LNCS 3796

Nigel P. Smart (Ed.)

Cryptography and Coding

10th IMA International Conference
Cirencester, UK, December 2005
Proceedings

Springer

Nigel P. Smart (Ed.)

Cryptography and Coding

10th IMA International Conference
Cirencester, UK, December 19-21, 2005
Proceedings

Springer

Volume Editor

Nigel P. Smart
University of Bristol
Department of Computer Science
Woodland Road, Bristol, BS8 1UB, UK
E-mail: nigel@cs.bris.ac.uk

Library of Congress Control Number: 2005936359

CR Subject Classification (1998): E.3-4, G.2.1, C.2, J.1

ISSN 0302-9743
ISBN-10 3-540-30276-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-30276-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11586821 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3796

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3706

please contact your bookseller or Springer

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Subseries LNAI).

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Subseries LNAI).

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Subseries LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Subseries LNAI).

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

Vol. 3770: J. Akoka, S.W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. van den Heuvel, M. Kolp, J. Trujillo, C. Kop, H.C. Mayr (Eds.), Perspectives in Conceptual Modeling. XXII, 476 pages. 2005.

Vol. 3768: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part II. XXVIII, 1088 pages. 2005.

Vol. 3767: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part I. XXVIII, 1022 pages. 2005.

Vol. 3766: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 231 pages. 2005.

Vol. 3765: Y. Liu, T. Jiang, C. Zhang (Eds.), Computer Vision for Biomedical Image Applications. X, 563 pages. 2005.

Vol. 3764: S. Tixeuil, T. Herman (Eds.), Self-Stabilizing Systems. VIII, 229 pages. 2005.

Vol. 3762: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. XXXI, 1228 pages. 2005.

Vol. 3761: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part II. XXVII, 653 pages. 2005.

Vol. 3760: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part I. XXVII, 921 pages. 2005.

Vol. 3759: G. Chen, Y. Pan, M. Guo, J. Lu (Eds.), Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. XIII, 669 pages. 2005.

Vol. 3758: Y. Pan, D.-x. Chen, M. Guo, J. Cao, J.J. Dongarra (Eds.), Parallel and Distributed Processing and Applications. XXIII, 1162 pages. 2005.

Vol. 3757: A. Rangarajan, B. Vemuri, A.L. Yuille (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. XII, 666 pages. 2005.

Vol. 3756: J. Cao, W. Nejdl, M. Xu (Eds.), Advanced Parallel Processing Technologies. XIV, 526 pages. 2005.

Vol. 3754: J. Dalmau Royo, G. Hasegawa (Eds.), Management of Multimedia Networks and Services. XII, 384 pages. 2005.

Vol. 3753: O.F. Olsen, L.M.J. Florack, A. Kuijper (Eds.), Deep Structure, Singularities, and Computer Vision. X, 259 pages. 2005.

Vol. 3752: N. Paragios, O. Faugeras, T. Chan, C. Schnörr (Eds.), Variational, Geometric, and Level Set Methods in Computer Vision. XI, 369 pages. 2005.

Vol. 3751: T. Magedanz, E.R. M. Madeira, P. Dini (Eds.), Operations and Management in IP-Based Networks. X, 213 pages. 2005.

Vol. 3750: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part II. XL, 1018 pages. 2005.

Vol. 3749: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part I. XXXIX, 942 pages. 2005.

Vol. 3748: A. Hartman, D. Kreische (Eds.), Model Driven Architecture – Foundations and Applications. IX, 349 pages. 2005.

Vol. 3747: C.A. Maziero, J.G. Silva, A.M.S. Andrade, F.M.d. Assis Silva (Eds.), Dependable Computing. XV, 267 pages. 2005.

Vol. 3746: P. Bozanis, E.N. Houstis (Eds.), Advances in Informatics. XIX, 879 pages. 2005.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005. (Subseries LNBI).

Vol. 3744: T. Magedanz, A. Karmouch, S. Pierre, I. Venieris (Eds.), Mobility Aware Technologies and Applications. XIV, 418 pages. 2005.

