RSA Laboratories' CryptoBytes. Volume 5, No. 1 – Winter/Spring 2002, pages 9–19.

How to Encrypt Properly with RSA

David Pointcheval

Dépt d'Informatique, ENS – CNRS, 45 rue d'Ulm, 75230 Paris Cedex 05, France
E-mail: David.Pointcheval@ens.fr – URL: http://www.di.ens.fr/users/pointche

Abstract. In 1993, Bellare and Rogaway formalized the concept of a random oracle, imported from complexity theory for cryptographic purposes. This new tool allowed them to present several asymmetric encryption and signature schemes that are both efficient and provably secure (in the random oracle model). The Optimal Asymmetric Encryption Padding (OAEP) is the most significant application of the random oracle model to date. It gives an efficient RSA encryption scheme with a strong security guarantee (semantic security against chosen-ciphertext attacks). After Bleichenbacher's devastating attack on RSA–PKCS #1 v1.5 in 1998, RSA–OAEP became the natural successor (RSA–PKCS #1 v2.0) and thus a *de facto* international standard. Surprisingly, Shoup recently showed that the original proof of security for OAEP is incorrect. Without a proof, RSA–OAEP cannot be trusted to provide an adequate level of security. Luckily, shortly after Shoup's discovery a formal and complete proof was found in joint work by the author and others that reaffirmed the strong level of security provided by RSA–OAEP. However, this new security proof still does not guarantee security for key sizes used in practice due to the inefficiency of the security reduction (the reduction to inverting RSA takes quadratic time). Recent alternatives to OAEP, such as OAEP$^+$, SAEP$^+$, and REACT, admit more efficient proofs and thus provide adequate security for key sizes used in practice.

1 Asymmetric Encryption

In 1978, Rivest, Shamir, and Adleman proposed the first candidate trapdoor permutation [30]. A trapdoor permutation primitive is a function f that anyone can compute efficiently; however, inverting f is hard unless we are also given some "trapdoor" information. Given the trapdoor information, inverting f becomes easy. Naively, a trapdoor permutation defines a simple public key encryption scheme: the description of f is the public key and the trapdoor is the secret key. Unfortunately, encryption in this naive public key system is deterministic and hence cannot be secure, as discussed below.

Before we can claim that a cryptosystem is secure (or insecure) we must precisely define what security actually means. The formalization of security notions started around the time when RSA was proposed and took several years to converge (see [18] for a survey on this topic). Today, the accepted security requirement for an encryption scheme is called "semantic security against an adaptive chosen-ciphertext attack" [29] or IND–CCA for short. To understand this concept we point out that security is always defined in terms of two parameters: (1) the attacker's capabilities, namely what the attacker can do during the attack, and (2) the attacker's goals, namely what the attacker is trying to do.

1. Attacker's capabilities: The strongest attacker capability in the standard model is called "adaptive chosen-ciphertext attack" and is denoted by (CCA) [29]. This means that the adversary has the ability to decrypt any ciphertext of his choice except for some challenge ciphertext (imagine the attacker is able to exploit a decryption box that will decrypt anything except for some known challenge ciphertext).

2. Attacker's goal: The standard security goal is called "semantic security" [19] (also known as "indistinguishability of ciphertexts"), and is denoted by (IND). Roughly speaking, the attacker's goal is to deduce just one bit of information about the decryption of some given ciphertext. We say that a system is semantically secure if no efficient attacker can achieve this goal. We note that a deterministic encryption algorithm can never give semantic security.

An encryption scheme that is semantically secure under an adaptive chosen-ciphertext attack is said to be IND–CCA secure. IND–CCA security implies that even with full access to the decryption oracle, the attacker is not able to deduce one bit of information about the decryption of a given challenge ciphertext. IND–CCA may seem very strong, but such attacks are possible in some real world scenarios. In fact, CCA-like attacks have been used to break practical implementations, as we will see later. Furthermore, semantic security is required for high confidentiality, namely when the message space is limited (such as "yes" or "no", "buy" or "sell"). As a consequence, IND–CCA is accepted as the required security level for practical encryption schemes.

