In [1]:
```
n = 15
```

In [2]:
```
Zn = IntegerModRing(n)
```

In [3]:
```
Zn.list()
```

Out[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

In [5]:
```
Zn(-19)
```

Out[5]: 11

In [6]:
```
Zn(10)+Zn(8)
```

Out[6]: 3

In [7]:
```
Zn(10)* Zn(8)
```

Out[7]: 5

In [8]:
```
Zn(2)*Zn(8)
```

Out[8]: 1

In [9]:
```
1/Zn(2)
```

Out[9]: 8

In [10]:
```
euler_phi(15)
```

Out[10]: 8

Sejam p e q primos distintos, e $n = pq$. Definimos $m = \varphi(n) = (p-1)(q-1)$. Seja $e \in \mathbb{Z}_m^*$; ou seja, $e \in \mathbb{Z}_m$ tal que $(e, m) = 1$.

A Chave Pública é (n, e).

In [11]:
```
p = random_prime(2^32, 2^30)
q = random_prime(2^32, 2^30)
p ,q
```

Out[11]: (3345955037, 1178504321)

In [12]:
```
n = p*q
```

In [15]:
```
#m = euler_phi(n)  !!! NÃO FAZER ISTO !!!
```

In [16]:
```
m = (p-1)*(q-1)
```

In [18]:
```
Zm = IntegerModRing(m)
e = Zm.random_element()
gcd(e, m)
```
Out[18]: 1

In [19]:
```
PubKey = (n, e)
```

Bob publica $PubKey = (n, e)$.

Alice quer enviar $mens = 1234$ a Bob.

Alice calcula $cifr = mens^e \mod n$ e envia $cifr$ a Bob.

In [20]:
```
mens = 1234
Zn = IntegerModRing(n)
cifr = Zn(mens)^e
cifr
```
Out[20]: 1283134510482137927

In [21]:
```
d = 1/e
d
```
Out[21]: 2765008457644335157

d é a chave privada de Bob.

Bob calcula $cifr^d \mod n$.

In [23]:
```
cifr^d
```
Out[23]: 1234

In [27]:
```
mod(55, 16)
```
Out[27]: 7

In [28]:
```
crt(13, 11, 16, 15)
```
Out[28]: 221

In [29]:
```
crt(5, 0, 16, 15)
```
Out[29]: 165

In [30]:
```
crt(8, 0, 16, 15)
```
Out[30]: 120

In [31]:
```
16*15*13
```
Out[31]: 3120

In [32]:
```
mod(1421, 16), mod(1421, 15), mod(1421, 13)
```
Out[32]: (13, 11, 4)

In [33]:
```
mod(1100, 16), mod(1100, 15), mod(1100, 13)
```
Out[33]: (12, 5, 8)

In [34]:
```
crt(crt(9, 1, 16, 15), 12, 16*15, 13)
```
Out[34]: 2521

In [35]:
```
1421+1100
```
Out[35]: 2521

In [ ]:
```

```