An Optimised Implementation of Public-Key Cryptography for a Smart-Card Processor

Braden Jace Phillips
B.E. (Hons.), B.Sc.

A dissertation submitted in the
Department of Electrical and Electronic Engineering,
University of Adelaide, to meet the requirements for
the award of the degree of Doctor of Philosophy

7 August 2000

Abstract

Smart-cards for public-key cryptography are conventionally based around an 8-bit microprocessor core with a hardware co-processor to accelerate long wordlength arithmetic. While the co-processor can deliver excellent public-key performance, the 8-bit processor and available RAM limit other smart-card applications.

This thesis examines RSA public-key arithmetic with the objective of achieving adequate cryptographic performance without a large hardware co-processor. The chip area thus saved can be used to incorporate a more powerful 32-bit microprocessor as well as extra RAM. The resulting smart-card will better support desirable applications such as on-line media processing or biometric identification.

In the absence of a hardware co-processor, suitable cryptographic performance is achieved through the application of a variety of arithmetic techniques. Significantly, the arithmetic is optimised for average-case rather than worst-case delay. (The timing attack, which exploits variations in delay, is circumvented by a few simple blinding operations.)

Sliding window number representation has been applied to improve average case performance. Sliding windows are studied in detail and a new family of signed sliding window representations is developed. Improved algorithms for multiplication, Montgomery reduction and optimised squaring are presented that take advantage of the new signed representation.

The application of these techniques results in a smart-card design that is novel in many respects: dedicated arithmetic hardware is replaced by extra general-purpose RAM; constant worst-case timing is replaced by average-case execution and blinding; and high radix algorithms requiring a fast hardware multiplier are replaced with low radix algorithms that do not require a multiplier at all.

Contents