# 伪造中本聪签名

## 伪造签名的依据：

- $\sigma = (r, s)$ is valid signature of $m$ with secret key $d$
- If only the hash of the signed message is required
- Then anyone can forge signature $\sigma' = (r', s')$ for $d$
- (Anyone can pretend to be someone else)
- Ecdsa verification is to verify:
- $s^{-1}(eG + rP) = (x', y') = R', r' = x' mod\ n == r$ ?
- To forge, choose $u, v \in \mathbb{F}_n^*$
- Compute $R' = (x', y') = uG + vP$
- Choose $r' = x'\ mod\ n$, to pass verification, we need
- $s'^{-1}(e'G + r'P) = uG + vP$
  - $s'^{-1}e' = u\ mod\ n \rightarrow e' = r'uv^{-1} mod\ n$
  - $s'^{-1}r' = v\ mod\ n \rightarrow s' = r'v^{-1}\ mod\ n$
- $\sigma' = (r', s')$ is a valid signature of $e'$ with secret key d

伪造签名函数：

```
def Pretend(r, s,G, P):
    u = 3
    v = 3
    r_forge = Point_Add(Multi(u, G), Multi(v, P))[0]
    print(u,v)
    e_forge = (r_forge * u * multi_inverse(v, mod_value)) %mod_value
    s_forge = (r_forge * multi_inverse(v, mod_value)) % mod_value
    if(Verify( r_forge, s_forge,e_forge, G, P)):
        return (r_forge,s_forge)
```

通过verify函数进行验证。

运行结果：

```
C:\Users\wynne\AppData\Local\Programs\Python\Python39\python.exe F:/practise/Forged_Satoshi_Signature/forged.
公钥为 (7, 1)
3 3
伪造通过
伪造的签名 (7, 15)

进程已结束,退出代码0
```

# 方法二，运用sage能快速伪造

```
sage: F = GF (0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F
....: )  #生成一个给定阶数的全局唯一有限域
....: C = EllipticCurve ([F (0), F (7)])     #构造出椭圆曲线
....:
....:
....: G = C.lift_x(0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F
....: 81798)
....: N = GF (C.order())
....: P = P=-C.lift_x(0x11db93e1dcdb8a016b49840f8c53bc1eb68a382e97b1482ecad7b148
....: a6909a5c) # block 9 coinbase payout key.
....:
....: def forge(c, a=-1):  # Create a forged 'ECDSA'  (hashless) signature
....:   # set a to something other than -1 to be less obvious
....:   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 ('hash1 = %d'%m)
....:   print ('r1 = %d'%(int(R.xy()[0])))
....:   print ('s1 = %d'%s)
....:
....: forge(1)
hash1 = 738919280936671005416662050438984412675877586577947667827507163075362860
39192
r1 = 419001611436490948819047799647894665852498056212801375998544468339818754551
45
s1 = 73891928093667100541666205043898441267587758657794766782750716307536286039 1
92
```

原理是相同的，SAGE运算的更快。SAGE参考代码：[https://www.zuocoin.com/a/news/experience/2018/1119/6754.html](https://www.zuocoin.com/a/news/experience/2018/1119/6754.html)