Abstract. We present the first implementation of a theorem prover which runs on a smart card. The prover is written in Java and implements a dual tableau calculus. Due to the limited resources available on current smart cards, the prover is restricted to propositional classical logic. It can be easily extended to full first-order logic. The potential applications for our prover lie within the context of security related functions based on trusted devices such as smart cards.

Keywords: automated deduction, tableaux, leanTAP, security, java, proof carrying code

1 Smart Cards: the Secure PC of Tomorrow

Smart cards are currently evolving into one of the most exciting and most significant technologies of the information society. Current smart cards on the market are in fact small computers consisting of a processor, ROM and RAM, an operating system, a file system, etc. This miniature computer resides within a tamper proof chip attached to a plastic card such as a credit card. The computer only lacks the typical I/O devices, but can communicate with a smart card reading machine, and has a small EEPROM (Electrically Erasable Programmable Read Only Memory) space analogous to computer disk space. Although their resources are still quite restricted, continuous advances in chip manufacturing will soon enable smart cards with 32 bit processors and up to 128 Kb of memory. Manufacturers are also thinking about integrating small keyboards and LCD displays on these plastic cards. Thus, the next generation of smart cards will be as powerful as PCs were a few years ago.

The evolution of smart card technology resembles the development of computer technology over the last 20 years: the separation of “physics” and “logic”. While early computers had to be programmed in machine language because each

* Supported by an Australian Research Council Queen Elizabeth II Fellowship

1 For further information about the system and obtaining the code see http://arp.anu.edu.au/~andrews/cardtap. An interactive simulation version will be made available in the future.
bit of memory and each instruction cycle was valuable, the increase of resources and processing power made it affordable to trade resources for higher level programming concepts and languages. This separation of software and hardware was the basis for the spread of computers into everyday life during this decade.

The same phenomenon is about to take place in smart card technology: as resources and processing power increase, it will become affordable to neglect the optimal use of the card processor and memory. The most promising move in this direction is Java smart cards, where a Java virtual machine is implemented inside the card. The software determining the function of the card is no longer tied to the particular card, but multiple applications can be loaded onto, and removed from, the card as desired.

The primary purpose of smart cards will probably continue to be security-related applications since they serve primarily as a trusted device for their owner. The most important applications to date are of a cryptographic nature like authentication and encryption, e.g. for electronic cash. Future applications running on more complex cards will be able to carry out more complex operations so that the smart card of the future will be a secure, personal computer.

Current smart cards have security-related applications hard-wired onto them. Future smart cards will serve multiple purposes and will be adaptable by downloading one or more applications. Interactions between such applications, and between the card and the outside world therefore become non-trivial. Security issues arise when new and hence untrusted code is introduced to the device, and when known code on the device is requested to perform a sequence of transactions that could result in a violation of security. In both cases we can test for malicious intent by verifying that insecure situations cannot eventuate. Formal logic is not only well-suited for modelling such complex interactions but is also ideal for describing a given security model. Consequently, a trusted, secure, personal device should be able to perform logical reasoning to ensure that the card complies to its owner's security model. A concrete example of the use of formal logic for the purposes of code-safety has been given in the context of proof-carrying code [1]. The significance of implementing a theorem prover capable of operating on a smart card is that we may determine the viability of using techniques from automated deduction to absolutely ensure security in situations where the flexibility of advanced smart card technology thwarts their popular use as a trusted device. At the practical level, untrusted code verification would require an automated deduction system, using an appropriately customised logic, to prove that a sequence of Java byte-code complies with a given security model. Additionally, we suggest that these small verification systems on smart cards will have applications for testing the legality of given input command sequences for a given smart card application and a particular owner's security model.

Automated theorem proving in classical logics is now a mature field, but automation of theorem proving in non-classical logics is a thriving field in artificial intelligence research. Extensions of our system will allow verification using communication protocols expressed in modal and authentication logics [2].

The challenge to implement a verification system for a smart card is to iden-