One can obtain many other security notions by combining different attacker goals with various attacker capabilities. For example, another security goal is called "non-malleability" [15, 7]. Here the attacker is given some ciphertext and his goal is to build another ciphertext such that the plaintexts are meaningfully related. Non-malleability is known to be equivalent to semantic security under an adaptive chosen-ciphertext attack [3]. For this reason, IND–CCA security is sometimes called non-malleability. Similarly, one can also consider different attacker capabilities based on the oracles given to the attacker [25, 29, 9, 20, 26]. As mentioned above, the most powerful attacker capability in the "classical" model is the decryption oracle itself, which decrypts any ciphertext (except the challenge ciphertext). This "classical" model gives the cryptographic engine to the adversary as a black box to which he can make queries and receive correct answers in constant time. It thus excludes timing attacks [21], simple and differential power analyses [22] as well, and other differential fault analyses [8, 12].

2 The RSA-based Cryptosystems

2.1 The Plain RSA

The RSA permutation, proposed by Rivest, Shamir and Adleman [30], is the most well known trapdoor permutation. Its one-wayness is believed to be as strong as integer factorization. The RSA setup consists of choosing two large prime numbers p and q, and computing the RSA modulus $n = pq$. The public key is n together with an exponent e (relatively prime to $\varphi(n) = (p-1)(q-1)$). The secret key d is defined to be the inverse of e modulo $\varphi(n)$. Encryption and decryption is defined as follows:

$$\mathcal{E}_{n,e}(m) = m^e \bmod n \qquad \mathcal{D}_{n,d}(c) = c^d \bmod n.$$

This primitive does not provide by itself an IND–CCA secure encryption scheme. Under a slightly stronger assumption than the intractability of the integer factorization, it gives a cryptosystem that is only one-way under chosen-plaintext attacks – a very weak level of security. Semantic security fails because encryption is deterministic. Even worse, under a CCA attack, the attacker can fully decrypt a challenge ciphertext $C = m^e \bmod n$ using the homomorphic property of RSA:

$$\mathcal{E}_{n,e}(m_1) \cdot \mathcal{E}_{n,e}(m_2) = \mathcal{E}_{n,e}(m_1 m_2 \bmod n) \bmod n.$$

To decrypt $C = m^e \bmod n$ using a CCA attack do: (1) compute $C' = C \cdot 2^e \bmod n$, (2) give C' $(\neq C)$ to the decryption oracle, and (3) the oracle returns $2m \bmod n$ from which the adversary can deduce m.

To overcome RSA this simple CCA attack, practical RSA-based cryptosystems randomly pad the plaintext prior to encryption. This randomizes the ciphertext and eliminates the homomorphic property.

2.2 The RSA–PKCS #1 v1.5 Encryption

A widely deployed padding for RSA-based encryption is defined in the PKCS #1 v1.5 standard: for any modulus $2^{8(k-1)} \leq n < 2^{8k}$, in order to encrypt an ℓ byte-long message m (for $\ell \leq k - 11$), one randomly chooses a $k - 3 - \ell$ byte-long random string r (with only non-zero bytes). Then, one defines the k-byte long string $M = 02\|r\|0\|m$ (see figure 1) which is thereafter encrypted with the RSA permutation, $C = M^e \bmod n$. When decrypting a ciphertext C, the decryptor applies RSA inversion by computing $M = C^d \bmod n$ and then checks that the result M matches the expected format $\mathsf{02}\| * \|\mathsf{0}\| *$. If so, the decryptor outputs the last part as the plaintext. Otherwise, the ciphertext is rejected.

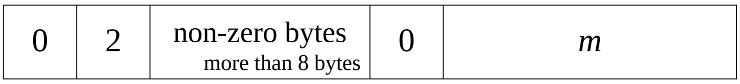


Fig. 1. PKCS #1 v1.5 Format

Intuitively, this padding seems sufficient to rule out the above weaknesses of the plain RSA system, but without any formal proof or guarantee. Surprisingly, in 1998, Bleichenbacher [9] showed that a simple active attack can completely break RSA–PKCS #1. This attack applies to real systems such as a Web server using SSL v3.0. These servers often output a specific "failure" message in case of an invalid ciphertext. This enables an attacker to test whether the two most significant bytes of a challenge ciphertext C are equal to '02'. If so, the attacker learns the following bound on the decryption of C:

$$2 \cdot 2^{8(k-2)} \leq C^d \bmod n < 3 \cdot 2^{8(k-2)}.$$

Due to the random self-reducibility of the RSA permutation, in particular the homomorphism $Cs^e = M^e s^e = (Ms)^e \bmod n$, the complete decryption of C can be recovered after a relatively small number of queries. Only a few million queries are needed with a 1024-bit modulus.

