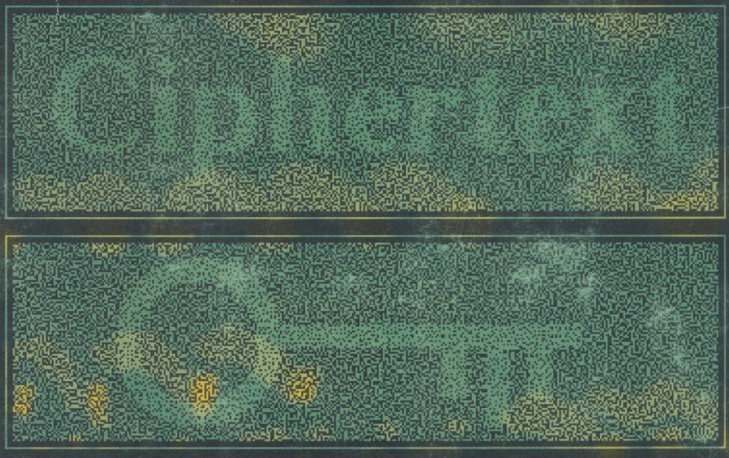
LNCS Tutorial

Ivan Damgård (Ed.)

Lectures on Data Security

Modern Cryptology in Theory and Practice



Springer

Ivan Damgård (Ed.)

Lectures on Data Security

Modern Cryptology in Theory and Practice

Springer

Cryptography

Volume Editor

Ivan Damgård
BRICS, University of Aarhus
Ny Munkegade, Building 540
DK-8000 Aarhus C, Denmark
E-mail: ivan@daimi.aau.dk

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Lectures on data security : modern cryptology in theory and practice / Ivan Damgård (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; 1561)
ISBN 3-540-65757-6

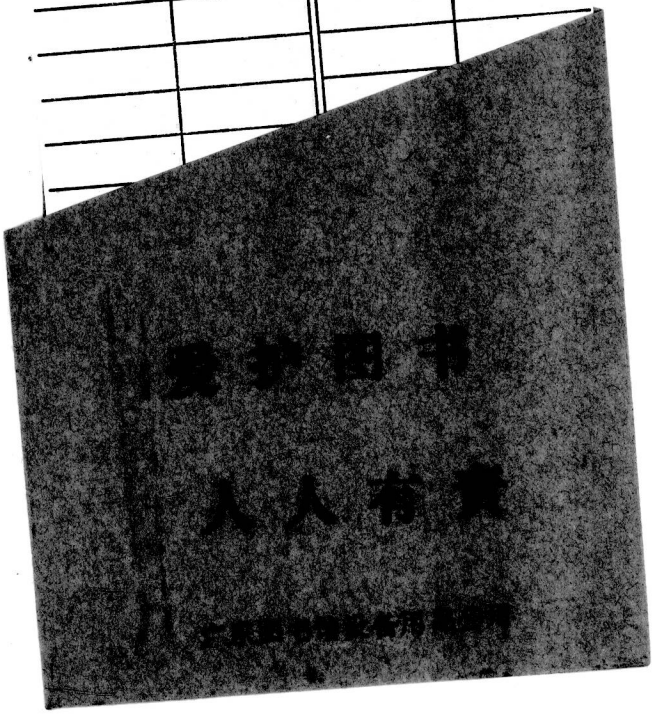
Cover illustration taken from the contribution by Stefan Wolf, pages 217 ff

CR Subject Classification (1998): E.3, G.2.1, D.4.6, K.6.5, F.2.1-2, C.2, J.1

ISSN 0302-9743
ISBN 3-540-65757-6 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10702913 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Lecture Notes in Computer Science 1561
Edited by G. Goos, J. Hartmanis and J. van Leeuwen



Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Preface

In July 1998, a summer school in cryptology and data security was organized at the computer science department of Aarhus University, Denmark. This took place as a part of a series of summer schools organized by the European Educational Forum, an organization consisting of the research centers TUCS (Finland), IPA (Holland) and BRICS (Denmark, Aarhus). The local organizing committee consisted of Jan Camenisch, Janne Christensen, Ivan Damgaård (chair), Karen Møller, and Louis Salvail. The summer school was supported by the European Union.

Modern cryptology is an extremely fast growing field and is of fundamental importance in very diverse areas, from theoretical complexity theory to practical electronic commerce on the Internet. We therefore set out to organize a school that would enable young researchers and students to obtain an overview of some main areas, covering both theoretical and practical topics. It is fair to say that the school was a success, both in terms of attendance (136 participants from over 20 countries) and in terms of contents. It is a pleasure to thank all of the speakers for their cooperation and the high quality of their presentations.

