2.6 Breaking Single Ciphertexts

Breaking a single ciphertext (without necessarily computing the private key) could be even easier: For a given ciphertext c we could have $E_e^r(c) = c$ even if $E_e^r \neq \mathbf{1}_M$. If a is the corresponding plaintext, $c = E_e(a)$, then the cryptanalyst can compute:

$$E_e^{r-1}(c) = D_e(E_e^r(c)) = D_e(c) = a.$$

From a mathematical viewpoint we have the situation:

- The group $\mathbb{M}_{\lambda(n)}$ acts on the set $M = \mathbb{Z}/n\mathbb{Z}$, as does its cyclic subgroup $G := \langle e \rangle \leq \mathbb{M}_{\lambda(n)}$.
- For $a \in M$ the orbit is $G \cdot a = \{a^{e^k} \mid 0 \leq k < s\}$ (where s is the order of e in the multiplicative group $\mathbb{M}_{\lambda(n)}$).
- The stabilizer $G_a = \{f \in G \mid a^f \equiv a \pmod{n}\}$ is a subgroup of G. We have a natural bijective correspondence between the sets $G \cdot a$ and G/G_a.
- For the orbit length $t = \#G \cdot a$ we have

$$t = \frac{s}{\#G_a}, \qquad t|s|\lambda(\lambda(n))$$
$$E_e^r(c) = c \iff E_e^r(a) = a \iff t|r.$$

- $G \cdot c = G \cdot a$ and $\#G_c = \#G_a$. (The two stabilizers are conjugate.)
- Finding the orbit length t of a and c is at least as difficult as breaking the ciphertext c.

This suggests yet another problem:

3. Under what conditions is $t = s$, in other words, which stabilizers G_a are trivial? Or at least quite small?

Answer once more (without proof): in most cases. For superspecial primes p and q where $\lambda(\lambda(n)) = 2p''q''$ we expect by similar considerations as in Section 2.5 that $t < p''q''$ only for a negligeable set of exceptions.

The following two papers show how low is the risk of hitting a small orbit length by pure chance, enabling an iteration attack:

- J. J. Brennan/ Bruce Geist, Analysis of iterated modular exponentiation: The orbits of x^α mod N. **Designs, Codes and Cryptography** 13 (1998), 229–245.
- John B. Friedlander/ Carl Pomerance/ Igor E. Shparlinski, Period of the power generator and small values of Carmichael's function. **Mathematics of Computation** 70 (2001), 1591–1606, + 71 (2002), 1803–1806.