Bleichenbacher's attack had an impact on many practical systems and standards bodies, which suddenly became aware of the importance of formal security arguments. Nevertheless, the weak PKCS #1 v1.5 padding is still used in the TLS protocol [33]. The TLS specification now appears to defend against Bleichenbacher's attack using a technique for which no proof of security has yet been published. Certain simple attacks are still possible (for example, plaintext-checking attacks [26] can be easily run, even if they seem ineffective). The lesson here is that standards should rely as much as possible on fully analyzed constructions and avoid ad-hoc techniques.

3 The Optimal Asymmetric Encryption Padding

For some time, people have tried to provide security proofs for cryptographic protocols in the “reductionist” sense [10]. To do so, one presents an algorithm that uses an effective adversary as a sub-program to break some underlying hardness assumption (such as the RSA assumption, or the intractability of the integer factorization). Such an algorithm is called a “reduction”. This reduction is said to be efficient, roughly speaking, if it does not require too many calls to the sub-program.

3.1 The Random Oracle Model

A few years ago, a new line of research started with the goal of combining provable security with efficiency, still in the “reductionist” sense. To achieve this goal, Bellare and Rogaway [4] formalized a heuristic suggested by Fiat and Shamir [16]. This heuristic consisted in making an idealized assumption about some objects, such as hash functions, according to which they were assumed to behave like truly random functions. This assumption, known as the “random oracle model”, may seem strong, and lacking in practical embodiments. In fact, Canetti et al. [13] gave an example of a signature scheme which is secure in the random oracle model, but insecure under any instantiation of the random oracle.

However, one can also consider random-oracle-based proofs under the assumption that the adversary is generic, whatever the actual implementation of the hash function or other idealized algorithms may be. In other words, we may assume that the adversary does/can not use any specific weakness of the hash functions used in practice. Thanks to this ideal assumption, several efficient encryption and signature schemes have been analyzed [5, 6, 27].

We emphasize that even formal analyses in the random oracle model are not strong security proofs, because of the underlying ideal assumption. They do, however, provide strong evidence for security and can furthermore serve as the basis for quite efficient schemes. Since people do not often want to pay more than a negligible price for security, such an argument for practical schemes is more useful than formal security proofs for inefficient schemes.

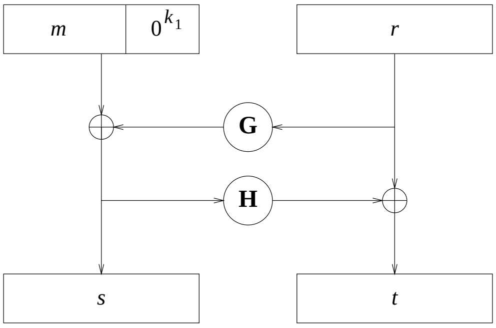


Fig. 2. OAEP Padding

3.2 Description of OAEP

At the time Bleichenbacher published his attack on RSA–PKCS #1 v1.5, the only efficient and "provably secure" encryption scheme based on RSA was the Optimal Asymmetric Encryption Padding (OAEP) proposed by Bellare and Rogaway [5]. OAEP can be used with any trapdoor permutation f. To encrypt a message m using the encryption scheme f–OAEP, first apply the OAEP procedure described in Figure 2 Here r is a random string and G, H are hash functions. The resulting values $[s\|t]$ are then encrypted using f, namely $C = f(s,t)$.

Bellare and Rogaway proved that OAEP padding used with any trapdoor permutation f provides a semantically secure encryption scheme. By adding some redundancy (the constant value 0^{k_1} at the end of the message, as shown in Figure 2), they furthermore proved it to be *weakly* plaintext-aware. Plaintext-awareness is a property of encryption schemes in the random oracle model which means that there exists a plaintext-extractor able to simulate the decryption oracle on any ciphertext (valid or not) designed by the adversary. The *weak* part in the definition proposed by Bellare and Rogaway was that the plaintext-extraction was just required to work while the adversary had not received any valid ciphertext from any source. Unfortunately, the adaptive chosen-ciphertext attack model gives the adversary a full-time access to the decryption oracle, even after receiving the challenge ciphertext about which the adversary wants to learn information. This challenge is a valid ciphertext. Therefore, semantic security together with *weak* plaintext-awareness only implies the semantic security against non-adaptive chosen-ciphertext attacks (a.k.a. lunchtime attacks [25], or *indifferent* chosen-ciphertext attacks), where the decryption oracle access is limited until the adversary has received the challenge ciphertext.