A total of 13 speakers gave talks: Mihir Bellare, University of California, San Diego; Gilles Brassard, University of Montreal; David Chaum, DigiCash; Ronald Cramer, ETH Zürich; Ivan Damgård, BRICS; Burt Kaliski, RSA Inc.; Lars Knudsen, Bergen University; Peter Landrock, Cryptomathic; Kevin McCurley, IBM Research, Almaden; Torben Pedersen, Cryptomathic; Bart Preneel, Leuven University; Louis Salvail, BRICS; Stefan Wolf, ETH Zürich.

It was natural to take the opportunity kindly offered by Springer-Verlag to publish a set of papers reflecting the contents of the school. Although not all speakers were able to contribute, due to lack of time and resources, this volume does cover all the main areas that were presented. The intention of all papers found here is to serve an educational purpose: elementary introductions are given to a number of subjects, some examples are given of the problems encountered, as well as solutions, open problems, and references for further reading. Thus, in general we have tried to give an up-to-date overview of the subjects we cover, with an emphasis on insight, rather than on full-detail technical presentations. Several results, however, are in fact presented with full proofs. The papers have not been refereed as for a journal.

I would like to thank all of the authors for their contributions and the hard work and time they have invested.

Ivan Damgård

Lecture Notes in Computer Science

For information about Vols. 1–1492
please contact your bookseller or Springer-Verlag

Vol. 1493: J.P. Bowen, A. Fett, M.G. Hinchey (Eds.), ZUM '98: The Z Formal Specification Notation. Proceedings, 1998. XV, 417 pages. 1998.

Vol. 1494: G. Rozenberg, F. Vaandrager (Eds.), Lectures on Embedded Systems. Proceedings, 1996. VIII, 423 pages. 1998.

Vol. 1495: T. Andreasen, H. Christiansen, H.L. Larsen (Eds.), Flexible Query Answering Systems. IX, 393 pages. 1998. (Subseries LNAI).

Vol. 1496: W.M. Wells, A. Colchester, S. Delp (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI'98. Proceedings, 1998. XXII, 1256 pages. 1998.

Vol. 1497: V. Alexandrov, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. Proceedings, 1998. XII, 412 pages. 1998.

Vol. 1498: A.E. Eiben, T. Bäck, M. Schoenauer, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature – PPSN V. Proceedings, 1998. XXIII, 1041 pages. 1998.

Vol. 1499: S. Kutten (Ed.), Distributed Computing. Proceedings, 1998. XII, 419 pages. 1998.

Vol. 1500: J.-C. Derniame, B.A. Kaba, D. Wastell (Eds.), Software Process: Principles, Methodology, and Technology. XIII, 307 pages. 1999.

Vol. 1501: M.M. Richter, C.H. Smith, R. Wiehagen, T. Zeugmann (Eds.), Algorithmic Learning Theory. Proceedings, 1998. XI, 439 pages. 1998. (Subseries LNAI).

Vol. 1502: G. Antoniou, J. Slaney (Eds.), Advanced Topics in Artificial Intelligence. Proceedings, 1998. XI, 333 pages. 1998. (Subseries LNAI).

Vol. 1503: G. Levi (Ed.), Static Analysis. Proceedings, 1998. IX, 383 pages. 1998.

Vol. 1504: O. Herzog, A. Günter (Eds.), KI-98: Advances in Artificial Intelligence. Proceedings, 1998. XI, 355 pages. 1998. (Subseries LNAI).

Vol. 1505: D. Caromel, R.R. Oldehoeft, M. Tholburn (Eds.), Computing in Object-Oriented Parallel Environments. Proceedings, 1998. XI, 243 pages. 1998.

Vol. 1506: R. Koch, L. Van Gool (Eds.), 3D Structure from Multiple Images of Large-Scale Environments. Proceedings, 1998. VIII, 347 pages. 1998.

Vol. 1507: T.W. Ling, S. Ram, M.L. Lee (Eds.), Conceptual Modeling – ER '98. Proceedings, 1998. XVI, 482 pages. 1998.

Vol. 1508: S. Jajodia, M.T. Özsu, A. Dogac (Eds.), Advances in Multimedia Information Systems. Proceedings, 1998. VIII, 207 pages. 1998.

Vol. 1510: J.M. Zytkow, M. Quafafou (Eds.), Principles of Data Mining and Knowledge Discovery. Proceedings, 1998. XI, 482 pages. 1998. (Subseries LNAI).

Vol. 1511: D. O'Hallaron (Ed.), Languages, Compilers, and Run-Time Systems for Scalable Computers. Proceedings, 1998. IX, 412 pages. 1998.

Vol. 1512: E. Giménez, C. Paulin-Mohring (Eds.), Types for Proofs and Programs. Proceedings, 1996. VIII, 373 pages. 1998.

Vol. 1513: C. Nikolaou, C. Stephanidis (Eds.), Research and Advanced Technology for Digital Libraries. Proceedings, 1998. XV, 912 pages. 1998.

