CR10: Cryptanalysis Autumn 2018

Lecture NUMBER — December 4th, 2018

Lecturer G. Hanrot, E. Kirshanova *Scribe: Franois Pitois*

1 Remainder on $(p-1)$ method

Recall that in $(p-1)$ method, the idea is to find stuff that appends modulo p but not modulo N.

Consider the following quantity:

$$X(B) = \prod_{p \leq B} p^{\lfloor \frac{\log B}{\log p} \rfloor}$$

For $a \in \mathbb{Z}/N\mathbb{Z}$, if $\gcd(a, N) = 1$, then compute $\gcd(a^{X(B)} - 1, N)$.

A sufficient condition for some q/N to divide also $a^{X(B)} - 1$ is that $q - 1 | X(B)$, meaning that $q - 1$ has only small prime power factors. The algorithm can be improve with a "second phase" to deal with the case where $q - 1$ might have ONE prime factor.

Observation 1. *"Second phase" deals with the case where $q-1$ might have one prime factor within $[B, B^2]$. The idea is to compute all $\gcd(a^{lX(B)} - 1, N)$ for l prime in $[B, B^2]$, and compute their product modulo N.*

We use the fact that if $l < l'$ are two such consecutive primes, $a^{lX(B)} = a^{l'X(B)} a^{(l-l')X(B)}$. If $\beta = a^{X(B)} \mod N$, $\beta^l = \beta^{l'} \beta^{(l-l')}$. Then, pre-compute all possible value of $\beta^{(l-l')}$ to compute all $a^{lX(B)} - 1$ fast. Recall that $(l - l') = \mathcal{O}(\log^2(B))$..

2 $(p + 1)$ method

This method is due to Hugh C. Williams in 1982 [Wil82].

Let $G_d(N) = \{(a, b) \in \mathbb{Z}/N\mathbb{Z} \mid a^2 + db^2 = 1 \mod N\} \subseteq (\mathbb{Z}/N\mathbb{Z})\sqrt{-d}]$.

Claim 2. *There is a group structure on $G_d(N)$ if N is prime, where:*

- *the neutral element is $(1, 0)$,*
- *product is defined as $(a, b) \times (a', b') = (aa' - dbb', ab' + a'b)$.*

The idea here is to think of (a, b) as $a + b\sqrt{-d}$).

Claim 3. *Let p be a prime. If $-d$ is a square modulo p, then $\#G_d(p) = p - 1$. If $-d$ is not a square modulo p, then $\#G_d(p) = p + 1$.*

Algorithm 1 $p + 1$ Algorithm

Input: N

Output: A prime factor of N, or **fail**

1: Pick $a, b \in \mathbb{Z}/N\mathbb{Z}$ randomly
2: Put $d = \frac{1-a^2}{b^2} \mod N$
3: Compute $(u, v) = (a, b)^{X(B)}$ in $G_d(N)$ ▷ Is $(u, v) == (1, 0) \mod p$ for some $p \mid N$?
4: **return** $\gcd(u - 1, v, N)$

The success condition can be:

- $-d$ is a square, thus $p - 1|X(B)$
- $-d$ is not a square, thus $p + 1|X(B)$

We are in the second case.

3 ECM (Elliptic Curve Method)

This method is due to Lenstra Jr and Hendrik W in 1987 [LJ87].

An Elliptic Curve parametrized with a and b is based on the ground set:

$$E_{a,b}(N) = \{(x, y) \mid y^2 = x^3 + ax + b\} \cup \{\infty\}.$$

For the curve not to be singular, we assume that $(4a^3 + 27b^2, N) = 1$.

Claim 4. *When p is prime, there is a group structure over $E_{a,b}(p)$, defined by:*

- *three aligned points sum to zero (counted with multiplicities),*
- *neutral elements is ∞.*

Theorem 5 (Hasse)**.** *If p is prime, $|\#E_{a,b}(p) - (p + 1)| \leq 2\sqrt{p}$.*

Algorithm 2 ECM Algorithm

Input: N

Output: A prime factor of N, or **fail**

1: Pick $(x, y) \in \mathbb{Z}/N\mathbb{Z}$, pick a and $b = y^2 - x^3 - ax \mod N$
2: Check that $\gcd(4a^3 + 27b^2, N) = 1$
3: Compute $(u, v) = X(B) \cdot (a, b)$

During the computation, we hope that at some point an inverse $\mod N$ (slope of (PQ) or of tangent at T) will be impossible, meaning that the number we are trying to invert is not coprime to N ($\Rightarrow$ often get a factor of N).

