Guest Editorial Overview

INFORMATION security plays an important and increasingly critical role in society. It is, therefore, essential that we have effective tools and techniques to design and evaluate secure systems and demonstrate that they meet their security requirements. The application of rigorous methods to the specification, modeling, analysis, and design of security-critical systems has made considerable strides in recent years, and the field is rapidly gaining in maturity.

The aim of this special edition is to bring together some of the best, recent, developments in the use of mathematically well-founded techniques and tools for the modeling and analysis of security-critical systems. We believe that the time is now ripe for such a snapshot: the area has reached a sufficient degree of maturity that such tools and techniques are now regularly applied to real, commercial products and designs. At the same time there are still many open questions and challenges and it remains a very dynamic area of research.

It was recognized early on that to achieve predictability, and hence dependability, in the design and implementation of critical systems it is necessary to apply more systematic techniques to their development and evaluation than simple *ad hoc* code cutting. Many approaches have been proposed that vary widely and involve differing degrees of rigor and formality. Thus, at the more informal end of the spectrum, we have structured systems, analysis, and design methodology (SSADM), the Jackson Development method, and so on. However, for systems whose correct operation was considered critical, the need for more rigorous and mathematical techniques and concepts was recognized.

The security community appreciated early on the need for rigorous, ideally formal, approaches to the modeling and analysis of designs and implementations. Much of the early work in developing automated support for the verification of designs against security requirements was stimulated and funded by the organizations such as the National Security Agency (NSA). The Gypsy system developed at Computational Logic, Inc. (CLI) and the hierarchical development method (HDM) developed at Stanford Research Institute (SRI) are prime examples.

In the 1970s, a number of major projects had formal methods applied in the development process. By and large these experiences were disappointing: the time and effort involved was very high and the benefits derived highly debatable. A number of reasons for this lack of success have been proposed but it seems likely that the methods were at that stage too immature and the projects they were applied to too ambitious. Attempts were made, for example, to verify trusted operating systems that were way beyond the techniques of the time, or probably even of current techniques. In particular, since security properties are generally not functional capabilities, but pervasive constraints on the way all parts of a system behave, it is difficult to verify them conclusively from a description of a limited aspect of a system.

A further difficulty was the fact that design and evaluation processes tended to be conducted quite separately. This meant that the evaluation had little or no chance to influence the design, in particular little opportunity to steer the design toward architectures more amenable to analysis. The most damning observation is that in several cases the deployed design had drifted so far from the design that had been handed over to the analysts that the output was rendered virtually irrelevant.

These rather bad early experiences gave rise to a crisis of confidence in the community and a rather poor image to the whole idea of proofs of system dependability. Around this time a pair of workshops sponsored by the U.S., U.K., and Canadian governments were convened to discuss the role and relevance of formal methods in critical systems [1], [4]. As R. Morris, then Chief Scientist at the National Computer Security Center (NCSC) put it: "The question is: 'Where do I put my extra $5 of verification money?'". The conclusions were, very broadly, that formal methods do have a useful role to play but must be used in conjunction with other techniques like testing. It was also observed that notations like Vienna development method and the Z notation (VDM, Z), etc., could be usefully deployed even without necessarily performing proofs: the process of casting requirements and high-level designs in a formal notation itself served to greatly sharpen understanding and reduce bugs in implementations.

After this rather unpromising start, the community has, over the past decade or so, turned things around and chalked up some significant successes. This is due to the growing maturity and effectiveness of the tools and methods. Theorem provers have moved on from the Boyer–Moore days of "pushing on one end of the string" and are now much more usable, with more intuitive interfaces and better automation of the tedious steps of the proofs. Model-checkers have also made major strides and have been making a significant impact beyond their original area of application to hardware verification.

In addition, success has followed more careful focus on specific modeling contexts. Verifying the security of an operating

Digital Object Identifier 10.1109/JSAC.2002.806120

system is intractable partly because interactions occur at many different levels of abstraction, so that the verification is not complete until they have all been considered. By contrast, working within a particular modeling framework focuses attention on a particular class of problems that can arise; it is significant progress to prove that a protocol or smart card avoids a class of problems. Many of the papers in this issue devote a good deal of attention to finding a useful modeling framework. A modeling framework must be somewhat restrictive, so that verification goals can be sharply defined, and strong methods be developed to achieve them. However, it must also be realistic, so that a wide range of practical troubles have been eliminated when correctness is proved within the model. A natural part of scientific development is discovering more inclusive models that remain tractable to work with, or alternative models that can eliminate new classes of problems.