Vol. 3740: T. Srikanthan, J. Xue, C.-H. Chang (Eds.), Advances in Computer Systems Architecture. XVII, 833 pages. 2005.

Vol. 3739: W. Fan, Z.-h. Wu, J. Yang (Eds.), Advances in Web-Age Information Management. XXIV, 930 pages. 2005.

Vol. 3738: V.R. Syrotiuk, E. Chávez (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XI, 360 pages. 2005.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005. (Subseries LNAI).

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005. (Subseries LNAI).

Vol. 3733: P. Yolum, T. Güngör, F. Gürgen, C. Özturan (Eds.), Computer and Information Sciences - ISCIS 2005. XXI, 973 pages. 2005.

Vol. 3731: F. Wang (Ed.), Formal Techniques for Networked and Distributed Systems - FORTE 2005. XII, 558 pages. 2005.

Vol. 3729: Y. Gil, E. Motta, V. R. Benjamins, M.A. Musen (Eds.), The Semantic Web – ISWC 2005. XXIII, 1073 pages. 2005.

Vol. 3728: V. Paliouras, J. Vounckx, D. Verkest (Eds.), Integrated Circuit and System Design. XV, 753 pages. 2005.

Vol. 3726: L.T. Yang, O.F. Rana, B. Di Martino, J.J. Dongarra (Eds.), High Performance Computing and Communications. XXVI, 1116 pages. 2005.

Vol. 3725: D. Borrione, W. Paul (Eds.), Correct Hardware Design and Verification Methods. XII, 412 pages. 2005.

Vol. 3724: P. Fraigniaud (Ed.), Distributed Computing. XIV, 520 pages. 2005.

Vol. 3723: W. Zhao, S. Gong, X. Tang (Eds.), Analysis and Modelling of Faces and Gestures. XI, 4234 pages. 2005.

Vol. 3722: D. Van Hung, M. Wirsing (Eds.), Theoretical Aspects of Computing – ICTAC 2005. XIV, 614 pages. 2005.

Vol. 3721: A.M. Jorge, L. Torgo, P.B. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005. (Subseries LNAI).

Vol. 3720: J. Gama, R. Camacho, P.B. Brazdil, A.M. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005. (Subseries LNAI).

Vol. 3719: M. Hobbs, A.M. Goscinski, W. Zhou (Eds.), Distributed and Parallel Computing. XI, 448 pages. 2005.

Vol. 3718: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XII, 502 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005. (Subseries LNAI).

Vol. 3716: L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, Ó. Pastor (Eds.), Conceptual Modeling – ER 2005. XVI, 498 pages. 2005.

Vol. 3715: E. Dawson, S. Vaudenay (Eds.), Progress in Cryptology – Mycrypt 2005. XI, 329 pages. 2005.

Vol. 3714: H. Obbink, K. Pohl (Eds.), Software Product Lines. XIII, 235 pages. 2005.

Vol. 3713: L.C. Briand, C. Williams (Eds.), Model Driven Engineering Languages and Systems. XV, 722 pages. 2005.

Vol. 3712: R. Reussner, J. Mayer, J.A. Stafford, S. Overhage, S. Becker, P.J. Schroeder (Eds.), Quality of Software Architectures and Software Quality. XIII, 289 pages. 2005.

Vol. 3711: F. Kishino, Y. Kitamura, H. Kato, N. Nagata (Eds.), Entertainment Computing - ICEC 2005. XXIV, 540 pages. 2005.

Vol. 3710: M. Barni, I. Cox, T. Kalker, H.J. Kim (Eds.), Digital Watermarking. XII, 485 pages. 2005.

Vol. 3709: P. van Beek (Ed.), Principles and Practice of Constraint Programming - CP 2005. XX, 887 pages. 2005.

Vol. 3708: J. Blanc-Talon, W. Philips, D.C. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems. XXII, 725 pages. 2005.

Vol. 3707: D.A. Peled, Y.-K. Tsay (Eds.), Automated Technology for Verification and Analysis. XII, 506 pages. 2005.

