0.1 Paillier Encryption

Paillier encryption relies on the hard problem that is computing $n'th$ residuosity classes. A typical Paillier encryption scheme consists of the following algorithms.

- KeyGen(p, q): It takes two prime numbers p, q as input, it computes $n = p \cdot q$. It selects an integer $g \in Z_{n^2}^*$, such that n and $L(g^\lambda mod n^2)$ are coprime. The function L is defined as: $Z_{n^2}^* \rightarrow Z_n$, $u \rightarrow (u-1)/n$, and λ is the Carmichael function $\lambda(p \cdot q) = lcm(p-1, q-1)$. The output is the public key $pk = (n, g)$, and secret key $sk = (p, q)$.

- Enc(m, pk): It takes the messaage m and public key pk as input. Firstly, it chooses a random number $r \in Z_{n^2}^*$, then computes the encryption as $c = g^m \cdot r^n$ mod n^2. Fincally, this algorithm outputs $c \in Z_{n^2}^*$.

- Dec(c, sk): On input the ciphertext c, it decrypts the message by computing m=$L(c^\lambda \bmod n^2)/L(g^\lambda \bmod n^2) \bmod n$.

0.1.1 Homomorphic Property

The Paillier encryption scheme has the additivly homomorphic property, which plays an inportant roal in threshold signature schemes. For example, suppose we have c_1=Enc(m_1, pk)=$g^{m_1} \cdot {r_1}^n \bmod n^2$, c_2=Enc(m_2, pk)=$g^{m_2} \cdot {r_2}^n \bmod n^2$, then, $c_1 \cdot c_2$=$g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \bmod n^2$. This means that Enc$(m_1, pk)$+Enc$(m_2, pk)$=Enc$(m_1 + m_2, pk)$