
Professional statement: **Phillip Rogaway**

For further information visit <http://www.cs.ucdavis.edu/~rogaway/>

Research

PERSPECTIVE. My research is in **cryptography**: the science of secure communication. Over the last decade I have worked to create and refine a branch of cryptography that is theoretically-well-founded but nonetheless shaped by, and directly contributes to, cryptographic practice. Let me explain.

In the early 1980s Goldwasser and Micali first put forward the remarkable idea that security could be *proved* under well-believed, complexity-theoretic assumptions. The methodology they developed is called **provable security**. Achieving it for a given problem entails providing: (i) a definition of the goal; (ii) a protocol, this protocol making use of some lower-level primitive; (iii) a proof (called a “reduction”) that the protocol meets the definition if its primitive does its (similarly defined) job.

Provable security transformed cryptography from an art to a science, and became the main thread of theoretical work in our field. All the same, the actual users of cryptography tended to view the provable-security approach—when they knew of it at all—as having little or no applicability.

In the early 1990s, after being trained in the provable-security tradition at MIT, I went to work at an IBM development lab. There I tried to figure out what factors were causing provable security to be so uninfluential in shaping cryptographic practice. I came to understand that, during these early years of our science, it had been convenient to develop provable-security cryptography in a way that just so happened to limit the area’s actual and perceived utility. Among the issues: (a) traditional “starting points” for our reductions, like one-way functions, didn’t match the desirable ones for creating a practical science; (b) our (asymptotic) theorem statements said nothing of obvious use; (c) the efficiency of our protocols (and our reductions) was often quite abysmal; and (d) central cryptographic problems then occupying practitioners, like entity authentication and session-key distribution, had never received a provable-security treatment.

With these issues in mind, I set out to adjust the “way” that I did provable security, in order that my output could be more directly useful. The resulting approach wouldn’t be any less rigorous or substantive, but it would feel different. Over the course of several papers, often with Mihir Bellare, we developed what we came to call **practice-oriented provable security**. It’s still about definitions, protocols, and proofs; the provable-security “core” is all in tact. But there are significant differences in the way that we do these things. These differences reflect a shift in goals: I am not working to unravel the provable-security relationships between different cryptographic goals—others finished this, by and large, before my time. Rather, I am working work to make definitions and reductions become powerful **tools** for the design and analysis of practical protocols.

Among its aspects, practice-oriented provable security often: (a) uses finite pseudorandom permutations (block ciphers) as a starting point in reductions (and, in general, confusion/diffusion primitives are handed as smoothly as number-theoretic ones); (b) pays close attention to concrete analysis (no asymptotics), not only to make clear how good or bad an analysis is, but also to provide guidance in choosing among practical protocols, to inspire better protocols, and to engender sharper underlying notions; (c) adopts the random-oracle paradigm when obtaining provable security in the standard model would involve a loss of efficiency or simplicity so great as to render the

methods undesirable in the real world; (d) brings provable security to bear on practical problems like entity authentication and session-key distribution.

SELECTED CONTRIBUTIONS. Papers [A24–A35, B1–B3] were published or accepted for publication during the current review period. Each embodies practice-oriented provable security. I sketch some of this work and its context.

(1) In [A7, A11] we gave the first provable-security treatment for entity authentication and session-key distribution for the distributed-systems setting. In [A31] we move on to give the first provable-security treatment of authenticated key exchange secure against dictionary attacks. In this setting the underlying long-lived key is an easily-remembered password. Security in the face of dictionary attacks is a central problem of cryptographic practice.

(2) In [A9] and [A12] we initiated the concrete-security analysis of block-cipher using constructions. We continue this approach in [A20, A28, A32, A34, B2]. We analyze old techniques (CBC MAC, DESX) and new techniques of our own (XOR MAC, XCBC, OCB). Our most recent work [A32, B2] has yielded particularly elegant schemes for message authentication and authenticated encryption.

(3) In [A6] we put forward the *random-oracle paradigm*, where provable-security is carried out in a model of computation that gives all parties (including the adversary) access to a public random oracle, and where this oracle is then instantiated as a final, “heuristic” step. The paradigm is highly useful for achieving protocols with efficiency characteristics the equal of ad. hoc practice, but with better assurance guarantees. This paper has become my most referenced (16% of all citations to me.) Papers [A25, A31, A35] give further results under this now-popular model of computation.