In 1998, Bellare, Desai, Rogaway and the author [3] corrected this initial definition of plaintext-awareness, requiring the existence of a plaintext-extractor able to simulate the decryption oracle on any ciphertext submitted by the adversary, even after seeing some valid ciphertexts not encrypted by the adversary himself. This stronger definition is a more accurate model of the real world, where the adversary may have access to ciphertexts via eavesdropping. We furthermore proved that this new property (which can only be defined in the random oracle model) actually provides the encryption scheme with the strongest security level, namely semantic security against (adaptive) chosen-ciphertext attacks (IND–CCA). However, no one ever provided OAEP with such a new plaintext-extractor. Therefore, even if everybody believed in the strong security level of OAEP, it had never been proven IND–CCA under the one-wayness of the permutation alone.

3.3 The OAEP Security Analyses

In fact, the only formally proven security result about OAEP was its semantic security against lunchtime attacks, assuming the one-wayness of the underlying permutation. Until very recently OAEP was widely believed to also be IND–CCA.

Shoup's Result Shoup [32] recently showed that it was quite unlikely that OAEP is IND–CCA assuming only the one-wayness of the underlying trapdoor permutation. In fact, he showed that if there exists a trapdoor one-way permutation g for which it is easy to compute $g(x \oplus a)$ from $g(x)$ and a, then OAEP cannot be IND–CCA secure for an arbitrary trapdoor permutation f. Referring to this special property of g as "XOR malleability", let us briefly present Shoup's counter-example. Let $s\|t$ denote the output of the OAEP transformation on a plaintext message m. Define the one-way permutation f as $f(s\|t) = s\|g(t)$. Then encrypting m using f–OAEP gives the ciphertext $C = [s\|g(t)]$.

What Shoup showed is that under these conditions the adversary can use C to construct a ciphertext C' of a plaintext message m' that is closely related to the message m. In particular, for any string δ, the adversary can construct C' which is the encryption of $m' = m \oplus \delta$. Thus, the scheme is malleable and hence not IND–CCA – giving C' to the decryption oracle will reveal $m' = m \oplus \delta$, from which the adversary can obtain m.

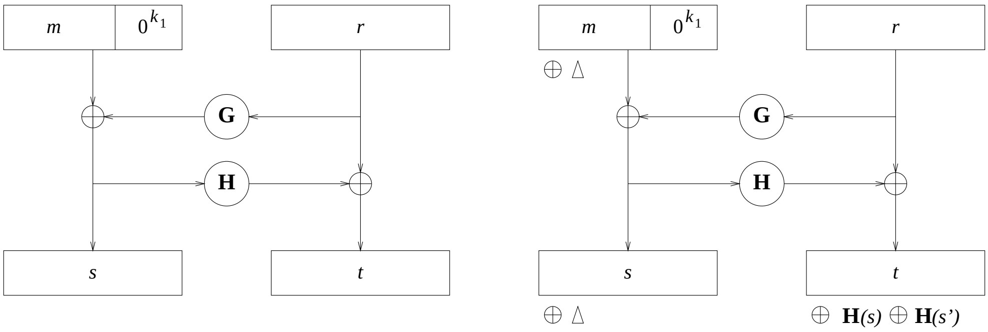


Fig. 3. Shoup's Attack

To construct C', the idea is for the adversary to exploit the explicit appearance of s in the ciphertext C. The adversary first computes $s' = s \oplus \Delta$, where $\Delta = \delta\|0^{k_1}$; essentially, Δ is simply a padded rendering of δ. The adversary then computes $D = H(s) \oplus H(s')$ using explicit knowledge of s and s' and access to the random oracle for H. Finally, by exploiting the "XOR malleability" of g, the adversary computes $g(t')$, where $t' = t \oplus D$. It is easy to see now that $C' = s'\|g(t')$ is a valid encryption of the message m'. Hence, the non-malleability of f–OAEP is broken.

