ECDSA 签名伪造实验报告

实验目的

本实验旨在验证 ECDSA（椭圆曲线数字签名算法）的数学可伪造性，通过中本聪创世区块的真实公钥参数，演示攻击者如何在不持有私钥的情况下构造有效签名。实验依据 ECDSA 签名公式的数学变形特性实现伪造，并验证其有效性。

ECDSA 签名原理

ECDSA 签名流程基于椭圆曲线密码学：

签名生成：$s = k^{-1}(e + dr) \mod n$
签名验证：$R = s^{-1}e \cdot G + s^{-1}r \cdot P$，验证$R_x \equiv r \mod n$

k 为临时随机数（nonce）

e 为消息哈希

d 为私钥

P=d·G 为公钥

n 为椭圆曲线阶

伪造原理

通过选择随机数 *u* 和 *v*（满足 *gcd*(*u*,*n*)=1 且 *gcd*(*v*,*n*)=1），构造：

$$
\begin{aligned}
R &= u \cdot G + v \cdot P \\
r &= R_x \\
s &= r \cdot v^{-1} \mod n \\
e &= u \cdot r \cdot v^{-1} \mod n
\end{aligned}
$$

可证明(*r*,*s*)是消息 *e* 的有效签名，实现无私钥伪造。

实验环境

Python 3.10

- ecdsa 0.18.0 库

- 曲线参数：secp256k1 (比特币标准曲线)

中本聪公钥获取方法

公钥参数来源自比特币创世区块（高度 0）的 coinbase 交易：

区块链浏览器查询

通过区块链浏览器获取创世区块原始数据：

01000000010000000000000000000000000000000000000000000000000000000000000000000000

ffffffff4d04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368

616e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f75742066

6f722062616e6b73ffffffff0100f2052a01000000434104678afdb0fe5548271967f1a671

30b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384

df7ba0b8d578a4c702b6bf11d5fac00000000

公钥提取

从输出脚本 41...ac 中解析未压缩公钥：

X 坐标：0x678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb6

Y 坐标：0x49f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5f

实验过程

```
def forge_signature():
    # 生成互质随机数 u, v
    u = random.randint(1, N-1)
    v = random.randint(1, N-1)

    # 构造伪造点 R = u·G + v·P
    R_point = u * G + v * PK

    # 计算伪造签名参数
    r = R_point.x()
    v_inv = pow(v, -1, N)
    s = (r * v_inv) % N
    e = (u * r * v_inv) % N
```

```
    # 验证伪造签名
    return verify(e, r, s)
```

构建中本聪公钥点

```
PK = ecdsa.ellipticcurve.Point(curve.curve, PK_x, PK_y, N)
```

实现签名验证函数

```
def verify(e, r, s):
    w = pow(s, -1, N)
    u1 = (e * w) % N
    u2 = (r * w) % N
    R_point = u1 * G + u2 * PK
    return R_point.x() == r
```

实验结果