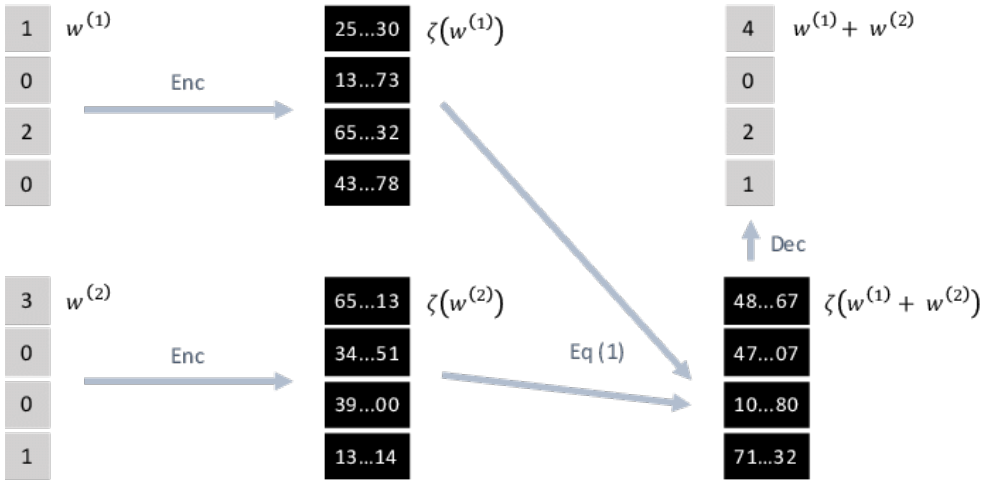
Secure Image Processing using Paillier Homomorphic Encryption

Abstart:

Securing data online against attackers is a big task to acheive. But we can acheive this with cryptographic algorithms. But when it comes to processing images it is a long process of encrypting and decrypting images several times to preserve confidentiality. But with Homomorphic encryptions like Paillier we can acheive processing ability with decrypting the data.

1. **Homomorphic Encryption** : It is a form of encryption that permits users to perform computations on it's encrypted data without decrypting it. These resulting computations are left in an encrypted form which, when decrypted, result is identical output to that produced had the operations been performed on unencrypted data.



2. **Paillier Encryption :** The Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumptions in the intractability hypothesis upon which this Paillier crytosystem is based. This cryptosystem is partially homomorphic that allows addition operations on it.

$$E(m1)+E(m2)\rightarrow E(m1+m2)$$

Algorithm:

1. Key Generation:

- Choose two large numbers p and q randomly and independently of each other such that $gcd(pq,(p-1)(q-1))=1$.

- Compute $n=pq$ and $\lambda=lcm(p-1,q-1)$. lcm is least comman multiple.

- Select random integer g where $g\in Z^*_{n^2}$

- Ensure n divides the order of g by checking the existance of the following modular multiplicative inverse : $\mu=(L(g^{\lambda} mod\, n^2))^{-1} mod\, n$ where function L is defined as $L(x)=(x-1)/n$
- Public key is (n, g)

- Private Key is (λ, μ)

2. Encryption:

- Let m be the message to be encrypted where $0 \le m < n$

- Select random r where $0 < r < n$ and $r \in Z_n^*$ (i.e ensure $gcd(r, n)=1$)

- Compute cipher text as $c = g^m . r^n mod\, n^2$

3. Decryption:

- Let c be the ciphertext to decrypt, where $c \in Z_{n^2}^*$

- Compute the plaintext as $m = L(c^{\lambda} mod\, n^2) . \mu mod\, n$

3. Homomorphic Properties:

1. Addition of Cipher Data:

$$D(E(m_1, r_1) . E(m_2, r_2) mod\, n^2) = m_1 + m_2 mod\, n$$

4. Project Details

1. Encrypting images data at pixel level and storing cipher value at each position.
2. Both RGB, Gray scale images are involved.
3. Allowing image processing techniques like increasing brightness, color etc.
4. Again decrypting to produce output image.

5. Software Stack Used:

1. Python3 >= 3.6.0
2. Numpy >= 1.16.0

6. Project Flow: