Minimizing the Use of Random Oracles in Authenticated Encryption Schemes

Mihir Bellare[1] and Phillip Rogaway[2]

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

[2] Dept. of Computer Science, Engineering II Bldg., University of California at Davis, Davis, CA 95616, USA. E-mail: rogaway@cs.ucdavis.edu. URL: http://wwwcsif.cs.ucdavis.edu/~rogaway.

Abstract. A cryptographic scheme is "provably secure" if an attack on the scheme implies an attack on the underlying primitives it employs. A cryptographic scheme is "provably secure in the random-oracle model" if it uses a cryptographic hash function F and is provably secure when F is modeled by a public random function. Demonstrating that a cryptographic scheme is provably secure in the random-oracle model engenders much assurance in the scheme's correctness. But there may remain some lingering fear that the concrete hash function which instantiates the random oracle differs from a random function in some significant way. So it is good to limit reliance on random oracles. Here we describe two encryption schemes which use their random oracles in a rather limited way. The schemes achieve semantic security and plaintext awareness under specified assumptions. One scheme uses the RSA primitive; another uses Diffie-Hellman. In either case messages longer than the modulus length can be safely and directly encrypted without relying on the hash functions modeled as random-oracles to be good for private-key encryption.

1 Introduction

1.1 Provable security and random oracles

A cryptographic scheme S based on a primitive P is said to be *provably secure* if the security of P has been demonstrated to imply the security of S. More precisely, we use this phrase when someone has formally defined the goal G_P for some primitive P; someone has formally defined and the goal G_S for some scheme S; and then someone has proven that the existence of an adversary A_S who breaks scheme S, in the sense of violating G_S, implies the existence of an adversary A_P who breaks protocol P, in the sense of violating G_P.

What provable security means is that as long as we are ready to believe that P is secure, then there are no attacks on S. This obviates the need to consider any specific cryptanalytic attacks on S. Provable security can vastly increase assurance in a cryptographic scheme. For this reason it is a highly desirable goal.

For many cryptographic goals there are protocols known which establish provable security. But achieving provable security is often quite difficult, and schemes which achieve it are usually more complex and less efficient than their not-provably-secure counterparts. To address this problem the current authors suggested a few years back that the *random oracle model* could provide an effective tool to simultaneously achieve efficiency and something which is "close to" provable security [3]. The idea is to assume during algorithm design and analysis that all parties have access to a *public random oracle*— that is, a publicly-know "black box" which, on input of a string x, returns a *random* string $R(x)$ of some appropriate length. The *random-oracle paradigm* is to do provable security in this enriched model of computation and then, after a protocol and proof have been worked out, to *instantiate* the random oracle with an SHA-like hash function. The *thesis* underlying the random oracle paradigm is that substantial assurance remains despite the not-theoretically-justified instantiation step. For more details on this approach, see [3].

The buying of provable-security-style assurance without loss of efficiency has made the random oracle model an attractive choice for doing rigorous yet practical work in several cryptographic domains. In particular, the approach has been followed for asymmetric encryption [3,4] and digital signatures [3,5,19]. It is a particularly attractive approach for designing the sort of simple, efficient, as-high-assurance-as-possible schemes one wants for cryptographic standards.

One such standards effort is currently going on. The IEEE working group known as "P1363" has been drafting a *Standard for Public-Key Cryptography* [11]. This will be the first document owned by any standard-setting authority which provides general-purpose, bit-level specification for doing public key encryption, digital signatures, and key agreement using public-key techniques.

The P1363 committee has been considering several schemes (for both encryption and digital signatures) which are provably secure in the random-oracle model. This is a major gain in assurance over *ad. hoc.* design. All the same, some concerns have been voiced about the use of random oracles in P1363 schemes. The concerns are of two types: general questions about what a proof in the the random oracle model really means; and specific concerns about the way in which random oracles have been used in particular candidate schemes. In the next two subsections we address each of type of concern, in turn.

1.2 The security guarantee from proofs in the random-oracle model

It is important to neither over-estimate nor under-estimate what the random-oracle paradigm buys you in terms of security guarantees. Here we explain some of the issues and guarantees. See also [3].

Provable security in the random-oracle model is significantly different from (and fundamentally weaker than) provable-security in the standard model. At issue is the fact that when a scheme is designed assuming a random oracle R, and then this oracle is replaced by a concrete hash function F, there is no "standard" assumption on F which is adequate to ensure that F is a good-enough instantiation of R to cause no problems for the particular scheme. So