System Description: cardTAP: The First Theorem Prover on a Smart Card

Rajeev Goré¹, Joachim Posegga², Andrew Slater¹, and Harald Vogt²

¹ Automated Reasoning Project, Australian National University, 0200, Canberra
² Deutsche Telekom AG, Technologiezentrum, IT Security, D-64276 Darmstadt

Abstract. We present the first implementation of a theorem prover running 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.

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. Although their resources are still quite restricted, continuous advances in chip manufacturing will soon enable to market smart cards with 32 bit processors and up to 128 KByte 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 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 are 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 continue to be security-related applications since they will serve 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. Today's smart cards are used essentially as simple authentication devices in these contexts. Future smart cards applications, however, will take advantage of the fact that 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. 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 is the use of formal logic in the context of proof-carrying code [1].

Here we outline the first successful implementation of a theorem prover on a Java smart card.

2 Implementation Details

cardT^A_P is a theorem prover for propositional logic that uses a dual tableaux method based on leanT^A_P [2]. cardT^A_P was specifically designed to reside on a smart card; the program executable size is less than 2 KByte, and the stack usage, heap space and allocated memory is minimal. To achieve this, cardT^A_P naively simulates Prolog's run time stack and backtracking environment using recomputation. The trade off is efficiency: some work must be repeated since we cannot save all of the prover's previous states.\footnote{Although backtracking is not necessary for propositional classical logic, cardT^A_P has been implemented for extensions to other logics.} The theorem prover resides on the smart card as a Java applet which can download a formula from a card reader and determine its theoremhood.

Due to space constraints we only allow formulae in Negated Normal Form using Reverse Polish Notation. The current prover is limited by statically defined restrictions on the length and complexity of the formula determined by the limited memory resources of the smart card [3]. Specifically, a formula can contain up to 26 distinct propositional variables, at most 20 disjunctions, and at most 20 nested conjunctions, with a total length of 126 symbols. Future cards with greater resources will be less restrictive.

Formulae are written to an EEPROM file; excess EEPROM space is used as virtual memory. The efficiency of accessing the formula is somewhat enhanced by using a smaller buffer in local memory as a small "window" into the formula.