This observation shows that it is unlikely that one can prove that f–OAEP is IND–CCA secure for arbitrary trapdoor permutations f by assuming only the one-wayness of f.

Repairing the OAEP Proof of Security To construct a valid ciphertext C' in the above attack it seems that the adversary has to query the hash function H at $H(s)$. But this seems to imply that given C the adversary can figure out the value s used to create C (recall that s is the left hand side of $f^{-1}(C)$). Thus, it appears that in order to mount Shoup's attack the adversary must be able partly to invert f – given $f(s,t)$, the adversary must be able to expose s.

We say f is partial-domain one-way if no efficient algorithm can deduce s from $C = f(s,t)$. For such trapdoor permutations f, one could hope that Shoup's attack will fail and that f–OAEP is IND–CCA secure. Fujisaki, Okamoto, Stern and the author [17] formally proved this fact: If f is partial-domain one-way, then f–OAEP is IND–CCA secure. We note that partial-domain one-wayness is a stronger property than one-wayness: a function might be one-way but still not partial-domain one-way.

Fortunately, the homomorphic properties of RSA enable us to prove that the RSA permutation is partial-domain one-way if and only if RSA is one-way. More precisely, an algorithm that can expose half of $\mathrm{RSA}^{-1}(C)$ given C can be used to completely invert the RSA permutation. Altogether, this proves the widely believed IND–CCA security of RSA–OAEP assuming that RSA is a trapdoor permutation. For security parameters, and t (whose formal definitions are omitted here), we obtain the following result [17]:

> Let $\mathcal{A}$ be a CCA-adversary against the "semantic security" of RSA–OAEP with running time bounded by t and advantage ε. Then, the RSA function can be inverted with probability greater than approximately $\varepsilon^2/4$ within time bound $2t$.

Unfortunately, the security reduction from an RSA-inversion into an attack is quite inefficient for practical sizes (more precisely, it is quadratic in the number of oracle queries). Hence, this reduction is meaningless unless one uses a modulus large enough so that the RSA-inversion (or the factorization) requires much more than 2^{150} computational effort. With current factorization techniques [23, 14], one needs to use a modulus of length more than 4096 bits to make the reduction meaningful (see [24] for complexity estimates of the most efficient factoring algorithms). Viewed another way, this reduction shows that a 1024-bit modulus just provides a provable security level of 2^{40}, which is clearly inadequate given currently prevalent levels of computing power. (We note, however, that this does not mean that there is an attack with this low complexity, only that one cannot be ruled out by the available proofs of security.)

4 OAEP Alternatives

4.1 The OAEP$^+$ Padding

Shoup also proposed a formal security proof of RSA–OAEP with a much more efficient security reduction, but in the particular case where the encryption exponent e is equal to 3. However, many people believe that the RSA trapdoor permutation with exponent 3 may be weaker than with greater exponents. Therefore, he also proposed a slightly modified version of OAEP, called OAEP$^+$ (see Figure 4), which can be proven secure under the one-wayness of the permutation alone. It uses the variable redundancy $R(m,r)$ instead of the constant 0^{k_1}. It is thus a bit more intricate than the original OAEP. The security reduction for OAEP$^+$ is efficient, but still runs in quadratic time.

