Project16 代码说明

在python中实现了ECDSA的签名以及伪造

可能需要安装hashlib及libnum库才能正常运行

运行成功截图如下：

在python中的伪造的签名并非中本聪 只是一个函数正确性的验证

中本聪签名的伪造在sagemath中进行了实现

代码如下：

```
sage: F = FiniteField
(0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F)
sage: F
Finite Field of size
115792089237316195423570985008687907853269984665640564039457584007908834671663
sage: C = EllipticCurve ([F (0), F (7)])
sage: C
Elliptic Curve defined by y^2 = x^3 + 7 over Finite Field of size
115792089237316195423570985008687907853269984665640564039457584007908834671663
sage: G =
C.lift_x(0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798)
sage: G
(55066263022277343669578718895168534326250603453777594175500187360389116729240 :
83121579216557378445487899878180864668798711284981320763518679672151497189239 :
1)
sage: N = FiniteField (C.order())
sage: N
Finite Field of size
115792089237316195423570985008687907852837564279074904382605163141518161494337
sage: P = P=-
C.lift_x(0x11db93e1dcdb8a016b49840f8c53bc1eb68a382e97b1482ecad7b148a6909a5c);P
```

```
(80772785790619904002497599521352676923512680340858642894518802994327118546844
34883007453703041530665294287464619720895014397849163628292634352974843079052 :
1)
sage: def forge(c, a=-1):
....:     a = N(a)
....:     R = c*G + int(a)*P
....:     s = N(int(R.xy()[0]))/a
....:     m = N(c)*N(int(R.xy()[0]))/a
....:     print("hash:",m)
....:     print("r:",int(R.xy()[0]))
....:     print("s:",s)
....:
sage: forge(5)
hash: 0xbf354f45c6c957e7ec37fc27c2317229de8b812fffc6738af97338bc9080f056
r: 0xa68ef0253ea48804d0c19a5e72c2e929cf3cca1559126914012ad9e489ab3756
s: 0x59710fdac15b77fb2f3e65a18d3d16d4eb7212d156363727bea784a8468b09eb
```

得到了如下伪造签名：
hash: 0xbf354f45c6c957e7ec37fc27c2317229de8b812fffc6738af97338bc9080f056
r: 0xa68ef0253ea48804d0c19a5e72c2e929cf3cca1559126914012ad9e489ab3756
s: 0x59710fdac15b77fb2f3e65a18d3d16d4eb7212d156363727bea784a8468b09eb