Chapter 1
About Cryptographic Engineering

Çetin Kaya Koç

1.1 Introduction

Cryptographic engineering is the name we have coined to refer to the theory and practice of engineering of cryptographic systems, i.e., encryption and decryption engines, digital signature and authentication hardware and software systems, key generation, distribution, and management systems, and random number generators. A cryptographic engineer designs, implements, tests, and validates cryptographic systems. She is also interested in cryptanalyzing them for the purpose of checking their robustness and their strength against attacks, and also building countermeasures in them in order to thwart such attacks by reducing their probability of success.

This is a subject barely taught in our undergraduate and graduate schools. Most courses in cryptography deal with theory, generally introducing mathematically expressed algorithms without showing (or knowing) how they are realized in actual software or hardware. As expected, the devil is in the details: The fastest and most practical implementation of the RSA algorithm requires the implementation of Montgomery multiplication. However, the last step in this algorithm (the so-called final subtraction) yields information which allows an attacker capable of observing, recording, and analyzing the timings of the process to learn some of the private bits. One cannot deduce this information by looking at a mathematical description of the RSA algorithm found in a textbook.

Cryptographic engineering material is scattered among many journal and conference papers, and the practitioners are too busy to write books. A group of us got together in Lausanne, Switzerland, in 2002, and began to teach short courses to engineers and researchers from industry and academia. The idea of putting our course notes into a book was born there and then.

City University of Istanbul & University of California Santa Barbara
e-mail: koc@cryptocode.net

Ç.K. Koç (ed.) *Cryptographic Engineering*, DOI 10.1007/978-0-387-71817-0_1,

© Springer Science+Business Media, LLC 2009

Cryptographic engineering is a fast-moving field. Every year in conferences such as the CHES (Cryptographic Hardware and Embedded Systems) Workshop, new innovative hardware and software realizations of cryptographic algorithms are introduced or new attacks to cryptanalyze these actual hardware and systems are proposed. This explains the unwillingness of researchers in cryptographic engineering to write books; we are more interested in designing new cryptographic systems or breaking the systems designed by our colleagues!

However, people who are new to this exciting field need good introductions. Engineers from industry and students from our colleges and graduate schools can use this book as a first step to cryptographic engineering.

1.2 Chapter Contents

This book has 18 chapters. It can be divided into 4 parts; however, the sections are intimately interconnected and there is a logical construction of the sections starting from the first chapter. There are also chapters which can belong to more than one part, as one might expect.

Chapters 2, 3, and 4 constitute the *first part* of the book. These chapters investigate and uncover the roles of random numbers in cryptography, and propose evaluation methods and practical designs for random number generators. Random numbers are used in other sciences; for example, the so-called Monte Carlo methods use random numbers to simulate physical or mathematical systems. In cryptography, random numbers provide the uncertainty and unpredictability upon which we build the secrecy of our cryptographic keys. For us, their most important property is requirement R2 (see, Chapter 2) which says that the full knowledge of a current bit does not help us to guess its past or future companions better than 50% chance. Chapter 2 examines the general definitions, requirements, and classifications of random numbers while Chapter 3 proposes an evaluation criteria for true random number generators (TRNGs). The ideas behind Chapter 3 produced the world's first evaluation methodology for TRNGs, called AIS.

Chapter 4, on the other hand, proposes a few practical TRNG designs suitable for implementation using ASIC and reconfigurable logic blocks. There is no doubt that, as we improve our understanding of the evaluation of TRNGs, more practical (low power, small circuit area, etc.) TRNG designs will be produced. I believe we are just entering this exciting field of TRNG designs, which requires collaboration by analog and digital circuit designers and cryptographers.

The *second part* of the book (Chapters 5–9) concentrates on implementation (i.e., hardware and software realizations) of public-key cryptographic systems, such as RSA, Diffie-Hellman, and elliptic curve cryptography, and their underlying arithmetic which includes large-integer arithmetic, arithmetic in prime fields and binary extension fields. Chapter 5 gives a general introduction to finite field arithmetic and describes the basic algorithms. Chapter 6 introduces the so-called unified arithmetic (which is also called dual-field arithmetic). The unified arithmetic allows one to

design a single hardware unit with negligible additional cost that performs arithmetic in both $GF(p)$ and $GF(2^k)$.

