Lecture Notes in Computer Science 870
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Jonathan S. Greenfield

Distributed Programming Paradigms with Cryptography Applications

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Author

Jonathan S. Greenfield
Distributed Computing Environments Group, M/S B272
Los Alamos National Laboratory
Los Alamos, New Mexico 87545, USA

CR Subject Classification (1991): D.1.3, D.2.m, E.3, G.3

ISBN 3-540-58496-X Springer-Verlag Berlin Heidelberg New York

CIP data applied for

Typesetting: Camera-ready by author
SPIN: 10479104 45/3140-543210 - Printed on acid-free paper

Foreword

Jonathan Greenfield's thesis work in distributed computing, presented in this monograph, is part of my research project on programming paradigms for parallel scientific computing at Syracuse University. The starting point of this research is the definition of a *programming paradigm* as "a class of algorithms that solve different problems but have the same control structure." Using this concept, two or more parallel programs can be developed simultaneously.

First, you write a *parallel generic program* that implements the common control structure. The generic program includes a few unspecified data types and procedures that vary from one application to another. A *parallel application program* is obtained by replacing these types and procedures with the corresponding ones from a sequential program that solves a particular problem. The clear separation of the issues of parallelism from the details of applications leads to parallel programs that are easy to understand.

Jonathan Greenfield has convincingly demonstrated the power of this programming methodology in this monograph on *Distributed Programming Paradigms with Cryptography Applications*. The work makes several contributions to parallel programming and cryptography:

1. The monograph describes five *parallel algorithms* for the *RSA cryptosystem*: enciphering, deciphering, modulus factorization, and two methods for generating large primes.
2. The *parallel performance* of these algorithms was tested on a reconfigurable *multicomputer*—a Computing Surface with 48 transputer processors and distributed memory.
3. *RSA enciphering* and *deciphering* are very time-consuming operations that traditionally use a serial method (cipher block chaining) to break long messages into shorter blocks. Jonathan has invented a new method, called *message block chaining*, which permits fast enciphering and deciphering of message blocks on a parallel computer.
4. The RSA cryptosystem requires the generation of large primes. The primality of large random integers has traditionally been certified by the probabilistic Miller-Rabin algorithm. Jonathan has proposed a new algorithm that uses *deterministic primality testing* for a particular class of large random integers.

5. The five distributed programs are derived as instances of two *programming paradigms* invented by Jonathan: the *synchronized servers* and the *competing servers* paradigms. The latter is particularly interesting, since it demonstrates that non-determinism is necessary for efficient use of parallel computing in some user applications—which was an unexpected surprise for me.

The work is an appealing combination of the *theory* and *practice* of parallel computing. From the point of view of a computer user, the most important contribution is the new method of fast RSA enciphering and deciphering, which can be implemented in hardware. From the point of view of a computer scientist, it is an amazing feat to recognize five different aspects of the same application as instances of two simple programming paradigms for parallel computing. In addition, the monograph is well-written and easy to understand.

June 1994 Per Brinch Hansen

Acknowledgments

"When a true genius appears in this world you may know him by the sign that the dunces are all in confederacy against him."

Jonathan Swift

This monograph is a revised version of my doctoral thesis, prepared under the direction of Per Brinch Hansen at Syracuse University. I owe many thanks to Professor Brinch Hansen who introduced me to programming paradigms and to the problem of parallel primality testing. The latter sparked my interest in cryptography, while the former gave me a direction for my research.

Per Brinch Hansen has kindly granted permission to include revised versions of his occam program of June 9, 1992 for primality testing with multiple-length arithmetic.

In addition, I am grateful to Professor Brinch Hansen for the enormous amount of time he spent critiquing earlier versions of this monograph. His many useful suggestions have improved this work considerably. I must also thank Geoffrey Fox, David Gries, Salim Hariri, Carlos Hartmann, Nancy McCracken and Ernest Sibert for their helpful comments.

Finally, I thank my wife Georgette, my sons Channon and Alexzander, and especially my parents Judith and Stuart, for their support and toleration.

June 1994 Jonathan Greenfield

Table of Contents