¥566.40元

Preface

The 10th in the series of IMA Conferences on Cryptography and Coding was held at the Royal Agricultural College, Cirencester, during 19–21 December 2005. As usual, the venue provided a relaxed and informal atmosphere for attendees to discuss work and listen to the collection of talks.

The program consisted of four invited talks and 26 contributed talks. The invited talks where given by Tuvi Etzion, Ueli Maurer, Alfred Menezes and Amin Shokrollahi, and three of these invited talks appear as papers in this volume. Special thanks must go to these four speakers as they helped to set the tone, by covering all the areas the meeting aimed to cover, from cryptography through to coding. In addition the best speakers are often the hardest to persuade to come to a meeting, as they are usually the most busy. We therefore feel privileged to have had a meeting with four such distinguished speakers.

The contributed talks were selected from 94 submissions. This is nearly twice the number of submissions for the previous meeting in 2003. This is an indication of the strength of the subject and the interest in the IMA series of meetings as a venue to present new work. The contributed talks ranged over a wide number of areas, including information theory, coding theory, number theory and asymmetric and symmetric cryptography. Subtopics included a number of current "hot topics," such as algebraic cryptanalysis and cryptographic systems based on bilinear pairings.

Assembling the conference program and these proceedings required the help of a large number of individuals. I would like to thank them all here.

Firstly, thanks must go to the Program Committee and their subreferees who aided in evaluating the contributed papers and coming to a difficult decision as to what to include and what to exclude. We had to reject a number of papers simply due to lack of space.

For the first time the IMA Program Committee used the WebReview software from K.U. Leuven. This is an excellent program, which greatly helped myself as Program Chair in collating and mediating the referee reports. Thanks must go to the authors and maintainers of this program. Particular thanks must also go to Richard Noad, one of my Research Assistants at Bristol, who dealt with all the technical issues with mounting the servers and generally acted as a right-hand-man throughout the process.

Thanks must also go to the authors of the submitted papers, and in particular to those whose papers were accepted. The authors of the accepted papers cooperated in compiling this volume, often meeting very tight deadlines imposed by the publication schedule. Thanks must also go to the staff of Springer, in particular Alfred Hofmann who helped in a large number of ways.

Valuable sponsorship of the meeting was provided by Hewlett-Packard Laboratories and Vodafone. We thank them for their contributions.

Finally, thanks must go to the staff of the IMA, in particular Pamela Bye and Lucy Nye, who dealt with all the day-to-day issues and allowed the Program Committee to concentrate on the science. A conference such as this could not take place without their help.

September 2005

Nigel Smart
Program Chair

Cryptography and Coding 2005

December 19–21, 2005, Cirencester, United Kingdom

Sponsored by the
The Institute of Mathematics and its Applications (IMA)
in cooperation with
Hewlett-Packard Laboratories and *Vodafone Ltd.*

Program Chair

Nigel Smart	University of Bristol

Program Committee

Steve Babbage	Vodafone Group Services Ltd.
Bahram Honary	University of Lancaster
Steven Galbraith	Royal Holloway, University of London
Chris Mitchell	Royal Holloway, University of London
David Naccache	École Normale Supérieure
Matthew Parker	University of Bergen
Kenny Paterson	Royal Holloway, University of London
Ana Salagean	Loughborough University
Frederik Vercauteren	K.U. Leuven
Mike Walker	Vodafone Group Services Ltd.
Gilles Zemor	ENST, Paris

External Referees

Claude Barral
Eric Brier
Alister Burr
Julien Cathalo
Benoit Chevallier-Mames
Carlos Cid
Mathieu Ciet
Gerard Cohen
Nicolas Courtois
Alex Dent
Jean-François Dhem
Eran Edirisinghe
Nicolas Gresset
Helena Handschuh
Marc Joye
Caroline Kudla
Arjen Lenstra
John Malone-Lee
Keith Martin
Philippe Martins
Jean Monnerat
David M'Raihi
Gregory Neven
Katie O'Brien
Dan Page
Havard Raadum
Vincent Rijmen
Jasper Scholten
Martijn Stam
Emmanuel Thomé
Claire Whelan
Andreas Winter
Christopher Wolf

