Guest editor's preface

This issue of the *Journal of Computer Security* is drawn from papers presented at the 2000 European Symposium on Research in Computer Security (ESORICS 2000), held in Toulouse, France, 4–6 October 2000. The ESORICS symposia have been held every two years since 1990 and represent the main European forum for security research.

Several papers presented at the ESORICS 2000 Symposium were invited for submission to the Journal. Submitted papers were revised for journal publication and subjected to the normal rigorous review process of the Journal. This issue contains four papers selected for publication through this process.

"Manageable access control for CORBA", by Gerald Brose presents a language and its support for specifying and managing access control policies. This language provides a formal notation that allows the security administrators to express a wide range of practical security policies. This language called VPL for *View Policy Language* is based on the concept of role already widely used in the RBAC model. In this paper, roles have a strictly *functional interpretation* and groups are used to model organizational structure. VPL also uses the concept of view that is introduced as a grouping concept for providing a more comprehensive specification of access control policies. This paper then shows how to combine these concepts in the context of CORBA.

Gerhard Schellhorn and colleagues, in "Verified formal security models for multi-applicative smart cards", present two security models that are extensions of the classical Bell/LaPadula and Biba models. The first model is designed at a very abstract level and the second refines the first by inserting more practical issues that are useful for multiapplicative smart cards. These models include requirements for authentication and intransitive noninterference, and avoid the need for *trusted processes* that is generally viewed as a drawback of the Bell/LaPadula model. An interesting and useful contribution is that, unlike several theoretical papers on noninterference previously published, this paper describes how to use such a model in developing a practical system.

"Checking secure interactions of smart card applets: extended version", by Pierre Bieber and colleagues is a paper on a similar topic. In the context of a multiapplicative smart card, this paper shows how to verify that applets interact in a secure way. The suggested security policy is a MAC policy that associates labels to applet attributes and methods. The main contribution is then to define a technique based on model checking to verify that actual information flows between applets are authorized. This approach is illustrated in the context of an electronic purse running on Java Card.

Ian Welch and Robert Stroud, in "Using reflection as a mechanism for enforcing security policies on compiled code", show how a reflective version of Java, developed by the authors and called Kava, can be used to enforce security policies in an easy and abstract way. This is obtained by defining the *loader* class that, when a new class is loaded, inspects a user-defined meta-configuration file. This file specifies where access controls must take place in the application code. The object code is then modified by inserting suitable controls that switch the execution from the application code to the metaobject associated with each object. The main contribution is to show how to join bytecode transformation and metaobject protocol to enforce access controls.

In conclusion, I would like to thank the authors, for extending the initial versions of their papers and going through several revision cycles according to the referees' comments. And of course, special thanks are due to the anonymous referees for their invaluable help in assessing both contents and presentation of each paper.

Frédéric Cuppens
ONERA, Toulouse, France