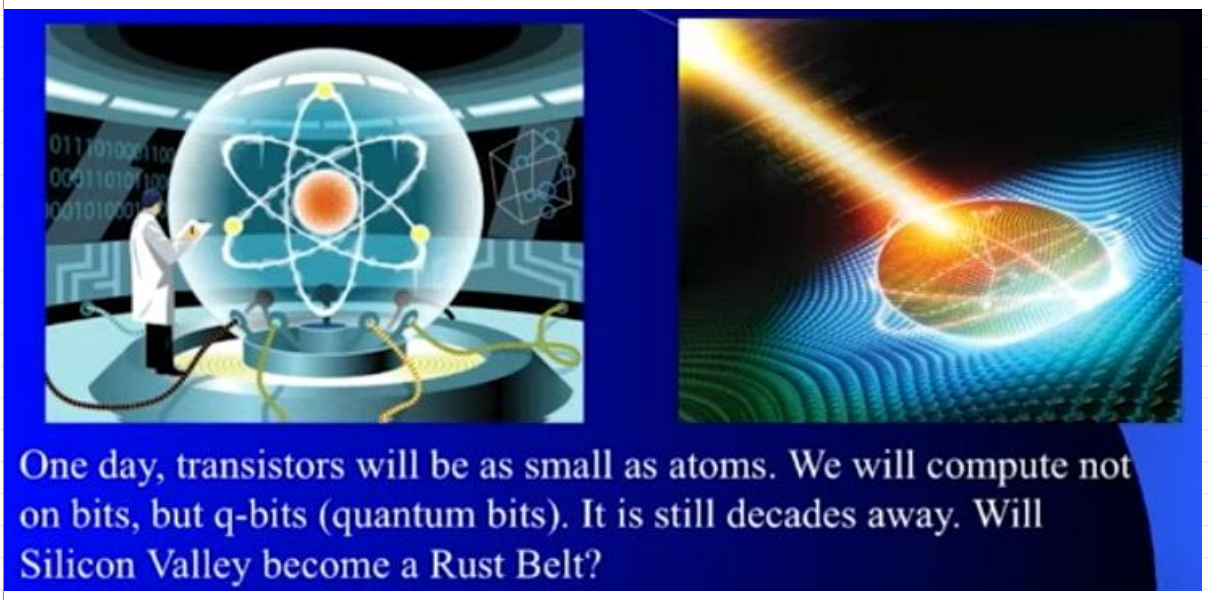
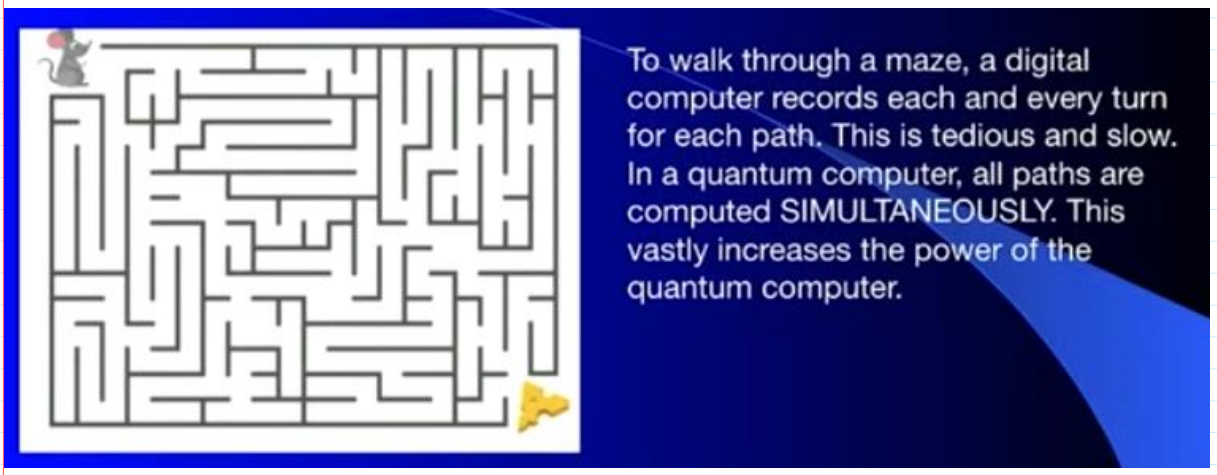
`127_011_Post-Quantum-1-2

