

Lecture Notes in Computer Science

756

Edited by G. Goos and J. Hartmanis

Advisory Board: W. Brauer D. Gries J. Stoer



Josef Pieprzyk Babak Sadeghiyan

Design of Hashing Algorithms

Springer-Verlag

Berlin Heidelberg New York

London Paris Tokyo

Hong Kong Barcelona

Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
D-76131 Karlsruhe, Germany

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Authors

Josef Pieprzyk
Department of Computer Science, Centre for Computer Security Research
University of Wollongong
Wollongong, N.S.W. 2500, Australia

Babak Sadeghiyan
Computer Engineering Department, Amir-Kabir University of Technology
Tehran, Iran

CR Subject Classification (1991): E.3-4, G.2.1, F.2.2, D.4.6, C.2.0

ISBN 3-540-57500-6 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57500-6 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1993
Printed in Germany

Typesetting: Camera-ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

Historically, computer security is related to both cryptography and access control in operating systems. Cryptography, although mostly applied in the military and diplomacy, was used to protect communication channels and storage facilities (especially the backups). In the seventies there was a breakthrough in cryptography - the invention of public-key cryptography. It started in 1976 when Diffie and Hellman formulated their public-key distribution system and formally defined public-key cryptosystems. Two years later two practical implementations of public-key cryptosystems were published. One was designed by Rivest, Shamir, and Adleman (called the RSA system); the authors based the system on the two “difficult” numerical problems: discrete logarithm and factorization. The other invented by Merkle and Hellman was based on the knapsack problem, which is even “harder” than these used in the RSA system. Since that time cryptography, traditionally seen as the theory of encryption algorithms, has extended its scope enormously. Now it comprises many new areas, namely authentication, digital signature, hashing, secret sharing, design and verification of cryptographic protocols, zero knowledge protocols, quantum cryptography, etc.

This work presents recent developments in secure hashing algorithm design. The main part of the work was written when the authors were with the Department of Computer Science, University of New South Wales, Australian Defence Force Academy, and Babak Sadeghiyan was a PhD student working with Josef Pieprzyk as his supervisor.

Hashing is a process of creating a short digest (i.e. 64 bits) for a message of arbitrary length, for example 20 Mbytes. Hashing algorithms were first used for searching records in databases. These algorithms are designed to create a uniform distribution of collisions (two messages collide if their digests

are the same). In cryptographic applications, hashing algorithms should be “collision-free”, i.e. finding two different messages hashed to the same digest should be computationally intractable. Hashing algorithms are central for digital signature applications and are used for authentication without secrecy.

There have been many proposals for secure hash algorithms, and some of them have been in use for a while. However, many of them have proved insecure. One of the major reasons for this is the progress in technology. The failed effort of many researchers suggests that we should work on some guidelines or principles for the design of hash functions. This work presents some principles for the design of secure hash algorithms. Hash algorithms are classified based on whether they apply a block cipher as the underlying one-way function or not.

For a block-cipher-based hash scheme, if the underlying block cipher is secure against chosen plaintext/ciphertext attack, the hash scheme is secure against meet-in-the-middle attack. We develop structures, based on DES-like permutations and assuming the existence of pseudorandom function generators, which can be adopted both for the structure of block-cipher-based hash schemes and for the underlying block ciphers to be used in such schemes.

Non-block-cipher-based hash functions include a spectrum of many different proposals based on one-way functions from different branches of mathematics. So, in the book, generalized schemes for the construction of hash functions are developed, assuming the existence of a one-way permutation. The generalized constructions are improvements upon the Zheng, Matsumoto and Imai’s hashing scheme, based on the duality between pseudorandom bit generators and hash functions, but they incorporate strong one-way permutations. It is shown that we can build such strong permutations with a three-layer construction, in a theoretical approach. Two schemes for the construction of families of strong one-way permutations are also proposed.

Acknowledgement

We were very fortunate to receive help from many people throughout the time of this project. Firstly, we would like to express our deep gratitude to Professor Jennifer Seberry for her critical comments, helpful suggestions and encouragement. Also we would like to thank Professor Tsutomu Matsumoto and Dr Rei Safavi-Naini for their thoughtful criticism, suggestions and corrections. We also received helpful comments about the work from Dr Lawrence Brown, Professor Andrzej Gościnski, Dr Thomas Hardjono, Dr Xian-Mo Zhang and Dr Yuliang Zheng. We thank all our friends from the Department of Computer Science, University College, University of NSW, for their friendliness and everyday support. In particular our thanks go to Dr George Gerrity, Mr Per Hoff, Mr Jeff Howard, Dr Jadwiga Indulska, Mr Martin Jaatun, Mr Ken Miles, Mr Andy Quaine and Mr Wen Ung. Finally we would like to thank Mrs Nilay Genctruck for proof-reading the final manuscript.

This project was partially supported by the Australian Research Council grant number A49131885.

