Euler totient function **ϕ**(***n***): defines number of numbers **z** less than **n** that gcd(***z***,***n***)=1.
ϕ(***n***) = **ϕ** ≡ **fy**.
If ***n***=***p****q* where ***p***,***q***-primes then **ϕ**(***n***) = **ϕ** = (***p***-1)*(***q***-1) ≡ **fy**.
Let ***n***=3*5=15 --> **ϕ**(***n***) = **ϕ** = (3-1)*(5-1) = 2*4 = 8 ≡ **fy**.

Euler theorem. If gcd(***z***,***n***)=1 then

$$z^{\phi} = 1 \bmod n$$

According to Euler theorem exponents are computing mod ϕ.

```
>> p=3;
>> q=5;
>> n=p*q
n = 15
>> z=2;
>> mod_exp(2,8,n)
ans = 1
>> mod_exp(2,16,n)
ans = 1
>> mod_exp(2,32,n)
ans = 1
>> mod(8,8)
ans = 0
>> mod(16,8)
ans = 0
```

```
>> p=genprime(14)
p = 12409
>> dec2bin(p)
ans = 11 0000 0111 1001
>> q=genprime(14)
q = 11959
>> dec2bin(q)
ans = 10 1110 1011 0111
>> n=p*q
n = 148399231
>> dec2bin(n)
ans = 1000 1101 1000 0110 0100 0111 >> f111
>> factor(n) = 11959  12409
```

Exponents of numbers in Z_n are computed mod **ϕ**.

```
>> fy=(p-1)*(q-1)
fy = 148374864
>> m=1234567
>> e=2^16+1
e = 65537          % e computation according to
                   % RSA standard
>> isprime(e)
ans = 1
>> gcd(e,fy)
ans = 1
>> d=mulinv(e,fy)
```

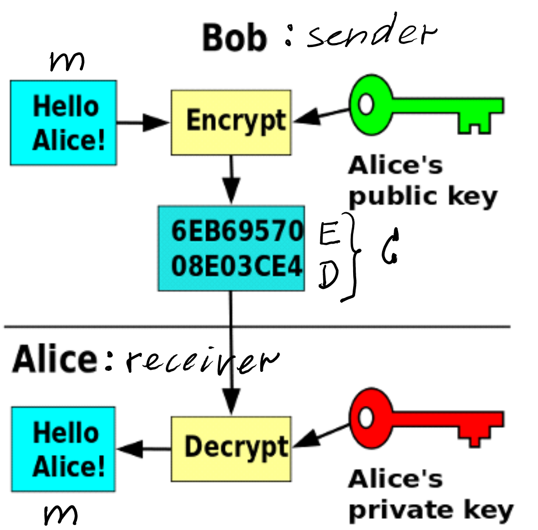
$$d = e^{-1} \bmod \phi \Rightarrow d \cdot e \bmod \phi = e \cdot d \bmod \phi = 1$$

RSA : $PuK=(n, e)$; $PrK=d \longrightarrow A$

Asymmetric Encryption - Decryption

$c = Enc(PuK_A, m)$

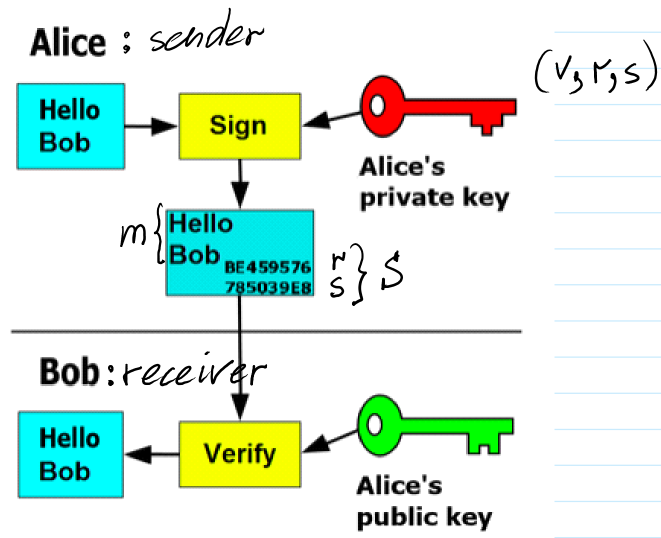
$m = Dec(PrK_A, c)$



Asymmetric Signing - Verification

$S = Sign(PrK_A, m)$

$V = Ver(PuK_A, m, s)$, $V \in \{\text{True, False}\} \equiv \{1, 0\}$



Encryption: $c = m^e \bmod n$

Decryption: $c^d \bmod n =$

$= (m^e)^d \bmod n = m^{ed \bmod \phi} \bmod n =$

$= m^1 \bmod n \overset{m<n}{=\!=} m$

```
>> m=111222
m = 111222
>> c=mod_exp(m,e,n)
c = 40923014
>> mm=mod_exp(c,d,n)
mm = 111222
```

Signing : $s = m^d \bmod n$

Verification: $V = s^e \bmod n =$

$= (m^d)^e \bmod n = m^{de \bmod \phi} \bmod n =$

$= m^1 \bmod n \overset{m<n}{=\!=} m$ // RSA signature with message recovery.

```
>> s=mod_exp(m,d,n)
s = 2893859
>> v=mod_exp(s,e,n)
v = 111222
```

To achieve security encrypt & sign paradigm is used to resist against so called Chosen Ciphertext Attack - CCA.

$\mathcal{A}$: $PuK_A = (n, e)$; $PrK_A = d$;

m - message to be sent to

1. $Enc(e_1, m) = c_1$ c_1, S_1

$\mathcal{B}$:

$PuK_B = (n_1, e_1)$

$PrK_B = d_1$

```
>> p1=genprime(14)
p1 = 9949
>> q1=genprime(14)
q1 = 10513
>> n1=p1*q1
```

m - message to be sent to ... $PuK_B = (n_1, e_1)$

>> q1=genprime(14)
q1 = 10513
>> n1=p1*q1
n1 = 104593837
>> fy1=(p1-1)*(q1-1)
fy1 = 104573376
e = 65537
>> d1=mulinv(e,fy1)
d1 = 18263681

1. $Enc(e_1, m) = c_1$
2. $Sign(d, c_1) = s_1$

$\xrightarrow{c_1,\ s_1}$

$PrK_B = d_1$

B: 1. Verifies signature s_1 on c_1

$Ver(PuK_A, s_1) = c_1$

$Ver(e, s_1) = c_1$

2. Derypts ciphertext c_1

$Dec(PrK_B, c_1) = m$

$Dec(d_1, c_1) = m$

To be continued during exercises lecture.

Masking with RSA: blind signature

A: want to withdraw money amount m from Bank B.

$PuK_B = (n_1, e_1)$

$PuK_B = (n_1, e_1)$

$PrK_B = d_1$

$r \leftarrow randi(\mathbb{Z}^*_{n1})$

Masking: $mask = (r^{e_1} \cdot m) \bmod n_1 \xrightarrow{mask}$ B:

$sign(d_1, mask) = s_1 =$

$= (r^{e_1} \cdot m)^{d_1} \bmod n_1 =$

$= (r^{e_1 d_1} \cdot m^{d_1}) \bmod n =$

$= (r^1 \cdot \underbrace{m^{d_1}) \bmod n_1}_{s_m}$

$\xleftarrow{s_1}$

$[(r^{-1}) \bmod n_1] \cdot s_1 \bmod n_1 =$

$= \cancel{r^{-1} \cdot r} \cdot m^{d_1} \bmod n_1 = s_m$

$Ver(PuK_B, s_m) = m.$