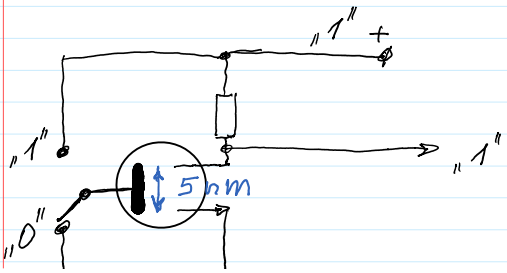
Michio Kaku
https://www.youtube.com/watch?v=_OjRClPzU6Y

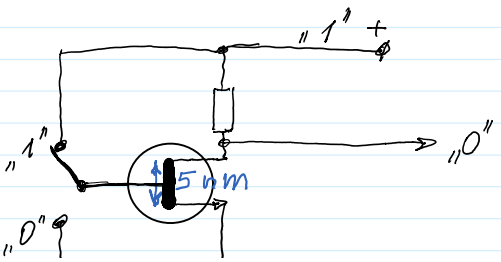


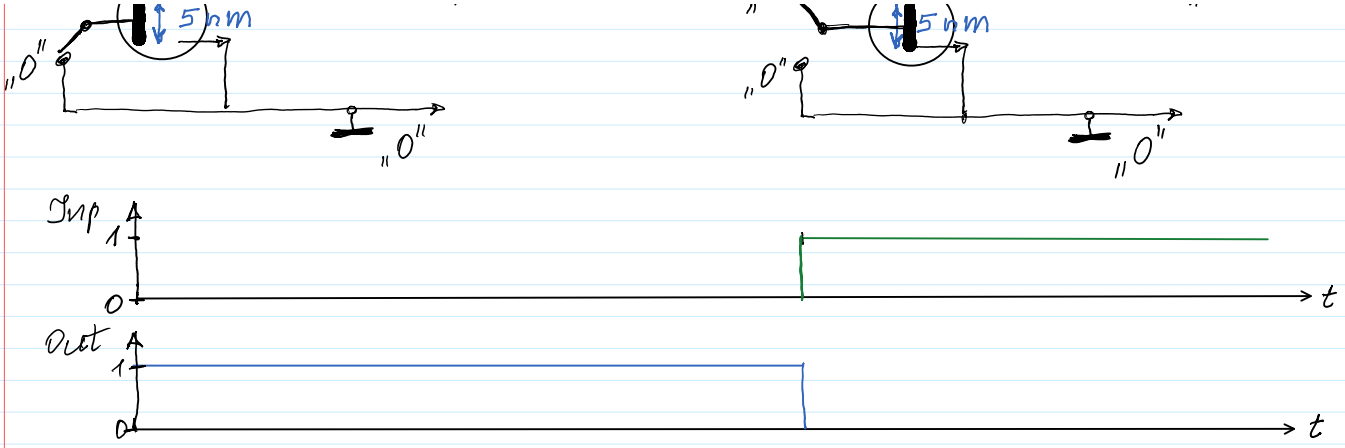


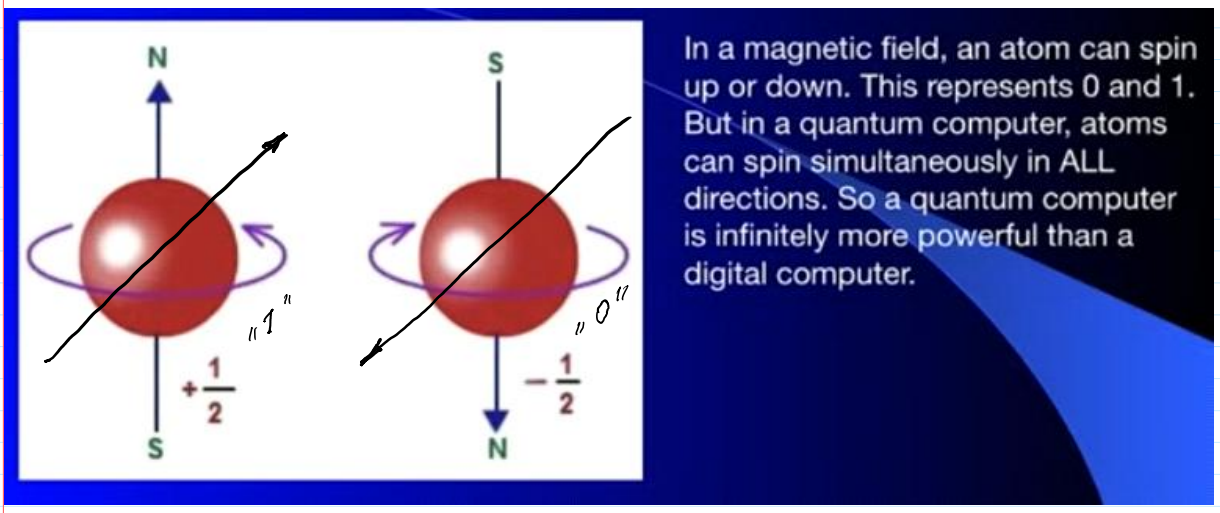
https://fb.watch/vIEogSoMB8/

23 min
https://www.youtube.com/watch?v=gfUEUhDbGXA

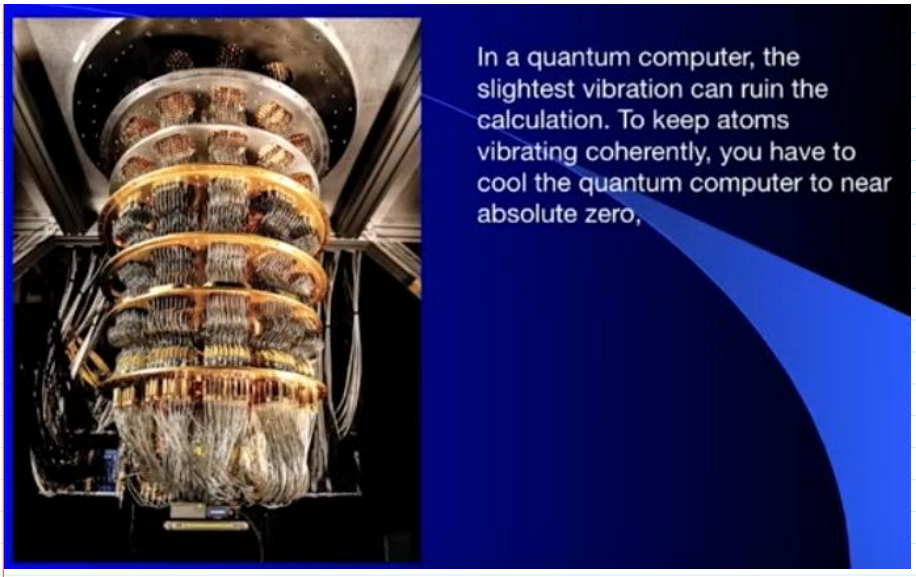


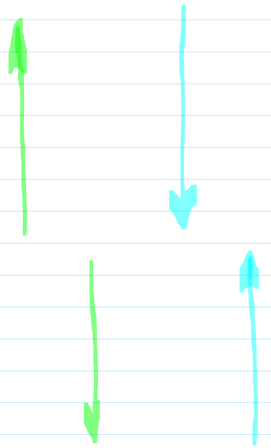



5 nm
"0"
"0"
5 nm
"0"
"0"
Inp
1
0
t
Out
1
0
t


N
S
"1"
+1/2
S
N
"0"
−1/2
In a magnetic field, an atom can spin up or down. This represents 0 and 1. But in a quantum computer, atoms can spin simultaneously in ALL directions. So a quantum computer is infinitely more powerful than a digital computer.


In a digital computer, each transistor is independent of each other.
In a quantum computer, all atoms are entangled with each other, with information flowing between them, increasing their power.





Alfred Menezes

Y2Q - Year to Quantum.
Physical q-bits are non-stable.
To create stable quantum computation it is required to create a lot physical q-bits creating so called Logical q-bit.
For 1 logical q-bit it is required hundreds of physical q-bits.
Then errors ocuuring during the computatioms are corrected - similarly as Error Correcting Codes.

PQC method based on Learning With Errors - LWE

System of linear equations: **Alex**, **Bob**, **Cecilia** - are programmers.
Has certain reward per hour in CBDC.
In Monday **Alex** worked 1 hour, **Bob** worked 2 hours, **Cecilia** worked 3 hours: they all earned 14 Ⓒ. SALARY
In Tuesday **Alex** worked 2 hours, **Bob** worked 1 hour, **Cecilia** worked 2 hours: they all earned 10 Ⓒ.
In Wednesday **Alex** worked 1 hour, **Bob** worked 2 hours, **Cecilia** worked 3 hours: they all earned 13 Ⓒ.

