Koliokviumas vyks **3 Lapkričio 13:30 506 a**.
Reikės realizuoti eBalsavimo sistemą.

https://docs.google.com/spreadsheets/d/1joAuG6oh1arcQqc7zPTPbLBH48wG8Fn6/edit?usp=sharing&ouid=111502255533491874828&rtpof=true&sd=true

P.Vardas	No	ri	ci	c	V_by_M: cMi	Dec(cMi)	Yes/ No	Mul(c)	Mul(c)*cMi	Dec(c*cMi)	Tot_S_of_V	N_of_V_Can1	N_of_V_Can2	N_of_V_Can3	Tot_N_of_V
ES	1	11692	12891778	57215918	132901443			133052728							
	2														
	3														
	4														
	5														
	6														
	7														
	8														
	9														
	10														
	11														
	12														
	13														
	14														
	15														
	16														

N=		14351
N_2=		205951201
fy=		14112
ri		Random number generated for Paillier encryption
ci		Your vote encrypted by Paillier encryption
c		The product of all encrypted votes in your polling station. **Provided by lecturer**
V_by_M: cMi		Encrypted Vote received by Mail: cMi. **Provided by lecturer**
c*cMi		Multiplied encrypted votes in polling station multiplied by cMi
Dec(c*cMi)		Decryption all multiplied votes
Tot_S_of_V		Total sum of votes
N_of_V_Can1		Number of votes for Can1
N_of_V_Can2		Number of votes for Can2
N_of_V_Can3		Number of votes for Can3
Tot_N_of_V		Total number of votes
Dec(c*Mi)		Decrypted vote cMi received by Mail
Acc/Dec cMi		Accept or Decline vote received by Mail. Input: **Acc** or **Dec**
Can1		Number of votes for Can1
Can2		Number of votes for Can2
Can3		Number of votes for Can3

N
N_2
fy

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.

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

$e_1 \bmod N^2 \quad e_2 \bmod N^2$

$c = e = e_1 \cdot e_2 \bmod N^2$

>> ri=genprime(14)

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

d_1

$d_2 \bmod N \qquad d_3 \bmod N$

$m = d_2 \cdot d_3 \bmod N$

```
>> d1=mod_exp(c,fy,N_2)
d1 = 197426708
>> d2=mod((d1-1)/N,N)
d2 = 13757
>> d3=mulinv(fy,N)
d3 = 5224
>> mm=mod(d2*d3,N)
mm = 11111
```

C:\Users\Eligijus\Documents\500 SOFTAS App 2023\Python 3.9.1\Paii_Enc Mult-mod_N2

mul N_2