在以太坊中，ECDSA 算法可以用于从签名中推导出公钥。这个功能在以太坊中的身份验证和签名验证过程中非常重要。当一个交易被广播到以太坊网络时，它必须被验证，以确保它是由合法的发送者发送的。ECDSA 算法可以用于验证交易的签名，并从签名中推导出发送者的公钥，从而验证发送者的身份。

1 推导技术概述

推导技术是一种利用已知信息和算法进行推理的方法，可用于从相关数据中推断出隐藏的信息或密钥。

2 ECDSA 算法描述

在以太坊（Ethereum）网络中，ECDSA（椭圆曲线数字签名算法）是一种常用的签名算法，用于验证交易和身份认证。

具体算法如下：

签名过程

(1) 选择一条椭圆曲线 Ep(a,b)，和基点 G；

(2) 选择私有密钥 k（k<n，n 为 G 的阶），利用基点 G 计算公开密钥 K=kG；

(3) 产生一个随机整数 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 SHA1 做为 hash，即 Hash=SHA1(原数据,x,y)；

(5) 计算 sffr - Hash * k (mod n)

(6) r 和 s 做为签名值，如果 r 和 s 其中一个为 0，重新从第 3 步开始执行。

验证过程

接受方在收到消息 (m) 和签名值 (r,s) 后，进行以下运算:

(1) 计算：sG+H(m)P=(x1,y1), r1ff x1 mod p。

(2) 验证等式：r1 ff r mod p。

(3) 如果等式成立，接受签名，否则签名无效。

(4) 从签名中恢复公钥

这个算法的思路可以简要概括如下：

(1) 使用签名中的 r 值，通过取模运算得到 x 坐标：x = r

(2) 根据椭圆曲线方程 y^2 = x^3 + ax + b，计算 y 坐标。

(3) 将消息 m 进行哈希运算，得到哈希值 e。

(4) 构造两个椭圆曲线点 P1 和 P2，分别为 (x, y) 和 (x, p-y)。利用签名中的 s 值和对应的点 P1 或 P2，计算临时私钥 sk1 或 sk2。

(5) 计算临时点 tmp，为消息哈希值 e 乘以椭圆曲线基点 G。计算 tmp_i，为临时点 tmp 的 y 轴取负。

(6) 计算临时点 tmp_1，为临时私钥 sk1 或 sk2 与 tmp_i 的加法。

(7) 使用临时私钥和 gcd(r, n) 乘法运算，计算出推导的公钥 pk1 或 pk2。