How many **Alex**, **Bob**, **Cecilia** are earning per hour?
The rewards per hour for **Alex**, **Bob**, **Cecilia**
We denote by x y z

Then to find 3 unknown variables x, y, z we have a system 3 of linear equations:

Then to find 3 unknown variables x, y, z we have a system 3 of linear equations:
1*x + 2*y + 3*z = 14 (1)
2*x + 1*y + 2*z = 10 (2)
3*x + 2*y + 2*z = 13 (3)

This system of equations can be written matrix form: ***Mw*** = ***s***,
Where ***w***′ = (x, y, z) is vector of unknown salaries x, y, z and ***s*** is a vector of total salaries of every day written in column.
Transposed vector ***s*** we denote by ***s***' then it can be written in a row: ***s***' = (14, 10, 13).
This data in Octave is formed in the following way:

This data in Octave is formed in the following way:

```
>> M=[1 2 3; 2 1 2; 3 2 2]
M =
   1   2   3
   2   1   2
   3   2   2
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 2 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 14 \\ 10 \\ 13 \end{pmatrix}$$

w

```
>> s=[14;10;13]
s =
   14
   10
   13
```

```
>> st=s'
st =
   14   10   13
```

$5*z = 15$

$5^{-1} * 5*z = 5^{-1} * 15 \rightarrow z = 3$

$m*z = s$

$m^{-1} * m*z = m^{-1} * s$

$1*z = m^{-1} * s$

$z = m^{-1} * s$

$M*w = s$

$M^{-1} * M*w = M^{-1} * s$

$I*w = M^{-1} * s$

$w = M^{-1} * s$

```
>> Mi=inv(M)
Mi =
  -0.4000   0.4000   0.2000
   0.4000  -1.4000   0.8000
   0.2000   0.8000  -0.6000
```

% M^{-1}

```
>> I=Mi*M
I =
   1.0000   0.0000   0.0000
  -0.0000   1.0000   0.0000
   0.0000        0   1.0000
```

```
>> ww=Mi*s
ww =
   1
   2
   3
```

In Thursday **Alex** worked 3 hour, **Bob** worked 3 hours, **Cecilia** worked 1 hour: they all earned 12 $\mathcal{D}$.

This 4-th condition yields the 4-th equation of the form:

$3*x + 3*y + 1*z = 12$ (4)

Then we have the following system of 4 equations:

$$\left.\begin{aligned} 1*x + 2*y + 3*z &= 14 \quad (1) \\ 2*x + 1*y + 2*z &= 10 \quad (2) \\ 3*x + 2*y + 2*z &= 13 \quad (3) \\ 3*x + 3*y + 1*z &= 12 \quad (4) \end{aligned}\right\}$$

Overdefined linear system of equations

Let's consider the system consisting of (1), (2), (3) equations and their matrix denoting by M123, and the system consisting of (2), (3), (4) equations and their matrix denoting by M234

$1*x + 2*y + 3*z = 14$ (1)
$2*x + 1*y + 2*z = 10$ (2)
$3*x + 2*y + 2*z = 13$ (3)

$2*x + 1*y + 2*z = 10$ (2)
$3*x + 2*y + 2*z = 13$ (3)
$3*x + 3*y + 1*z = 12$ (4)

The matrices of these two systems of equations we denote by M123 and M234 respectively.
The salaries vectors of these two systems of equations we denote by s123 and s234 respectively.
The unknown variables x, y, z of these systems we denote by the vectors w123 and w234 respectively.
It is evident that vectors w123 and w234 satisfying both systems are equal, i.e. w123 = w234 = w.

```
>> w=[1;2;3]
```

```
w =
   1
   2
   3
```

```
>> M123=[1 2 3; 2 1 2; 3 2 2]
M123 =
   1   2   3
   2   1   2
   3   2   2
```

```
>> s123=[14;10;13]
S123 =
   14
   10
   13
```

```
>> M234=[2 1 2; 3 2 2; 3 3 1]
M234 =
```

$$\begin{pmatrix} 2 & 1 & 2 \\ 3 & 2 & 2 \\ 3 & 3 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} = = \begin{pmatrix} 10 \\ 13 \\ 12 \end{pmatrix}$$

