RSA problem

In cryptography, the **RSA problem** summarizes the task of performing an RSA private-key operation given only the public key. The RSA algorithm raises a *message* to an *exponent*, modulo a composite number N whose factors are not known. Thus, the task can be neatly described as finding the e^{th} roots of an arbitrary number, modulo N. For large RSA key sizes (in excess of 1024 bits), no efficient method for solving this problem is known; if an efficient method is ever developed, it would threaten the current or eventual security of RSA-based cryptosystems—both for public-key encryption and digital signatures.

More specifically, the RSA problem is to efficiently compute P given an RSA public key (N, e) and a ciphertext $C \equiv P^{e} \pmod{N}$. The structure of the RSA public key requires that N be a large semiprime (i.e., a product of two large prime numbers), that $2 < e < N$, that e be coprime to $\varphi(N)$, and that $0 \leq C < N$. C is chosen randomly within that range; to specify the problem with complete precision, one must also specify how N and e are generated, which will depend on the precise means of RSA random keypair generation in use.

The most efficient method known to solve the RSA problem is by first factoring the modulus N, a task believed to be impractical if N is sufficiently large (see integer factorization). The RSA key setup routine already turns the public exponent e, with this prime factorization, into the private exponent d, and so exactly the same algorithm allows anyone who factors N to obtain the *private key*. Any C can then be decrypted with the private key.

Just as there are no proofs that integer factorization is computationally difficult, there are also no proofs that the RSA problem is similarly difficult. By the above method, the RSA problem is at least as easy as factoring, but it might well be easier. Indeed, there is strong evidence pointing to this conclusion: that a method to break the RSA method cannot be converted necessarily into a method for factoring large semiprimes.[1] This is perhaps easiest to see by the sheer overkill of the factoring approach: the RSA problem asks us to decrypt *one* arbitrary ciphertext, whereas the factoring method reveals the private key: thus decrypting *all* arbitrary ciphertexts, and it also allows one to perform arbitrary RSA private-key encryptions. Along these same lines, finding the decryption exponent d indeed *is* computationally equivalent to factoring N, even though the RSA problem does not ask for d.[2]

In addition to the RSA problem, RSA also has a particular mathematical structure that can potentially be exploited *without* solving the RSA problem directly. To achieve the full strength of the RSA problem, an RSA-based cryptosystem must also use a padding scheme like OAEP, to protect against such structural problems in RSA.

See also

- Strong RSA assumption
- RSA Factoring Challenge
- Rabin cryptosystem, whose equivalency to factoring is known

References

1. Boneh, Dan; Venkatesan, Ramarathnam (1998). "Breaking RSA may not be equivalent to factoring". *Advances in Cryptology — EUROCRYPT'98*. Lecture Notes in Computer Science. **1403**. Springer.

pp. 59–71. doi:10.1007/BFb0054117 (https://doi.org/10.1007%2FBFb0054117). ISBN 978-3-540-64518-4.

2. An algorithm for this is, for example, given in Menezes; van Oorschot; Vanstone (2001). "Public-Key Encryption" (http://www.cacr.math.uwaterloo.ca/hac/about/chap8.pdf) (PDF). *Handbook of Applied Cryptography*.

Further reading

- *Breaking RSA may be as difficult as factoring* (http://eprint.iacr.org/2005/380), D. Brown, 2005. This unrefereed preprint purports that solving the RSA problem using a Straight line program is as difficult as factoring provided *e* has a small factor.
- *Breaking RSA Generically is Equivalent to Factoring* (http://eprint.iacr.org/2008/260), D. Aggarwal and U. Maurer, 2008. This Eurocrypt 2009 paper (link is to a preprint version) proves that solving the RSA problem using a generic ring algorithm is as difficult as factoring.
- *When e-th Roots Become Easier Than Factoring* (http://eprint.iacr.org/2007/424), Antoine Joux, David Naccache and Emmanuel Thomé, 2007. This Asiacrypt 2007 paper (link is to a preprint version) proves that solving the RSA problem using an oracle to some certain other special cases of the RSA problem is easier than factoring.

Retrieved from "https://en.wikipedia.org/w/index.php?title=RSA_problem&oldid=994955809"

This page was last edited on 18 December 2020, at 12:15 (UTC).