Yu, and Winsborough (Irwin *et al.*, 2006) present a framework for modeling and analyzing *positive* obligations, *i.e.*, requirements that

particular actions be taken during a particular time window. Obligations might be imposed as a condition of allowing certain actions to be performed, and they complement preventive access-control measures. Noting that system participants typically cannot be compelled to satisfy their obligations, Irwin *et al.* seek systems in which it is possible to fulfill all obligations. In such systems, unfulfilled obligations represent violations by participants (instead of, *e.g.*, tasks made impossible by scheduling or resource restrictions), and the goal is to identify the participant that did not fulfill its obligations. Informally, Irwin *et al.* say that “[a] system is accountable if and only if all the obligations will be fulfilled supposing all the subjects are diligent. In other words, a secure state implies that any obligation violation can only be due to the lack of diligence of subjects.” They then study the feasibility of checking whether systems are accountable in this sense; we discuss this and subsequent work in Sec. 4.3.1.

Irwin *et al.* also say:

Rather than requiring that it be impossible for obligations to be violated, instead we assume that it is possible that obligations go unfulfilled, but when they do, we would like to clearly identify whose fault it is. Obviously, an obligation can go unfulfilled because a subject simply fails to take the required action before the deadline, even if he has sufficient privileges and resources. It is desirable that this is the only reason that an obligation will go unfulfilled.

Intuitively, if it is the case that all users have sufficient privileges and resource to carry out their obligations so long as every other user carries out his or her obligation, then a system is said to be in an *accountable state*, because we can know that whoever first fails to carry out an obligation is responsible for the violation and anything which results from it.

Datta *et al.* (2015) distinguish accountability from punishment (“determining which agents should be held accountable and appropriately punished is important to deter agents from committing future violations”) and identify assigning blame to an agent in a system with holding

agents accountable. This is expanded upon by Sharma (2015) in her dissertation, where she also provides an overview of other accountability-related work. We note that the view of Datta *et al.* does not encompass our notion of automatic punishment.

In connecting this perspective to technical, computer-science definitions, Datta *et al.* (2015, Sec. VI) argue that

accountability is not a trace property since evidence from the log alone does not provide a justifiable basis to determine accountable parties. Actual causation is not a trace property; inferring actions which are actual causes of a violating trace requires analyzing counterfactual traces[. . .] Accountability depends on actual causation and is, therefore, also not a trace property.

Kacianka, Beckers, Kelbert, and Kumari (Kacianka *et al.*, 2017) survey the literature on implementations of accountability mechanisms and techniques. One of their goals is to identify a common definition of “accountability.” While they find no such common definition in the literature they survey, they offer the following four elements of a “work-in-progress definition of accountability” that focuses on blaming entities for actions:

1. *Accountability* is a property of a system or a collection of systems and is ensured by an Accountability Mechanism.
2. An *Accountability Mechanism* is part of an *Accountable System* and reasons over a tamper-proof log to link effects of that system to entities.
3. An entity is (partially) *accountable* for a given effect if an *Accountability Mechanism* can prove a causal link between the entity’s action and the given effect.
4. The set of entities *accountable* for a given effect is the set of all entities for which an *Accountability Mechanism* can prove a causal link between the entities’ actions and the given effect.

2.2.6 Punishment and deterrence

In Sec. 2.2.3, we pointed out that some key definitions of “accountability” that focus on association between principals and actions tacitly assume the existence of a punishment mechanisms in the background. By contrast, the perspectives we review in this subsection make punishment itself an explicit focus of the definition.

Lampson (2005a) takes an early step toward the focus on punishment and deterrence:

Definition 2.12. Accountability is the ability to hold an entity, such as a person or organization, responsible for its actions.

He notes that “[a]ccountability is about punishment, not locks” and, in the context of an anti-spam policy, that “[s]enders must be able to make themselves accountable: This means pledging something of value.” In general, he identifies deterrence as the goal and punishment as a tool to achieve it, thereby conceptualizing online enforcement of system policies as analogous to real-world law enforcement. He later draws this connection explicitly in (Lampson, 2009, p. 27), where he says that “real-world security depends mainly on deterrence, and hence on the possibility of punishment.”

One can view this formulation as an evolution of the NRC definition used earlier in (Lampson, 2004). The focus shifts from “knowledge about access” to deterrence via punishment, but punishment may be *enabled through* knowledge about access. Indeed, Lampson (2005a) envisions using “a consistent identifier based upon a name, a pseudonym, or a set of attributes” to punish a principal by tarnishing its reputation. This illustrates the often close relationship between identification of principals with actions and punishment or deterrence.

We also recall the definition of Grant and Keohane (2005) that we noted in Sec. 1.2.1:

Definition 2.13. Accountability, as we use the term, 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.