```
>> s234=[10;13;12]
s234 =
```

```
>> M123i=inv(M123)
M123i =
  -0.4000   0.4000   0.2000
   0.4000  -1.4000   0.8000
   0.2000   0.8000  -0.6000
```

```
>> I=M123i*M123
I =
   1.0000   0.0000   0.0000
  -0.0000   1.0000   0.0000
   0.0000        0   1.0000
```

```
>> w123=M123i*s123
w123 =
   1
   2
   3
```

```
>> M234i=inv(M234)
M234i =
  -4   5  -2
   3  -4   2
   3  -3   1
```

```
>> I=M234i*M234
I =
   1.0000e+00  -8.8818e-16  -8.8818e-16
   6.6613e-16   1.0000e+00   2.2204e-16
   6.6613e-16   2.2204e-16   1.0000e+00
```

```
>> w234=M234i*s234
w234 =
   1
   2
   3
```

Let's change a little the initial system of 4 equations by adding to the right side -1, 1, or 0 at random.

1*x + 2*y + 3*z = 14	(1)	1*x + 2*y + 3*z = 14 - 1	1*x + 2*y + 3*z = 13 = s1e
2*x + 1*y + 2*z = 10	(2)	2*x + 1*y + 2*z = 10 + 0	2*x + 1*y + 2*z = 10 = s2e
3*x + 2*y + 2*z = 13	(3)	3*x + 2*y + 2*z = 13 +1	3*x + 2*y + 2*z = 14 = s3e
3*x + 3*y + 1*z = 12	(4)	3*x + 3*y + 1*z = 12 - 1	3*x + 3*y + 1*z = 11 = s4e

Then we introduce the erroneous vector for the system consisting of (1), (2), (3) equations denoting it by s123, and the erroneous vector for the system consisting of (2), (3), (4) equations denoting it by s234.
The corresponding matrices M123 and M234 are introduced above.

```
>> s123e=[13;10;14]
 s123e =
   13
   10
   14
```

```
>> s234e=[10;14;11]
s234e =
   10
   14
   11
```

Then the solution of the first system of equations can be found by computing the inverse matrix to the matrix M123 we denote by M123i, and the solution of the second system of equations can be found by computing the inverse matrix to the matrix M234 we denote by M234i

```
>> w123e=M123i*s123e
w123e =
 1.6000
   2.4000
   2.2000
```

```
>> w234e=M234i*s234e
w234e =
  6.0000e+00
 -2.0000e+00
  7.1054e-15
```

As we see solutions differs.

Let's change a little the initial system of 4 equations by adding to the right side by numbers -1, 1, or 0 at random.

1*x + 2*y + 3*z = 14	(1)	1*x + 2*y + 3*z = 14 - 1	1*x + 2*y + 3*z = 13 = s1e	15+
2*x + 1*y + 2*z = 10	(2)	2*x + 1*y + 2*z = 10 + 0	2*x + 1*y + 2*z = 10 = s2e	11
3*x + 2*y + 2*z = 13	(3)	3*x + 2*y + 2*z = 13 +1	3*x + 2*y + 2*z = 14 = s3e	13+
3*x + 3*y + 1*z = 12	(4)	3*x + 3*y + 1*z = 12 - 1	3*x + 3*y + 1*z = 11 = s4e	12+
(1) + (3) + (4) = 7		7*x + 7*y + 6*z = 7		=7 mod 11

Then we introduce the erroneous vector for the system corresponding to (1), (2), (3) equations denoting it by s123e, and the erroneous vector for the system corresponding to (2), (3), (4) equations denoting it by s234e.
The corresponding matrices M123 and M234 are the same.

Then the solution of the first system of equations can be found by computing the inverse matrix to the matrix M123 we denote by M123i, and
the solution of the second system of equations can be found by computing the inverse matrix to the matrix M234 we denote by M234i

```
>> M123=[1 2 3; 2 1 2; 3 2 2]
M123 =
   1   2   3
   2   1   2
   3   2   2
```

```
>> s123e=[13;10;14]
 s123e =
   13
   10
   14
```

```
>> M234=[2 1 2; 3 2 2; 3 3 1]
M234 =
   2   1   2        x
   3   2   2        y
   3   3   1        z
```

```
>> s234e=[10;14;11]
s234e =
   10
   14
   11
```

```
>> M123i=inv(M123)
M123i =
  -0.4000   0.4000   0.2000
   0.4000  -1.4000   0.8000
   0.2000   0.8000  -0.6000
```

