

RSA Problem

Ronald L. Rivest, MIT Laboratory for Computer Science

`rivest@mit.edu`

and Burt Kaliski, RSA Laboratories

`bkaliski@rsasecurity.com`

December 10, 2003

1 Introduction

In RSA public-key encryption [30], Alice encrypts a plaintext M for Bob using Bob's public key (n, e) by computing the ciphertext

$$C = M^e \pmod{n} . \tag{1}$$

where n , the *modulus*, is the product of two or more large primes, and e , the *public exponent*, is an odd integer $e \geq 3$ that is relatively prime to $\phi(n)$, the order of the multiplicative group \mathbf{Z}_n^* .

Bob, who knows the corresponding RSA private key (n, d) , can easily decrypt, since $de = 1 \pmod{\phi(n)}$ implies that

$$M = C^d \pmod{n} . \tag{2}$$

An adversary may learn C by eavesdropping, and may very well also know Bob's public key; nonetheless such an adversary should not be able to compute the corresponding plaintext M .

One may formalize the task faced by this adversary as the *RSA Problem*:

The RSA Problem: Given an RSA public key (n, e) and a ciphertext $C = M^e \pmod{n}$, to compute M .

To solve the RSA Problem an adversary, who doesn't know the private key, must nonetheless invert the RSA function.

The *RSA Assumption* is that the RSA Problem is hard to solve when the modulus n is sufficiently large and randomly generated, and the plaintext M (and hence the ciphertext C) is a random integer between 0 and $n - 1$. The assumption is the same as saying that the RSA function is a trapdoor one-way function (the private key is the trapdoor).

The randomness of the plaintext M over the range $[0, n - 1]$ is important in the assumption. If M is known to be from a small space, for instance, then an adversary can solve for M by trying all possible values for M .

The RSA Problem is the basis for the security of RSA public-key encryption as well as RSA digital signature schemes.

See also surveys by Boneh [10] and Katzenbeisser [24].

2 Relationship to integer factoring

The RSA Problem is clearly no harder than integer factoring, since an adversary who can factor the modulus n can compute the private key (n, d) from the public key (n, e) .

However, it is not clear whether the converse is true, that is, whether an algorithm for integer factoring can be efficiently constructed from an algorithm for solving the RSA Problem.

Boneh and Venkatesan [9] have given evidence that such a construction is unlikely when the public exponent is very small, such as $e = 3$ or 17. Their result means that the RSA Problem for very small exponents could be easier than integer factoring, but it doesn't imply that the RSA Problem is actually easier, i.e., efficient algorithms are still not known. For larger public exponents, the question of equivalence with integer factoring still open as of this writing.

3 Recovering the private key

Clearly, if the adversary could compute Bob's private key (n, d) from his public key (n, e) , then the adversary could decrypt C using equation (2).

However, de Laurentis [15] and Miller [27] have shown that computing an RSA private key (n, d) from the corresponding RSA encryption key (n, e) is as hard as factoring the modulus n into its prime factors p and q . As already noted, given the factors p and q , it is easy to compute d from e , and

conversely there is a probabilistic polynomial-time algorithm which takes as input n , e , and d , and which factors n into p and q . (See also Fact 1 in Boneh [10].)

If the modulus n was chosen as the product of two “sufficiently large” randomly-chosen prime numbers p and q , then the problem of factoring n appears to be intractable. Thus, the private exponent d is protected from disclosure by the difficulty of factoring the modulus n .

An adversary might also try to compute d using some method of solving the discrete logarithm problem. For example, an adversary could compute the discrete logarithm of M to the base $M^e \pmod{n}$. If d is too small (say, less than 160 bits), then an adversary might be able to recover it by the baby-step-giant step method.

Even if d is too large to be recovered by discrete logarithm methods, however, it may still be at risk.

For example, Wiener [33] has shown that if the secret exponent is less than $n^{1/4}/3$, an adversary can efficiently compute d given n and e . An improved bound of $n^{0.292}$ has been presented by Boneh and Durfee [8].