Vol. 1514: K. Ohta, D. Pei (Eds.), Advances in Cryptology – ASIACRYPT'98. Proceedings, 1998. XII, 436 pages. 1998.

Vol. 1515: F. Moreira de Oliveira (Ed.), Advances in Artificial Intelligence. Proceedings, 1998. X, 259 pages. 1998. (Subseries LNAI).

Vol. 1516: W. Ehrenberger (Ed.), Computer Safety, Reliability and Security. Proceedings, 1998. XVI, 392 pages. 1998.

Vol. 1517: J. Hromkovič, O. Sýkora (Eds.), Graph-Theoretic Concepts in Computer Science. Proceedings, 1998. X, 385 pages. 1998.

Vol. 1518: M. Luby, J. Rolim, M. Serna (Eds.), Randomization and Approximation Techniques in Computer Science. Proceedings, 1998. IX, 385 pages. 1998.

1519: T. Ishida (Ed.), Community Computing and Support Systems. VIII, 393 pages. 1998.

Vol. 1520: M. Maher, J.-F. Puget (Eds.), Principles and Practice of Constraint Programming - CP98. Proceedings, 1998. XI, 482 pages. 1998.

Vol. 1521: B. Rovan (Ed.), SOFSEM'98: Theory and Practice of Informatics. Proceedings, 1998. XI, 453 pages. 1998.

Vol. 1522: G. Gopalakrishnan, P. Windley (Eds.), Formal Methods in Computer-Aided Design. Proceedings, 1998. IX, 529 pages. 1998.

Vol. 1524: G.B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade. VI, 432 pages. 1998.

Vol. 1525: D. Aucsmith (Ed.), Information Hiding. Proceedings, 1998. IX, 369 pages. 1998.

Vol. 1526: M. Broy, B. Rumpe (Eds.), Requirements Targeting Software and Systems Engineering. Proceedings, 1997. VIII, 357 pages. 1998.

Vol. 1527: P. Baumgartner, Theory Reasoning in Connection Calculi. IX, 283. 1999. (Subseries LNAI).

Vol. 1528: B. Preneel, V. Rijmen (Eds.), State of the Art in Applied Cryptography. Revised Lectures, 1997. VIII, 395 pages. 1998.

Vol. 1529: D. Farwell, L. Gerber, E. Hovy (Eds.), Machine Translation and the Information Soup. Proceedings, 1998. XIX, 532 pages. 1998. (Subseries LNAI).

Vol. 1530: V. Arvind, R. Ramanujam (Eds.), Foundations of Software Technology and Theoretical Computer Science. XII, 369 pages. 1998.

Vol. 1531: H.-Y. Lee, H. Motoda (Eds.), PRICAI'98: Topics in Artificial Intelligence. XIX, 646 pages. 1998. (Subseries LNAI).

Vol. 1096: T. Schael, Workflow Management Systems for Process Organisations. Second Edition. XII, 229 pages. 1998.

Vol. 1532: S. Arikawa, H. Motoda (Eds.), Discovery Science. Proceedings, 1998. XI, 456 pages. 1998. (Subseries LNAI).

Vol. 1533: K.-Y. Chwa, O.H. Ibarra (Eds.), Algorithms and Computation. Proceedings, 1998. XIII, 478 pages. 1998.

Vol. 1534: J.S. Sichman, R. Conte, N. Gilbert (Eds.), Multi-Agent Systems and Agent-Based Simulation. Proceedings, 1998. VIII, 237 pages. 1998. (Subseries LNAI).

Vol. 1535: S. Ossowski, Co-ordination in Artificial Agent Societies. XV; 221 pages. 1999. (Subseries LNAI).

Vol. 1536: W.-P. de Roever, H. Langmaack, A. Pnueli (Eds.), Compositionality: The Significant Difference. Proceedings, 1997. VIII, 647 pages. 1998.

Vol. 1537: N. Magnenat-Thalmann, D. Thalmann (Eds.), Modelling and Motion Capture Techniques for Virtual Environments. Proceedings, 1998. IX, 273 pages. 1998. (Subseries LNAI).

Vol. 1538: J. Hsiang, A. Ohori (Eds.), Advances in Computing Science – ASIAN'98. Proceedings, 1998. X, 305 pages. 1998.

Vol. 1539: O. Rüthing, Interacting Code Motion Transformations: Their Impact and Their Complexity. XXI,225 pages. 1998.

Vol. 1540: C. Beeri, P. Buneman (Eds.), Database Theory – ICDT'99. Proceedings, 1999. XI, 489 pages. 1999.

Vol. 1541: B. Kågström, J. Dongarra, E. Elmroth, J. Waśniewski (Eds.), Applied Parallel Computing. Proceedings, 1998. XIV, 586 pages. 1998.