Table of Contents

Invited Papers

Abstract Models of Computation in Cryptography
U. Maurer 1

Pairing-Based Cryptography at High Security Levels
N. Koblitz, A. Menezes 13

Improved Decoding of Interleaved AG Codes
A. Brown, L. Minder, A. Shokrollahi 37

Coding Theory

Performance Improvement of Turbo Code Based on the Extrinsic Information Transition Characteristics
W.T. Kim, S.H. Kang, E.K. Joo 47

A Trellis-Based Bound on $(2,1)$-Separating Codes
H.G. Schaathun, G.D. Cohen 59

Tessellation Based Multiple Description Coding
C. Cai, J. Chen 68

Exploiting Coding Theory for Collision Attacks on SHA-1
N. Pramstaller, C. Rechberger, V. Rijmen 78

Signatures and Signcryption

Hash Based Digital Signature Schemes
C. Dods, N.P. Smart, M. Stam 96

A General Construction for Simultaneous Signing and Encrypting
J. Malone-Lee 116

Non-interactive Designated Verifier Proofs and Undeniable Signatures
C. Kudla, K.G. Paterson 136

Symmetric Cryptography

Partial Key Recovery Attacks on XCBC, TMAC and OMAC
C.J. Mitchell 155

Domain Expansion of MACs: Alternative Uses of the FIL-MAC
U. Maurer, J. Sjödin 168

Normality of Vectorial Functions
A. Braeken, C. Wolf, B. Preneel 186

Related-Key Differential Attacks on Cobra-H64 and Cobra-H128
C. Lee, J. Kim, J. Sung, S. Hong, S. Lee, D. Moon 201

Side Channels

The Physically Observable Security of Signature Schemes
A.W. Dent, J. Malone-Lee 220

On the Automatic Construction of Indistinguishable Operations
M. Barbosa, D. Page 233

Efficient Countermeasures for Thwarting the SCA Attacks on the Frobenius Based Methods
M. Hedabou 248

Algebraic Cryptanalysis

Complexity Estimates for the F_4 Attack on the Perturbed Matsumoto-Imai Cryptosystem
J. Ding, J.E. Gower, D. Schmidt, C. Wolf, Z. Yin 262

An Algebraic Framework for Cipher Embeddings
C. Cid, S. Murphy, M.J.B. Robshaw 278

Probabilistic Algebraic Attacks
A. Braeken, B. Preneel 290

Information Theoretic Applications

Unconditionally Secure Information Authentication in Presence of Erasures
G. Jakimoski 304

Generalized Strong Extractors and Deterministic Privacy Amplification
R. König, U. Maurer 322

On Threshold Self-healing Key Distribution Schemes
G. Sáez 340

Number Theoretic Foundations

Concrete Security of the Blum-Blum-Shub Pseudorandom Generator
A. Sidorenko, B. Schoenmakers 355

The Equivalence Between the DHP and DLP for Elliptic Curves Used in Practical Applications, Revisited
K. Bentahar 376

Pairings on Elliptic Curves over Finite Commutative Rings
S.D. Galbraith, J.F. McKee 392

Public Key and ID-Based Encryption Schemes

A Key Encapsulation Mechanism for NTRU
M. Stam 410

Efficient Identity-Based Key Encapsulation to Multiple Parties
M. Barbosa, P. Farshim 428

Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme
L. Chen, Z. Cheng 442

Author Index 461

Abstract Models of Computation in Cryptography

Ueli Maurer*

Department of Computer Science,
ETH Zurich, CH-8092 Zurich, Switzerland
maurer@inf.ethz.ch

Abstract. Computational security proofs in cryptography, without unproven intractability assumptions, exist today only if one restricts the computational model. For example, one can prove a lower bound on the complexity of computing discrete logarithms in a cyclic group if one considers only generic algorithms which can not exploit the properties of the representation of the group elements.

