NUMBER 9 — FEBRUARY 23, 1999

RSA Laboratories' Bulletin

News and advice on data security and cryptography

A Note on the Security of the OAEP-Enhanced RSA Public-Key Encryption Scheme

Matthew Robshaw and Jessica Staddon
RSA Laboratories, San Mateo, California
Version 1.1

This note provides some background information on the security of the OAEP scheme [4] used in RSA PKCS #1 v2.0 [13]. We discuss the primary security issues for encryption schemes, and use these as a basis of comparison for OAEP and some other provably secure public-key encryption schemes.

In choosing between different public-key encryption schemes, the two most important issues to consider are (1) security and (2) cost (practicality). The latter issue is readily quantified and most experts can agree on which systems are likely to offer performance advantages. Our purpose in this note is to clarify the primary considerations involved in analyzing security, as assessing the security of different cryptosystems is far less straightforward than determining their cost.

There are two approaches to assessing the security of a cryptosystem. One might be viewed as being *certificational*—a cryptosystem that withstands attack over several years might be viewed as secure. There are however no “guarantees” about the future. What makes assessing security tricky is that we do not know beforehand what kind of attack an adversary may decide to mount. One cannot test the system against all possible attacks, because there are infinitely many of them. Thus the certificational approach may fail because a clever adversary might find an attack that others did not think about. We have seen examples of this recently, such as Bleichenbacher’s attack on the use of RSA PKCS #1 encryption in SSL [7].

The second approach for assessing security is sometimes called the *provable security* approach, and was initiated by Goldwasser and Micali in the early eighties [11]. This approach provides better guarantees. Many cryptographers see it as the only reliable way to get some assurance that a scheme will not be broken in the future.

How is it possible to get such an assurance, given that we don’t know what attack an adversary might mount? To understand this, we must understand more what provable security is about. In particular, the term itself is a bit misleading. Provable security does not provide an absolute guarantee of security. (Indeed, that is not possible given the state of current mathematical knowledge.) What it does show is that an ability to break the scheme guarantees an ability to solve some *other* well-known problem, P. Problem P might be the problem of efficiently factoring large integers, or it might be the problem of efficiently inverting the RSA function on random points. The success of a provable security result is measured by the “tightness” of the reduction from a break of the scheme to a solution of P. More precisely, a reduction is tight if when an adversary can break a scheme with probability ε, then there exists an algorithm that solves P with probability ε', where ε' is very close to ε. In other words, the provable security approach might be used to prove that if an adversary is able to break a system, then this yields a fast algorithm for factoring. Since we believe it is

Matthew Robshaw is principal research scientist at RSA Laboratories, and can be reached at matt@rsa.com. Jessica Staddon is a research scientist at RSA Laboratories, and can be reached at jstaddon@rsa.com.

RSA Laboratories
A Division of RSA Data Security

hard to factor, we must then believe the system is unbreakable.

One should appreciate that this is actually a very strong guarantee, because it says that, as long as factoring remains hard, the adversary will fail in the future even though we do not know beforehand exactly what strategy the adversary will employ. Another way to view such a result is that it says the best possible attack on a provably secure RSA-based scheme is to try to factor the RSA modulus.

Provably secure systems are somewhat underused in practice. The reason is that high security guarantees have a cost: these schemes are often much less efficient than heuristic ones, leaving system designers with a difficult choice between security and pragmatism.

Recently, an approach was developed to get assurance benefits akin to provable security without paying the usual cost. It is called the random oracle paradigm [5]. The idea is to make use of hash functions that are assumed in the analysis to behave randomly. These hash functions are instantiated carefully and appropriately in the actual scheme with functions derived from cryptographic hash functions like SHA-1 [1]. The random oracle paradigm is a bridge between theory and practice. It makes strong assumptions about the hash functions, so that the provable security is not in the usual sense. Yet it continues to offer assurance benefits.

