End to End Semantic Accountability (E2ESA)

Lalana Kagal and Daniel Weitzner
Decentralized Information Group
MIT Computer Science and Artificial Intelligence Lab
{lkagal, djweitzner}@csail.mit.edu

What are we trying to do ? How is it done at present?

Though large-scale, decentralized systems like the Web provide ease of information flow, this information revolution comes with the challenges of inappropriate use. Excesses and abuses in the use of information are most commonly considered problems of information security. They are seen as consequences of unauthorized access, and accordingly, enormous effort in current information technology research and development is devoted to inventing more reliable methods for restricting access to information. However, even when access restriction can be perfectly and completely achieved, there are significant cases where policies implemented purely as up front controls are too rigid to faithfully reflect societal needs.

An alternative approach is to emphasize the design of systems that provide greater *information accountability* as judged against rules governing appropriate use. In a world where information is ever more easily copied and improperly passed on even by authorized users, and where automated correlations and inferences across multiple databases can uncover information even when it has not been explicitly revealed, access control is simply not enough and accountability must become a primary means by which society addresses issues of appropriate use. We propose to address risks to privacy protection and to extend the Web architecture to support transparency and accountability of data aggregation, inference, and use.

What is new about your approach? Why do you think it will be successful?

We believe that transparency and accountability can be supported by a set of technical mechanisms we call *Policy Awareness*. Policy Awareness is a property of information systems that provides all participants with accessible and understandable views of the policies associated with information resources, provides machine-readable representations of policies in order to facilitate compliance with stated rules, and enables accountability when rules are intentionally or accidentally broken. We propose that information accountability on the Web will emerge from the development of three basic capabilities: policy-aware audit logging, a policy language framework, and accountability reasoning tools.

In a decentralized system each endpoint will have to assume the responsibility of recording information usage events that may be relevant to current or future assessment of accountability to some set of policies. These logs will become the basis of assessing policy accountability either in real time or at some point in the future when such an assessment is needed. A policy-aware transaction log will initially resemble traditional network and database transaction logs, but also include data provenance, annotations about how the information was used, and what rules are known to be associated with that information.

Assessing policy compliance over a set of transactions logged at a heterogeneous set of Web endpoints by a diversity of human actors requires some common framework for describing policy rules and restrictions with respect to the information being used. We consider it improbable in the extreme that the entire world would ever agree on a single set of policy language primitives. However, drawing on Semantic Web techniques including ontologies and rules languages, we believe it will be possible for larger and larger overlapping communities on the Web to develop a shared policy vocabulary in a step-by-step, bottom-up fashion.

Accountable systems must assist users in seeking answers about compliance of data usage with specific policies. It seems likely that special purpose reasoners, based on specializations of general logic frameworks, will be needed to provide a scalable and open policy reasoner. An initial application of special reasoning techniques has been the AIR policy reasoner, which uses dependency tracking to generate explanations for violations of privacy policies.

Research Results As part of our initial investigation into developing accountable systems, we have developed (i) a rule-based policy language, AIR, for defining privacy policies about the appropriate uses of information, (ii) a reasoner that is able to identify and explain policy violations in transaction logs, and (iii) a justification user interface that provides users with a graphical view of the explanation for why the policy violation occurred.

AIR Policy Language: Accountability in RDF or AIR is a policy language grounded in Semantic Web technologies that exploits dependency tracking in order to provide explanations for policy decisions and violations [4]. In information networks, we expect heterogeneous formats for data from different domains. In order to provide shared semantics not only for the data but also the privacy policies, the use of Semantic Web technologies is critical. This not only allows the language and reasoner to support different data formats but also allows the integration of data from different domains. Semantic Web technologies include languages such as Resource Description Framework (RDF) [5] and Web Ontology Language (OWL) [1] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. The AIR language is represented in Turtle [2], which is a human readable format for RDF. For any given conclusion, it is useful to know the specific set of premises that it was derived from; this set is called the set of dependencies for the conclusion. The AIR reasoner uses dependency tracking to identify the premises of every policy result and integrates them with descriptions associated with policies to generate explanations. The AIR ontology expressed in OWL comprises several classes and properties that are used to define rule-based policies.

AIR Reasoner: The AIR Reasoner focuses on reasoning over transaction logs and privacy policies to provide explanations for policy violations. The reasoner tracks dependencies during the reasoning process in order to provide integrated justification support. Policy administrators are not required to manipulate these dependencies or justifications but if required, can modify them to provide customized explanations. Dependency tracking is the process of maintaining dependency sets for derived conclusions. Some dependency-tracking mechanisms provide additional features. For example, a Truth Maintenance System (TMS) [3] keeps track of the logical structure of a derivation, which is an effective explanation of the corresponding conclusion. Another useful feature, also provided by a TMS, is the ability to assume and retract hypothetical premises. AIR is able to produce a concise explanation for any result it computes. It is hard to overstate the importance of explanations: in many cases, the explanation for a result is more important than the result itself. For example, to someone being arrested by a police officer, an explanation is likely to be very important. And later, a detailed justification for the arrest may be crucial as well.

Justification User Interface: As explanations are usually in the form of proof trees, which might be incomprehensible to end users, we have developed a graphical Justification User Interface in Tabulator [6], a Firefox extension for Semantic Web browsing. The interface allows users to view the explanation provided by the AIR reasoner in different ways: (i) in a simple Semantic Web based rule language, (ii) in a graphical layout that highlights the result of the reasoning and shows both its natural language explanation as well as its specific premises (or dependencies) and allows these explanations to be explored, (iii) a textual view that presents the information in a format that is expected by lawyers.

When we succeed, what difference will it make?

By augmenting information with data about provenance and usage policies, and developing automated means for maintaining that provenance and interpreting policies, we will be able to create Web architectures that support transparent and accountable data usage. In its place, information accountability through policy awareness, while a departure from traditional information security techniques, is actually consistent with the way that legal rules traditionally work in democratic societies - we follow rules because we are aware that they are there and because we know there will be consequences if we violate them. We hope that, in a similar manner, having policy awareness in information architectures will encourage users to conform to the governing policies because they will understand the consequences of policy violation.

References

[1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference, W3C Recommendation. http://www.w3.org/TR/owl-ref/, February 2004.

[2] D. Beckett. Turtle - Terse RDF Triple Language. `http://www.dajobe.org/2004/01/turtle/`.

[3] J. Doyle. A Truth Maintenance System. *Journal of Artificial Intelligence, November 1979*.

[4] L. Kagal, C. Hanson, and D. Weitzner. Explanation Generation via Dependency Tracking. In *9th International IEEE Workshop on Policies for Distributed Systems and Networks (POLICY 2008)*, 2008.

[5] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004. `http://www.w3.org/TR/rdf-concepts/`, 2004.

[6] Tim Berners-Lee and Y. Chen and L. Chilton and D. Connolly and R. Dhanaraj and J. Hollenbach and A. Lerer and D. Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *SWUI06 Workshop at ISWC06*, 2006.