On a similar timescale, information security has moved from being primarily the preserve of governments and the military to being a major concern of society at large. This is due mainly to fact that we are increasingly moving into an age of information; the fabric of society depends increasingly on the secrecy, integrity, and availability of information. Another, more subtle factor is the revolution in cryptography brought about by the invention of public key cryptography [2], [3]. This turned cryptography from being a black art practiced in the confines of government establishments to a respectable and thriving academic discipline.

The result is that mathematical tools and techniques for the design and analysis of security critical systems now appear to have reached a good degree of maturity. In this issue, the formal analysis of a "real" (and indeed rather complex) electronic commerce protocol, secure electronic transactions (SET), is presented for example.

Mathematical analysis is not enough to ensure the security of deployed security systems, or other critical systems for that matter. The models on which the analysis is based will inevitably be abstractions and approximations of reality. Our system may be provably secure with respect to the model and assumptions and yet be vulnerable to timing or differential power attacks if these aspects of reality have been abstracted away from the models. Furthermore, many security failures are actually due to nontechnical attacks, social engineering and the like. Nonetheless, there is great value in mathematical modeling and analysis and making explicit underlying assumptions.

The fact that the field is of such great importance and that many challenges remain, both theoretical and practical, is ensuring that this remains an active and indeed growing research area. The reader will find indications of many of the open questions and challenges in the papers. The analysis of security protocols has been particularly active and Meadows' paper gives the reader an accurate indication of where the current activity is concentrated.

In this special issue, we bring together a collection of papers on a number of aspects of analysis techniques. In the first paper, "Language-Based Information-Flow Security," by Sabelfeld and Myers present a wide-ranging survey of the research from the late 1970s to the present, into the nature of information flow and techniques for achieving it. The paper particularly focuses on recently developed approaches based on programming-language based techniques, and identifies a number of open challenges for research in this area.

"Real-Time Information Flow Analysis," by Focardi *et al.* is concerned with the investigation of information flow properties in the presence of timing considerations. The paper builds on previous work in the security process algebra (SPA) by incorporating timing behavior into the model and considering the hierarchy of noninterference properties in this new setting. The paper demonstrates that timed analysis can identify insecurities that are not apparent in the untimed model.

In "A Nonfunctional Approach to System Integrity," by Foley, proposes a framework for analysing integrity properties of a system. He uses the process algebra communicating sequential processes (CSP) to provide a definition of integrity and shows how a number of integrity mechanisms, such as separation of duties, can be understood as ways of implementing integrity. Foley identifies a relationship between integrity and noninterference, which are both nonfunctional properties, and consequently enhances our understanding of integrity. The relationship allows for the possibility that techniques for establishing noninterference might also be applicable to integrity properties.

Verification methods for security protocols have seen an explosion of activity over the last decade, with the development of significant new ideas which have transformed ways in which such protocols can be analyzed. Meadows gives a comprehensive overview of this field in, "Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends," and identifies the problems that now need to be addressed to extend the application of the techniques and tools that have so far been developed.

The paper, "Posets and Protocols—Picking the Right Three-Party Protocol," by Ng, shows how security protocols can be optimized by transforming the protocol steps in ways that preserve its original goals. Formal justification for the transformation steps is grounded in the strand space framework for protocol analysis, providing a novel application of that protocol verification framework.

Design techniques are often made practical by the provision of tool support. The paper, "Enhanced Security Protocol Engineering Through a Unified Multidimensional Framework," by Saul and Hutchison, presents Security Protcol Engineering and Analysis Resource II (SPEAR II), a tool that integrates four components useful for protocol analysis: specification, a belief logic, an analysis engine for that belief logic, and a message round calculator. The paper demonstrates that these components integrate into a tool useful for protocol design and analysis.

Formal verification of commercial protocols is difficult because of the sheer size of the protocols involved. In "Verifying the SET Registration Protocols," by Bella *et al.*, apply the

inductive method in Isabelle/Hol to modeling the registration part of the SET e-commerce protocol and verifying secrecy properties of it. The paper gives a flavor of the scale of this task, not least the modeling process which enables the formal verification to proceed.

In "Authentication by Correspondence," Gollmann reviews the use of correspondence properties to characterize authentication. He explores the history of this approach to authentication, how it has evolved, and the variety of such properties that now abound in the literature. Gollmann argues that correspondence properties sometimes, but not always, capture authentication properties, and that their use in protocol verification is, therefore, not always appropriate: the use of correspondence properties as a formal specification of authentication requires careful consideration of the security requirements being modeled.

