Exercise Set 2
Solutions

Prof. Kenny Paterson
Kien Tuong Truong, Jan Gilcher

Problem 1. Recall the Gaussian heuristic from the lecture: this states that, for a "random" lattice $\mathcal{L}$ in n dimensions, we have:

$$\lambda_1 \approx \sqrt{\frac{n}{2\pi e}} \cdot \det(\mathcal{L})^{1/n}.$$

Consider the matrix B' of the form arising in Kannan's embedding:

$$\begin{bmatrix} B & 0 \\ \underline{w} & M \end{bmatrix}$$

where B is the $(n+1) \times (n+1)$ array arising in the CVP formulation of the HNP, namely:

$$\begin{bmatrix} q & 0 & 0 & \cdots & 0 & 0 \\ 0 & q & 0 & \cdots & 0 & 0 \\ 0 & 0 & q & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & q & 0 \\ t_1 & t_2 & t_3 & \cdots & t_n & 1/2^{L+1} \end{bmatrix}$$

Compute the determinants of B and B' (in the latter case, the result is a function of M), and from this, the expected value of λ_1 for the corresponding lattices (under the Gaussian heuristic).
Solution As both matrices are lower triangular, computing the determinant of each is just a matter of multiplying all elements in the diagonal together.

$$\det(\mathcal{L}_B) = q^n/2^{L+1} \qquad \det(\mathcal{L}_{B'}) = M \cdot q^n/2^{L+1}$$

Then, the expected values of λ_1 for will be:

$$\lambda_1(\mathcal{L}_B) \approx \sqrt{\frac{n+1}{2\pi e}} \det(\mathcal{L}_B)^{1/(n+1)} = \sqrt{\frac{n+1}{2\pi e}} \frac{q^{n/(n+1)}}{2^{(L+1)/(n+1)}}$$

$$\lambda_1(\mathcal{L}_C) \approx \sqrt{\frac{n+2}{2\pi e}} M^{1/(n+2)} \frac{q^{n/(n+2)}}{2^{(L+1)/(n+2)}}$$

Problem 2. (Solving this problem will again be useful in the lab, especially for implementing the last steps in the solution using the Kannan embedding technique and SVP.)

Recall that, in the Kannan embedding approach to solving CVP, a shortest vector in the lattice $\mathcal{L}'$ is expected to be of the form $(\underline{f}, M)$ where $\underline{f} = \underline{w} - \underline{v}$. Here $\underline{w}$ is the target input vector for CVP, and $\underline{v}$ is the solution.. Recall also that in our specific application to breaking ECDSA, $\underline{w}$ is of the form $(u_1, u_2, \ldots, u_n, 0)$ while $\underline{v}$ is likely to be of the form $(-\ell_1, -\ell_2, \ldots, -\ell_n, x) \cdot B = (u_1 + e_1, u_2 + e_2, \ldots, u_n + e_n, x/2^{L+1})$ (since we showed that this $\underline{v}$ is close to $\underline{w}$).

- Given this information, show that $\underline{f} = (-e_1, -e_2, \ldots, -e_n, -x/2^{L+1})$.
  Solution $\underline{f} = \underline{w} - \underline{v} = (u_1, ..., u_n, 0) - (u_1 + e_1, ..., u_n + e_n, x/2^{L+1}) = (-e_1, ..., -e_n, -x/2^{L+1})$
- Given what you know (from class) about the size of the e_i, compute an upper bound on the norm of $(\underline{f}, M) \in \mathcal{L}'$.
  Solution $\|\underline{f}, M\|^2 = \|\underline{f}\|^2 + \|M\|^2 = \|\underline{f}\|^2 + M^2$.
  $\|\underline{f}\|^2 = \left(\sum_{i=0}^{n} e_i^2\right) + \frac{x^2}{2^{2(L+1)}}$. Then, we use the fact that $e_i \leq 2^{N-(L+1)}$ and that $x < q < 2^N$:

$$\|\underline{f}\|^2 \leq n \cdot 2^{2(N-(L+1))} + 2^{2(N-(L+1))} = (n+1)2^{2(N-(L+1))}$$

Thus:

$$\|\underline{f}\| \leq \sqrt{(n+1)2^{2(N-(L+1))}} \quad \text{and} \quad \|\underline{f}, M\| \leq \sqrt{(n+1)2^{2(N-(L+1))} + M^2}$$

- Compare the result in the previous part to what you obtained in the previous problem using the Gaussian heuristic. What does this suggest about how to set M in order to ensure the success of the Kannan embedding approach to solving CVP in this case?
  Solution This is a tricky question! At face value, one might consider the upper bound of the previous answer and the lemma from the lectures that tells us to set $M = \|\underline{f}\|$. The problem with this is that we merely have an upper bound of $\|\underline{f}\|$ rather than the actual value. Even comparing our results with the Gaussian heuristic for λ_1 might not work: the Gaussian heuristic usually starts working consistently for large n ($n \geq 45$) and for random lattices. Our lattices are often small and not-so-random, making the Gaussian heuristic possibly imprecise.