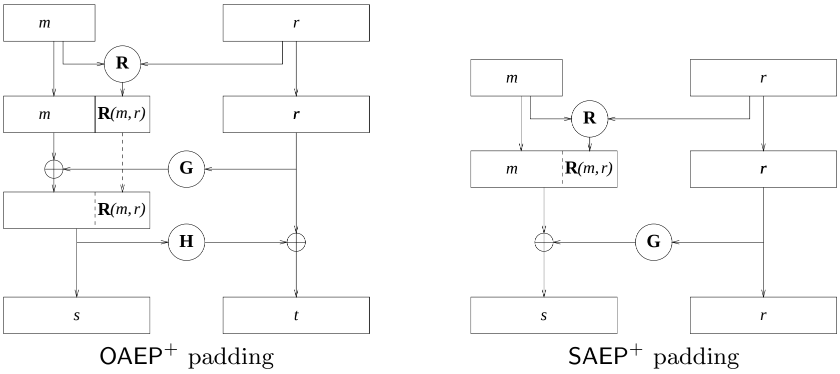


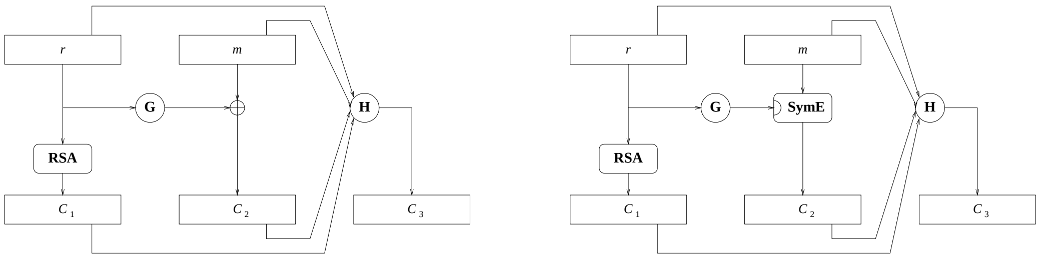
Fig. 4. OAEP$^+$ and SAEP$^+$ Paddings

4.2 SAEP$^+$ Padding

Boneh [11] recently proposed a new padding scheme, SAEP$^+$, to be used with the Rabin primitive [28] or RSA. It is simpler than OAEP, hence the name Simplified Asymmetric Encryption Padding: whereas OAEP is a two-round Feistel network, SAEP$^+$ is a single-round. SAEP$^+$ has a linear time reduction for the Rabin system (i.e., $e = 2$). For larger exponents, SAEP$^+$ has a quadratic time reduction. Hence, for larger exponents ($e > 2$), SAEP$^+$ does not guarantee security for practical parameters (less than two thousand bits).

4.3 The REACT Construction

Another alternative to OAEP is the REACT construction, proposed by Okamoto and the author [26] (see Figure 5). It provides an IND–CCA encryption scheme from any weakly secure one (more precisely, a one-way primitive, against plaintext-checking attacks), such as the RSA primitive. Therefore, the RSA–REACT scheme is IND–CCA secure under the RSA assumption.



Basic encryption

Hybrid encryption

Fig. 5. REACT

Furthermore, the security reduction is very efficient, since it is in linear time without any loss in the success probability, whatever the exponent. Consequently, it guarantees perfect equivalence with RSA inversion for moduli which require just a bit more than 2^{70} effort to be factored. This is the case for 1024 bit-long moduli, the minimal currently advised key size. In comparison to previous proposals, REACT is a full scheme and not just a pure padding applied to the message before the RSA function.

Consequently, the ciphertext is a bit longer. However, even when used for key transport, it allows integration of a symmetric encryption scheme (SymE) to achieve very high encryption rates, as shown in the hybrid construction. In the specific case of RSA, REACT can be optimized, as explained below.

4.4 Simple RSA

In an ISO report [31], Shoup suggested a possible alternative, based on ideas from Bellare and Rogaway [4] that provide a secure encryption scheme from any trapdoor one-way permutation f. Roughly speaking, "simple RSA", as it is called, consists of first encrypting a random string r using f to obtain C_0 (thus $C_0 = r^e \bmod n$), and then parsing $G(r)$ as $k_0 \| k_1$, where G is some hash function (modeled by a random oracle). Thereafter, one encrypts the message m using a symmetric encryption scheme with the key k_0 to get C_1 (e.g., $C_1 = m \oplus k_0$), and authenticates the ciphertext with a MAC function H using the key k_1 to get a tag $T = H(k_1, C_1)$. The ciphertext is the triple (C_0, C_1, T). This construction is a special case of REACT, optimized for RSA, and hence is IND–CCA under the RSA assumption. It provides a very efficient linear time reduction. Moreover, thanks to the random self-reducibility of RSA (which can only be used with this latter construction, but cannot with the OAEP and SAEP variants), this construction provides a high security level even when encrypting many plaintexts [1, 2].

5 Conclusion

RSA–OAEP is a practical RSA encryption scheme with provable security in the random oracle model. For practical security, the cost of the reductions cannot simply be shown to be polynomial time (as in asymptotical analyses), since the reduction efficiency directly impacts the security parameters needed for the scheme. Hence, when evaluating cryptographic constructions, one must take into account the efficiency of the security proof. Inefficient proofs of security do not give security guarantees for real world parameters.

