Exercises on lattice-based cryptography

Daniel J. Bernstein
(with some exercises from Tanja Lange)

12 July 2022

1. Notation: In the following exercises, $N \geq 2$ is an integer; $Q \geq 2$ is a power of 2; R_0 means the ring $\mathbf{Z}[x]/(x^N - 1)$; R_m means $(\mathbf{Z}/m)[x]/(x^N - 1)$ for any $m \geq 1$; and "short" elements of R_0 mean elements that are sums or differences of a small number of powers of x.

2. For $N = 11$, give an example of two nonzero elements of R_0 with product 0, and give an example of two nonzero elements of R_8 with product 0.

3. ["Find more equivalences."] An NTRU public key is an element of R_Q. A secret key for a public key G means a pair (e, a) of short elements of R_0 such that e is invertible in R_Q; a is invertible in R_Q and in R_3; and $G = 3e/a$ in R_Q. Show that if (e, a) is a secret key for G then (xe, xa), (x^2e, x^2a), etc. are also secret keys for G. Show that there are other secret keys for G.

4. [Simpson's method, usually miscredited to Newton.] Simpson (1740) introduced the following iteration for finding roots of a function φ, unifying and generalizing previous iterations for polynomials φ: if x is close to a root then, under reasonable assumptions, $x - \varphi(x)/\varphi'(x)$ is closer, where φ' is the derivative of φ. Fix $f \neq 0$, and observe that the iteration $g \mapsto 2g - fg^2$ maps $1/f$ to $1/f$. Show that this iteration is a special case of Simpson's iteration.

5. ["Figure out how `invertmodpowerof2` works."] Consider the problem of inverting an element f of the ring R_Q. Assume that $g \in R_Q$ satisfies $fg = 1$ in R_2. Consider replacing g with $2g - fg^2$, and repeating this replacement $\log_2 \log_2 Q$ times. Show that the final g satisfies $fg = 1$ in R_Q. (Hint: Start with the case $Q = 4$.)

6. Computer exercise: Pick a specific example of short $e, a \in R_0$ for $N = 11$ and $Q = 256$, using sums or differences of 3 powers of x. Compute $H = e/a$ in R_Q. Check that $aH = e$.

7. After computing H in the previous exercise, use Sage's lattice-basis reduction (e.g., LLL) to find a short nonzero linear combination of the rows of the following matrix:

$$\begin{matrix}
Q & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & Q & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Q & 0 & 0 & \cdots & 0 \\
H_0 & H_1 & \cdots & H_{N-1} & 1 & 0 & \cdots & 0 \\
H_{N-1} & H_0 & \cdots & H_{N-2} & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
H_1 & H_2 & \cdots & H_0 & 0 & 0 & \cdots & 1
\end{matrix}$$

Compare the results to the $2N$ coefficients of (e, a).

8. Math exercise: An NTRU ciphertext for a public key $G \in R_Q$ is an element $bG + d \in R_Q$, where b, d are short secret elements of R_0. Construct a lattice L and a vector v such that searching for (b, d) is equivalent to finding an element of L close to v. Also construct a lattice L' such that searching for (b, d) sounds equivalent to finding a short nonzero element of L'.

9. Computer exercise: For $N = 701$, for each $Q \in \{2, 4, 8, 16, \ldots, 16384\}$, generate 1000 examples of a, b, d, e with coefficients chosen independently and uniformly at random from $\{-1, 0, 1\}$. How many times, out of these 1000 experiments, does $3be + ad$ have coefficients between $-Q/2$ and $Q/2 - 1$?

10. In the previous exercise, also implement key generation ($G = 3e/a$ in R_Q), encryption ($C = bG + d$ in R_Q), and decryption (multiply C by a in R_Q, take coefficients in R_0 between $-Q/2$ and $Q/2 - 1$, reduce modulo 3, multiply by $1/a$ in R_3, see whether this produces d). Check that decryption works whenever inversion succeeds and $3be + ad$ has coefficients between $-Q/2$ and $Q/2 - 1$. How often does inversion fail?

11. [Speed of a subset-sum attack.] The objective here, given integers $S, K_1, K_2, \ldots, K_N$, is to see whether there are bits $b_1, b_2, \ldots, b_N$ such that $S = b_1K_1 + b_2K_2 + \cdots + b_NK_N$. In Sage, implement a collision search that

- chooses $M = \lfloor N/2 \rfloor$,
- sorts a list of all $b_1K_1 + b_2K_2 + \cdots + b_MK_M$,
- sorts a list of all $S - b_{M+1}K_{M+1} - \cdots - b_NK_N$, and then
- merges the sorted lists.

Check that the attack works for each $N \in \{1, 2, \ldots, 40\}$, when $K_1, K_2, \ldots, K_N$ are chosen as random $2N$-bit integers and S is chosen as some $b_1K_1 + b_2K_2 + \cdots + b_NK_N$. Measure the time taken for the attack, with multiple experiments for each N. How much variance is there in the time for each N? As N grows, does the time seem to grow as $2^{N/2}$, or $2^{N/2}N$, or something else? What would you expect to happen when N is much larger?