Vol. 1542: H.I. Christensen (Ed.), Computer Vision Systems. Proceedings, 1999. XI, 554 pages. 1999.

Vol. 1543: S. Demeyer, J. Bosch (Eds.), Object-Oriented Technology ECOOP'98 Workshop Reader. 1998. XXII, 573 pages. 1998.

Vol. 1544: C. Zhang, D. Lukose (Eds.), Multi-Agent Systems. Proceedings, 1998. VII, 195 pages. 1998. (Subseries LNAI).

Vol. 1545: A. Birk, J. Demiris (Eds.), Learning Robots. Proceedings, 1996. IX, 188 pages. 1998. (Subseries LNAI).

Vol. 1546: B. Möller, J.V. Tucker (Eds.), Prospects for Hardware Foundations. Survey Chapters, 1998. X, 468 pages. 1998.

Vol. 1547: S.H. Whitesides (Ed.), Graph Drawing. Proceedings 1998. XII, 468 pages. 1998.

Vol. 1548: A.M. Haeberer (Ed.), Algebraic Methodology and Software Technology. Proceedings, 1999. XI, 531 pages. 1999.

Vol. 1550: B. Christianson, B. Crispo, W.S. Harbison, M. Roe (Eds.), Security Protocols. Proceedings, 1998. VIII, 241 pages. 1999.

Vol. 1551: G. Gupta (Ed.), Practical Aspects of Declarative Languages. Proceedings, 1999. VIII, 367 pgages. 1999.

Vol. 1552: Y. Kambayashi, D.L. Lee, E.-P. Lim, M.K. Mohania, Y. Masunaga (Eds.), Advances in Database Technologies. Proceedings, 1998. XIX, 592 pages. 1999.

Vol. 1553: S.F. Andler, J. Hansson (Eds.), Active, Real-Time, and Temporal Database Systems. Proceedings, 1997. VIII, 245 pages. 1998.

Vol. 1555: J.P. Müller, M.P. Singh, A.S. Rao (Eds.), Intelligent Agents V. Proceedings, 1998. XXIV, 455 pages. 1999. (Subseries LNAI).

Vol. 1557: P. Zinterhof, M. Vajteršic, A. Uhl (Eds.), Parallel Computation. Proceedings, 1999. XV, 604 pages. 1999.

Vol. 1558: H. J.v.d. Herik, H. Iida (Eds.), Computers and Games. Proceedings, 1998. XVIII, 337 pages. 1999.

Vol. 1559: P. Flener (Ed.), Logic-Based Program Synthesis and Transformation. Proceedings, 1998. X, 331 pages. 1999.

Vol. 1560: K. Imai, Y. Zheng (Eds.), Public Key Cryptography. Proceedings, 1999. IX, 327 pages. 1999.

Vol. 1561: I. Damgård (Ed.), Lectures on Data Security.VII, 250 pages. 1999.

Vol. 1563: Ch. Meinel, S. Tison (Eds.), STACS 99. Proceedings, 1999. XIV, 582 pages. 1999.

Vol. 1567: P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, S. Sastry (Eds.), Hybrid Systems V. X, 445 pages. 1999.

Vol. 1568: G. Bertrand, M. Couprie, L. Perroton (Eds.), Discrete Geometry for Computer Imagery. Proceedings, 1999. XI, 459 pages. 1999.

Vol. 1569: F.W. Vaandrager, J.H. van Schuppen (Eds.), Hybrid Systems: Computation and Control. Proceedings, 1999. X, 271 pages. 1999.

Vol. 1570: F. Puppe (Ed.), XPS-99: Knowledge-Based Systems. VIII, 227 pag s. 1999. (Subseries LNAI).

Vol. 1572: P. Fischer, H.U. Simon (Eds.), Computational Learning Theory. Proceedings, 1999. X, 301 pages. 1999. (Subseries LNAI).

Vol. 1575: S. Jähnichen (Ed.), Compiler Construction. Proceedings, 1999. X, 301 pages. 1999.

Vol. 1576: S.D. Swierstra (Ed.), Programming Languages and Systems. Proceedings, 1999. X, 307 pages. 1999.

Vol. 1577: J.-P. Finance (Ed.), Fundamental Approaches to Software Engineering. Proceedings, 1999. X, 245 pages. 1999.

Vol. 1578: W. Thomas (Ed.), Foundations of Software Science and Computation Structures. Proceedings, 1999. X, 323 pages. 1999.

Vol. 1579: W.R. Cleaveland (Ed.), Tools and Algorithms for the Construction and Analysis of Systems. Proceedings, 1999. XI, 445 pages. 1999.

Vol. 1580: A. Včkovski, K.E. Brassel, H.-J. Schek (Eds.), Interoperating Geographic Information Systems. Proceedings, 1999. XI, 329 pages. 1999.

