Research Statement

Wilfredo Marrero

My research has been motivated by the desire to address real world problems. For the most part, I have chosen problems that deal with large, complex systems. When analyzing such systems, requirements and correctness become a major issue. It is my experience that tools such as modeling, logic, abstraction, formal reasoning, and computer automation can be of great help when trying to understand and analyze these systems. Ultimately, however, the success of research in this area is measured by how much it helps people to design and to build large systems.

As a doctoral student at Carnegie Mellon University my research has dealt with system design, modeling, and analysis. My research began with model checking and hardware verification. I was also interested in real-time systems, and how a model checker could be used to analyze the performance of these systems in addition to verifying correctness. Recently, however, I have devoted myself to modeling and verifying computer security protocols including authentication protocols and electronic commerce protocols. Below I summarize this research and provide some areas of research I wish to pursue after graduation.

Past Research

My research has spanned a number of topics, both theoretical and practical, but it has always been directed by real world problems. While an undergraduate at Columbia University, I was selected to participate in the Minority Summer Research Program where I was first exposed to computer science research. That summer and the following summer, I helped with formalizing and implementing the object consistency model in MARVEL, a rule-based, object oriented software development environment. The problem involved insuring that a collection of objects remains consistent with respect to a set of rules as the objects and the rules evolve.

When I became a student at Carnegie Mellon University, I joined Professor Edmund Clarke's group. While in this group I investigated a number of issues related to model checking. On the more theoretical side I investigated how to improve the worst-case running time for model checking the modal μ-calculus. On the more practical side, I looked at applications of the μ-calculus, and I also investigated how to extend the ideas of model checking to include performance evaluation of real-time systems. I was also exposed to some of the great successes of model checking in the field of electronic circuit verification. I spent one summer at Intel Corporation investigating the use of symbolic model checking to verify arithmetic circuits. While there, I worked as part of a larger group investigating the use of formal methods to validate microprocessors. The following summer I worked at Cadence Berkeley Labs investigating the use of an model checker for verifying super scalar microprocessors. This background in requirements analysis helped to influence the direction of my thesis research.

Thesis

The rapid growth of such entities as "the Internet" and "the World Wide Web" has made computer security of paramount importance. The increase in the number of users has made intrusions more numerous. A larger problem, however, is the increase in the number and kinds of services provided on these insecure channels. Such services as electronic commerce, online auctions, key distribution, and even electronic mail require security guarantees such as privacy, authenticity, non-repudiation, integrity, and anonymity. Designers have turned to cryptography in an attempt to guarantee these properties; however, cryptography is not sufficient. Even if the cryptosystems used could not be compromised, the cryptography must be used correctly in order to provide these guarantees.

When arguing the correctness of a security protocol, the reasoning can quickly get out of hand. The errors that can occur in security protocols can be very subtle and difficult to catch. Compounding the problem is the necessity to make trade-offs in the protocol design in an attempt to make the protocols more efficient by using smaller or fewer messages. It is precisely in problem domains such as this that formal methods have proven useful in analyzing and debugging systems. For this reason, my thesis research is centered on the construction of a model checker for security protocols and on the analysis of a number of authentication and electronic commerce protocols.

Future Work

There are still a number of research topics in computer security, system analysis, and systems in general that I wish pursue. Below is a list of some of the immediate and some of the more long term research problems I am interested in investigating after graduation.

Immediate

There are some obvious extensions that can be made to my model checker that I would like to pursue after graduation. I would like to increase the kinds of actions allowed as as to be able to model protocols with greater detail. I would also like to add new operations on messages that so that other kinds of encryption and other kinds of protocols could be modeled. I also have ideas about how to increase the efficiency of the model checker so that larger systems can be analyzed.

Near Term

There are still other systems and properties that I am interested in analyzing. For example, I would like to investigate protocols for online auctions, electronic voting, and digital cash. These examples may require a large extension to the model used, but I believe they are amenable to this kind of analysis. I am also very interested in the area of information survivability. Here it is not sufficient to check security properties. One must also guarantee a certain level of service even in the presence of intrusions and failures. I believe a good first step would be a formalization of what is meant by survivability. Once this has been done, we can investigate how to model survivable systems, and we can develop tools to help design, analyze, and build them.

Long Term

I also have some long term research interests in computer security. More specifically, these revolve around the idea of how to help people to build secure systems. One way of doing this is by composing subsystems. Even if these subsystems can be shown to be secure independently (for some notion of secure), it is not clear how they can be composed in such a way as to guarantee global security properties. The other way of building a secure system is to "do it yourself." A significant number of security flaws occur at the implementation level, so I am extremely interested in ways to help insure the implementation of secure systems. This could include hardware support, operating system support, and programming language support. Additionally, with the increase in the amount of downloaded code and of remotely executed code, the area of proof carrying code also strikes me as terribly fascinating.