Research Statement

Vicky Weissman

I am interested in using formal methods to solve problems in security. More generally, I want to identify problems that are, or are likely to be, of practical significance within the next ten years. Then I want to propose solutions that are, in some sense, provably correct.

My dissertation is about security policies, especially as they are used for DRM (digital rights management). A policy describes the conditions under which an action, such as copying, modifying, or distributing digital content, is permitted or forbidden. Digital content providers write policies to govern the use of their works. For example, the ACM (Association for Computing Machinery) has a set of policies that regulate access to their digital library. These policies include statements such as 'members of the ACM are permitted to access the articles in the library for personal use' and 'members may not republish articles without explicit permission from the ACM'. Digital content providers want their policies to be intelligible to human readers and enforceable by computers. In addition, they want the task of writing policies to be as easy as possible. My dissertation is a step towards meeting these goals.

Giving Policy Languages a Formal Foundation

A number of languages have been proposed for reasoning about the policies of digital content providers (e.g., digital libraries, certain online stores, and even healthcare providers). The languages that seem most likely to be used in practice include XrML, ODRL, and MPEG-21 REL. XrML has been endorsed by several companies including Microsoft, Hewlett-Packard, and Xerox; ODRL is arguably the favorite of the open source community; and MPEG-21 REL is an international standard based on XrML. When I began my research, none of these languages had formal semantics. As a result, policies written in the languages were open to interpretation and, thus, were neither intelligible to human readers nor enforceable by computers.

To address this problem, Joe Halpern and I proposed the first formal semantics for XrML. We also proposed the first formal semantics for the March 2003 version of MPEG-21 REL, which was the most recent version when I began my work and the last version before the final release. At the same time, Riccardo Pucella and I proposed the first formal semantics for ODRL. For each language, we provided formal semantics by giving a translation from policies in the languages to formulas in first-order logic. We chose this approach because it allowed us to build directly on previous work done by the formal methods community.

When creating the translations, we found substantial problems with each language. I discussed our findings with Xin Wang, who is one of the creators of XrML and of MPEG-21 REL, and with Renato Iannella, who is the creator of ODRL. Since these discussions, the final version of MPEG-21 REL has been released. In that version, all of our concerns are addressed and, in fact, the language has formal semantics. I believe that these improvements were made, at least in part, because of my talks with Xin. As for ODRL, a new version is currently under development and a requirements document for it was released in November. According to the document, formal semantics is one of seven key features that the new version should have. I have been invited to join the ODRL working group to help meet this goal.

The Importance of Saying No

When creating a policy language, the standard approach is to include only statements about what is permitted. It is assumed that an action is forbidden, unless it is explicitly permitted. A shortcoming of this approach is that we cannot distinguish actions that are forbidden from the ones that the policy maker simply does not care about. This becomes a problem when multiple policy sets govern access to the same resource. For example, suppose that access to a document in a digital library is regulated by the library's policies and the policies of the document's donor. To enforce the policies correctly, we need to detect if the library permits an action that the donor forbids (or vice-versa) and resolve the conflict accordingly. However, if every action that is not explicitly permitted is forbidden, then we cannot detect conflicts. To see why, notice that if the library allows its patrons to access the card catalog and a donor does not mention the catalog, then the two policy sets conflict because the donor forbids access to the catalog by not explicitly permitting it. In general, two policy sets will conflict unless they are identical. It follows that we usually cannot enforce multiple policy sets that govern access to the same resource, unless the policy sets explicitly state what is permitted and what is forbidden.

We can easily extend policy languages to include statements about what is forbidden. The challenge is to do so in such a way that the policies can be enforced in a reasonable period of time. To create an appropriate language, I first collected policies from various sources, including libraries, universities, and a database of government legislation. Once I had a sense of what people wanted to say, Joe Halpern and I discovered a tractable fragment of first-order logic that could capture these policies. Now we are using our results to find suitable extensions to XrML, ODRL, and MPEG-21 REL.

