2 Asymmetric Encryption

In the following we describe a simple description of an asymmetric encryption mechanism. The idea is to represent encryption and decryption with table lookups. To this end, three types of tables are used:

M1 is just a sequence of N elements and contains a random permutation of all integers between 1 and N. For example $m1 = (4, 3, 2, 5, 1)$ could be an example for $N = 5$. It is used to construct a public key from a private key. For example, if we assume that our private key is 2, then the corresponding public key, according to our example table $m1$, is given by $m1(2) = 3$.

M2 is an $N \times N$ matrix such that each row represents a random permutation of all integers between 1 and N. For example, for $N = 5$ we may have:

$$m2 = \begin{matrix} 3 & 1 & 2 & 5 & 4 \\ 1 & 4 & 3 & 2 & 5 \\ 4 & 5 & 2 & 3 & 1 \\ 3 & 2 & 1 & 4 & 5 \\ 2 & 3 & 5 & 1 & 4 \end{matrix}$$

It is used for encryption. For example, to encrypt 3 using our table $m2$ and key 2, we get $m2(2, 3) = 3$.

M3 is an $N \times N$ matrix such that each column represents a random permutation of all integers between 1 and N. It is used for decryption.

The tables must be constructed in a way such that for all k and p, with $1 \le k, p \le N$ the following property holds:

$$M3(M2(M1(k), p), k) = p \tag{1}$$

2.1 Tasks

T2.1 Construct an example of M1, M2, and M3 for $N = 5$. Hint: first, randomly create M1 and M2 and then construct M3 such that property Eq. (1) holds.

T2.2 Encrypt the number 3 and then decrypt it again.

T2.3 Is the scheme secure? Explain why/why not.

T2.4 Implement the key generation scheme in Python. It should be called `myPKE` and take one input parameter which represents N. It should then generate three random tables $m1$, $m2$, and $m3$ which satisfy Eq. 1. For example:

```
>myPKE 5
>m1:
>4,3,2,5,1
>m2:
>3,1,2,5,4
>1,4,3,2,5
>4,5,2,3,1
>3,2,1,4,5
>2,3,5,1,4
>m3:
>........
>........
>........
>........
>........
```