https://docs.google.com/spreadsheets/d/10YMCa5YVRE3PG9hghgoUudEO9ib7XQQl/edit?usp=sharing&ouid=111502255533491874828&rtpof=true&sd=true

CONSTRUCTION 11.32

Let GenModulus be a polynomial-time algorithm that, on input 1^n, outputs (N, p, q) where $N = pq$ and p and q are n-bit primes (except with probability negligible in n). Define a public-key encryption scheme as follows:

- **Gen**: on input 1^n run GenModulus(1^n) to obtain (N, p, q). The public key is N, and the private key is $\langle N, \phi(N) \rangle$ = $\langle N, \phi \rangle$.
- **Enc**: on input a public key N and a message $m \in \mathbb{Z}_N$, choose a random $r \leftarrow \mathbb{Z}_N^*$ and output the ciphertext

$$c := [(1+N)^m \cdot r^N \bmod N^2].$$

- **Dec**: on input a private key $\langle N, \phi(N) \rangle$ and a ciphertext c, compute

$$m := \left[\frac{[c^{\phi(N)} \bmod N^2] - 1}{N} \cdot \phi(N)^{-1} \bmod N \right].$$

The Paillier encryption scheme.

```
>> N=14351
N = 14351
>> fy=14112
fy = 14112
>> N_2=int64(N*N)
N_2 = 205951201
>> m=256;
```

% Z_N*={z | gcd(z,N)=1}

```
>> r=int64(randi(N))
r = 2274
>> gcd(r,N)
ans = 1
```

$$c := [\underbrace{(1+N)^m}_{c_1} \cdot \underbrace{r^N}_{c_2} \bmod N^2].$$

$$m := \left[\underbrace{\frac{\overbrace{[c^{\phi(N)} \bmod N^2]}^{d_1} - 1}{N}}_{d_2} \cdot \underbrace{\phi(N)^{-1}}_{d_3} \bmod N \right].$$

```
>> c1=mod_exp(1+N,m,N_2)
c1 = 3673857
>> c2=mod_exp(r,N,N_2)
c2 = 185095907
>> c=mod(c1*c2,N_2)
cL = 39605469
```

```
>> d1=mod_exp(cL,fy,N_2)
d1 = 151704422
>> d2=mod((d1-1)/N,N)
d2 = 10571
>> fy_m1=mulinv(fy,N)
fy_m1 = 5224
>> mod(fy*fy_m1,N)
ans = 1
>> d3=fy_m1
d3 = 5224
```

mm = d_2 * d_3 mod N

```
>> mm=mod(d2*d3,N)
mm = 256
```

Total sum of votes
Tot_S_of_V = 785

```
>> N_of_V_Can1=floor(785/256)
N_of_V_Can1 = 3
>> N_of_V_Can12=785-3*256
N_of_V_Can12 = 17
>> N_of_V_Can2=floor(N_of_V_Can12/16)
```

```
N_of_V_Can2 = 1
>> N_of_V_Can1=785-3*256-1*16
N_of_V_Can3 = 1

>> Tot_N_of_V=N_of_V_Can1+N_of_V_Can2+N_of_V_Can3
Tot_N_of_V = 5
```