Vol. 1587: J. Pieprzyk, R. Safavi-Naini, J. Seberry (Eds.), Information Security and Privacy. Proceedings, 1999. XI, 327 pages. 1999.

Table of Contents

Practice-Oriented Provable-Security

Mihir Bellare[1]

Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir.

1 Introduction

This short article is intended to complement my talk. I would like to try to introduce you to a certain, relatively new sub-area of cryptography that we have been calling *practice-oriented provable-security*. It is about applying the ideas of "provably security" to the derivation of practical, secure protocols. I believe it is a fruitful blend of theory and practice that is able to enrich both sides and has by now had some impact on real world security.

A few years ago, provable security was largely known only to theoreticians. This has been changing. We are seeing a growing appreciation of provable security in practice, leading in some cases to the use of such schemes in preference to other ones. Indeed it seems standards bodies and implementors now view provable security as an attribute of a proposed scheme. This means that a wider audience needs an understanding of the basic ideas behind provable security.

This article is directed at practioners and theoreticians alike. For the first I hope it will help to understand what provable security is and isn't, why it is useful, how to evaluate the provable security of a scheme, and where to look for such schemes. For the second group, it can serve to acquaint them with how the ideas with which they are familiar are being applied.

I will begin by describing the basic idea behind provable security. (For many of you, this will be mostly recall, but some novel viewpoints or examples may enter.) Next, I will discuss the practice-oriented approach. I will discuss its main ideas, the problems it has addressed, and briefly survey known results. I hope to leave you feeling there is scope here both for interesting research and for application.

2 Protocols, primitives, proofs and practice

The basic task in cryptography is to enable to parties to communicate "securely" over an insecure channel, namely in a way that guarantees privacy and authenticity of their transmissions. (There are many other tasks as well, but we will begin by thinking about this basic one.)

2.1 Protocols and primitives: the problem

PROTOCOLS: THE END GOAL. To enable secure communication, one wants cryptographic *protocols* or *schemes*. For example, an encryption scheme enables users to communicate privately. Such a scheme is specified by a pair $(\mathcal{E}, \mathcal{D})$ of algorithms. The first, run by the sender, takes a *key* and the *plaintext* M to create a *ciphertext* C, which is transmitted to the receiver. The latter applies $\mathcal{D}$, which takes a key and the received ciphertext to recover the plaintext. (Roughly, the security property desired is that an adversary can't learn anything useful about the plaintext given the ciphertext, but we will get into this more later.) They key could be a shared one (this is the private key or symmetric setting) or the keys for encryption and decryption could be different (the public key or asymmetric setting). Designing an encryption scheme means designing the two algorithms $\mathcal{E}$ and $\mathcal{D}$.

Similarly, a message authentication scheme (or protocol) enables parties to tag their data so that the recipient is assured that the data originates with the person claiming to have sent it and has not been tampered with on the way.

The design of such protocols is the end goal for the cryptographer. However, it is not an easy one to reach. What makes it reachable at present is that we have very good *primitives* on which to *base* these protocols.

PRIMITIVES: THE TOOLS. Julius Caesar also wanted to design protocols. He had a much harder time than we do today, because he didn't have DES or the RSA function.

The latter are examples of what I will call *atomic primitives*. Certainly, they are cryptographic objects of some sort. What is it that distinguishes them from protocols? The distinction is that in their purest and rawest state, atomic primitives don't solve any cryptographic problem we actually care about. We must *use* them appropriately to construct protocols to solve the problems that matter. For example, DES based CBC encryption is a way of using DES to do symmetric encryption. By first hashing a message and then decrypting under RSA we have a possible way to do digital signatures based on the RSA function. (Whether these ways are good or bad ways of accomplishing the goal is another question, to be addressed later.) Thus, atomic primitives are simple building blocks that must be put together to yield protocols.

Good atomic primitives are rare, as are people who understand their workings. Certainly, an important effort in cryptography is to design new atomic primitives and cryptanalyze them and old ones. This, however, is not the part of cryptography I want to talk about. The reason is that the design (or discovery) of good atomic primitives is more an art than a science. On the other hand, I'd like to claim that the design of protocols can be made a science.

THE QUESTION. We will view a cryptographer as an engine for turning atomic primitives into protocols. That is, we focus on protocol design under the assumption that good atomic primitives exist. Some examples of the kinds of questions we are interested in are these. What is the best way to encrypt a large text file using DES, assuming DES is secure? What is the best way to design a signature

scheme using the RSA function, assuming the latter is one-way? How "secure" are known methods for these tasks? What do such questions even mean, and can we find a scientific framework in which to ask and answer them?