We propose an abstract model of computation which allows to capture such reasonable restrictions on the power of algorithms. The algorithm interacts with a black-box with hidden internal state variables which allows to perform a certain set of operations on the internal state variables, and which provides output only by allowing to check whether some state variables satisfy certain relations. For example, generic algorithms correspond to the special case where only the equality relation, and possibly also an abstract total order relation, can be tested.

We consider several instantiation of the model and different types of computational problems and prove a few known and new lower bounds for computational problems of interest in cryptography, for example that computing discrete logarithms is generically hard even if an oracle for the decisional Diffie-Hellman problem and/or other low degree relations were available.

1 Introduction and Motivation

1.1 Restricted Models of Computation

Proving the security of a certain cryptographic system means to prove a lower bound on the hardness of a certain computational problem. Unfortunately, for general models of computation no useful lower bound proofs are known, and it is therefore interesting to investigate reasonably restricted models of computation if one can prove relevant lower bounds for them.

In a restricted model one assumes that only certain types of operations are allowed. For example, in the monotone circuit model one assumes that the circuit performing the computation consists only of AND-gates and OR-gates, excluding NOT-gates. Such a restriction is uninteresting from a cryptographic viewpoint since it is obvious that an adversary can of course perform NOT-operations.

* Supported in part by the Swiss National Science Foundation.

N.P. Smart (Ed.): Cryptography and Coding 2005, LNCS 3796, pp. 1–12, 2005.

Nevertheless, some restricted models are indeed meaningful in cryptography, for example the generic model which assumes that the properties of the representation of the elements of the algebraic structure (e.g. a group) under consideration can not be exploited. In view of the fact that for some problems, for example the discrete logarithm problem on general elliptic curves, exploiting the representation is not known to be of any help and hence generic algorithms are the best known, such an assumption is reasonable from a practical viewpoint.[1]

The purpose of this paper is to provide a simple framework for such restricted models of computation and to prove some lower bounds. Generic algorithms are the simplest case. Some of the presented results are interpretations and generalizations of previous results, for instance of [10] and [4].

1.2 Generic Algorithms and Computing Discrete Logarithms

In order to compute with the elements of a set S (e.g. a group), one must represent the elements as bitstrings (without loss of generality). A representation is a bijective mapping from S to the set of bitstrings. A generic algorithm works independently of the representation. The term generic means that one can not exploit non-trivial properties of the representation of the elements, except for two generic properties that any representation has. First, one can test equality of elements, and second one can impose a total order relation $\preceq$ on any representation, for example the usual lexicographic order relation on the set of bitstrings. However, one can generally not assume that the representation is dense or satisfies any regularity or randomness condition.

In order to motivate the model to be introduced, we briefly discuss generic algorithms for computing discrete algorithms in a cyclic group G. A cyclic group G of order n, generated by a generator g, is isomorphic to the additive group $\mathbf{Z}_n$. A generic algorithm for computing the discrete logarithm (DL) x of an element $b = g^x$ to the base g in G can be modeled as follows. The algorithm is given a black-box which contains x. It can also input constants into the box[2] and add values in the box. The only information reported back from the box is when an equality (collision) between two computed elements occurs. The algorithm's task is to extract x by provoking collisions and computing x from the collision pattern. The order relation allows to establish ordered tables of the generated values and thus reduces the number of equality tests required, but it does not allow to reduce the number of computed values and is ignored in most of the following.

If one is interested in proving a lower bound on the number of operations for any generic algorithm, then one can consider the simpler objective of only provoking a *single* collision and that all equalities of elements are reported for free. Since only additions and the insertion of constants are allowed, every value computed in the box is of the form $ax + b$ (modulo n) for known values a and

[1] In contrast, for computing discrete logarithms in Z_p^* for a prime p, quite sophisticated algorithms are known (e.g. index calculus) which exploit that the elements are integers that can be factored into primes.

