2.2 Computing the Key and Factorization

Question: *How to compute the private RSA exponent d, given the public exponent e and the module n?*

Answer: Each of the following tasks (A) – (D) is efficiently reducible to each of the other ones:

(A) Computing the private key d.

(B) Computing $\lambda(n)$ (Carmichael function).

(C) Computing $\varphi(n)$ (Euler function).

(D) Factoring n.

Breaking RSA is the (possibly properly) easier task:

(E) Computing e-th roots in $\mathbb{Z}/n\mathbb{Z}$.

The "proof" (not an exact proof in the mathematical sense) follows the roadmap:

$$\begin{tikzcd} C \arrow[r] & A \arrow[r] \arrow[dl] & E \\ D \arrow[u] \arrow[r] & B \arrow[u] & \end{tikzcd}$$

We always assume that n and the public exponent e are known, and $n = p_1 \cdots p_r$ with different primes $p_1, \ldots, p_r$.

Clearly "A $\longrightarrow$ E": Taking an e-th root means raising to the d-th power. So if d is known, computing e-th roots is easy.

Note that the converse implication is unknown: *Breaking RSA could be easier than factoring.*

"D $\longrightarrow$ C": $\varphi(n) = (p_1 - 1) \cdots (p_r - 1)$.

"D $\longrightarrow$ B": $\lambda(n) = \mathrm{kgV}(p_1 - 1, \ldots, p_r - 1)$.

"B $\longrightarrow$ A": Compute d by congruence division from $de \equiv 1 \pmod{\lambda(n)}$.

"C $\longrightarrow$ A": Since $\varphi(n)$ has exactly the same prime factors as $\lambda(n)$, also $\varphi(n)$ is coprime with e. From $de \equiv 1 \pmod{\varphi(n)}$ we get a solution for d by congruence division. This might not be the "true" exponent, but works in the same way as private key since a forteriori $de \equiv 1 \pmod{\lambda(n)}$.

"A $\longrightarrow$ D" is significantly more involved. Moreover we only construct a probabilistic algorithm.

Preliminary Remarks

1. *It suffices to decompose n into two proper factors.*

   (a) Let $n = n_1 n_2$ be a proper decomposition, and assume for simplicity that $n_1 = p_1 \cdots p_s$ with $1 < s < r$. Then

   $$\lambda(n_1) = \mathrm{kgV}(p_1 - 1, \ldots, p_s - 1) \,|\, \mathrm{kgV}(p_1 - 1, \ldots, p_r - 1) = \lambda(n),$$

   thus also $de \equiv 1 \pmod{\lambda(n_1)}$. This reduces the problem to the analoguous ones for n_1 and n_2.

   (b) Since the number of prime factors of n is at most $\log_2(n)$ the recursive reduction suggested by (a) is efficient.

2. *How can a residue class $w \in \mathbb{Z}/n\mathbb{Z}$ help with factoring n?*

   (a) Finding a $w \in [1 \ldots n-1]$ with $\gcd(w, n) > 1$ decomposes n since $\gcd(w, n)$ is a proper divisor of n.

   (b) Finding a $w \in [2 \ldots n-2]$ with $w^2 \equiv 1 \pmod{n}$ (that is a nontrivial square root of 1 in $\mathbb{Z}/n\mathbb{Z}$) likewise decomposes n:
   Since $n | w^2 - 1 = (w + 1)(w - 1)$ and $n \nmid w \pm 1$ we have $\gcd(n, w + 1) > 1$, and this decomposes n by (a).

Now let (d, e) be a pair of RSA exponents. Then also $u := ed - 1 = k \cdot \lambda(n)$ is known (with unknown k and $\lambda(n)$). Since $\lambda(n)$ is even we may write

$$u = r \cdot 2^s \qquad \text{with } s \geq 1 \text{ and } r \text{ odd.}$$

If we choose a random $w \in [1 \ldots n - 1]$, then we have to deal with two possibilities:

- $\gcd(w, n) > 1$—then n is decomposed.
- $\gcd(w, n) = 1$—then $w^{r2^s} \equiv 1 \pmod{n}$.

In the second case we efficiently find the minimal $t \geq 0$ with

$$w^{r2^t} \equiv 1 \pmod{n}.$$

Again we distinguish two cases:

- $t = 0$—bad luck, choose another w.
- $t > 0$—then $w^{r2^{t-1}}$ is a square root $\neq 1$ of 1 in $\mathbb{Z}/n\mathbb{Z}$.

In the second case we distiguish:

- $w^{r2^{t-1}} \equiv -1 \pmod{n}$—bad luck, choose another w.

- $w^{r2^{t-1}} \not\equiv -1 \pmod{n}$—then n is decomposed by preliminary remark 2.

Thus every choice of $w \in [1 \ldots n-1]$ has one of four possible outcomes, two of them decompose n, and the other two flop. Denote the last two events by

$$\begin{array}{ll} (\mathrm{E}_{n,u}(w)/\mathrm{I}) & w^r \equiv 1 \pmod{n} \\ (\mathrm{E}_{n,u}(w)/\mathrm{II}) & w^{r2^{t-1}} \equiv -1 \pmod{n} \quad \text{for a } t \text{ with } 1 \leq t \leq s. \end{array}$$

Altogether this yields a tree-like structure:

$w \in [1 \ldots n-1] \longrightarrow$
  $\gcd(w, n) > 1 \longrightarrow n$ decomposed SUCCESS
  $w \in \mathbb{M}_n \longrightarrow$
    $w^r \equiv 1 \pmod{n} \longrightarrow (\mathrm{E}_{n,u}(w)/\mathrm{I})$ FLOP
    $w^r \not\equiv 1 \pmod{n} \longrightarrow$
      $w^{r2^{t-1}} \equiv -1 \pmod{n} \longrightarrow (\mathrm{E}_{n,u}(w)/\mathrm{II})$ FLOP
      $w^{r2^{t-1}} \not\equiv -1 \pmod{n} \longrightarrow n$ decomposed SUCCESS

Thus our procedure decomposes n "with high probability" if there are only "few" "bad" integers w with $(\mathrm{E}_{n,u}(w)/\mathrm{I,II})$. The next section will provide an upper bound for their number.