Question 5

Recall from lecture notes the proto-Dilithium:

Algorithm 1 proto-Dilithium KeyGen

Require: $k, l \in \mathbb{Z}$, polynomial ring R_q, secret distribution $\chi_s = \mathcal{B}(n, p)$, y distribution $\chi_y = R_{[-\gamma,\gamma]}$, and hash function H that hashes from the message space to $R_{0,1}$ with exactly τ coefficients being ones

$\mathbf{s} \leftarrow \chi_s^l$
$A \leftarrow R_q^{k \times l}$
$\mathbf{t} \leftarrow A\mathbf{s}$
return $\text{sk} \leftarrow \mathbf{s}, \text{pk} \leftarrow (A, \mathbf{t})$

Algorithm 2 Dilithium.Sign

Require: pk, sk, m

$\sigma \leftarrow \perp$
while $\sigma = \perp$ **do**
  $\mathbf{y} \leftarrow \chi_y^l, \mathbf{w} \leftarrow A\mathbf{y}$
  $c \leftarrow H(m)$
  $\mathbf{z} \leftarrow \mathbf{y} + c \cdot \mathbf{s}$
  if $\|\mathbf{z}\|_\infty \geq \gamma - np\tau$ **then**
    $\sigma \leftarrow \perp$
  else
    $\sigma \leftarrow (\mathbf{z}, c, \mathbf{w})$
  end if
end while

Algorithm 3 Dilithium.Verify

Require: pk, m, σ

assert $\|\mathbf{z}\|_\infty < \gamma - np\tau$
assert $A\mathbf{z} = \mathbf{w} + c\mathbf{t}$

(1)

Observe that $A, \mathbf{t}$ are both public information. In addition, any adversary can sample $\mathbf{y} \leftarrow \chi_y$ on its own, and produce a valid $\mathbf{w}$. For any chosen message m, an adversary can sample $\mathbf{w}$ and compute $c \leftarrow H(\mathbf{w}, m)$, then pass $(A, \mathbf{w} + c\mathbf{t})$ to the provided module-ISIS solver. The output from this solver, denoted by $\mathbf{z}$, is such that

$$\begin{cases} A\mathbf{z} = \mathbf{w} + c\mathbf{t} \\ \|\mathbf{z}\|_\infty < \gamma - np\tau \end{cases}$$

In other words, $(m, (\mathbf{w}, c, \mathbf{z}))$ will pass the verification for the challenge keypair, making it a valid forgery.

(2)

Knowing that $\mathbf{y}$'s infinity norm is bounded by γ, we can use the given module-ISIS solver to recover $\mathbf{y}$ from the relation $\mathbf{w} = A\mathbf{y}$. Once $\mathbf{y}$ is recovered, it is possible to recover the secret key $\mathbf{s}$ using the relation $\mathbf{z} = \mathbf{y} + c\mathbf{s}$.

Note that because A is wide, $A\mathbf{y} = \mathbf{w}$ might not have unique solution. We can speculate the condition on which solutions are unique. One sufficient condition is that the linear mapping is injective, which requires

that the domain is smaller than the co-domain. The domain of the linear mapping is $R^l_{[-\gamma,\gamma]}$ (l polynomials of degree n each having coefficients in the range $[-\gamma, \gamma]$), whose size is approximately:

$$|R^l_{[-\gamma,\gamma]}| = (2\gamma + 1)^{ln}$$

On the other hand, there is no restriction on $\mathbf{w} \in R^k_q$:

$$|R^k_q| = q^{nk}$$

For the linear mapping to be injective, $|R^l_{[-\gamma,\gamma]}| < |R^k_q|$, which results in a bound on γ:

$$\gamma < \frac{1}{2}(q^{\frac{k}{l}} - 1)$$

Finally, here is the key recovery attack:

Algorithm 4 Key-recovery attack under CMA

Require: pk, module-ISIS solver

1: Generate random message m, and query its signature $\sigma = (\mathbf{w}, c, \mathbf{z})$
2: Use module-ISIS solver to solve for $\mathbf{y}$ in $A\mathbf{y} = \mathbf{w}$
3: Compute $\mathbf{s} \leftarrow (\mathbf{z} - \mathbf{y}) \cdot c^{-1}$
return s is the secret key