[2] One can also assume that the box contains only 1 and x initially and constants must be computed explicitly from 1 by an addition-and-doubling algorithm.

b. For uniform x the probability that two such values $ax+b$ and $a'x+b'$ collide is easily seen to be at most $1/q$, where q is the largest prime factor of n. Hence the total probability of provoking a collision is upper bounded by $\binom{k}{2}/q$ and therefore the running time of any algorithm with constant success probability is at least $O(\sqrt{q})$.

The simplest non-trivial generic DL algorithm is the so-called baby-step giant-step algorithm with complexity $O(\sqrt{n}\log n)$. It need not know the group order n, an upper bound on n suffices, and it is the best known algorithm when the group order is unknown. The Pohlig-Hellman algorithm [7] is also generic and a bit more sophisticated. It makes use of the prime factorization of n and has complexity $O(\sqrt{q}\log q)$, which is essentially optimal.

1.3 Discussion and Generalization of the Model

This view of a generic algorithm appears to be simpler than the model usually considered in the literature, introduced by Shoup [10], where one assumes that access to group elements is via a randomly selected representation. This complicates the random experiment in which the algorithm's success probability is to be analyzed. Also, in a realistic setting one has no guarantee that the representation corresponds in any way to a random mapping.

As a generalization of the described approach, one can also model that one can exploit certain additional information from the representation of the elements, for instance that one can test certain relations efficiently. As an example, one can imagine that one can efficiently test for any three elements x, y and z whether $xy = z$, which corresponds to assuming the availability of a decisional Diffie-Hellman (DDH) oracle. For this setting one can still prove an $O(\sqrt[3]{q})$ lower bound for the discrete logarithm problem.

2 An Abstract Model of Computation

2.1 The Model

We consider an abstract model of computation characterized by a black-box **B** which can store values from a certain set S (e.g. a group) in internal state variables $V_1, V_2, \ldots, V_m$. The storage capacity m can be finite or unbounded.

The initial state consists of the values of $V^d := [V_1, \ldots, V_d]$ (for some $d < m$, usually d is $1, 2$, or 3), which are set according to some probability distribution P_{V^d} (e.g. the uniform distribution).

The black-box **B** allows two types of operations, computation operations on internal state variables and queries about the internal state. No other interaction with **B** is possible.[3] We give a more formal description of these operations:

[3] This model captures two aspects of a restricted model of computation. The computation operations describe the types of computations the black-box can perform, and the state queries allow to model precisely how limited information about the representation of elements in S can be used. A quantum computer is another type of device where only partial information about the state can be obtained, but it could not be captured in our model.

- **Computation operations.** For a set Π of operations on S of some arities (nullary, unary, binary, or higher arity), a computation operations consist of selecting an operation $f \in \Pi$ (say t-ary) as well as the indices $i_1, \ldots, i_{t+1} \leq m$ of $t+1$ state variables.[4] **B** computes $f(V_{i_1}, \ldots, V_{i_t})$ and stores the result in $V_{i_{t+1}}$.[5]
- **Queries.** For a set Σ of relations (of some arities) on S, a query consist of selecting a relation $\rho \in \Sigma$ (say t-ary) as well as the indices $i_1, \ldots, i_t \leq m$ of t state variables. The query is replied by $\rho(V_{i_1}, \ldots, V_{i_t})$.

A black-box **B** is thus characterized by S, Π, Σ, m, and d. As mentioned above, one can include an abstract total order relation $\preceq$.

2.2 Three Types of Problems

We consider three types of problems for such black-boxes, where the problem instance is encoded into the initial state of the device.

- **Extraction:** Extract the initial value x of V_1 (where $d = 1$).[6]
- **Computation:** Compute a function $f : S^d \to S$ of the initial state within **B**, i.e., one must achieve $V_i = f(x_1, \ldots, x_d)$ for some (known) i, where $x_1, \ldots, x_d$ are the initial values of the state variables $V_1, \ldots, V_d$.
- **Distinction:** Distinguish two black-boxes **B** and **B**$'$ of the same type with different distributions of the initial state V^d.

An algorithm for solving one of these problems is typically assumed to be computationally unbounded, but it is restricted in terms of the number k of interactions with the black-box it can perform. The memory capacity m can also be seen as a parameter of the algorithm.

