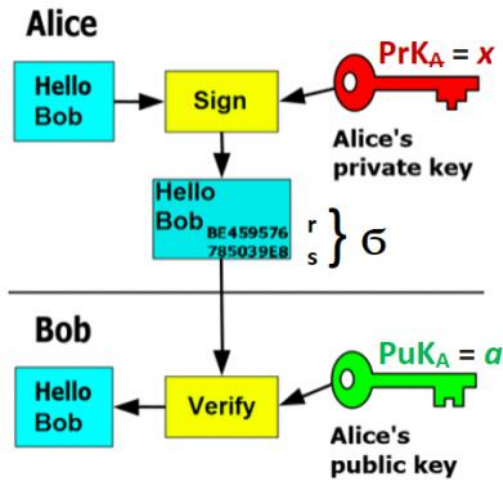




2. El-Gamal Signature scheme

Signature creation.

- Generate $i \leftarrow \text{randi}(Z_{p-1})$; verify if $\text{gcd}(i, p-1) = 1$.
 - Compute $i^{-1} \bmod (p-1)$; check if $i \cdot i^{-1} \bmod (p-1)$; is equal to 1?
 - Compute $r = g^i \bmod p$
 - Compute $s = (h - x \cdot r) \cdot i^{-1} \bmod (p-1) \rightarrow h = xr + is \bmod (p-1)$
- Signature $\sigma = (r, s)$

[illegible]

Signature verification.

Bob calculates $V1 = g^h \bmod p$ and $V2 = a^r r^s \bmod p$, and verifies if $V1 = V2$.

```
>> p=int64(268435019)      >> i=int64(randi(p-1))      >> r=mod_exp(g,i,p)      >> g_h=mod_exp(g,h,p)
p = 268435019              i = 201156232              r = 172536234          g_h = 241198023
>> g=2;                   >> gcd(i,p-1)          >> hmxr=mod(h-x*r,p-1)  >> V1=g_h
>> x=int64(randi(p-1))     ans = 2              hmxr = 20262153      V1 = 241198023
x = 65770603              >> i=int64(randi(p-1))  >> s=mod(hmxr*i_m1,p-1)
>> a=mod_exp(g,x,p)        i = 35395315         s = 44575091        >> a_r=mod_exp(a,r,p)
a = 232311991            >> gcd(i,p-1)          >> a_r = 49998673
>> M='Hello Bob...'       ans = 1              >> r_s=mod_exp(r,s,p)
M = Hello Bob...         >> i_m1=mulinv(i,p-1)  r_s = 111993804
>> h=hd28(M)              i_m1 = 192754179    >> V2=mod(a_r*r_s,p)
h = 150954921            >> mod(i*i_m1,p-1)    V2 = 241198023
ans = 1
```

ECDSA in Ethereum

Ethereum

GasLimit is evaluated by Wey.

1 Wey = 10^{-18} Eth --> 1 Eth = 10^{18} Wey.

Usually 1Gas is dozens of thousands of Wey's.

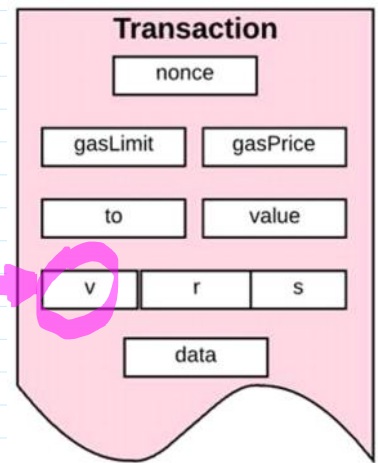
Parameters (**v**, **r**, **s**) is an Ethereum signature:

v - version;

r - first component of signature;

s - second component of signature.

Version



$Tx = 'nonce || gasLimit || gasPrice || to || value || v || r || s || data'$

$h = H(Tx) \rightarrow \text{Sign}(\text{PrK}, h) = \mathbf{s} = (r, s)$

ECDSA using **secp256k1** software:

1. Install Python 3.9.1.

2. Launch script Packages for joining a libraries.

3. Launch file ECC.

4. If window is escaping, then open hidden windows in icon near the Start icon.

 Packages	2021.12.05 18:23	Python File	1 KB
 ECC	2021.12.09 19:06	Python File	9 KB

Documents > 500 SOFTAS 2023 > Python 3.9.1 > 111.ECDSA 2023.09

☐	Name	Date modified	Type	Size
	Archyvas	2023-09-28 19:26	File folder	
	111.ECDSA.zip	2023-09-28 19:21	Compressed (zippe...	4 KB
	App_PrK.txt	2023-10-27 13:41	Text Document	1 KB
	App_PuK.txt	2023-10-27 13:41	Text Document	1 KB
	App_Signature.txt	2023-10-27 13:49	Text Document	1 KB
☒	ECC.py	2023-09-21 19:15	PY File	9 KB
	Instrukcija.txt	2021-12-15 14:29	Text Document	1 KB
	Packages.py	2021-12-05 18:23	PY File	1 KB

C:\Users\Eligijus\AppData\Local\Programs\Python\Launcher\py.exe

```
ECDS python app
Please input required command:
1 - Generate new ECC private and public keys
2 - Export private and public keys
3 - Export private key
4 - Export public key
5 - Load private key
6 - Load data file
7 - Sign loaded file
8 - Load public key
9 - Verify signature
10 - Export signature
11 - Load signature
12 - Draw secp256k1 graph in real numbers
13 - Draw secp256k1 graph over finite field
exit/e - Exit app
Input command: 1
ECC private key loaded/generated
ECC public key loaded/generated
ECDS python app
Please input required command:
```

PrK:

B64e8c0c98b2aaaf36caea44a452cb1c7a22b88859cf08a5b4f7d45b165ef82d

PuK:

x = 424da40e7cfe5cccc2204aa1785bced9aef8e2845843479dbc1f128c8ae08810

y = ed83908cb8352ce098598416be3957f45e8f24553dc7adec3cd523a7ffff2873