(4) In [A10] we developed OAEP, which uses the RSA (or any other) trapdoor permutation and a cryptographic hash function to construct an encryption scheme. We proved that, in the random-oracle model, with an arbitrary trapdoor permutation, OAEP achieves indistinguishability under chosen-plaintext attack plus a weak form of plaintext awareness. OAEP was adopted by influential standards of the IEEE and RSA PKCS. An entire session at CRYPTO '01 was devoted to follow-on work on OAEP. In [A35] we do for Diffie-Hellman what OAEP does for RSA. The paper gives an analysis of DHIES, our encryption scheme based on the Diffie-Hellman assumption (in any group). Under specified assumptions, we prove security against chosen-ciphertext attack. DHIES has been rapidly adopted: it is in standards and draft standards of ANSI, IEEE, ISO, and SEC.

(5) In [A27] we introduce UMAC, a message authentication code that authenticates messages (in software, on contemporary machines) roughly ten times faster than conventional practice. This was a major piece of “crypto-engineering.” UMAC uses the Carter-Wegman paradigm, employing a new universal-hash-function, NH, which can exploit the SIMD instructions of modern microprocessors. UMAC makes clear that practice-oriented provable security isn’t just for assurance any more: it can give rise to methods more efficient than any that would emerge without such a tool.

(6) In [A29, B2] we begin to bridge the provable-security treatment of cryptography that evolved within my community and the formalistic treatment of cryptography that evolved within the security/formal-methods community. We give a soundness theorem that shows that, under appropriate complexity-theoretic conditions, formal ciphertexts which are deemed “equivalent” with respect to a simple “algebra” of rules represent computationally indistinguishable ensembles. The hope is that this work will be substantially extended, providing a way to get meaningful (complexity-theoretic) guarantees as a byproduct of higher-level reasoning about protocols.

(7) In [B1] we introduce OCB, a block-cipher mode of operation inspired by the emergence of the Advanced Encryption Standard (AES) and by a paper of C. Jutla. OCB protects both the privacy and authenticity of messages, which it does at a cost within a few percent of providing privacy-protection alone. Many efficiency and usability characteristics are built into OCB, such a parallelizability and minimal ciphertext-expansion. We expect OCB to “take off,” since it performs what is probably the most widely needed cryptographic operation about twice as fast as the current art. OCB is already in a draft standard of IEEE 802.11 (the standard for Wireless LAN networks).

INFLUENCE. Practice-oriented provable security has definitely caught on. Provably-secure schemes are at the center of modern standards and contemporary practice, and the phrase “provable security” has acquired such a cachet that one now hears people claim to have achieved this even when they don’t understand what the term means. Some “evidence” of having moved my field: ▷ Nowadays, a large number of papers influenced by my work appear at every major cryptography conference. For example, 16 of 33 papers at CRYPTO ’01 reference at least one paper of mine, as do 16 of 32 papers at CRYPTO ’00, 14 of 39 papers at CRYPTO ’99, 11 of 32 papers at EUROCRYPT ’01, and 17 of 39 papers at EUROCRYPT ’00.¹ ▷ A database search finds **1247** academic papers which reference my work.² ▷ My work has had a large impact in the world of cryptographic standards. Four of my protocols—OAEP [A10], PSS/PSS-R [A15], DHIES [A21, A35], and OCB [B3]—are in standards or draft standards of ANSI, IEEE, ISO, PKCS, SEC. Nobody in my field has been more effective at getting his work into cryptographic standards.

