Research Statement

Joshua A. Kroll (kroll@berkeley.edu)

Software systems are an increasingly critical part of the world, deciding and recommending everything from credit outcomes to job candidates, friendships and dates, disease diagnoses, ratings of individuals' public safety risks in law enforcement and national security contexts, and more. The shift towards automation exacerbates gaps between traditional governance and oversight processes and the realities of software-driven decision-making, leading to calls for more fairness, transparency, and accountability in computer systems to ensure that they respect and conform to social, political, and legal norms. This governance gap can exist even for the software engineers, data scientists, and system operators who design, build, deploy, and manage the software-mediated systems that mediate our modern lives, especially where data-driven techniques are used during development. Using data-driven approaches to building prediction, recommendation, and decision-support software makes these systems more opaque and harder to govern, undermining human intuition while giving decisions an unearned patina of data-driven legitimacy. Still, there is a broad societal interest in ensuring the good governance of these technologies, ensuring both that technical artifacts are built according to the right goals, capturing the values of different stakeholder groups, and that they actually measure up to those requirements. **My research addresses this by developing *accountable algorithms* [4, 10], composing technical capabilities with human processes to create sociotechnical systems that comport with governance norms.** For example, accountable algorithms should keep a record of their behavior in a way that supports the function and operation of governance and oversight processes. My work bridges the gap between the detailed specification requirements of software systems and the interpretive space required by legal and policy development and analysis, and has helped turn discussions of computer system governance away from the idea of accountability through transparency and towards a technology-informed call to design systems to uphold key requirements.

It has famously been said that in discussions of technology and information policy that technologists defer too much to what can be done with social, political, or legal solutions, and that non-technologists defer too much to the promise of technology. It is common for technologists to believe that the human context in which their software is situated can be reduced to a few rules, equations, or requirements specified clearly before a technology is built or deployed. Engineers often outsource questions about the appropriate use of the artifacts they design and create to others. Similarly, it is common for non-technologists to argue that technology can do anything or solve any problem without regard for the feasibility of the desired solution. Policymakers, legal practitioners, and scholars outside technical fields often consider the design and function of technical artifacts beyond their purview or imagine that a perfect and complete technology solution exists for a messy social problem, even if they can't describe how to build it. Each group assumes the other can solve the problems it doesn't understand or doesn't want to tackle, and attempts to push ideas across this divide often lead to acrimony. It is easy and tempting to mistake feasibility of achieving some result in software for a full solution to a messy sociotechnical problem.

As a computer scientist with an interdisciplinary approach, my research builds bridges across this divide. In particular, I study how the tools of computer science can be used to address the full scope of sociotechnical problems with technical interventions. While tools from computer science can solve many kinds of problems, in many cases pure technology—even new, purpose-created technology—does not provide a complete solution. For example, technical tools for machine learning fairness adjust the distribution of outcomes without considering process-oriented understandings of fairness, which are common in law. Conversely, a software-driven decision rule might be well calibrated to its task (e.g., assessing risk in an accurate way), but nonetheless be grossly unfair as deployed (this is famously the case for recidivism risk prediction instruments, which by accurately reflecting the best proxy for recidivism, namely re-arrest, over-rate the risk of people of color who are disproportionately targeted for policing). To make the most of technical interventions, I bring to bear a broad toolbox of analysis methods, including law [10, 3], public policy [1, 11], economics [8], education [2], organizational management [6], engineering culture and processes [7], public activism [5], or whatever methods best address the problem at hand. Sociotechnical problems are nearly always amenable to a range of solutions that combine technical, legal, and social mechanisms in different ways,[1] and it is critical to establish the best way to deploy tools from computer science to greatest effect.[2]

[1] For example, risk to election system integrity may best be solved by reallocating auditing resources [9].

[2] As an example, while cryptographic guarantees are excellent for detecting violations of the rules of cryptocurrencies such as Bitcoin, they are a poor substitute for governance either of what those rules should be or how a cryptocurrency's community should respond to detected rule violations [8].

Research Statement

Joshua A. Kroll (kroll@berkeley.edu)

Accountable Algorithms

The accountability and governance mechanisms that regulate increasingly automated consequential processes have not kept pace with technology. The tools currently available for policymakers, legislators, and courts were designed to oversee human decision-makers and often fail when applied to computers instead: for example, it is not possible to judge the intent of a piece of software, but unclear that intent can reasonably be ascribed to a system's designers, operators, or programmers. Those who must understand such systems from the outside, including entities charged with oversight such as regulators but also purchasers and even the general public, often do not understand where or why there are gaps between what they can see and what computer systems are actually doing. These problems only become more acute as decision making algorithms move from comprehensible sets of rules coded by human programmers to inscrutable machine learning models with thousands or millions of parameters.