```
>> M234i=inv(M234)
M234i =
  -4   5  -2
   3  -4   2
   3  -3   1
```

```
>> w123e=M123i*s123e
w123e =
 1.6000
```

```
>> w234e=M234i*s234e
w234e =
```

```
>> w123e=M123i*s123e
w123e =
  1.6000
  2.4000
  2.2000
```

```
>> w234e=M234i*s234e
w234e =
   6.0000e+00
  -2.0000e+00
   7.1054e-15
```

As we see solutions differs.

Learning With Errors - LWE Encryption



Time is computed modulo 12.
We will perform computations modulo 11.

In classical cryptography computations are performed modulo ***p***, when ***p*** is a large prime number.
Let ***p*** is prime then for all positive integers ***z***, expression ***z*** mod ***p*** is a reminder of ***z*** divided by ***p***.

Let ***p***=11.
Then computations of equations (1), (2), (3) and equations (2), (3), (4) we perform modulo 11.
Matrices M123, M234 and s123, s234 we leave the same and operations we perform modulo 11.
Matrix operations mod p are realized in .m files.
Library of .m files



We are using functions **matinv.m** and **matmult.m**
To find inverse matrix **Mi** mod ***p*** to matrix **M** we use
>> Mi=matinv(M,p)
To multiply inverse matrix **Mi** with vector **s** mod ***p*** to obtain vector **w** we use command
>> w=matmult(Mi,s,p)

Let's **again** change a little the initial system of 4 equations by adding to the right side the numbers -1, 1, or 0 at random.

Initial equations		Equations with errors	Final equations with errors
1*x + 2*y + 3*z = 14	(1)	1*x + 2*y + 3*z = 14 - 1	1*x + 2*y + 3*z = 13 = s1e
2*x + 1*y + 2*z = 10	(2)	2*x + 1*y + 2*z = 10 - 0	2*x + 1*y + 2*z = 9 = s2e
3*x + 2*y + 2*z = 13	(3)	3*x + 2*y + 2*z = 13 + 1	3*x + 2*y + 2*z = 14 = s3e
3*x + 3*y + 1*z = 12	(4)	3*x + 3*y + 1*z = 12 + 1	3*x + 3*y + 1*z = 13 = s4e

```
>> M123=[1 2 3; 2 1 2; 3 2 2]      >> s123e=[13;10;14]
M123 =        w123e=               s123e =
  1  2  3       x                    13
  2  1  2       y                    10
  3  2  2       z                    14

>> M123i=matinv(M123,p)
M123i =
  4  7  9
  7  3  3
  9  3  6

>> I123=matmult(M123i,M123,p)
I123 =
  1  0  0
  0  1  0
  0  0  1

>> w123e=matmult(M123i,s123e,p)
w123e =
  6
  9
  0
```

```
>> M234=[2 1 2; 3 2 2; 3 3 1]      >> s234e=[10;14;11]
M234 =        w=234e               s234e =
  2  1  2       x                    10
  3  2  2       y                    14
  3  3  1       z                    11

>> M234i=matinv(M234,p)
M234i =
  7  5  9
  3  7  2
  3  8  1

>> I234=matmult(M234i,M234,p)
I234 =
  1  0  0
  0  1  0
  0  0  1

>> w234e=matmult(M234i,s234e,p)
w234e =
  8
  7
  10
```

As we see solutions differs as well since right sides of equations are corrupted by errors.
System of equations (1), (2), (3), (4) is inconsistent since if solution w123e satisfies one part of equations (1), (2), (3) it does not satisfies the other part of equations (2), (3), (4).

Exercises

Public parameters - **PP** is a system of 4 linear equations
with errors in the right-side computed mod 11:

1*x + 2*y + 3*z = 13 mod11 = 2 = s1e
2*x + 1*y + 2*z = 9 mod 11 = 9 = s2e
3*x + 2*y + 2*z = 14 mod 11 = 3 = s3e
3*x + 3*y + 1*z = 13 mod 11 = 2 = s4e

Learning with errors - LWE encryption is performed for a single bit ***b***={0, 1} at once.

The private key is a vector of unknown but **exact** wages per hour: **PrK** = **w'** = (x, y, z) = (1, 2, 3),
where **w'** is written as a row vector and is a transpose vector of vector column **w**.
w' = [1,2,3] = **PrK**.

To encrypt bit ***b*** a random sub-system of equations is selected, e.g. let's take (1), (2), (3) sub-system of equations.
Then erroneous vector for the sub-system corresponding to (1), (2), (3) equations we denote by s123e.
The corresponding matrix M123 is the same.
The equations and erroneous vectors are of the form.

1*x + 2*y + 3*z = 2 = s1e (1)
2*x + 1*y + 2*z = 9 = s2e (2)
3*x + 2*y + 2*z = 3 = s3e (3)