Chapter 7 introduces a new and compelling research area: the use of discrete Fourier transforms over finite rings in order to design parallel functional units for modular arithmetic. While the use of Fourier transforms to perform fast multiplication is well known, this chapter proposes the first spectral algorithm for modular multiplication.

Chapter 8 provides a high-level, mathematical view of elliptic and hyperelliptic curve arithmetic; it is also a good introduction to vulnerabilities of and attacks on elliptic and hyperelliptic curve cryptography. Finally, Chapter 9 provides a detailed account of instruction set architectures for cryptography, for both secret-key and public-key cryptographic algorithms. Chapter 8 provides a smooth transition from public-key cryptography to secret-key cryptography, a topic which we deal with in the subsequent part of the book.

The *third part* of the book studies implementation aspects of secret-key cryptographic algorithms, which are in Chapters 10, 11, and 12. The emphasis of these chapters is that they concenrate on hardware realizations of secret-key ciphers and their simple (ECB, CBS) and advanced (CCM) modes of operations. Chapter 10 covers both ASIC and FPGA realizations, while Chapter 11 particularly deals with FPGA implementations, exploiting logic structures more efficiently. Chapter 12, on the other hand, is a good summary on modes of operation, with special concentration on modern modes. The most important mode seems to be the CCM mode, which is an authenticated encryption mode used particulary in wireless communication protocols.

The final and *fourth part* of the book is the longest part (Chapters 13–18), and deals with the important topics of side-channel cryptanalysis and countermeasures against such attacks. Chapter 13 gives a brief introduction to the side-channel analysis. It covers the basic principles of side-channel cryptanalysis and introduces simple countermeasures to prevent side-channel leakage.

Chapter 14 delves into more advanced topics and shows how only a fraction of the information obtained from a side-channel can be used to cryptanalyze a practical system. Chapter 15 explains a particular type of side-channel: electromagnetic emanations from physical systems can be collected and analyzed by an attacker in order to capture messages not intended for others to see.

Chapters 16 and 17 show how algorithmic properties can be modeled and utilized to guess the bits of private keys. Chapter 16 focuses on how Montgomery multiplication leaks information, while Chapter 17 introduces methods to make the job of the attacker infeasible by using randomized exponentiations.

Finally, Chapter 18 introduces microarchitectural side-channel attacks, which allow an attacker to obtain information about cryptographic key bits from a crypto-process running on a client or server computer, by sneaking an unprivileged spy process into the same processor. These attacks slightly differ from the classic side-channel attacks, and are shown to be quite effective. It is very likely that future processors will have to be designed with hardware countermeasures against these microarchitectural side-channel attacks.

1.3 Exercises and Projects

Whenever appropriate, a chapter ends with two sections for the purpose of checking the reader's understanding of the chapter's technical material and leading her into research by describing a few doable projects. If the book is used in a graduate-level course, the *exercises* can be given as homework assignments. On the other hand, the *projects* are suitable for small groups (1 or 2 individuals) to implement.

Acknowledgements I would like to express my gratitude to all authors of the chapters in the book. Without their dedication, this book would not have come into existence.

I would also like to thank the staff of Springer US, particularly, Jason Ward, Caitlin Womersley, and Katelyn Stanne for helping me through the steps to completing the book, and also being patient with me and my co-authors, as we struggled to create time for this book from our other duties.

I thank my dear friends Vlado Valence and Caroline Huber of Mead Education, the co-organizers of the EPFL lectures in cryptographic engineering. Above all, I am indebted to Gabor Temes, who encouraged us to organize these lectures in the first place.

I thank my former and current students for contributing to this book by co-authoring chapters, reading and correcting portions of the material, and being life-long collaborators. I would like to thank particularly Gökay Saldamlı, Serdar Erdem, Tuğrul Yanık, Erkay Savaş, and Sultan Selçuk for helping in the creation and correction of the manuscript.

Finally, I thank my family for their constant patience and love.

Copyrights and Permissions A great many of figures and tables, and much of the other material found in this book are from the proceedings of the following conferences and workshops: Cryptographic Hardware and Embedded Systems, Cryptography and Coding, Computational Science and Its Applications, Field-Programmable Logic and Applications, Information Security and Cryptology, Information and Communications Security, CT-RSA, ASIACRYPT, and INDOCRYPT.

The above conference proceedings are published by Springer in the Lecture Notes of Computer Science series. This material is printed in this book with the kind permission of Springer Science+Business Media.