FUNDING. ▷ A gift from CISCO, November 2000, \$80,000. ▷ NSF CAREER Award (1996–2000): *Practice-Oriented Provable Security*. \$200,000. ▷ MICRO grant (1997–98) from Certicom and RSA Data Security. *Provable-Security Design and Analysis for Emerging Cryptographic Standards*. \$64,600 (Certicom: \$20,000, RSA: \$20,000, UC Matching: \$24,600). ▷ MICRO grant (1998–1999) from ORINCON and RSA Data Security. *Provable-Security Design and Analysis for IEEE Cryptographic Standards*. \$67,672 (ORINCON: \$15,000, RSA: \$25,000, UC Matching: \$27,672). ▷ MICRO grant (1999–2000) from ORINCON. *Provable-Security Design and Analysis for IEEE Cryptographic Standards*. \$36,000 (ORINCON: \$20,000, UC Matching: \$16,000).

OUTLETS. Most of my publications are in competitive conferences. This is customary in my area, where the best papers in cryptography usually appear in the proceedings for our top conferences.³ Submissions to our top conferences are thoroughly reviewed (usually by at least three referees) and acceptance rates are lower than with most journals.⁴ The proceedings of these conferences are published as books by Springer-Verlag (in the *Lecture Notes in Computer Science* series) and these books are carried by all major libraries. My remaining publications are in our top journals. In general, I disseminate my research in the customary way for the best work in my field.

¹ The proceedings of these two conferences is usually considered the top publication venue for my field.

² To put the number in perspective: (Rivest, 6506), (Abadi, 4065), (Shamir, 2951), (Bellare, 2221), (Krawczyk, 1239), (Coppersmith, 1160), (Shoup, 647), (Boneh, 572), Leaving my field, other highly-cited faculty members within my department include (Bai, 1316), (Mukherjee, 737), (Laub, 727), (Levitt, 478). (Gusfield, 389), (Hamann, 356), (Martel, 343). All reference counts are from <http://citeseer.nj.nec.com/cs> as of 25 Sept 2001. Database queries were made using last name only, except for Bai, Mukherjee, and Shamir, where this resulted in a substantial number of false matches. For those cases, queries were: `z bai or z d bai or zhaojun bai`; `b mukherjee or biswanath mukherjee`; `a shamir or adi shamir`.

³ CRYPTO and EUROCRYPT are considered the top conferences, but several other venues also get good papers. Among all outlets in computer science, impact ratings are: (J. of Cryptology, #35, top 4%) (CRYPTO, #97, top 11%), (ACM-CCS, #104, top 12%), (EUROCRYPT, #180, top 21%). This data from April 2001 compilation of <http://citeseer.nj.nec.com/impact.html>. Impact ratings look at how often articles from each venue get cited.

⁴ Acceptance rates: CRYPTO (’00, 27%), (’01, 21%); EUROCRYPT (’00, 26%), (’01, 21%); CCS (’01, 18%).

Teaching

I taught six courses at UC Davis from F98 to S01, receiving average scores of **6.9**, **8.1**, **8.7**, **8.9**, **9.6**, **9.7** for ECS 20—Discrete Math (S00), ECS 120—Theory of Computation (W99, S99), ECS 122A—Algorithms (S00), and ECS 227—Cryptography (W00, W99), respectively. The low number of courses for a three-year period is due to being on sabbatical or leave of absence during five quarters (F98, F99, F00, W01, S01); no buy-out was taken during this period.

I espouse the same pedagogic philosophy for both undergraduate and graduate teaching: that my goal is to engender creative thinking and problem-solving skills (for a style of thinking characteristic of theoretical computer science). I regard the material being taught as the “stuff” that is used to get the students to think; installing it in the student’s heads isn’t what my courses are about. My teaching style was strongly influenced by Manuel Blum (UCB) and Silvio Micali (MIT).

The emphasis on thinking and problem solving skills, coupled with my trying to target the top students, gives rise to courses that are regarded as very difficult. This is fine by me. Still, I have increasingly tried to make what I say and do accessible to “average” students.

At the graduate level, my cryptography course is nearly unique, covering the subject in a way that is as rigorous as the cryptography courses that trained me at MIT, but much more concrete. I completely avoid asymptotic complexity theory (an approach that reflects one vein of my research) and emphasize modeling and definitions over proofs (though I certainly do proofs, too.) I see definitions as especially important, since the skill of crafting good ones is surely applicable across much of computer science. Course notes have been evolving into a textbook.

