1 The $p-1$ Factoring Method

Another factoring algorithm, the $p-1$ method, is not a general-purpose factoring algorithm, but it can be used to quickly find factors of n that are of a certain form. Additionally, this method is the basis for many modern factoring techniques, especially those used in elliptic curve analysis. Also GIMPS uses the $p-1$ algorithm for its search. Keep in mind the $p-1$ algorithm only finds factors of a certain type and to explain that type first a definition must be established.

Definition 1 *Let B be a positive integer. A positive integer n will be said to be B-*smooth *if all the prime divisors of n are less than or equal to B. We will say that n is B-*powersmooth *if all prime powers dividing n are less than or equal to B.*

Suppose p is an unknown divisor of N. Consider a an integer such that $(a, N) = 1$ and $a > 1$. So $a^{p-1} \equiv 1 \mod p$. Now let lcm[1..B] be the least common multiple of the integers from 1 to B. Suppose $p-1$ is B-powersmooth. Then $p-1 | \operatorname{lcm}[1..B]$, and so $a^{\operatorname{lcm}[1..b]} \equiv 1 \mod p$. Thus

$$(a^{\operatorname{lcm}[1..B]} - 1, N) > 1$$

Now it is highly unlikely that $(a^{\operatorname{lcm}[1..B]} - 1, N) = N$if one gradually increases B since this implies for all p_i dividing n that $Q \max_{q_j | p_i} q_j$ for some common Q where each p_i and q_j is prime. The algorithm is essentially as follows (this form of the $p-1$ algorithm is due to Pollard):

The $p-1$ **Algorithm**: Input N and a bound B. Next form a list of primes $p[1], \ldots, p[k]$ which are all primes up to B.

1. [Initialize] Set $x \leftarrow 2, y \leftarrow x, c \leftarrow 0, i \leftarrow 0$, and $j \leftarrow i$.
2. [Next prime] Set $i \leftarrow i + 1$.
   - If $i > k$, compute $g \leftarrow (x-1, N)$.
     - If $g = 1$ output "Splitting N failed."
     - Otherwise $i \leftarrow j, x \leftarrow y$ go to step 5.
   - Otherwise if $i \leq k$ set $q \leftarrow p[i], q_1 \leftarrow q, l \leftarrow \lfloor B/q \rfloor$.
3. [Compute power] While $q_1 \leq l$, set $q_1 \leftarrow qq_1$
   - Then set $x \leftarrow x^{q_1} \mod N, c \leftarrow c + 1$ and if $c < 20$ go to step 2.
4. [Compute GCD] Set $g \leftarrow (x-1, N)$.
   - If $g = 1$, set $c \leftarrow 0, j \leftarrow I, y \leftarrow x$, and go to step 2.
   - Otherwise set $i \leftarrow j$ and $x \leftarrow y$.
5. [Backtrack] Set $i \leftarrow i + 1, q \leftarrow p[i]$ and $q_1 \leftarrow q$.
6. [Finished?] Set $x \leftarrow, g \leftarrow (x-1, N)$.
   - If $g = 1$, then set $q_1 \leftarrow qq_1$
     - If $q_1 \leq B$, go to step 6.
     - Otherwise go to step 5.
   - Otherwise
     - If $g < N$ then output g and terminate.
     - Otherwise if $g = N$ then
       - output that the algorithm failed and terminate.