Sufficient condition of successor is that for some $p \mid N, \#E_{a,b}(p) \mid X(B)$.

Heuristic 6 (False)**.** *For random x, y, a as in the algorithm, the probability that $E_{a,b}(p)$ is B-smooth is the same as for a random integer in $[p/2, 3p/2]$, namely*

$$p_{B-smooth} \approx \frac{1}{u^u}, \text{ where } u = \frac{\log p}{\log B}.$$

The expected number of curves to get a success is one over this probability, i.e. u^u. The cost of testing one curve is $\log(X(B)) \approx Bx \operatorname{poly}(\log N)$. Hence, the total cost is $Bu^u \operatorname{poly}(\log N)$. The goal is now to estimate the optimal B. Let's consider the log of this cost:

$$\log(Bu^u) = \log B + \frac{\log p}{\log B} \log \frac{\log p}{\log B}$$

Let $x = \frac{\log p}{\log B}$. Then

$$\log(Bu^u) = \frac{1}{x} \log p + x \log x \quad \text{and} \quad (\log(Bu^u))' = -\frac{1}{x^2} \log p + \log x + 1$$

Hence, the optimal value is obtained when $x^2(1 + \log x) = \log p$. For convenience, let's look for an x such that $x^2 \log x = \log p$.

$$x = \sqrt{\frac{\log p}{\log x}} = \sqrt{\frac{\log p}{\frac{1}{2} \log \frac{\log p}{\log x}}} = \sqrt{2 \frac{\log p}{\log \log p - \log \log x}} \approx \sqrt{2 \frac{\log p}{\log \log p - 0}}$$
$$\log B = \frac{\log p}{x} = \sqrt{\frac{1}{2} \log p \log \log p}$$

Hence, based on the false heuristic 6, the total cost of ECM is $\mathcal{O}\left(\exp\left(\sqrt{\frac{1}{2} \log p \log \log p}\right) \times \operatorname{poly}(N)\right)$.

4 Congruence-based methods

Idea 7. *Find (x, y) with $x \neq \pm y \mod N$, and $x^2 = y^2 \mod N$. Then N can be factorized as $N = \gcd(x - y, N) \times \gcd(x + y, N)$, hoping that both $\gcd$ are not 1 neither N.*

Example 8. Let $N = 143$. To find x and y, one idea is to find some x^2 that are congruent modulo N to a small number (i.e. lower than B for some B). To find it, let's check all first x, and consider $B = 5$ for example.

x	x^2	$x \mod N$
⋮	⋮	⋮
13	169	26
14	196	53
15	225	82
16	256	30
17	289	3
⋮	⋮	⋮

Doing so, we find that $17^2 = 3 \mod N$. Thus, it would be great to find y such that $y^2 = 3 \mod N$, or even of the form $y^2 = 3 \times k^2 \mod N$ for some k. Let's continue to explore the table:

x	x^2	$x \mod N$
$\vdots$	$\vdots$	$\vdots$
15	225	82
16	256	30
17	289	3
18	324	38
19	361	$75 = 3 \times 5^2$
$\vdots$	$\vdots$	$\vdots$

Finally, we found that $17^2 \times 19^2 = 3 \times (3 \times 5^2) = 15^2 \mod N$, which means that $37^2 = 15^2 \mod N$. Hence, we can easily factorize $N = 143$: we have $37 - 15 = 22$ and $\gcd(22, N) = 11$; and $37 + 15 = 52$ and $\gcd(52, N) = 13$. Thus, $N = \gcd(22, N) \times \gcd(52, N) = 11 \times 13$.