Instead, **we must design technologies to facilitate *accountability* of their controllers**; it must be possible to hold the designers, implementers, and operators of computer systems responsible for the behaviors of those systems just as if those behaviors were taken by people in a human-mediated process. Although my work describes how to control technical artifacts, it is people and non-computerized structures such as organizations and governments that are held responsible. Accountability begins as *accounting*, or record keeping which describes faithfully the action of a decision-making process. Even high-fidelity records alone are insufficient to hold people responsible for the outcomes of decision processes they control; competent oversight is necessary to connect those outcomes to the norms and values they are meant to reflect, to mete out punishments or corrections of misapplied processes, and to make process controllers answerable for their decisions. Although the best computer-mediated process may take completely different form to acceptable human-mediated process, automation must not shield humans from responsibility for the outcomes of any important decision process.

There are two paradigmatic models for accountability. In the first, stakeholders agree on a set of rules *ex ante* and it is necessary only to confirm that implementations uphold those rules correctly. It is in general possible to achieve this with technology. The second model involves setting a high-level policy *ex ante* to preserve flexibility in implementation. Checks on implementations—certifications, testing regimes, assessments, and formal guarantees—serve to ensure that the methods used further the overall aims of the policy. Such checks could be part of the development process or conducted by an external oversight authority which can separate acceptable behaviors from unacceptable ones *ex post*. This requires connecting information about implementations, behaviors of systems, and the goals, norms, and values abstractly embedded in high-level policies. Space for compromise often arises from agreement on a process for setting or enforcing the rules (for example, legislatures defer both to administrative agencies to write specific rules and to courts to determine when rules or standards are violated). The integrity of this process can often be assured by technology, even in the absence of a complete policy to enforce on the original problem. Technology (and even evaluation of the capabilities or correctness of technology) generally requires a complete *ex ante* specification of what it is supposed to do. Yet the interpretive space in laws and policy documents is necessary to create the consensus that brings those rules their normative force. Separate from validating fidelity either to rules or high-level goals is the question of how to establish good and useful sets of specific rules or high-level policies that constrain the behavior of computer systems, and who should be involved in and charged with developing them. Governance of technology requires both, at a minimum.

My work challenges the once-dominant position in the legal literature that disclosure of the internals of these systems will solve this problem by itself. Certainly, disclosures about the operation of computer systems must be part of the solution, but this information on its own is neither necessary nor sufficient for accountability, and it is often undesirable to reveal due to countervailing values such as the privacy of individuals or trade secrecy. For example, a software lottery is a fully transparent system in which transparency does not facilitate accountability, since a lottery could be re-run in secret with only the (valid) run that produced a desired winner revealed. It is important to understand the role of various disclosures—certifications, specifications, requirements documents, testing plans, impact assessments, or formal guarantees—in oversight, to establish their value in facilitating accountability, and to determine what must be disclosed to whom to provide the basis for trust in automated decision making systems. There are many approaches to establishing (and verifying for outsiders) that a computer system is fit for purpose. For example, systems can bind disclosures to specific assurances about their behavior in many ways: audit logs that allow decisions

to be fully justified [4]; traditional software verification [13] or cryptographic invariants [14] as have long been studied by computer scientists; or the use of machine learning techniques with desirable formal fairness guarantees [10]. Critically, such assurances must connect the technical guarantees with applicable norms and values, disclosing relevant facts about a system rather than simply revealing its internals.

Specifically, I develop new cryptographic protocols to facilitate the operation of oversight processes. These cryptographically enhanced records can then certify to oversight entities, decision subjects, and the public at large that an automated decision process satisfies a criterion I call procedural regularity. Procedural regularity implies that all subjects of a decision process are treated under the same policy, that the policy is determined in advance of seeing the subject's data (so that it cannot be engineered to prefer or disfavor particular participants), that a full accounting of how any particular decision was made exists and is recorded for later review, and that any random choices required by a decision policy are made outside the control of the decision-maker. This protocol allows verification of procedural regularity *without requiring the disclosure* of sensitive data or of the source code of the decision policy itself. Instead, these secrets can be demanded later by an oversight entity, which can determine if practices were acceptable or not using after-the-fact review, and members of the public can verify the consistency of their view of what happened with the oversight entity's. This allows political trust in the oversight entity to be leveraged into trust in the automated decision system itself, creating a kind of social contract for automated decision making without the need for full disclosure. Using this approach more closely matches the process of oversight via courts or legislative committees and allows accountability even in processes where the exact line between acceptable and unacceptable behavior cannot be specified in advance. The work applies traditional methods (commit-and-prove protocols) realized with cutting edge methods (recent succinct non-interactive zero-knowledge proof schemes, zk-SNARKs) while developing a new protocol specific to this application and a new theory of procedural regularity.