When we begin to consider multiple policy sets, a number of interesting questions arise. For example, suppose we have two policy sets; the policy maker is confident that the first captures her intentions and the second is easier to implement. Can we prove that the two sets permit and forbid the same actions? Similarly, given two policy sets, can we prove that the first set permits an action only if the second set permits it? What are the consequences of changing a policy set while the system is running? Riccardo Pucella and I developed a policy language and showed how we could answer these questions for all policy sets written in the language. Less than a year after our work was published, Joe Halpern and I began working with Thomas Macklin at NRL to find a suitable policy language for one of their systems. Although we are still in the preliminary stages of this project, it appears that their needs include the type of policy comparisons that Riccardo and I investigated.

Usability

The policy languages mentioned in the preceding sections are expressive and enforceable by computers. For many policy writers and readers, however, they are not intuitive. Using a policy language that is based on formal logic or that follows XML conventions requires training and, even with training, reading XML syntax is not easy. There are policy languages that are intuitive, but to the best of my knowledge, none of them are highly expressive. For example, in a standard UNIX environment, it is easy to formulate the statement ‘Alice may read file f’, but we cannot say ‘if an individual pays five dollars, then she may read file f’. Similar practical limitations exist when writing policies using the Creative Commons approach and when using the interfaces provided as part of Microsoft Office 2003.

Carl Lagoze and I have created a language, called Rosetta, that we believe strikes a better balance between expressivity and ease of use. Rosetta is essentially a set of templates for English sentences. A policy writer creates a policy by selecting the appropriate template and filling it in. Since the resulting policy is a sentence in English, the only skills that we are assuming policy writers and readers have are (1) the ability to fill in forms and (2) a knowledge of English, which is not universal but is at least more likely than familiarity with formal logic or XML conventions. To facilitate both enforcement by computers and formal analysis, we provide a translation from statements in Rosetta to statements in the XML-based and logic-based languages discussed previously.

Future Work

Part of my future work will, of course, be to continue my current projects. These include being a member of the ODRL working group and finding an appropriate language for one of the NRL systems. I am also planning on extending my research in a few straightforward ways. For example, I am interested in whether we can combine the various policy languages to get the best of each. Also, having considered how to compare and enforce policy sets, I would now like to determine how we can answer other types of questions, such as ‘what can an individual do (if anything) so that a desired permission follows from a given policy set’. When considering new research directions, my focus is on reasoning about policies in a broader context and meeting the needs of particular classes of applications. Some details are given below.

We can think of a policy set as a program that is chosen by the policy maker to satisfy a specification. For example, Congress passed the HIPAA legislation (the Health Insurance Portability and Accountability Act of 1996 and related laws), to better meet the specification that medical records are kept confidential. Similarly, a digital library might adopt certain access control policies to protect minors. In this context, standard programming language questions arise. In particular, what is a good language for capturing the specifications that are of practical interest; can we determine whether a specific policy set satisfies a certain specification; and, given a specification, can we generate a satisfying policy set. A less common question that I find especially interesting in this setting is ‘given a policy set, can we infer the specification with a reasonable degree of confidence’. If the answer to this last question is ‘yes’, then we can use the information to help policy makers discover unintentional biases and avoid misunderstandings with customers. If the answer is ‘no’, then we can guarantee that, in some sense, the policy maker’s privacy is preserved (e.g., given Alice’s policies, we cannot determine who she does or does not trust). Over the next five years, I would like to address all of these questions.

I am also interested in creating policy languages that are tailored to particular classes of applications. For example, PDAs and other small devices have minimal storage capacity and processing power. One of my research goals is to design a policy language that is well-suited to this environment. A natural extension is to consider DRM in sensor networks. Even without technological restrictions, sometimes less is more. For example, many digital content providers might prefer a language in which they could *not* write illegal policies, particularly if reducing the options made policy writing easier. I would like to find a simple language that is just expressive enough for a wide range of applications.

To answer the questions that I have outlined above in a meaningful way requires collaborations with lawyers, engineers, and end-users. For example, to create an appropriate specification language, I need to understand what types of specifications are of practical interest. An obvious way to get this insight is to talk with end-users about their needs. Also, if I am going to create a policy language that is well suited to small devices, then it makes sense for me to work with an engineer in the field. Because DRM is inherently a multi-disciplinary field, an important part of my research agenda is to develop relationships across disciplines. I have taken some preliminary steps towards this goal by taking classes in the law school, working with policy language developers, and co-chairing a privacy workshop that brought together people in medicine, finance, and digital libraries. I believe this type of interaction will not only help me address the questions discussed above, but will also lead me to future research that is important and exciting.