Algorithm 3 Meta-Algorithm

Input: N
Output: A prime factor of N, or **fail**

1: B a bound, $\mathcal{B} = \{p \leq B\}$ ▷ $\mathcal{B}$ is the factor base
2: i $\leftarrow$ 0
3: **while** i $\leq \#\mathcal{B}$ **do**
4: Pick x_i
5: If $x^2 \mod N$ factors as $\prod_{p_j \in \mathcal{B}} p_j^{u_{i,j}}$, then increment i
6: Solve the linear system $u_{i,j}^t \times v = 0 \mod 2$

Proposition 9. *If we have:*

$$Y = \prod_i x_i^{v_i} \mod N \quad \textit{and} \quad Z = \prod_j p_j^{\frac{1}{2}\sum_j u_{i,j} v_i} \mod N$$

then $Y^2 = Z^2 \mod N$.

Proof. Indeed,

$$Y^2 = \prod_i (x_i^2)^{v_i} = \prod_i \left(\prod_j p_j^{u_{i,j}} \right)^{v_i} = \prod_j p_j^{\sum_j u_{i,j} v_i} = Z^2 \mod N$$

□

4.1 Dixon's algorithm

Let specify this meta-algorithm. In Dixon's algorithm [Dix81], x_i's are picked randomly, factorizing $x_i^2 \mod N$ is done by trial division, and solving the linear system is done by Gaussian elimination.

To analyze Dixon's algorithm, let's make two assumptions:

Assumption 10. *Suppose that $x_i^2 \lessapprox N^\alpha$ for some α.*

Assumption 11. *Suppose the cost of factorizing x_i^2 is roughly B^θ.*

The number of relations needed is approximatively $\#\mathcal{B} \approx B^{1+o(1)}$, and the cost of trying one is $x_i = B^\theta$.

Heuristic 12. $x_i^2 \mod N$ *behaves as a random integer in* $[0, N^\alpha]$. *Hence, probability of success for one* x_i *is*

$$p_{success} = \frac{1}{u^u}, withu = \frac{\log N^\alpha}{\log B}.$$

Thus, the total cost is $\max(u^u B^\theta B^{1+o(1)}, B^3)$, where B^3 comes form linear algebra solving.

This is optimal when:

$$\log B = \sqrt{\frac{\alpha}{2(1+\theta)} \log N \log\log N}$$

Finally, the total cost is:

$$\max\left(B^3, \exp\left(\sqrt{2\alpha(1+\theta)\log N \log\log N}\right)\right)$$

In Dixon's algorithm, we take the values $\alpha = 1$ and $\theta = 1$, which leads to a total cost of:

$$\max\left(B^3, \exp\left(2\sqrt{\log N \log\log N}\right)\right).$$

One can be smarter in the choice of α and θ. In the Quartic Sieve algorithm, we choose $\alpha = 1/2$ and $\theta = 0$. Hence,

$$\log B = \sqrt{\frac{1}{4}\log N \log\log N}$$

and the total cost becomes

$$\max\left(\exp\left(\sqrt{\log N \log\log N}\right), \exp\left(\sqrt{\log N \log\log N}\right)\right)$$

where the first argument of the max comes from linear algebra, and the second one come from previous relations.

Good News: the matrix of the linear system is sparse! At most $\mathcal{O}(\log N)$ nonzero coefficients per rows for $\exp\left(\sqrt{\log N \log\log N}\right)$ columns $\Rightarrow$ linear algebra can be done in $B^{2+o(1)}$ instead of B^3

Introduce $P(X) = (X + \lfloor\sqrt{N}\rfloor)^2 - N$. If $i \ll N$, then, $P(i) \approx 2i\sqrt{N}$. So if $i = N^{o(1)}$,

$$P(i) \approx N^{1/2+o(1)}$$

(we are still in the case where $\alpha = 1/2$).

One can use $P(i)$ for x_i. The number of x_i used by the algo is

$$u^u B^\theta = \exp\left(c\sqrt{\log N \log\log N}\right) = N^{o(1)}.$$

References

[Dix81] John D Dixon. Asymptotically fast factorization of integers. *Mathematics of computation*, 36(153):255–260, 1981.

[LJ87] Hendrik W Lenstra Jr. Factoring integers with elliptic curves. *Annals of mathematics*, pages 649–673, 1987.

[Wil82] Hugh C Williams. A p+1 method of factoring. *Mathematics of Computation*, 39(159):225–234, 1982.