For example, it is difficult ahead of time to develop a full specification for a decision process that does not improperly discriminate with respect to a protected status attribute such as race or gender. One could attempt to specify heuristics for fairness, such as demographic parity, the requirement that an equal fraction of the group with each value of the protected attribute receive each possible outcome from the decision process. However, any such heuristic will be fraught (for example, demographic parity says nothing about which members of each group should receive which outcome, preserving the ability to treat groups differently), and anti-discrimination law often functions on disparities that only appear after decisions are made.

To move this work beyond a purely technical advance, I led an interdisciplinary team of lawyers, social scientists, and computer scientists that developed a coherent theory of how accountable algorithms would support better governance of technology, stronger oversight of machine learning techniques, and greater fidelity to social, political, and legal norms. This collaborative effort produced two articles published in top law journals [10, 3]. The first of these articles was recognized in the Future of Privacy Forum's "Privacy Papers for Policymakers" series as one of the top pieces of policy scholarship in 2016.

Following on this work, I continued investigating how interventions during the design and implementation of a software system could improve its fidelity to normative goals, arguing that inscrutability in computer systems as realized in the world is purely a consequence of power relationships between actors [7]. Opacity is never the result of technological incapacity either to understand how a system will perform or to design it to perform to certain standards; the fine detail of a system's operation is largely irrelevant to evaluations of its ability to further normative goals or comply with high-level policies that reflect appropriate values. Rather, the idea that a system must be inscrutable shields technologists from responsibility, distracts from debates about what policies the system should effectuate, and limits system governance.

Many important social and political values are conceptualized differently by disparate communities, and may be essentially contested. This makes discussing which values should be required by or reflected in computer systems difficult across communities with different disciplinary approaches, especially as these discussions often conflate terms such as fairness, accountability, transparency, and others. Computer scientists largely see software systems as existing abstractly and being embodied by machines, and so look to meanings of these terms (and interventions around these terms) that apply within machines (for example, by defining the fairness of a machine learning system in terms of distributions of output or in terms of mathematically well-defined allocations of resources). Other disciplines are more apt to take a sociotechnical view, including people and context in the evaluation of technology, though perhaps still disagreeing on how to achieve this (for example, fairness in social science often speaks to opportunities available at different positions within societies, while in law it often means something about the integrity of a process). This terminological con-

fusion underpins debates about the proper interventions, since it affects both what problem people believe they are trying to solve and their biases towards or away from particular solutions. To address this, together with an interdisciplinary team, I explore the ways in which key terms in the study of values-in-technology are conceptualized by different fields [12]. This work has been presented as a tutorial at the FAT* conference, workshopped at the Privacy Law Scholars Conference, and forms the groundwork for a tutorial to be presented at the Neural Information Processing Symposium (NIPS), the largest annual machine learning research gathering. It is also currently being circulated as a working paper.

Fairness, Accountability, and Transparency

In order to create the opportunity for true interdisciplinary scholarship on the problem of governing software systems, I worked with others to make the Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML) a regularly held forum, co-located with major machine learning research conferences, and a respected venue for research on the normative impacts and governance of machine learning systems. I served on the executive committee for FAT/ML from 2015-2017, and was the program chair in 2017. Also in 2017, I helped create the freestanding computer science Conference on Fairness, Accountability, and Transparency (FAT*). These events have become competitive, top-tier research venues eagerly anticipated by a new community of researchers across computer science as well as other disciplines including sociology, law, public policy, and philosophy. The continued success and impact of these events demonstrates that interdisciplinary work on the social and political values and governance of software systems has home within computer science. My research is shaping work in this field and I am using my leadership position within it to grow a global community of researchers across many specialties and disciplines.

Research Agenda

Many entities have proposed tools, practices, or sets of principles that attempt to capture what it would mean to build technology that is responsive to human values.[3] But these measures are only worthwhile to the extent that they can be operationalized in technologies and organizations in a way that recognizably improves the resultant systems in the eyes of a broad set of stakeholder communities. To the extent that systems should be made more fair, more private, more secure, or to better facilitate accountability, we must find a way to test whether proffered mechanisms for improving technology along these values can be implemented to improve outcomes in sociotechnical systems.

At present, such values-oriented design is purely the province of heroes, experts who by their individual efforts and the largess of their skill and intellect seek to improve the extent to which technologies conform to desired values. But such important work must be achievable through repeatable and transferable processes if we are to demand through public policy that technology better reflect desired values. Such processes must robustly convince those outside technology development that the generated artifacts embody the proper values; my previous work provides a foundation [4, 10].

