Poster: Defining Accountability using Causation and Evidence

Anupam Datta
Carnegie Mellon University
Email: danupam@cmu.edu

Dilsun Kaynar
Carnegie Mellon University
Email: dilsunk@cmu.edu

Divya Sharma
Carnegie Mellon University
Email: divyasharma@cmu.edu

Arunesh Sinha
Carnegie Mellon University
Email: aruneshs@cmu.edu

I. Introduction

Accountability mechanisms complement preventive security and privacy mechanisms by *detecting* policy violations after they occur, identifying agents to *blame* for violations, and *punishing* the violators. The importance of accountability has been recognized in a wide range of areas in security and privacy including cryptographic protocols (e.g., contract-signing [1], auctions, voting [8], anonymity protocols including electronic cash and group signatures), computer systems (e.g., using audit logs for access control systems [9]), copyright protection [12], and privacy protection on the Web, healthcare, financial and other sectors [2], [12], [4], [3].

In this work, we seek to develop a definition of accountability that provides a semantic basis for identifying agents to blame for a policy violation. An important desideratum for this definition is that it be general, i.e., applicable to a broad set of application domains including the ones mentioned above. We plan to explore the possibility of algorithmically checking whether a given mechanism satisfies this definition and to use the definition to guide the design of new accountability mechanisms. Finally, we plan to examine the relationships between our definition and prior work on blame assignment (e.g., [1], [8], [7]).

In this poster, we report on progress towards such a definition. Our view is that an agent should be blamed for a violation on an execution if that agent were at *fault* and the agent's actions *caused* the violation. We formalize *fault* as deviations from expected behavior (e.g., not following the program that an honest participant in a cryptographic protocol is expected to follow or not discharging the responsibilities in an enterprise workflow). We formalize *cause* using ideas from counterfactual definitions of actual causation [6], [11], [10], [5], [13]. Since these definitions of causation are not properties of single executions, our definition of accountability is also not a trace property in contrast to some prior definitions in the literature [1], [8]. Also, in contrast to the prior work on causation using structural equations [11], [6], our definition is formalized using a model that explicitly captures the execution semantics of concurrent multi-agent systems. The level of abstraction of our model is thus closer to models for security protocols and enterprise workflows. Furthermore, while actual causation has been an extremely difficult concept to formalize in its full generality, we believe that by focusing on the restricted class of security and privacy mechanisms, we have a better chance of overcoming the various difficulties that this line of work has faced over the years.

II. Model

Our formal model captures multiple interacting agents who perform actions—either following the expected behavior or deviating from it. Since linking actions to their performers is important for blame assignment, we need a formalism that makes explicit which agent has performed any given action. Among several formalisms that support this feature, we choose to work with *Causal Concurrent Game Structures*—a restricted class of Concurrent Game Structures (CGS) that we define, and a temporal logic supported by CGS because (i) it allows us to model the actions of multiple interacting agents explicitly (ii) the generality of these structures enables their use as a model for expressive logics (iii) use of these general structures allow integration of our work on blame assignment with our prior work on detecting policy violations [3]. We model an execution of the system (as recorded, for example, on an audit log) as a computation in the concurrent game structure.

III. Definitions

We hold an agent accountable for a violation, if the agent is at fault and the agent's action is determined to be part of a sequence of actions that is a cause of the violation.

In order to counterfactually reason whether a sequence of actions is a cause of a violation, we check whether the sequence leads to the violation and if it is the 'minimal' sequence that is sufficient for the violation. For testing non-redundancy of a sequence S, we test whether $S - [a]$ is sufficient for the violation where a represents an action in the sequence S. We repeat the test for every action a in sequence S, in order to establish the necessity of the action a for the sequence S. In this manner, we obtain a sequence of actions, which is *sufficient* to cause the violation, and each of the elements in the sequence is *non-redundant* for the sequence. For a computation α in the model, we define $\vec{P}_\alpha$ to be the sequence of actions for the states in α (modeled as a sequence of propositions corresponding to a sequence of states α in the CGS).