THE PROBLEM. The problem with protocol design is that a poorly designed protocol can be insecure *even though the underlying atomic primitive is good.* An example is ECB (Electronic Code-Book) mode encryption with a block cipher. It is not a good encryption scheme because partial information about the plaintext leaks. Yet this is no fault of the underlying atomic primitive (typically DES). Rather, the atomic primitive was mis-used.

Indeed, lots of protocols are broken. Yet the good atomic primitives, like DES and RSA, have never been convincingly broken. We would like to build on the strength of atomic primitives in such a way that protocols can "inherit" this strength, not loose it!

2.2 Provable security: Reductions

The idea of provable security was introduced in the pioneering work of Goldwasser and Micali [26]. They developed it in the particular context of asymmetric encryption, but it soon spread to be applied to other tasks. (Of these, the most basic were pseudorandomness [16,40,25] and digital signatures [27]).

WHAT IS PROVABLE SECURITY? The paradigm is as follows. Take some goal, like achieving privacy via encryption. The first step is to make a formal adversarial model and *define* what it *means* for an encryption scheme to be secure. With this in hand, a particular scheme, based on some particular atomic primitive, can be analyzed from the point of view of meeting the definition. Eventually, one shows that the scheme "works" via a *reduction.* The reduction shows that the *only way* to defeat the protocol is to break the underlying atomic primitive. In other words, there is no need to directly cryptanalyze the protocol: if you were to find a weakness in it, you would have unearthed one in the underlying atomic primitive. So you might as well focus on the atomic primitive. And if we believe the latter is secure, we *know*, without further cryptanalysis of the protocol, that the protocol is secure.

An important sub-part of the last step is that in order to enable a reduction one must also have a formal notion of what is meant by the security of the underlying atomic primitive: what attacks, exactly, does it withstand? For example, we might assume RSA is a one-way function.

Here is another way of looking at what reductions do. When I give you a reduction from the one-wayness of RSA to the security of my protocol, I am giving you a transformation with the following property. Suppose you claim to be able to break my protocol. Let P be the program that does this. My transformation takes P and puts a simple "wrapper" around it, resulting in a protocol P'. This protocol P' provably breaks RSA. Conclusion? As long as we believe you can't break RSA, there could be no such program P. In other words, my protocol is secure.

Those familiar with the theory of NP-completeness will recognize that the basic idea of reductions is the same. When we provide a reduction from SAT to some problem we are saying our problem is hard unless SAT is easy; when we provide a reduction from RSA to our protocol, we are saying the latter is secure unless RSA is easy.

Here, I think, is a beautiful and powerful idea. Some of us by now are so used to it that we can forget how innovative it was. And for those not used to it, it can be hard to understand (or, perhaps, believe) at first hearing, perhaps because it delivers so much. Protocols designed this way truly have superior security guarantees.

NOMENCLATURE. In some ways the term "provable security" is misleading. As the above indicates, what is probably the central step is providing a model and definition, which does not involve proving anything. And one does not "prove a scheme secure:" one provides a reduction of the security of the scheme to the security of some underlying atomic primitive. For that reason, I sometimes use the term "reductionist security" to refer to this genre of work.

THE COMPLEXITY-THEORETIC APPROACH. The precise formalization of provable security can take many forms. The theoretical literature has chosen, for the most part, to develop it in a complexity theoretic framework where one talks about "polynomial time" adversaries and transformations, and "negligible success probabilities." This approach was convenient for a field striving to develop a technical idea of great depth. Complexity-based cryptography has been remarkably successful, coming up with definitions for many central cryptographic primitives, and constructions based on "minimal assumptions." For a brief introduction to this body of work, refer to the recent survey by Goldreich [24].

IN PRACTICE? The potential for the idea of provable security to impact practice is large. Yet its actual impact had been disappointingly small, in the sense that these ideas were reflected almost not at all in protocols used in practice. Here are some possible reasons.

In practice, block ciphers are the most popular atomic primitive, especially for private key cryptography. Yet the provable security line of work (prior to the development of the practice-oriented variant) omitted any treatment of schemes based on block ciphers: only number-theoretic atomic primitives were deemed adequate as a basis for protocol design. In particular some of the world's most used protocols, such as CBC MAC [1] or encryption [32,2], seemed to be viewed as outside the domain of provable security.[1]

The main generic disadvantage of the schemes delivered by the traditional provable security approach is that they are inefficient.[2] This is due in part to the complexity of the constructions. But it is also due in part to a reliance on inefficient atomic primitives. For example, a MAC would be constructed out of

[1] Luby and Rackoff [31] studied the Feistel structure behind DES, but what I am talking about is to look at protocols that use DES and ask about their security.

[2] Typically the gap relative to what is desirable in practice is enormous. In some cases it is small, but still seems enough to preclude usage.

a one-way function like RSA rather than out of a block cipher. This takes us back to the above.