To fill this gap, I propose to develop and validate reliable system development and lifecycle processes that aim to capture difficult-to-specify values such as privacy, fairness, and trustworthiness in ways which are evident to relevant consumers. It is critical that teams and organizations be able to take on the problem of constructing computer systems that align with social and political values as well as regulations. This program requires analytical study to ground it in existing literature and theory (which I have already begun as described above); empirical work on the processes in use to determine which ones actually improve system outcomes; and the development of new methodologies and tools that apply the best techniques from computer science in cryptography, software verification, and machine learning to create systems that reliably reflect the desired values of stakeholders, regulations, context, etc.

My current projects reflect this program of exploring how to capture values using technology and to measure how effectively existing technologies capture values when they promise to. I have received a grant from the Hewlett-funded Center for Long-Term Cybersecurity to assess the question of malicious subversion

[3] For example, Google has released a set of guiding principles (`https://www.blog.google/technology/ai/ai-principles/` and a more concrete set of "Responsible AI Practices" (`https://ai.google/education/responsible-ai-practices`); a group of academics also released a framework for evaluating the social impact of machine learning (`http://www.fatml.org/resources/principles-for-accountable-algorithms`).

of machine learning model creation. While the question of detecting and correcting for unintended bias in machine learning models has received much research attention, little to no attention has been given to the questions of whether and how an adversarial modeler can build such problems into a model intentionally. Under this grant, I have undertaken a project to measure how sensitive word embedding biases are to choices made during training in order to demonstrate how issues of representational bias in natural language processing systems respond to choices about training data and training parameters. Similarly, I have undertaken theoretical machine learning work that seeks to determine how likely it is that training processes will yield values-reflective models. I am also part of a team that received a grant from the NSA's Science of Security office to investigate how to build data governance models that adequately capture privacy concerns given the twin risks of reidentification and unpredictable inferences from data. Through this, I have begun a project examining how work on software engineering process maturity and software development methodology can be applied to data science and machine learning with a focus on how to capture values in these technologies.

References

[1] Solon Barocas, Edward W. Felten, Joanna Huey, Joshua A. Kroll, and Arvind Narayanan. Big data and consumer privacy in the internet economy. Comment to the NTIA Big Data Request for Comments, 79 Fed. Reg. 32714, August 2014.

[2] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. *Proceedings of IEEE Security and Privacy*, 2015.

[3] Deven Desai and Joshua A. Kroll. Trust but verify: A guide to algorithms and the law. *Harvard J. of Law and Tech.*, 31(1), 2018.

[4] Joshua A. Kroll. *Accountable Algorithms*. PhD thesis, Princeton University, 2015.

[5] Joshua A. Kroll. The Cyber Conundrum: Why the current policy for national cyber defense leaves us open to attack. *The American Prospect*, 2(9), 2015.

[6] Joshua A. Kroll. Data science data governance. *IEEE Security & Privacy*, 2018. Forthcoming, DOI: 10.1109/MSEC.2018.2875329.

[7] Joshua A. Kroll. The fallacy of inscrutability. *Philosophical Transactions of the Royal Society A*, 376(2133), 2018.

[8] Joshua A. Kroll, Ian C. Davey, and Edward W. Felten. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *Proceedings of the Workshop on the Economics of Information Security*, volume 12, page 11, 2013.

[9] Joshua A. Kroll, J Alex Halderman, and Edward W Felten. Efficiently auditing multi-level elections. *6th International Conference on Electronic Voting (EVOTE 2014)*, 2014.

[10] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. Accountable algorithms. *University of Pennsylvania Law Review*, 165(3), 2017.

[11] Joshua A. Kroll, Nitin Kohli, Guillermo Monge, Nathan Good, and Paul Laskowski. Provable privacy: Modern methods and the FTC's role. Comment to the FTC Competition and Consumer Protection in the 21st Century Hearings, 83 Fed. Reg. 38307, August 2018.

[12] Joshua A. Kroll, Nitin Kohli, and Deirdre K. Mulligan. Lost in translation: Disciplinary confusion in conceptions of values in technology. *Manuscript*, 2018.

[13] Joshua A. Kroll, Gordon Stewart, and Andrew W Appel. Portable software fault isolation. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*. IEEE, 2014.

[14] Joshua A. Kroll, Joe Zimmerman, David J. Wu, Valeria Nikolaenko, Edward W. Felten, and Dan Boneh. Accountable cryptographic access control. *Workshop on Encryption and Surveillance, CRYPTO 2018*, 2018.