However, it does appear to be the case that if the RSA parameters were chosen large enough, then the adversary can not solve the RSA Problem by computing the private RSA exponent of the recipient.

4 Self-reducibility

It is conceivable that someone could devise a clever procedure for solving the RSA Problem without factoring the modulus n or determining the private key d . An adversary might, for example, have a procedure that decrypts a small fraction of “weak” ciphertexts. However, the RSA procedure enjoys a certain kind of “self-reducibility”, since it is multiplicative:

$$(MR)^e = M^e R^e \pmod{n} .$$

An adversary can transform a given ciphertext M^e into another one $(MR)^e$ by multiplying it by the encryption R^e of a randomly chosen element R of \mathbf{Z}_n^* . Since the result has a chance of being a “weak” ciphertext, it follows that if there is an adversarial procedure A that can decrypt a fraction ϵ of ciphertexts, then there is another (randomized) adversarial procedure A' that can decrypt all ciphertexts in expected running time that is polynomial

in the running time of A , in $1/\epsilon$, and in $\log n$ (see polynomial time). (See Motwani and Raghavan [28, Section 14.4].)

Self-reducibility is a double-edged sword. On the one hand, it provides assurance that “all” random ciphertexts are equally hard to invert. This property has been helpful in the security proofs for several public-key encryption and signature schemes based on the RSA Problem. On the other hand, self-reducibility provides an avenue for an adversary to gain information about the decryption of one ciphertext from the decryption of other ciphertexts (see “chosen ciphertext attacks”) below.

5 Low public exponent RSA

A user of the RSA cryptosystem may reasonably wish to use a public exponent e that is relatively short: common choices are $e = 3$ or $e = 2^{16} + 1 = 65537$. Using a short public exponent results in faster public-key encryption and faster public-key signature verification. Does this weaken RSA?

If the public exponent is small and the plaintext M is very short, then the RSA function may be easy to invert: in particular, if $M < \sqrt[e]{N}$, then $C = M^e$ over the integers, so M can be recovered as $M = \sqrt[e]{C}$.

Håstad [22] shows that small public exponents can be dangerous when the same plaintext is sent to many different recipients, even if the plaintext is “padded” in various (simple) ways beforehand.

Coppersmith et al.[12] give a powerful “related messages” attack, which is effective when the public exponent is small, based on the LLL algorithm [25] for lattice reduction.

Because of these concerns, small public exponents are sometimes avoided in industry standards and in practice. However, the concerns can also be addressed with appropriate padding schemes (see “chosen ciphertext attacks” below), provided they are correctly implemented. For digital signature schemes, small public exponents are generally not an issue.

6 Strong RSA Assumption

The *Strong RSA Assumption* was introduced by Barić and Pfitzmann [3] and by Fujisaki and Okamoto [18] (see also [13]).

This assumption differs from the RSA Assumption in that the adversary

can select the public exponent e . The adversary’s task is to compute, given a modulus n and a ciphertext C , *any* plaintext M and (odd) public exponent $e \geq 3$ such that $C = M^e \pmod{n}$. This may well be easier than solving the RSA Problem, so the assumption that it is hard is a stronger assumption than the RSA Assumption. The Strong RSA Assumption is the basis for a variety of cryptographic constructions.

7 Bit-security of RSA encryption

It is conceivable that RSA could be “secure” in the sense that the RSA Assumption holds (i.e. RSA is hard to invert), yet that RSA “leaks” information in that certain plaintext bits are easy to predict from the ciphertext. Does RSA provide security to individual bits of plaintext?

Goldwasser et al. [21] first studied the bit-security of RSA, showing that an adversary who could reliably extract from a ciphertext the least significant bit of the plaintext would in fact be able to decrypt RSA efficiently (i.e. obtain the entire plaintext efficiently).