Finally, some aspects of the complexity-theoretic approach unfortunately distanced provable security from practice. For example, practioners need numbers: how many cycles of adversary computation can the scheme withstand, how many bits is the security parameter? These are only loosely captured by "polynomials" or "negligible probabilities." To make provable security useful, reductions and security analyses must be concrete. Theoreticians will say, correctly, that this information can be obtained by looking at their proofs. But this view obscures the importance of working on improving the security of reductions.[3]

Practice-oriented provable security attempts to remedy this by appropriate paradigm shifts.

3 Practice-oriented provable security

Practice-oriented provable security as I discuss it was introduced in a set of papers authored by myself and Phil Rogaway [8,7,6]. We preserve and focus on the two central ideas of the provable security approach: the introduction of *notions*, or *definitions* that enable us to think about protocols and atomic primitives in a systematic way, and the idea of doing reductions. But we modify the viewpoints, models, and problems treated. Here are some elements of the approach and work to date.

3.1 Using block ciphers

Block ciphers like the DES are the most ubiquitous tool in practical cryptographic protocol design. However, as indicated above, traditionally nothing was proved about protocols that use them. An important element of our line of work is to integrate block ciphers into the fabric of provable security. On the one hand we analyze existing schemes that use block ciphers to assess how well they meet strong, formal notions of security; on the other hand we design new schemes based on block ciphers and show they meet such notions. In the first category are our analyses of the CBC MAC [7] and analyses of various modes of operation of a block cipher [5]. In the second category are constructions like the XOR MAC [6] or the cascade [4].

Key to these results (and perhaps more important than any individual result) is that we treat block ciphers systematically by formally modeling them in some way. Specifically, the suggestion of [7], followed in the other works, was to model a block cipher as a *finite pseudorandom function* (FPRF) family. (The fundamental notion of a pseudorandom function family is due to Goldreich, Goldwasser and Micali [25]. The finite variant was introduced in [7].) Roughly, we are

[3] This is not to say concrete security has always been ignored. One person who from the beginning has systematically addressed concrete security in his works is Claus Schnorr. See any of his papers involving cryptographic reductions.

assuming that as long as you don't know the underlying key, the input-output behavior of a block cipher closely resembles that of a random function.

Thus, the theorems in the mentioned papers say that a scheme (eg. CBC MAC) is secure unless one can detect some deviation from random behavior in the underlying block cipher. Underlying this claim is a reduction, as usual in the provable security approach, showing how to break the cipher given any way to break the scheme based on it.

The idea of treating block ciphers as pseudorandom functions provides a fresh way of looking at block ciphers from both the design and usage perspective. On the one hand, this view can form the basis for analyses of many other block cipher based schemes. On the other hand, we suggest it be a design criterion for future block ciphers (a view that new efforts such as AES do seem to support) and that existing ciphers should be cryptanalyzed to see how well they meet this goal.

3.2 Concrete security

Practice oriented provable security attempts to explicitly capture the inherently *quantitative* nature of security, via a *concrete* or *exact* treatment of security. Rather than prove asymptotic results about the infeasability of breaking a protocol in polynomial time, we present and prove "exact" or "concrete" reductions. Our results have the form: "If DES withstands an attack in which the adversary gets to see 2^{36} plaintext-ciphertext pairs, then our protocol is secure against an adversary who can run for t steps, for the following value of t." This enables a protocol designer to know exactly how much security he/she gets. And it brings a new dimension to protocols: rather than just being secure or non-secure, one can be "more" secure than another.

For example, the theorem of [7] characterizing the security of the CBC MAC says that an adversary who runs for time t and sees q correctly MACed messages has chance at most $\epsilon + (3q^2n^2 + 1)/2^l$ of correctly forging the MAC of a new message, where l is the block length of the underlying cipher, n is the number of blocks in any message to which the MAC applies, and ϵ captures the security of the cipher, specifically being the chance of detecting a deviation of the cipher from random behavior in time $t + O(nql)$ given nq input-output examples of the cipher under the same key. (This ϵ is of course a function of the key length of the underlying cipher, but the latter does not need to appear explicitly.) Thus, a user sees exactly how the chance of forgery increases with the number of messages MACed.

Another aspect of the concrete security treatment is to try to preserve as much as possible of the strength of the underlying atomic primitive in transforming it to the protocol. This means we aim for reductions as *strong* as possible. This is important because reduction strength translates directly to protocol efficiency in practice. A weak reduction means that to get the same level of security in our protocol we must use larger keys for the underlying atomic primitive, and this means slower protocols. If the reduction is strong, shorter keys will suffice

and the protocol is more efficient. Reduction quality plays a significant role in [7,6,10,12,4,5] all of which achieve tight or close to tight reductions.