September 1993

Josef Pieprzyk

Babak Sadeghiyan

Contents

1	Introduction	1
1.1	Background and Aims	1
1.1.1	Introductory Comments	1
1.1.2	Discussion of Public-key and Private-key Cryptography	2
1.1.3	Digital Signature	5
1.1.4	RSA Cryptosystem and Digital Signature	9
1.1.5	Signature-Hashing Scheme	10
1.1.6	Other Applications of Hash Functions	13
1.2	Contents of the Book	14
2	Overview of Hash Functions	18
2.1	Introduction	18
2.2	Properties of Secure Hash Functions	19
2.3	Definitions	20
2.3.1	Strong and Weak Hash Functions	20
2.3.2	Message Authentication Codes and Manipulation De- tection Codes	22
2.3.3	Block-cipher-based and Non-block-cipher-based Hash Functions	23
2.4	Block-cipher-based Hash Functions	24

2.4.1	Rabin's Scheme	25
2.4.2	Cipher Block Chaining Scheme	26
2.4.3	CBC with Checksum Scheme	26
2.4.4	Combined Plaintext-Ciphertext Chaining Scheme . . .	27
2.4.5	Key Chaining Scheme	28
2.4.6	Winternitz' Key Chaining Schemes	29
2.4.7	Quisquater and Girault's 2n-bit Hash Function	30
2.4.8	Merkle's Scheme	31
2.4.9	N-hash Algorithm	32
2.4.10	MDC2 and MDC4	33
2.5	Non-block-cipher-based Hash Functions	34
2.5.1	Cipher Block Chaining with RSA	35
2.5.2	Schemes Based on Squaring	36
2.5.3	Schemes Based on Claw-Free Permutations	38
2.5.4	Schemes Based on the Knapsack Problem	39
2.5.5	Schemes Based on Cellular Automata	40
2.5.6	Software Hash Schemes	41
2.5.7	Matrix Hashing	43
2.5.8	Schnorr's FFT Hashing Scheme	44
2.6	Design Principles for Hash Functions	45
2.6.1	Serial Method	45
2.6.2	Parallel Method	46
2.7	Conclusions	46
3	Methods of Attack on Hash Functions	48
3.1	Introduction	48

3.2	General Attacks	49
3.3	Special Attacks	50
3.3.1	Meet-in-the-middle Attack	51
3.3.2	Generalized Meet-in-the-middle Attack	52
3.3.3	Correcting Block Attack	53
3.3.4	Attacks Depending on Algorithm Weaknesses	53
3.3.5	Differential Cryptanalysis	54
3.4	Conclusions	54
4	Pseudorandomness	56
4.1	Introduction	56
4.2	Notation	58
4.3	Indistinguishability	58
4.4	Pseudorandom Bit Generators	60
4.5	Pseudorandom Function Generators	62
4.6	Pseudorandom Permutation Generators	66
4.6.1	Construction	66
4.6.2	Improvements and Implications	69
4.6.3	Security	72
4.7	Conclusions	76
5	Construction of Super-Pseudorandom Permutations	77
5.1	Introduction	77
5.2	Super-Pseudorandom Permutations	78
5.3	Necessary and Sufficient Conditions	79
5.4	Super-Pseudorandomness in Generalized DES-like Permutations	92
5.4.1	Feistel-Type Transformations	93

5.4.2	Super-Pseudorandomness of Type-1 Transformations . . .	96
5.5	Conclusions and Open Problems	103
6	A Sound Structure	105
6.1	Introduction	105
6.2	Preliminaries	106
6.3	Perfect Randomizers	112
6.4	A Construction for Super-Pseudorandom Permutation Generators	116
6.4.1	Super-Pseudorandomness of $\psi(h, 1, f, h, 1, f)$	117
6.4.2	Super-Pseudorandomness of $\psi(f^2, 1, f, f^2, 1, f)$	124
6.5	Conclusions and Open Problems	130
7	A Construction for One Way Hash Functions and Pseudorandom Bit Generators	132
7.1	Introduction	132
7.2	Notation	134
7.3	Preliminaries	135
7.4	Theoretic Constructions	137
7.4.1	Naor and Yung's Scheme	138
7.4.2	Zheng, Matsumoto and Imai's First Scheme	138
7.4.3	De Santis and Yung's Schemes	139
7.4.4	Rompel's Scheme	140
7.5	Hard Bits and Pseudorandom Bit Generation	140
7.6	A Strong One-Way Permutation	146
7.7	UOWHF Construction and PBG	151
7.7.1	UOWHF Based on the Strong One-way Permutation	152

7.7.2	Parameterization	153
7.7.3	Compressing Arbitrary Length Messages	154
7.8	A Single construction for UOWHF and PBG	155
7.9	Conclusions and Extensions	156
8	How to Construct a Family of Strong One Way Permutations	157
8.1	Introduction	157
8.2	Preliminary Comments	159
8.3	Strong One Way Permutations	162
8.3.1	A Scheme for the Construction of Strong Permutations	164
8.3.2	A Three-layer Construction for Strong Permutations .	166
8.4	Conclusions	168
9	Conclusions	170
9.1	Summary	170
9.2	Limitations and Assumptions of the Results	174
9.3	Prospects for Further Research	177
	Bibliography	179
	Index	191

List of Symbols

C	Ciphertext
M	Message or a plaintext
K	Key
Σ	Alphabet
Σ	Summation
\subset	Subset
\rightarrow	Mapping
\circ	Composition of functions
\in	Set membership
\equiv	Congruence
$!$	Factorial
$ $	Such that (set notation)
\oplus	Exclusive-or (of Booleans)
\vee	Or (of Booleans)
\wedge	And (of Booleans)
\parallel	Concatenation
$O(f)$	Big-Oh of the function f
\cup	Union
$\lceil x \rceil$	Smallest integer greater than x
$\lfloor x \rfloor$	Greatest integer smaller than x
N	The set of natural numbers
Z_n	The set of integers modulo n
$ x $	Absolute value of the number x
$\#S$	Number of elements in the set S
\log_n	Logarithm to the base n