This line of research was pursued by other researchers. For example, Vazirani et al. [32]) showed that the adversary could still decrypt even with an lsb procedure that was only $0.732 + \epsilon$ accurate. They also showed that the low-order $\log(\log(n))$ bits of plaintext are $3/4 + \epsilon$ secure.

Chor and Goldreich [11] improved this result to show that the least-significant bit of RSA plaintext can not be predicted with probability better than $1/2 + 1/\text{poly}(\log(n))$ (under the RSA Assumption). Alexi et al. [1, 2] completed this result to show that the least-significant $\log(\log(n))$ bits are secure in the same sense. (Fischlin and Schnorr [17] provide a simpler and tighter proof of this result.)

Håstad and Näslund [23] have shown that *all* of the plaintext bits are well-protected by RSA, in the sense that having a nontrivial advantage for predicting any one plaintext bit would enable the adversary to invert RSA completely.

The results about bit-security of RSA generally involve a *reduction* technique (see computational complexity theory), where an algorithm for solving the RSA Problem is constructed from an algorithm for predicting one (or more) plaintext bits. Like self-reducibility, bit-security is a double-edged sword. This is because the security reductions also provide an avenue of attack on a “leaky” implementation. If an implementation of an RSA de-

encryption operation leaks some bits of the plaintext, then an adversary can potentially solve the RSA Problem for *any* ciphertext just by observing the implementation's behavior on some number of other ciphertexts. Such attacks have been described by Bleichenbacher [7] and by Manger [26].

8 Chosen ciphertext attacks

An adversary may be able to decrypt an RSA ciphertext C if he can obtain decryptions (e.g. from the legitimate recipient) of other ciphertexts C_1, C_2, \dots, C_k (which may or may not be related to C). Such attacks are known as chosen ciphertext attacks (CCA1 and CCA2, depending on whether the C_i 's are allowed to depend upon C (of course they can't be equal to C)); see Bellare et al. [4] for details.

(The attacks related to bit-security are a special case of chosen-ciphertext attacks in which the adversary only obtains partial information about the decryption, not the full plaintext.)

Davidson [14] first studied chosen ciphertext attacks for RSA, utilizing the multiplicative property of RSA.

Desmedt and Odlyzko [16] provided another chosen ciphertext attack, based on obtaining the decryption of many small primes.

To defeat chosen ciphertext attacks, researchers have turned to (possibly randomized) "padding" schemes that (reversibly) transform a plaintext before encryption.

One such proposal is Optimal Asymmetric Encryption Padding (OAEP) [5] which has been proven secure for chosen ciphertext attacks by Fujisaki et al. [19] under the RSA assumption. Other proposals that also avoid chosen ciphertext attacks have better security properties [29, 31]. See also [RSA public-key encryption](#) for related discussion.

Chosen-ciphertext attacks on [digital signature schemes](#) are the analogue to chosen ciphertext attacks on public-key encryption, and various padding schemes have been developed to defeat them as well, such as the Probabilistic Signature Scheme (PSS) of Bellare and Rogaway [6] and the scheme of Gennaro et al. [20]. See also [RSA digital signature scheme](#).

9 Conclusions

The RSA Problem is now over a quarter century old. The elegant simplicity of the problem has led to numerous observations over the years, some yielding attacks, others avoiding them. Public-key encryption schemes and digital signature schemes have been developed whose strength is derived fully from the RSA Problem. The remaining open question, still, is how closely the security of the RSA Problem depends on integer factoring, and as with any hard problem in cryptography, whether any methods faster than those currently available for solving the problem will ever be discovered.

References

- [1] W. B. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA/Rabin bits are $1/2 + 1/\text{poly}(\log(N))$ secure. In *Proc. FOCS '84*, pages 449–457, Singer Island, 1984. IEEE.
- [2] W. B. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA/Rabin functions: certain parts are as hard as the whole. *SIAM J. Computing*, 17(2):194–209, April 1988.
- [3] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, 1997.
- [4] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption. In H. Krawczyk, editor, *Proceedings Crypto '98*, pages 26–45. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.
- [5] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption—how to encrypt with RSA. In A. DeSantis, editor, *Proceedings Eurocrypt '94*, pages 92–111. Springer-Verlag, 1994. Lecture Notes in Computer Science No. 950.
- [6] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures—how to sign with RSA and Rabin. In U. Maurer, editor, *Proc. Eurocrypt '96*, pages 399–416. Springer Verlag, 1996.

