RECOVERING HIDDEN SNFS POLYNOMIALS

PAUL ZIMMERMANN

ABSTRACT. Given an integer N constructed with an SNFS trapdoor, i.e., such that $N = |\mathrm{Res}(f, g)|$ with $f = a_d x^d + a_{d-1}x^{d-1} + \cdots + a_0$ having small coefficients $a_i = O(B)$, and $g = \ell x - m$, we can recover f and g in $O(BF(\ell))$ arithmetic operations, assuming $B^2\ell^2 \ll a_d m$, where $F(\ell)$ is the number of arithmetic operations to extract a prime factor of same size as ℓ. This partially answers an open problem from [1].

We use the following algorithm, where the transform $N' \leftarrow d^d a_d^{d-1} N$ and the translation $x \to x - a_{d-1}/(da_d)$ are inherited from [3].

Algorithm 1

Input: an integer N, a degree d, a leading coefficient a_d, a bound L
Output: $f = a_d x^d + \cdots + a_0, g = \ell x - m$ such that $N = |\mathrm{Res}(f, g)|$ and $\ell < L$, or FAIL

$N' \leftarrow d^d a_d^{d-1} N$
$m' \leftarrow \lfloor N'^{1/d} \rceil$
$r \leftarrow N' - m'^d$
search using ECM prime factors of r smaller than L
for ℓ in known divisors(r) **do**
  if $r \bmod \ell^2 = 0$ **then**
    decompose $N = a_d m^d + a_{d-1} m^{d-1}\ell + \cdots + a_0 \ell^d$ where m, a_{d-1} satisfy $m' = da_d m + a_{d-1}\ell$
    return $f = a_d x^d + \cdots + a_0, g = \ell x - m$
return FAIL

Lemma 1. *If $N = a_d m^d + a_{d-1} m^{d-1}\ell + \cdots + a_1 m \ell^{d-1} + a_0 \ell^d$, with $a_i = O(B)$ and $B^2\ell^2 \ll a_d m$, then Algorithm 1 unveils $f = a_d x^d + \cdots + a_0$ and $g = \ell x - m$ in $O(F(\ell))$ arithmetic operations.*

Proof. We have $N = a_d m^d + a_{d-1} m^{d-1}\ell + R$ with $R = O(Bm^{d-2}\ell^2)$. Then:

$$\begin{aligned} N' &= d^d a_d^{d-1}(a_d m^d + a_{d-1} m^{d-1}\ell + R) \\ &= (da_d m + a_{d-1}\ell)^d - S + d^d a_d^{d-1} R, \end{aligned}$$

where $S = \sum_{i=0}^{d-2} \binom{d}{i} (da_d m)^i (a_{d-1}\ell)^{d-i} = O(d^d a_d^{d-2} B^2 m^{d-2} \ell^2)$, and $d^d a_d^{d-1} R = O(d^d a_d^{d-1} B m^{d-2}\ell^2)$, thus $N' = m'^d + O(d^d a_d^{d-2} B^2 m^{d-2}\ell^2)$ with $m' = da_d m + a_{d-1}\ell$. Since $B^2\ell^2 \ll a_d m$ and $m' \approx da_d m$, we get $N' - m'^d \ll dm'^{d-1}$, which ensures that the rounded d-th root of N' is m'. Now both R and S are divisible by ℓ^2, thus the divisor ℓ of r will be found in time $O(F(\ell))$, and the rest follows from Lemma 2.1 of [2]. □

Date: October 2019.

Example. Consider this innocent-looking 1024-bit prime produced by Emmanuel Thomé:

$$
\begin{aligned}
N \quad = \quad & 10125975488959488438636448139388738111384370034580126872774623167983065095763618 \\
& 44716875429364100448228034431031042649131921103572845443219053574589128101877982 \\
& 01444275956478694551535584037776691110761982172617916831503906052571224968894093 \\
& 3317113399977964690443112336421914513022902451215280589953974768870 83.
\end{aligned}
$$

We search for f of degree 6, with coefficients bounded by 1000 in absolute value. This search will in particular consider $a_6 = 883$. We then get

$$
\begin{aligned}
m' \quad &= \quad 369281866289223731963395973054879619878608371115794 0498, \\
r \quad &= \quad 82879887764694366348912168791836341837049570452618174403026264656774533779857170 \\
& \quad\quad 37239452504338734757522396248499672667034561347930357160942512349898884824251878 \\
& \quad\quad 72235920062471226328786567796505070700605282371914362200427993013634248968829556 \\
& \quad\quad 0116730782294875432021758 08000.
\end{aligned}
$$

Dividing out primes less than one million we get:

$$r = 2^9 \cdot 3^{13} \cdot 5^3 \cdot 17^2 \cdot 71 \cdot 137^2 \cdot q_{251},$$

where q_{251} is a 251-digit composite number. With GMP-ECM [4] we find the following prime factors of q_{251}:

$$q_{251} = 3513299 \cdot 2258358157748717 \cdot 36004635722054299^2 \cdot q_{196}.$$

Among the divisors of r we try $\ell = 1358447704865964290410 2 = 2 \cdot 3^4 \cdot 17 \cdot 137 \cdot 36004635722054299$, which yields the polynomials:

$$
\begin{aligned}
f \quad &= \quad 883x^6 - 202x^5 + 779x^4 - 990x^3 + 374x^2 - 886x + 316, \\
g \quad &= \quad 1358447704865964290410 2x - 69702126517407272926273305597424316027744676463 2799.
\end{aligned}
$$

A full search for $1 \leq a_6 \leq 1000$ takes about 280 minutes of cpu time on an Intel Xeon CPU E7-4850 running at 2.2GHz.

References

[1] Fried, J., Gaudry, P., Heninger, N., and Thomé, E. A kilobit hidden SNFS discrete logarithm computation. In *36th Annual International Conference on the Theory and Applications of Cryptographic Techniques - Eurocrypt 2017* (Paris, France, Apr. 2017), J.-S. Coron and J. B. Nielsen, Eds., vol. 10210 of *Advances in Cryptology – EUROCRYPT 2017*, Springer.

[2] Kleinjung, T. On polynomial selection for the general number field sieve. *Mathematics of Computation 75* (2006), 2037–2047.

[3] Kleinjung, T. Polynomial selection. slides presented at the CADO workshop on integer factorization, 2008.

[4] Zimmermann, P. GMP-ECM: yet another implementation of the Elliptic Curve Method (or how to find a 40-digit prime factor within $2 \cdot 10^{11}$ modular multiplications). In *Workshop* Computational Number Theory *of FoCM'99 (Foundations of Computational Mathematics)* (Oxford, United Kingdom, 1999).