We found that *improving* the concrete security was a rich and rewarding line of work, and thinking about it greatly increases understanding of the problem.

In [5] we also concern ourselves with how different formalizations of a notion (in this case, secure encryption) are affected when concrete security is an issue.

3.3 Security versus attacks

Practitioners typically think only about concrete attacks; theoreticians ignore them, since they prove the security. Under the practice oriented provable security approach, attacks and security emerge as opposite sides of the same coin, and complement each other. Attacks measure the degree of insecurity; our quantitative bounds measure the degree of security. When the two meet, we have completely characterized the security of the protocol.

For example, the security of the CBC MAC shown in [7] is the flip-side of attacks like those of Preneel and Van Oorschot [37]. (The latter say that the CBC MAC can be broken once $2^{l/2}$ messages have been MACed, where l is the block length of the underlying cipher. We say, roughly, that it *can't* be broken when *fewer* than this many messages are MACed.) Thus the results of [7,37] complement each other very well. Yet, the literature on these subjects does not reflect this duality appropriately.

We found that even when proofs are provided, much is to be gained by finding the best possible attacks. We find *new kinds* of attacks, which break the system as measured by our more stringent notions of security: an encryption scheme is broken of you can tell whether the message encrypted was 0 or 1, not just if you find the key. This is actually important in practice. Meanwhile, these attacks provide, effectively, the lower bounds to our concrete security analyses, telling us whether the proven security is optimal or not. Publications in which we assess the optimality of our reductions via attacks include [6,4,5].

3.4 The random oracle model

Sometimes, using pseudorandom function families or one-way functions alone, we are not able to find schemes efficient enough for practice. This is true for example in the case of public key encryption or signatures. In such cases, we turn to the random oracle paradigm.

The random oracle paradigm was introduced in [9] as a bridge between theory and practice. The idea is a simple one: namely, provide all parties —good and bad alike— with access to a (public) function h; prove correct a protocol assuming h is truly random, ie. a random oracle; later, in practice, set h to some specific function derived in some way from a standard cryptographic hash function like SHA-1 [33] or RIPEMD-160 [21].

We used the random oracle paradigm most importantly to design OAEP [10] and PSS [12]. These are schemes for (public key) encryption and signature

(respectively), the most popular versions of which use RSA as the underlying primitive. (Both OAEP and PSS are, more accurately, padding or formatting mechanisms which are applied to a message before the appropriate RSA operation is applied.) They are as efficient as previously used or standardized schemes, but, unlike them, provably achieve strong notions of security in the random oracle model, assuming RSA is a one-way function.

RSA Corporation publishes a standard for RSA based encryption called PKCS#1. (It is a widely used standard, implemented in Netscape and other browsers, and used in SSL.) Much publicity was given recently to a chosen-ciphertext attack on PKCS#1 that was discovered by Bleichenbacher [15]. RSA Corporation has now revised the protocol, adopting OAEP in PKCS#1 v2.0 [38]. The rationale for that move is that our protocol had been *proven* to resist chosen-ciphertext attacks (indeed Bleichenbacher's attacks do not work on OAEP, even though at the time of the design of OAEP we had not thought of these specific attacks), and furthermore OAEP is just as practical as the original PKCS#1 protocol.

OAEP is also included in SET, the electronic payment protocol of MasterCard and Visa, where it is used to encrypt credit card numbers. Both OAEP and PSS are being proposed for the IEEE P1363 standard.

What's the point of the random oracle paradigm, and what does it buy you? It buys efficiency, plus, we claim, security guarantees which, although not at the same level as those of the standard provable security approach, are arguably superior to those provided by totally ad hoc protocol design. The last point merits some more discussion.

The random oracle paradigm should be used with care and understanding. It is important to neither over-estimate nor under-estimate what this paradigm buys you in terms of security guarantees. First, one must be clear that this is not standard provable security. The function h that we actually use in the final scheme is not random. Thus the question is: what has it bought us to have done the proof in the first place?

The overly skeptical might say the answer is "nothing." This is not quite true. Here is one way to see what it buys. In practice, attacks on schemes involving a SHA-1 derived h and number theory will often *themselves treat h as random.* We call such attacks *generic attacks.* In other words, cryptanalysis of these "mixed" schemes is usually done by assuming h is random. But then the proofs apply, and indeed show that such generic attacks will fail unless the underlying number-theoretic problems are easy. In other words, the analysis at least provably excludes a certain common class of attacks, namely generic ones.

It is important to choose carefully the instantiating function h. The intuition stated in [9] is that the resulting scheme is secure as long as the scheme and the hash function are sufficiently "independent," meaning the scheme does not itself refer to the hash function in some way. This is a fuzzy guideline which we hope to understand better with time.

An important step in our understanding of the random oracle model was taken by Canetti, Goldreich and Halevi [19]. They indicate that there exist