OAEP-enhanced RSA [4] is an encryption scheme that has significant security within the random oracle paradigm. First, the security of the scheme can be tightly bound to the security of the RSA function (i.e. a provable security type of result). Second, it resists the strongest types of attacks, namely, adaptive chosen-ciphertext attacks. Moreover it is just as efficient as previous schemes that do not provide these assurances, such as that of RSA PKCS #1 v1.5.

How effective are these random oracle paradigm provable-security guarantees? The approach has worked well in practice, in the sense that schemes that are provably secure in this way have always resisted attack. OAEP is a case in point. It was designed in 1994. Since then several novel attacks on cryptographic protocols have been proposed, but in each case it was found that OAEP resists these attacks. This resistance is in spite of the fact that the designers of OAEP did not anticipate the specific attacks that would emerge. A case in point is Bleichenbacher's attack [7]. Even though the designers of OAEP did not know about such an attack when they designed their scheme in 1994, it turns out that OAEP is resistant to Bleichenbacher's attack. This protection against new and unanticipated attacks is the guarantee provable security provides.

How does OAEP compare to other provably secure schemes? As previously mentioned, security and cost are the primary areas of consideration. With respect to security, there are three main points of comparison. The first is *assumptions*. Remember that any provably secure scheme is proven secure based on some assumption, such as factoring being hard or RSA being a one-way function. The more confidence we have in the validity of the assumption, the better. That is, the weaker the assumption, the better. It is also important to consider the benefits gained by making the assumption. Therefore, the second point is the quantitative strength (or tightness) of the provable security result that can be obtained with the assumption. OAEP-enhanced RSA obtains quite a strong provable security result based on the assumption that hash functions behave randomly and the RSA function is one-way.

Moving to a different level, a third point of consideration is the type of attack that can be mounted on an encryption scheme. An adaptive chosen-ciphertext attack is perhaps the most powerful attack an adversary can mount. A provably secure scheme that is resistant to adaptive chosen ciphertext attack is, hence, very desirable. We know that breaking such a scheme is as hard as solving some previously known hard problem (the provable security) and that a class of strong attacks are all ineffective against the scheme. Within the random oracle paradigm, security against such attacks is implied by the property of *plaintext-awareness*. Informally, an encryption scheme is plaintext-aware if it is infeasible to construct a valid ciphertext without already knowing the corresponding plaintext. More precisely, in a plaintext-aware scheme it is possible to decrypt a ciphertext generated by an adversary by simply observing the oracle queries that the adversary makes (recall that the oracle is public). OAEP-enhanced RSA is plaintext-aware. Therefore, in addition to being as hard to break as the RSA function, it is known that (within the random oracle model) a powerful class of attacks fail against OAEP-enhanced RSA.

Dolev, Dwork and Naor (DDN) [10] have designed schemes that, like OAEP, resist adaptive chosen-ciphertext attack. The security of an RSA-based version of their scheme can be tightly reduced to the assumption that RSA is a one-way function. The provable security guarantee provided in the DDN scheme is superior to that of OAEP, because the assumption under which security is proven is weaker: the OAEP analysis assumes (in addition to assuming RSA is one-way) that some hash functions behave randomly.

Like OAEP, the DDN scheme is also resistant to chosen-ciphertext attack. The scheme achieves this through a security concept that is similar to plaintext-awareness but is defined in the standard (i.e. random oracle devoid) model: *non-malleability*. An encryption scheme is non-malleable if it is computationally infeasible to create a ciphertext whose plaintext relates in some predictable way to the plaintext of a previously known ciphertext. Interestingly, non-malleability has been shown to be equivalent to security against adaptive chosen-ciphertext attacks (see [3] and [10, 1998]), however, a comparison of non-malleability and plaintext-awareness is somewhat problematic because of the different models in which they are defined.

As we have seen, the DDN scheme offers similar security to OAEP with a weaker assumption. The two schemes differ dramatically, however, when we consider the final issue, cost. The cost of the DDN scheme is prohibitive. The scheme involves expensive zero-knowledge proofs and signatures, and is many orders of magnitude slower than OAEP. Indeed, it was to try to get these kinds of guarantees without the large cost that the random oracle paradigm was introduced.

