CPA-security for Padded RSA

Tingxiang Zou

RSA is one of the first practicable and most widely used public-key encryption scheme. It was first publicly introduced by (also named after) Ron Rivest, Adi Shamir and Leonard Adleman in 1977, despite that in 1973, an English mathematician Clifford Cocks had designed an equivalent system, which was classified information, and therefore not released to public until 1997.[3]

Ron Rivest(left), Adi Shamir(middle) and Leonard Adleman(right)

Clifford Cocks

The security of RSA is basically based on one assumption, known as the **RSA assumption** (Definition 7.46 in [1]): given N, e and a randomly chosen $y \in Z_N^*$, where N is the product of two large primes p and q, e is coprime with $\phi(N) := (p-1)(q-1)$, no probabilistic polynomial-time Turing machine can find with non-negligible probability the $x \in Z_N^*$, such that $y = [x^e \mod N]$.

The basic RSA scheme (often called plain or textbook RSA) is the following: (Construction 10.15 in [1])

• **Key-generation:** on security parameter n, generate public key (N, e) and secret key (N, d), such that $N = p \cdot q$, p, q primes with length n, e coprime with $\phi(N) = (p-1)(q-1)$ and $d = [e^{-1} \mod \phi(N)]$;

• **Encryption:** given message $m \in Z_N^*$, $Enc_{(N,e)}(m) := [m^e \mod N]$

• **Decryption:** given ciphertext $c \in Z_N^*$, $Dec_{(N,d)}(c) := [c^d \mod N]$.

However, plain RSA is vulnerable to all kinds of attacks. In particular, it is a deterministic encryption scheme, hence can never be chosen-plaintext secure (CPA-secure). One simple way to modify is to randomly pad the message before encrypting. We call this kind of schemes

padded RSA.

The problem is how long the padded random string should be in order to get a CPA-secure encryption scheme. Let $l(n)$ be the length of the random string padded before the message for encryption. If $l(n) = O(\log n)$, then there is no way to make the resulting scheme CPA-secure. If $l(n) = c \cdot n$ for some constant $c < 2$, it is widely believed that this kind of padded RSA is CPA-secure, if the RSA assumption is true. While for the case $l(n) = ||N|| - O(\log n)$, there are proofs based on the RSA assumption, that the padded RSA with padding length $l(n)$ is CPA-secure. (See Theorem 10.19 in [1])

In particular, in 1988, Werner Alexi et al. in [2] proved that in the plain RSA encryption scheme, if there exists a probabilistic polynomial-time Turing machine that can determine the least significant of the message from the ciphertext and public key with the success probability $\frac{1}{2}$ plus some non-negligible function, then the RSA assumption is false. Hence, under the RSA assumption, the padded RSA with message length 1 is CPA-secure. In the same paper, they also showed that this method can be generalised to the j-least significant bits, as long as j is $O(\log n)$. As a consequence, encoding messages of length $O(\log n)$ using the padded RSA scheme is CPA-secure. However, it is rather inefficient to encode messages in this way, since the length of padded string would be exponential to the length of message. And in practice, no such scheme is used, although it is proven to be secure.

On the other hand, the security of all kinds of RSA schemes are based on the RSA assumption, while this assumption is rather weak in the sense that it is no more difficult than *prime factorization* (factoring a composite number into the product of primes). And in quantum computing, there are even algorithms (e.g. Shor's algorithm [4]) that can solve the prime factorization problem efficiently. Therefore, in general, we only have a weak guarantee of the security of RSA encryption schemes.

References

[1] Jonathan Katz, Yehuda Lindell. *Introduction to modern cryptography.* Chapman and Hall/CRC, 2008.

[2] W. Alexi, B. Chor, O. Goldrech, C. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194–209, 1988.

[3] Wikipedia. RSA (cryptosystem).

[4] Wikipedia. Shor's algorithm.