The final paper in this Special Issue is concerned with the important problem of intrusion detection, for which formal frameworks are now beginning to emerge. "Determining the Operational Limits of an Anomaly-Based Intrusion Detector," by Tan and Maxion, investigates the anomaly detector "stide" and its underlying algorithm and obtains a theoretical justification for empirical observations about the minimal information required for stide to identify intrusions effectively. This information is given in terms of the window length used to consider the behavior of processes running on a system and the paper establishes the lower bound of six by reasoned rather than *ad hoc* means. The paper provides a showcase of how a formal framework can be developed and applied to understanding aspects of intrusion detection.

The editors would like to thank all the authors who submitted papers to this Special Issue. Thanks are also due to the anonymous referees for their reviews and to J-SAC for their assistance in preparing this special issue.

LI GONG, *Guest Editor*
SUN Microsystems
Engineering and Research Institute
Beijing, China

JOSHUA GUTTMAN, *Guest Editor*
The MITRE Corporation
Division for Information Security and Operations
Bedford, MA 01730-1420

PETER RYAN, *Guest Editor*
Newcastle University
School of Computing Science
Claremont Tower
Newcastle upon Tyne NE1 7RU, U.K.

STEVE SCHNEIDER, *Guest Editor*
Royal Holloway, University of London
Department of Computer Science
Egham, Surrey TW20 0EX, U.K.

W. BUX, *J-SAC Board Representative*

REFERENCES

[1] D. Craigen and K. Summerskill, Eds., *Formal Methods for Trustworthy Computer Systems*. New York: Springer-Verlag, 1989.
[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, pp. 644–654, 1976.
[3] J. H. Ellis. (1987) The story of non-secret encryption. [Online]. Available: http://www.cesg.gov.uk/publications/index.htm
[4] P. Y. A. Ryan and C. Sennett, Eds., *Formal Methods in Systems Engineering*. New York: Springer-Verlag, 1991.

Li Gong graduated from Tsinghua University, Beijing, China, and received the Ph.D. degree from the University of Cambridge, U.K.

He is Managing Director of Sun Microsystems's Engineering and Research Institute, Beijing, P.R. China. Previously, he managed the development of Java Security, Java Embedded Server, and JXTA, for which he has three books and six U.S. patents. He did extensive research in distributed systems and security, with over 60 papers. He is an Associate Editor of *ACM Transactions on Information and Systems Security*.

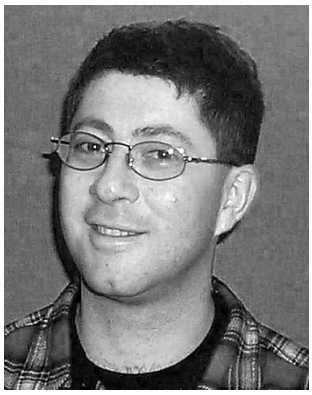
Dr. Gong is an Associate Editor-in-Chief of *IEEE Internet Computing*. He received the Leonard Abraham Award from the IEEE Communications Society.

Joshua Guttman received the A.B. degree from Princeton University, Princeton, NJ, in 1975 and the Ph.D. degree from the University of Chicago, IL, in 1984.

He is Senior Principal Scientist at The MITRE Corporation, Bedford, MA. Having worked in mechanized reasoning and compiler verification, he now focuses on applying rigorous methods to cryptographic protocols, network security, and operating system security.

Peter Ryan is a Principal Research Associate, Newcastle University, U.K., where he heads up the security strand of the dependability interdisciplinary research collaboration (DIRC) project funded by the EPSRC. His areas of interest include information security, formal methods, cryptography, security protocols, dependability and, more recently, the interdisciplinary aspects of security and dependability.

He is the Chair of the Steering Committee of ESORICS, a Member of the International Federation of Information Processing (IFIP) Working Group 1.7, Laxenburg, Austria, on information security theory and has served on numerous programs and organizing committees, notably CSFW and the *IEEE Symposium on Security and Privacy*.



Steve Schneider received the D.Phil. degree in computer science, University of Oxford, Oxford, U.K.

He is a Professor of computer science, Royal Holloway, University of London, Egham, Surrey, U.K., and leads the Formal Methods Group in the Department of Computer Science. He is the main developer of the "rank function" approach to security protocol analysis and verification based on the process algebra communicating sequential processes, and is one of the authors of *Modeling and Analysis of Security Protocols*. His interests include formal methods for security protocols, noninterference, as well as process algebra, real-time systems, and integrating formal methods.

Prof. Schneider served as Program Chair on the *IEEE Computer Security Foundations Workshop*, from 2001 and 2002.