1.4 Suitable Parameters for RSA

Proposition 3 *Let $n \geq 3$ be an integer. The following statements are equivalent:*

(i) *n is squarefree.*

(ii) *There exists an $r \geq 2$ with $a^r \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$.*

(iii) **[RSA equation]** *For every $d \in \mathbb{N}$ and $e \in \mathbb{N}$ with $de \equiv 1 \pmod{\lambda(n)}$ we have $a^{de} \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$.*

(iv) *For each $k \in \mathbb{N}$ we have $a^{k\cdot\lambda(n)+1} \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$.*

Proof. "(iv) $\Longrightarrow$ (iii)": Since $de \equiv 1 \pmod{\lambda(n)}$, we have $de = k \cdot \lambda(n) + 1$ for some k. Hence $a^{de} \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$.

"(iii) $\Longrightarrow$ (ii)": Since $n \geq 3$, we have $\lambda(n) \geq 2$. Choosing an arbitrary d with $\gcd(d, \lambda(n)) = 1$ and a corresponding e by congruence division $\bmod \lambda(n)$ we get (ii) with $r = de$.

"(ii) $\Longrightarrow$ (i)": Assume there is a prime p with $p^2 | n$. Then by (ii) we have $p^r \equiv p \pmod{p^2}$. But because of $r \geq 2$ we have $p^r \equiv 0 \pmod{p^2}$, contradiction.

"(i) $\Longrightarrow$ (iv)": By the chinese remainder theorem we only have to show that $a^{k\cdot\lambda(n)+1} \equiv a \pmod{p}$ for all prime divisors $p|n$.

Case 1: $p|a$. Then $a \equiv 0 \equiv a^{k\cdot\lambda(n)+1} \pmod{p}$.

Case 2: $p \nmid a$. Because of $p-1|\lambda(n)$, we have $a^{\lambda(n)} \equiv 1 \pmod{p}$, hence $a^{k\cdot\lambda(n)+1} \equiv a \cdot (a^{\lambda(n)})^k \equiv a \pmod{p}$. $\diamond$

Corollary 1 *The RSA procedures work for a module n if and only if n is squarefree.*

To find suitable exponents d and e we have to know $\lambda(n)$ or, better yet (and necessarily as it will turn out) the prime decomposition of n. Then the procedure of key generation suggests itself:

1. Choose different primes $p_1, \ldots, p_r$ and form the module $n := p_1 \cdots p_r$.

2. Compute $\lambda(n) = \operatorname{lcm}(p_1-1, \ldots, p_r-1)$ using the Euclidean algorithm.

3. Choose the public exponent $e \in \mathbb{N}_2$, coprime with $\lambda(n)$.

4. Compute the private exponent d with $de \equiv 1 \pmod{\lambda(n)}$ by congruence division.

Then take the pair (n, e) as public key, and the exponent d as private key.

Corollary 2 *Who knows the prime decomposition of n can compute the private key d from the public key (n, e).*

Practical Considerations

1. The usual choice is $r = 2$. Then the module has only two prime factors p and q that, as a compensation, are very large. Factoring this kind of integers $n = pq$ seems especially hard. It is crucial that the primes are chosen completely at random. Then an attacker has no hint for a guess.

2. For e we may choose a prime with $e \nmid \lambda(n)$, or a "small" integer say $e = 3$—more on the dangers of this choice later. A common standard choice is the prime $e = 2^{16} + 1$, provided $e \nmid \lambda(n)$. The binary representation of this integer contains only two 1's, making the binary power algorithm for enryption very fast. (For digital signature this is the verification of the signature.) However this choice of e doesn't make decryption (or generating a digital signature) more efficient.

3. After generating the keys we don't need p, q, and $\lambda(n)$ anymore, so we could destroy them.

   However: Since d is a "random" integer in the interval $[1 \ldots n]$ taking d-th powers is costly even with the binary power algorithm. It becomes somewhat faster when the owner of the private key computes $c^d \bmod p$ and $\bmod q$—using integers of about half the size—and then composes the result $\bmod n$ with the chinese remainder theorem. This procedure yields a small advantage in speed for decryption (or generating a digital signature).

4. Instead of $\lambda(n)$ we could use its multiple $\varphi(n) = (p-1)(q-1)$ for calculating the exponent.

   Advantage: We save (one) lcm computation.

   Drawback: In general we get a larger exponent d, slowing down each single decryption.

5. Notwithstanding Corollary 1 the RSA procedure works in a certain sense even if the module n is not squarefree. Decryption using the chinese remainder theorem is slightly more complex, involving an additional "Hensel lift." However decryption breaks down for plaintexts a that are multiples of a prime p with $p^2 | n$. Note that this effect is compatible with Corollary 1!

   The danger of hitting a plaintext divided by a multiple prime factor of n by chance is negligeable but grows with the number of prime factors. Even for a squarefree module n a plaintext divided by a prime factor would immediately yield a factorization of n, and hence reveal the private key.

Attention

The cryptanalytic approaches of the following chapter result in a set of side conditions that should be strictly respected when generating RSA keys.

Exercises

1. Let p and q be two different odd primes, and $n = p^2q$. Characterize the plaintexts $a \in \mathbb{Z}/n\mathbb{Z}$ that satisfy the RSA equation $a^{de} \equiv a \pmod{n}$. Generalize the result to arbitrary n.

2. Show that an integer $d \in \mathbb{N}$ is coprime with $\lambda(n)$ if and only if d is coprime with $\varphi(n)$.