Distributed ElGamal Encryption

- KeyGen $(1^\lambda, n)$; choose a large prime p and $g \xleftarrow{\$} \mathbb{Z}_p^*$.

  Each player chooses $s_i \xleftarrow{\$} \mathbb{Z}_p^*$.

  computes $y_i = g^{s_i}$ (along with ZKP, PoK_{DL} of y_i)

  output $pk = (p, g, \Pi y_i)$ and $sk = (\Sigma s_i)$

- Enc (pk, m) ; choose a random $r \xleftarrow{\$} \mathbb{Z}_p^*$.

  output $(g^r, m \cdot y^r)$ where $y = \Pi y_i$.

- Dec (sk_i, c) ; parse c as (c_1, c_2).

  (partial decryption) Each player computes and publish $c_1^{s_i}$. (along with ZKP. PoK_{EQ} of y_i and $c_1^{s_i}$)

  (msg reconstruction) output $c_2 \cdot (\Pi c_1^{s_i})^{-1}$.

Note.

- In order to multiply two ciphertexts, the underlying public key should be the same.
- Every player should take part in the decryption phase.