Only OAEP with exponents 2 or 3, SAEP$^+$ with exponent 2, and RSA–REACT (or the optimization “simple RSA”) with any exponent, admit formal proofs with linear time reductions in the random oracle model. Hence only these schemes guarantee semantic security against chosen-ciphertext attacks for practical modulus sizes (even less than 1024 bits). The provable security for other padding schemes is meaningful only for much larger moduli (more than 4096 bits).

Acknowledgments

I warmly thank my co-authors, Mihir Bellare, Anand Desai, Eiichiro Fujisaki, Tatsuaki Okamoto, Phil Rogaway and Jacques Stern for the interesting works we did on asymmetric encryption, as well as Dan Boneh, Pierre-Alain Fouque, Victor Shoup and Yves Verhoeven for the fruitful discussions we had.

References

1. O. Baudron, D. Pointcheval, and J. Stern. Extended Notions of Security for Multicast Public Key Cryptosystems. In *Proc. of the 27th ICALP*, LNCS 1853, pages 499–511. Springer-Verlag, Berlin, 2000.
2. M. Bellare, A. Boldyreva, and S. Micali. Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Eurocrypt '00*, LNCS 1807, pages 259–274. Springer-Verlag, Berlin, 2000.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
4. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
5. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
6. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
7. M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.

8. E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *Crypto '97*, LNCS 1294, pages 513–525. Springer-Verlag, Berlin, 1997.
9. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
10. M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudorandom Bits. *SIAM Journal on Computing*, 13:850–864, 1984.
11. D. Boneh. Simplified OAEP for the RSA and Rabin Functions. In *Crypto '01*, LNCS 2139, pages 275–291. Springer-Verlag, Berlin, 2001.
12. D. Boneh, R. DeMillo, and R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *Eurocrypt '97*, LNCS 1233, pages 37–51. Springer-Verlag, Berlin, 1997.
13. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracles Methodology, Revisited. In *Proc. of the 30th STOC*, pages 209–218. ACM Press, New York, 1998.
14. S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, Ch. Putnam, Cr. Putnam, and P. Zimmermann. Factorization of a 512-bit RSA Modulus. In *Eurocrypt '00*, LNCS 1807, pages 1–18. Springer-Verlag, Berlin, 2000.
15. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
16. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, Berlin, 1987.
17. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Secure under the RSA Assumption. In *Crypto '01*, LNCS 2139, pages 260–274. Springer-Verlag, Berlin, 2001.
18. O. Goldreich. On the Foundations of Modern Cryptography. In *Crypto '97*, LNCS 1294, pages 46–74. Springer-Verlag, Berlin, 1997.
19. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
20. C. Hall, I. Goldberg, and B. Schneier. Reaction Attacks Against Several Public-Key Cryptosystems. In *Proc. of ICICS '99*, LNCS, pages 2–12. Springer-Verlag, 1999.
21. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Crypto '96*, LNCS 1109, pages 104–113. Springer-Verlag, Berlin, 1996.
22. P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Crypto '99*, LNCS 1666, pages 388–397. Springer-Verlag, Berlin, 1999.
23. A. Lenstra and H. Lenstra. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
24. A. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. In *PKC '00*, LNCS 1751, pages 446–465. Springer-Verlag, Berlin, 2000.
25. M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In *Proc. of the 21st STOC*, pages 33–43. ACM Press, New York, 1989.
26. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT – RSA '01*, LNCS 2020, pages 159–175. Springer-Verlag, Berlin, 2001.
27. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
28. M. O. Rabin. Digitalized Signatures. In R. Lipton and R. De Millo, editors, *Foundations of Secure Computation*, pages 155–166. Academic Press, New York, 1978.
29. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
30. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
31. V. Shoup. A Proposal for an ISO Standard for Public-Key Encryption, december 2001. ISO/IEC JTC 1/SC27.
32. V. Shoup. OAEP Reconsidered. In *Crypto '01*, LNCS 2139, pages 239–259. Springer-Verlag, Berlin, 2001.
33. T. Dierks and C. Allen. The TLS Protocol, january 1999. RFC 2246
Available from `http://www.ietf.org/rfc.html`.