Algorithm r5_cca_kem_encapsulate(pk)

parameters: Integers $p, t, q, n, d, \overline{m}, \overline{n}, \mu, b, \kappa, f, \tau$; $\xi \in \{\Phi_{n+1}(x), x^{n+1} - 1\}$
input : $pk \in \{0,1\}^\kappa \times \mathcal{R}_{n,p}^{d/n \times \overline{n}}$
output : $ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v}, g) \in \mathcal{R}_{n,p}^{\overline{m} \times d/n} \times \mathbb{Z}_t^\mu \times \{0,1\}^\kappa, k \in \{0,1\}^\kappa$

$m \xleftarrow{\$} \{0,1\}^\kappa$
$(L, g, \rho) = G(m || pk)$
$(\tilde{\boldsymbol{U}}, \boldsymbol{v}) =$ r5_cpa_pke_encrypt(pk, m, ρ)
$ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v}, g)$
$k = H(L || ct)$
return (ct, k)

Algorithm r5_cca_kem_decapsulate(ct, sk)

parameters: Integers $p, t, q, n, d, \overline{m}, \overline{n}, \mu, b, \kappa, f, \tau$; $\xi \in \{\Phi_{n+1}(x), x^{n+1} - 1\}$
input : $ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v}, g) \in \mathcal{R}_{n,p}^{\overline{m} \times d/n} \times \mathbb{Z}_t^\mu \times \{0,1\}^\kappa, sk = (sk_{CPA-PKE}, y, pk) \in \{0,1\}^\kappa \times \{0,1\}^\kappa \times (\{0,1\}^\kappa \times \mathcal{R}_{n,p}^{d/n \times \overline{n}})$
output : $k \in \{0,1\}^\kappa$

$m' =$ r5_cpa_pke_decrypt$(sk_{CPA-PKE}, (\tilde{\boldsymbol{U}}, \boldsymbol{v}))$
$(L', g', \rho') = G(m' || pk)$
$(\tilde{\boldsymbol{U}}', \boldsymbol{v}') =$ r5_cpa_pke_encrypt(pk, m', ρ')
$ct' = (\tilde{\boldsymbol{U}}', \boldsymbol{v}', g')$
if $(ct = ct')$ **then**
    return $k = H(L' || ct)$
else
    return $k = H(y || ct)$
end if

Algorithm r5_cpa_pke_encrypt(pk, m, ρ)

parameters: Integers $p, t, q, n, d, \overline{m}, \overline{n}, \mu, b, \kappa, f, \tau$; $\xi \in \{\Phi_{n+1}(x), x^{n+1} - 1\}$
input : $pk = (\sigma, \boldsymbol{B}) \in \{0,1\}^\kappa \times \mathcal{R}_{n,p}^{d/n \times \overline{n}}, m, \rho \in \{0,1\}^\kappa$
output : $ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v}) \in \mathcal{R}_{n,p}^{\overline{m} \times d/n} \times \mathbb{Z}_t^\mu$

$\boldsymbol{A} = \boldsymbol{f}_{d,n}^{(\tau)}(c\,,$
$\boldsymbol{R} = f_R(\rho)$
$\boldsymbol{U} = \mathrm{R}_{q \to p, h_2}(\langle \boldsymbol{A}^T \boldsymbol{R} \rangle_{\Phi_{n+1}})$
$\tilde{\boldsymbol{U}} = \boldsymbol{U}^T$
$\boldsymbol{v} = \langle \mathrm{R}_{p \to t, h_2}(\mathrm{Sample}_\mu(\langle \boldsymbol{B}^T \boldsymbol{R} \rangle_\xi)) + \frac{t}{b} xef\_compute_{\kappa,f}(m) \rangle_t$
$ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v})$
return ct

Algorithm r5_cpa_pke_decrypt(sk, ct)

parameters: Integers $p, t, q, n, d, \overline{m}, \overline{n}, \mu, b, \kappa, f$; $\xi \in \{\Phi_{n+1}(x), x^{n+1} - 1\}$
input : $sk \in \{0,1\}^\kappa, ct = (\tilde{\boldsymbol{U}}, \boldsymbol{v}) \in \mathcal{R}_{n,p}^{\overline{m} \times d/n} \times \mathbb{Z}_t^\mu$
output : $\hat{m} \in \{0,1\}^\kappa$

$\boldsymbol{v}_p = \frac{p}{t} \boldsymbol{v}$
$\boldsymbol{S} = f_s(sk)$
$\boldsymbol{U} = \tilde{\boldsymbol{U}}^T$
$\boldsymbol{y} = \mathrm{R}_{p \to b, h_3}(\boldsymbol{v}_p - \mathrm{Sample}_\mu((\boldsymbol{S}^T(\boldsymbol{U} + h_4 \boldsymbol{J}))_\xi))$
$\hat{m} = xef\_decode_{\kappa,f}(\boldsymbol{y})$
return $\hat{m}$