- [7] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In H. Krawczyk, editor, *Proc. CRYPTO '98*, pages 1–12. Springer, 1998.
- [8] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46(4):1339–1349, July 2000.
- [9] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In K. Nyberg, editor, *Proc. EUROCRYPT '98*, pages 59–71. Springer, 1998.
- [10] Dan Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.
- [11] Benny Chor and Oded Goldreich. RSA/rabin least significant bits are $\frac{1}{2} + 1/\text{poly}(\log n)$ secure. In G. R. Blakley and D. C. Chaum, editors, *Proc. CRYPTO '84*, pages 303–313. Springer, 1985. Lecture Notes in Computer Science No. 196.
- [12] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-exponent RSA with related messages. In *Proc. EUROCRYPT 1996*, pages 1–9. Springer-Verlag, 1996. Lecture Notes in Computer Science No. 1070.
- [13] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.
- [14] G. Davida. Chosen signature cryptanalysis of the RSA (MIT) public key cryptosystem. Technical Report Tech Report TR-CS-82-2, Dept of EECS, University of Wisconsin, Milwaukee, Oct 1982.
- [15] J. M. DeLaurentis. A further weakness in the common modulus protocol for the RSA cryptoalgorithm. *Cryptologia*, 8:253–259, 1984.
- [16] Y. Desmedt and A. M. Odlyzko. A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes. In H. C. Williams, editor, *Proc. CRYPTO '85*, pages 516–522. Springer, 1986. Lecture Notes in Computer Science No. 218.

- [17] Roger Fischlin and Claus-Peter Schnorr. Stronger security proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244, 2000.
- [18] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *Proc. CRYPTO '97*, volume 1294 of *LNCS*, pages 16–30. Springer-Verlag, 1997.
- [19] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. *Journal of Cryptology*, ??(??):??–??, to appear.
- [20] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *Proc. EUROCRYPT '99*, pages 123–139. Springer-Verlag, 1999. Lecture Notes in Computer Science No. 1592.
- [21] S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network. In *Proc. FOCS '82*, pages 134–144, Chicago, 1982. IEEE.
- [22] J. Hastad. Solving simultaneous modular equations of low degree. *SIAM J. Computing*, 17:336–341, 1988.
- [23] Johan Håstad and Mats Näslund. The security of individual RSA bits. In *IEEE Symposium on Foundations of Computer Science*, pages 510–521, 1998.
- [24] Stefan Katzenbeisser. *Recent Advances in RSA Cryptography*. Kluwer Academic Publishers, 2001.
- [25] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
- [26] J. Manger. A chosen ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS #1 v2.0. In J. Kilian, editor, *Proc. CRYPTO 2001*, pages 260–274. Springer, 2001.
- [27] Gary L. Miller. Riemann’s hypothesis and tests for primality. *Journal of Computer and Systems Sciences*, 13(3):300–317, 1976.

- [28] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [29] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In D. Naccache, editor, *Proc. Cryptographers' Track RSA Conference (CT-RSA) 2001*, pages 159–175. Springer, 2001.
- [30] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [31] V. Shoup. *A Proposal for an ISO Standard for Public Key Encryption (Version 2.1)*. Manuscript, December 20, 2001. Available from <http://shoup.net/papers/>.
- [32] Umesh Vazirani and Vijay Vazirani. RSA bits are $.732 + \epsilon$ secure. In D. Chaum, editor, *Proc. CRYPTO '83*, pages 369–375. Plenum Press, 1984.
- [33] M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Trans. on Inform. Theory*, 36(3):553–558, May 1990.