I have advised a small number of Ph.D students. Two of my students have graduated, going off to good jobs, and one new student is working with me now. The relatively low output of Ph.D students reflects the fact that few of our computer science graduate students have the inclination and mathematical maturity necessary to do well in a theory-oriented area like cryptography. I haven’t taken any Master’s students, and probably won’t, as a master’s degree for cryptography is not a particularly appropriate pairing. Besides cryptography, I work hard to teach good writing to my advisees, as success in my field seems to require clear writing nearly as much as it requires creativity and clear thinking.

Some sample quotes from student evaluations: ▷ Each proof/theorem/idea is meticulously thought out and explained well in detail (227, W00). ▷ Very clear, detailed, and thought provoking (227, W00). ▷ Great lecturer, very interesting (122A, W00). ▷ One of the most engaging instructors I’ve had in 9 years of University coursework (120, W99). [What I like most is] ▷ His infectious passion for the subject & for scholarship in general (227, W00). ▷ I came in thinking this was the class from hell, but Rogaway showed how a difficult problem could be thought out to be easy (122A, S00). ▷ Smart, funny, chooses good problems to analyze (122A, S00). ▷ Very happy to teach. He makes me want to come to class (120, S99). ▷ Instructor is very nice and friendly, he makes people feel comfortable talking to him (120, S99). ▷ This class is incredibly well organized and presented (120, S99). ▷ He’s very bright and quite passionate about the subject (120, S99). ▷ Hands down, one of the “hardest” classes in the CS major (120, S99) ▷ other courses were learn and regurgitate. This was no such course. It actually required thinking! (122A, S00). ▷ This class is hard! Rogaway is great, though (120, W99). ▷ I can think differently (better) and this [is] one of the advantages of taking this class (120, S99). ▷ when you first said that ECS 120 will be one of the funnest ECS class, I found it hard to believe... now I couldn’t agree more... this class has given me a new insight about computation ... (120, S99)

Service

The following outlines recent service activities.

SERVICE TO THE DEPARTMENT AND COLLEGE. ▷ Member, CSUGA (Computer Science Undergraduate Affairs Committee). I have regularly served on this Departmental-level committee which handles most issues concerning undergraduate education. Activities include undergraduate student advising. ▷ Member (CS Representative), UGSC (Undergraduate Study Committee). This is the college-level committee that handles most issues concerning undergraduate education and policy. We meet regularly and deal with all matters involving the curricula of programs within the college. ▷ Member (CS Representative), Student Relations. This is the college-level committee that handles student petitions. We meet regularly to approve or reject the students' latest attempts to sidestep our arcane and ever-changing requirements.

PROFESSIONAL SERVICE. ▷ I served on the Program Committee for CRYPTO '98, CRYPTO '99, and CRYPTO '00, and for some more minor conferences, ▷ I do at least my share of conference and journal refereeing. ▷ I have been a consultant to Certicom, IBM, TIS, other companies. ▷ I have been a consultant to the government of Japan (ongoing). ▷ I am on the advisory board for an organization called SEC (Standard for Efficient Cryptography). ▷ I have contributed to the making of cryptographic standards. Besides the work mentioned already, I often give informal advice to members of the standards bodies, have provided several writeups for the IEEE and NIST, and have given talks to the IEEE and NIST in support of their standardization work. ▷ I convinced UC Legal to freely license PSS, helping to bring this technique to practice. ▷ Invited lectures at conferences, most recently to PKS '99 and ECC '00. ▷ Invited lectures at MIT. ▷ I have taught mini-classes in cryptography, by invitation, in Estonia and in Italy, and at UCSD.

SERVICE IN THAILAND. During two quarters of sabbatical and a one-year leave of absence, I have done my work at Chiang Mai University (CMU), in northern Thailand. ▷ I helped their Department of Computer Science carry out a revision of their undergraduate curriculum and then their graduate curriculum. ▷ I did much teaching at CMU: four one-semester courses. ▷ I also gave a 25-hour series of lectures at Chulalongkorn University, the top university in Thailand. ▷ I taught a group of top high school students who were competing in the International Computer Science Olympiad, delivering about a dozen lectures on discrete math, all of them, to my own astonishment, in Thai.

Phillip Rogaway

Date