Further, our notion of accountability is based on evidence in an audit log: we cannot establish accountability of an agent by considering possible behaviors that have not occurred in reality. Therefore, we identify those behaviors (modeled as computations in CGS) that occur on the log by only considering those actions as potential causes that occur on the log. We use these concepts to define cause:

Sketch of Definition of Cause: Let M denote a concurrent game structure, ϕ be a logic formula (representing a violation), and α be a computation of M such that $\alpha \models \phi$ (representing a violation). We say that a sequence of actions $\vec{P}_\beta$ is a cause of ϕ on α, if Property 1 and Property 2 hold:

1) **(Sufficiency)** The sequence of states in β is a subsequence of the states on the log α and there exists a computation α' of the model $\mathcal{M}$ such that $\alpha' \models \phi$ and $\vec{P}_\beta = \vec{P}_{\alpha'}$.
2) **(Non-redundancy)** If we consider a proper subsequence $\vec{P}_{\beta'}$ of actions in $\vec{P}_\beta$, either the corresponding computation $\beta' \notin M$ or $\beta' \models \neg\phi$.

The definition of cause consists of two properties. Property 1 captures the sufficiency condition that the sequence of actions denoted by β leads to a violation. Property 2 ensures that each of the actions in the sequence β are non-redundant, i.e., each of the actions is necessary for the sufficient sequence β found in Property 1. The sequence of actions $\vec{P}_\beta$ could contain a single action, in which case the action is a cause of the violation. The sequence of actions $\vec{P}_\beta$ could contain multiple actions, in which case each of the constituent actions would jointly be the cause of the violation. Additionally, there could be several different sequences of actions that satisfy the definition, in which case we would obtain independent causes of the violation. Combining this definition of cause with the notion of fault, we define accountability:

Sketch of Definition of Accountability: Given a model M and a violation ϕ, let α be a computation of M such that $\alpha \models \phi$. If an agent A is at fault, and the action a by the agent A is part of a sequence of actions $\vec{P}_\beta$ which is determined to be a cause of violation ϕ, then A is accountable for ϕ.

If an agent A is at fault, and the action a by the agent A is the only action in a sequence of actions $\vec{P}_\beta$ determined to be a cause of violation ϕ, then we hold A *individually accountable* for ϕ. If a sequence of actions $\vec{P}_\beta$ which is determined to be a cause of violation ϕ contains multiple actions and a subset of at least two agents controlling those actions are at fault, then that subset of agents is *jointly accountable* for ϕ. If several sequences of actions are found to be causes, then the corresponding sets of accountable agents are *independently accountable* for the violation.

REFERENCES

[1] M. Backes, A. Datta, A. Derek, J. C. Mitchell, and M. Turuani, "Compositional analysis of contract-signing protocols," *Theor. Comput. Sci.*, vol. 367, no. 1-2, pp. 33–56, 2006.

[2] A. Barth, J. C. Mitchell, A. Datta, and S. Sundaram, "Privacy and utility in business processes," in *CSF*, 2007, pp. 279–294.

[3] A. Datta, J. Blocki, N. Christin, H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Sinha, "Understanding and protecting privacy: formal semantics and principled audit mechanisms," in *Proceedings of the 7th international conference on Information Systems Security*, ser. ICISS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 1–27.

[4] J. Feigenbaum, A. D. Jaggard, and R. N. Wright, "Towards a formal model of accountability," in *Proceedings of the 2011 workshop on New security paradigms workshop*, ser. NSPW '11. New York, NY, USA: ACM, 2011, pp. 45–56.

[5] J. Y. Halpern, "Defaults and Normality in Causal Structures," *Artificial Intelligence*, vol. 30, pp. 198–208, 2008. [Online]. Available: http://arxiv.org/abs/0806.2140

[6] J. Y. Halpern and J. Pearl, "Causes and explanations: a structural-model approach: part i: causes," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, ser. UAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 194–202.

[7] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely, "Towards a theory of accountability and audit," in *ESORICS*, 2009, pp. 152–167.

[8] R. Küsters, T. Truderung, and A. Vogt, "Accountabiliy: Definition and Relationship to Verifiability," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010)*. ACM Press, 2010, pp. 526–535.

[9] B. Lampson, "Computer security in the real world," *Computer*, vol. 37, no. 6, pp. 37 – 46, june 2004.

[10] J. L. Mackie, "Causes and Conditions," *American Philosophical Quarterly*, vol. 2, no. 4, pp. 245–264, 1965.

[11] J. Pearl, *Causality: models, reasoning, and inference*. New York, NY, USA: Cambridge University Press, 2000.

[12] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman, "Information accountability," *Commun. ACM*, vol. 51, no. 6, pp. 82–87, Jun. 2008.

[13] R. Wright, "Causation in tort law," *California Law Review 73*, pp. 1735–1828, 1985.