

***Project10: report on the application of this deduce technique in Ethereum with ECDSA**

说明：

一、ECDSA 概述

ECDSA 是 Elliptic Curve Digital Signature Algorithm（椭圆曲线数字签名算的缩写。它是一种基于椭圆曲线密码学的数字签名算法。ECDSA 结合了椭圆曲线上的离散对数问题和哈希算法的安全性特性，提供了一种高效、安全的数字签名方案。

ECDSA 被广泛应用于各种加密和认证场景，包括数字证书、电子支付系统、区块链技术等。它的安全性基于椭圆曲线离散对数问题的困难性，即在给定椭圆曲线上的基点和一个点的情况下，计算出这个点的离散对数是困难的。这使得 ECDSA 具有相对较短的密钥长度和较高的计算效率。

ECDSA 的基本原理是使用私钥对消息进行签名，生成数字签名，然后使用对应的公钥对签名进行验证，确保签名的完整性和真实性。ECDSA 通过椭圆曲线上的点运算实现签名和验证操作，同时结合了哈希函数以增强安全性。

二、ECC 的实现

签名过程：

1. 选择一条椭圆曲线 $E_p(a, b)$ 和基点 G 。
2. 选择私钥 k ($k < n$)，其中 n 是基点 G 的阶数，通过计算公钥 $K = kG$ 。
3. 生成随机数 r ($r < n$)，计算点 $R = rG$ 。
4. 将原数据和点 R 的坐标值 x, y 作为参数，计算哈希值 $Hash = Hash(原数据, x, y)$ （通常使用哈希函数）。
5. 计算 $s \equiv r - Hash * k \pmod{n}$ 。
6. 如果 r 或 s 中有一个为 0，则重新从步骤 3 开始执行。

验证过程：

1. 接收方在收到消息 m 和签名值 (r, s) 后进行以下运算。
2. 计算 $sG + H(m)P = (x_1, y_1)$ ，其中 $H(m)$ 是对消息 m 进行哈希计算的结果。
3. 验证等式： $r_1 \equiv x_1 \pmod{n}$ ，其中 r_1 为计算得到的临时值。
4. 如果等式成立，则接受签名；否则，签名无效。

三、ECDSA 在以太坊中的应用：

ECDSA 作为以太坊中重要的加密算法之一，广泛用于验证交易、地址生成以及智能合约的部署。通过深入了解 ECDSA 在以太坊中的应用，我们可以更好地理解以太坊的安全机制和保障用户资产的安全。

(1) 交易验证和身份认证：以太坊中的交易是通过 ECDSA 签名进行验证的。每笔交易都需要发送者使用其私钥对交易内容进行签名，然后接收者可以使用发

送者的公钥验证签名的有效性。这确保了交易的身份认证和完整性，防止了篡改和伪造交易。

(2) 地址生成与管理：以太坊地址是由公钥生成的，并通过哈希函数转化为可识别的形式。这意味着用户可以使用其私钥生成一个唯一的地址，用于接收以太币或其他代币。ECDSA 的安全性保证了地址生成过程的安全性，同时也确保了地址和私钥之间的一一对应关系。

(3) 智能合约的安全性：智能合约是以太坊上的自动化合约，也可以使用 ECDSA 进行验证和授权。当智能合约需要执行涉及安全权限的操作时，可以使用 ECDSA 签名来确保只有经过授权的用户才能执行这些操作，从而增强智能合约的安全性。

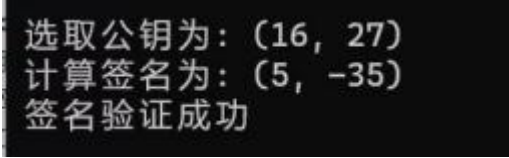
(4) 链上身份认证：ECDSA 还可以用于在以太坊上实现分布式的身份认证系统。用户可以使用其私钥对一些身份信息进行签名，从而证明自己的身份。这在去中心化身份验证和数字身份领域具有潜在的应用。

(5) 多重签名钱包：以太坊支持多重签名钱包，这意味着需要多个私钥的持有者共同签署一笔交易才能使其有效。这可以通过 ECDSA 实现，确保了资金的更高安全性，尤其在需要多方参与决策的情况下。

(6) 隐私保护：虽然 ECDSA 本身并不是专门用于隐私保护的技术，但它可以与其他隐私技术结合使用，如环签名、零知识证明等，从而在以太坊交易中增加一定程度的隐私保护。

实现方式：c

结果：在自己电脑上 CPU: 11 代 i7



```
选取公钥为: (16, 27)
计算签名为: (5, -35)
签名验证成功
```

分工：自己独立完成