  The best answer, and the one that is closer to what happens in practice during cryptanalysis, is to make educated guesses about what the value of M should be, experimenting around with it until it works consistently. In other words: let the theory guide you, but do not let it constrain you! Sometimes you don't have the instruments to fulfill al the conditions that theory requires, but more often than not, cryptanalytic tools work better than the theory predicts them to do.

  Also remember that Kannan embedding is pretty forgiving: there is not only one value that works, but many values may work. You may use any value that you believe works consistently across instances of the problem.

- By considering $(-t_1, -t_2, -t_3, \ldots, -t_n, q, 0) \cdot B'$, show that $\mathcal{L}'$ also contains the vector $\underline{z} = (0, 0, \ldots, 0, q/2^{L+1}, 0)$. Compute the norm of $\underline{z}$ and compare to the bound you obtained on the norm of $(\underline{f}, M)$ earlier, showing that it is typically about $\sqrt{n+1}$ times smaller (irrespective of the choice of M).
  Solution $\underline{z} = (-t_1, ..., -t_n, q, 0) \cdot B' = (-t_1 \cdot q + q \cdot t_1, ..., -t_n \cdot q + q \cdot t_n, q/2^{L+1}, 0) = (0, ..., 0, q/2^{L+1}, 0)$. This vector has norm $q/2^{L+1}$. Using the same approximation as before ($x < q < 2^N$), this yields an upper bound for the squared norm of $\underline{z}$:

  $$\|\underline{z}\|^2 \leq 2^{2(N-(L+1))}$$

  Comparing this with the approximation for the squared norm of $(\underline{f}, M)$, you may see that it at least $n+1$ times smaller, if we ignore the value of M. By applying the square root, the result follows.

This problem shows that $\mathcal{L}'$ contains unusually short vectors, so solving SVP will not directly yield a solution to the original CVP for the particular lattices we have constructed here. On the other hand, perhaps the required solution will appear elsewhere in the basis obtained from performing lattice reduction on $B' \ldots$

Problem 3. (This problem serves as a reminder of the RSA encryption scheme, which will be useful for the content of next week's lecture and and lab.)

Recall, the textbook RSA scheme:

$\mathsf{RSA\text{-}KGen}(k) \mapsto pk, sk$	$\mathsf{RSA\text{-}Enc}(pk, m) \mapsto c$	$\mathsf{RSA\text{-}Dec}(sk, c) \mapsto m$
generate k-bit primes p, q	with $pk = (N, e)$	with $sk = (N, d)$
$N \leftarrow pq$	$c \leftarrow m^e \mod N$	$m \leftarrow c^d \mod N$
$\varphi(N) := (p-1)(q-1)$	**return** c	**return** m
$e \leftarrow\$ \mathbb{N}$ s.t. $\gcd(e, \varphi(N)) = 1$		
$d \leftarrow e^{-1} \mod \varphi(N)$		
return $(N, e), (N, d)$		

As presented, this scheme is obviously not secure (why? **Solution:** because it's not randomized it is trivially not IND-CPA secure). In the following, we will see some simple attacks:

1. Let $e = 3$. Assume we encrypt a message $m < N^{\frac{1}{3}}$. Show that we can recover the plaintext without knowledge of the secret key.
   Solution: If $m < N^{\frac{1}{3}}$, then $m^e < N$ and therefore no modular reduction is performed when computing the ciphertext. Thus we can simply compute $\sqrt[3]{c}$ over $\mathbb{N}$ (or even numerically over $\mathbb{R}$) to recover the plaintext

2. Again, let $e = 3$, but this time without any restrictions on the message length. (Maybe we implicitly applied some padding to each message to ensure they are full length.) Assume the message is sent to several recipients with public keys (e, N_i). Show that you can still recover the message m from $c_i = \mathrm{Enc}((e, N_i), m)$ and the public keys all the N_i are coprime. How many different ciphertexts/recipients are necessary?
   Solution: The attacker obtains $c_i = m^e \mod N_i$. Let $N = N_1 \cdot N_2 \cdots N_i$. If all N_i are coprime, then by the Chinese Remainder Theorem there exists a unique $c \in \mathbb{Z}_N$ that fulfills the set of congruences:

$$\begin{aligned} c &\equiv c_1 \mod N_1 \\ &\vdots \\ c &\equiv c_i \mod N_i \end{aligned}$$

   Furthermore, it holds that $c = m^e \mod N$. To be a valid message it must hold that $m < N_i$ for all N_i. Therefore if we have 3 different ciphertexts/recipients (or more), we are back in the case of the previous task, where computing $c = m^e \mod N$ does not require a modular reduction, and can thus again solve simply compute the root over $\mathbb{N}$.

3. Assume now that two people share a modulus N, i.e. their public keys are (N, e_1) and (N, e_2) with e_1 and e_2 coprime. Show how an eavesdropper can recover a message m, given the the two ciphertexts of the message under the two different public keys.
   Solution: Since e_1 and e_2 are coprime we can use the extended Euclidean algorithm to compute two integers a, b s.t. $ae_1 + be_2 = 1$.
   We can then simply compute $c_1^a \cdot c_2^b = m^{ae_1} \cdot m^{be_2} = m^{ae_1 + be_2} = m \mod N$