One is often only interested in the computation queries, especially when proving lower bounds, and can then assume that, for every (say t-ary) relation $\rho \in \Sigma$, **B** provides all lists $(i_1, \ldots, i_t)$ such that $\rho(u_{i_1}, \ldots, u_{i_t})$ for free. We prove lower bounds in this model.

The success probability of an algorithm is taken over the choice of the initial state $V_1, \ldots, V_d$ and the (possible) randomness of the algorithm. The advantage of a distinguisher is defined as usual.

3 Concrete Settings

In this section we consider a few concrete instantiations of the model which are of interest in cryptography.

[4] This information is the input to **B**.

[5] A special case are constant functions, i.e., the operation of setting an internal state variable V_i to a particular value $c \in S$. If m is unbounded, then one can assume without loss of generality that each new result is stored in the next free state variable.

[6] More generally, one could consider the problem of extracting more general information about the initial state. This can be formalized by a function $g : S^d \to \mathcal{Q}$ for some $\mathcal{Q}$, where the task is to guess $g(V_1, \ldots, V_d)$.

3.1 Notation

We introduce some notation. Let $\mathcal{C}$ denote the set of constant (nullary) operations, which correspond to inserting a constant into the black-box. For a ring S, let $\mathcal{L}$ denote the set of linear functions (of the form $a_1V_1 + \cdots + a_dV_d$) on the initial state V^d. For a multiplicatively written operation (e.g. of a ring) S, let *square* denote the binary relation $\{(x,y) : y = x^2\}$, let *power*(e) denote $\{(x,y) : y = x^e\}$, and let *prod* denote the ternary relation $\{(x,y,z) : z = xy\}$.

For a given set Π of operations, let $\overline{\Pi}$ be the set of functions on the initial state that can be computed using operations in Π.

3.2 Extraction Problems with Constant and Unary Operations

The simplest case of an extraction problem to consider is when $\Pi = \mathcal{C}$ and $\Sigma = \{=\}$, i.e., one can only input constants and check equality.[7] It is trivial that the best strategy for the extraction problem is to randomly guess, and the success probability of any k-step algorithm is bounded by $k/|S|$, i.e.,. the complexity for achieving a constant success probability is $O(|S|)$. This bound holds independently of whether one counts equality checks or whether one assumes a total order $\preceq$ on S. This bound is trivially achievable with constant memory m.

If one would also allow to check a more general relation than equality (i.e., $\Sigma = \{=, \rho\}$ for some ρ), then better algorithms may exist. But the above upper bound generalizes easily to $kd/|S|$, where

$$d = \max_{u \in S} |\{v \in S : u\rho v \vee v\rho u\}|$$

is the maximal vertex degree of the relation graph. Note that $d = 1$ for the equality relation. If d is large, there can exist efficient algorithms. For example, if $\Sigma = \{=, \leq\}$ and S is totally ordered by the relation $\leq$, then one can use the binary search algorithm with running time $O(\log|S|)$, which is optimal.[8] It may be interesting to consider other relations.

We return to the case $\Sigma = \{=\}$ but now allow some unary operations.

Theorem 1. *Let $\star$ be a group operation on S, let $\Pi = \mathcal{C} \cup \{x \mapsto x \star a \mid a \in S\}$ consist of all constant functions and multiplications by constants, and let $\Sigma = \{=\}$. The success probability of every k-step algorithm for extraction is upper bounded by $\frac{1}{4}k^2/|S|$, and by $km/|S|$ if m is bounded.*

Proof. We use three simple general arguments which will be reused implicitly later. First, we assume that as soon as some collision occurs (more generally, some relation in Σ is satisfied for some state variables) in the black-box, the algorithm

[7] This corresponds to a card game where one has to find a particular card among n cards and the only allowed operation is to lift a card, one at a time.

[8] Note that the previously discussed order relation $\preceq$ can not be used to perform a binary search because it is not known explicitly, but only accessible through an oracle.