Recently, Cramer and Shoup [9] introduced a new provably secure encryption scheme that also thwarts chosen-ciphertext attacks through non-malleability. Unlike the schemes we have been discussing up to now, it is not RSA based: it assumes the hardness of a certain version of the Diffie-Hellman problem. In terms of cost, the CS scheme is much cheaper than the DDN scheme, but is still more expensive than OAEP. (Encryption in OAEP is only a few multiplications if a small RSA exponent is used; while in the CS scheme it is a few exponentiations. Decryption in the CS scheme is about five times as costly as in OAEP.)

How does the security of the CS scheme compare to that of OAEP? That is more difficult to assess. The CS scheme does not use the random oracle paradigm, which is a plus. But it assumes the hardness of the so-called decisional Diffie-Hellman problem. In particular, it is shown that the CS scheme is non-malleable by proving that contradicting this property would require the ability to solve the decisional Diffie-Hellman problem. (See [6] for a nice discussion of this problem.) This is a strong assumption, and a relatively new and unstudied one in comparison to the assumption that RSA is one-way. (It would be much more surprising if the RSA assumption failed than if the decisional Diffie-Hellman assumption failed.) In particular, we do not know how this assumption compares to the assumption underlying OAEP. So, while the fact that the CS scheme avoids random oracles is a point in its favor, it is not really possible to say that one of these schemes has better security guarantees than the other in practice, because the assumptions are incomparable.

While some researchers rightfully point out the need to exercise caution in using the random oracle paradigm [8], it appears today that this approach is yielding more efficient schemes than those provided by the standard provable-security approach. This is evidenced by the practical advantages of OAEP over the CS scheme.

Readers who are interested in knowing more about provable security in practice are referred to the survey [2].

Acknowledgments

The authors would like to thank Mihir Bellare and Phil Rogaway for a careful review of this note, and many insightful comments on the strengths and weaknesses of the random oracle paradigm.

References

[1] ANSI, “X9.30:2, Public Key Cryptography, The Secure Hash Algorithm (SHA-1)”, 1997.

[2] M. Bellare. Practice-Oriented Provable-Security. In Proceedings of the 1997 Information Security Workshop (ISW).

[3] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In Advances in Cryptology – Crypto ’98, pp. 26-45, Springer-Verlag, 1998.

[4] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption—How to Encrypt with RSA. In Advances in

Cryptology – Eurocrypt ’94, pp. 92-111, Springer-Verlag, 1994.

[5] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. Proceedings of the 1st ACM conference on Computer and Communication Security, ACM, 1993.

[6] D. Boneh. The Decision Diffie-Hellman Problem. Invited paper for the Third Algorithmic Number Theory Symposium (ANTS), Lecture Notes in Computer Science Vol.1423, Springer-Verlag, 1998.

[7] D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In Advances in Cryptology – Crypto ’98, pp. 1-12, Springer-Verlag, 1998.

[8] R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, Revisited. Proceedings of the 30th Annual Symposium on the Theory of Computing, ACM, 1998.

[9] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Advances in Cryptology – Crypto ’98, pp. 13-25, Springer-Verlag, 1998.

[10] D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography, 1998. (updated, full length version of paper that appeared in the 23rd Annual ACM Symposium on Theory of Computing, pp. 542-552, 1991).

[11] S. Goldwasser and S. Micali. Probabilistic Encryption. Journal of Computer and System Sciences 28, pp. 270-299, April 1984.

[12] R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2), pp. 120-126, February 1978.

[13] RSA Laboratories. PKCS #1 v2.0: RSA Cryptography Standard. September 1998.

For more information on this and other recent security developments, contact RSA Laboratories at one of the addresses below.

RSA Laboratories
20 Crosby Drive
Bedford, MA 01730 USA
781/687-7000
781/687-7213 (fax)
rsa-labs@rsa.com
http://www.rsa.com/rsalabs/