```
>> M123=[1 2 3; 2 1 2; 3 2 2]
M123 =
   1   2   3      x
   2   1   2      y
   3   2   2      z
```

```
>> s123e=[13;10;14]
 s123e =
    2
    9
    3
```

Encryption

Let **Alice** is intending to encrypt bit ***b***={0, 1} and send it to **Bob.**
By having public parameters **PP**, consisting of system of 4 equations, **Alice** chooses **at random** some sub-system consisting of 3 equations.
Let this sub-system consist of equations (1), (2), (3).
Having this sub-system, **Alice** adds (1), (2), (3) equations together mod 11:

```
   1*x + 2*y + 3*z = 2 = s1e  (1)
+  2*x + 1*y + 2*z = 9 = s2e  (2)
   3*x + 2*y + 2*z = 3 = s3e  (3)
-------------------------------
   6*x + 5*y + 7*z = 14 mod 11 = 3
```

If **Alice** intends to encrypt bit ***b***=**0** then she sends original equation 6*x + 5*y + 7*z = **3** corresponding to ciphertext $\mathbf{c_0}$ to **Bob**.

If **Alice** intends to encrypt bit ***b***=**1** then she sends equation 6*x + 5*y + 7*z = **3** + **5** = **8** corresponding to ciphertext $\mathbf{c_1}$ to **Bob**.
This means that for ***b***=1 encryption **Alice** adds to the right side of equation number **5**.

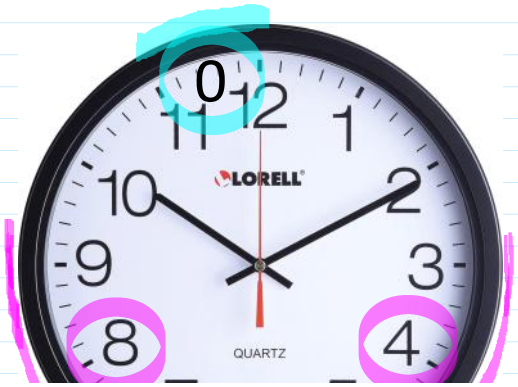
This number corresponds to the rounded down number $\lfloor p/2 \rfloor = \lfloor 11/2 \rfloor = \lfloor 5.5 \rfloor = 5$, i.e. it is equal to **5**.

Decryption

For decryption **Bob** is using his **PrK** = **w'** = (x, y, z) = (1, 2, 3).

Let **Bob** received ciphertext $\mathbf{c_0}$: 6*x + 5*y + 7*z = **3**.
Then **Bob** puts his **PrK** values (1, 2, 3) to the left side of $\mathbf{c_0}$ equation and obtains:
6*1 + 5*2 + 7*3 = 6 + 10 + 21 = 37 mod 11 = **4**
The number **4** is close to the right side number **3** of the received ciphertext $\mathbf{c_0}$: 6*x + 5*y + 7*z = **3**
This means that **Bob** decrypted bit ***b***=**0**.

Let **Bob** received ciphertext $\mathbf{c_1}$: 6*x + 5*y + 7*z = **8**.
Then **Bob** puts his **PrK** values (1, 2, 3) to the left side of $\mathbf{c_1}$ equation and obtains:
6*1 + 5*2 + 7*3 = 6 + 10 + 21 = 37 mod 11 = **4**
The number **4** is far from to the right side number **8** of the received ciphertext $\mathbf{c_1}$: 6*x + 5*y + 7*z = **8**.
The difference is **8** - **4** = **4** and it is more than the errors values fluctuating in the range [-1, 1].
This means that **Bob** decrypted bit ***b***=**1**.





$7*x + 7*y + 6*z = 7 = c_0$

w' = [1,2,3] = **PrK**.
7*1 + 7*2 + 7*3 = 42 mod 11 = 9

>> mod(15+13+12,11)
ans = 7
>> mod(42,11)
ans = 9

15+
11
13+
12+

=7 mod 11