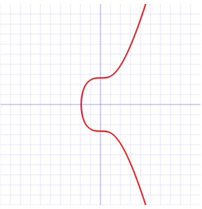
Ethereum Blockchain CryptoGraphy

Ethereum blockchain uses information as a hash, which in this process uses the SHA-3(Keccak-256) algorithm, the private and public keys to create a digital signature in this blockchain using the ECDSA algorithm.

The ECDSA algorithm is used to generate private and public keys as well as to create digital signatures and verify digital signatures.
this Blockchain use **secp256k1** Elliptic curve.



How an Ethereum address is generated :

A random private key of 64 (hex) characters (256 bits / 32 bytes) is generated first. For example:

0xf4a2b939592564feb35ab10a8e04f6f2fe0943579fb3c9c33505298978b74893

A 128 (hex) character (64 bytes) public key is then derived from the generated private key using Elliptic Curve Digital Signature Algorithm (ECDSA). For example:

0x04345f1a86ebf24a6dbeff80f6a2a574d46efaa3ad3988de94aa68b695f09db9ddca37439f99548da0a1fe4acf4721a945a599a5d789c18a06b20349e803fdbbe

The Keccak-256 hash function is then applied to (128 characters / 64 bytes) the public key to obtain a 64 character (32 bytes) hash string. The last 40 characters / 20 bytes of this string prefixed with 0x become the final Ethereum address. For example:

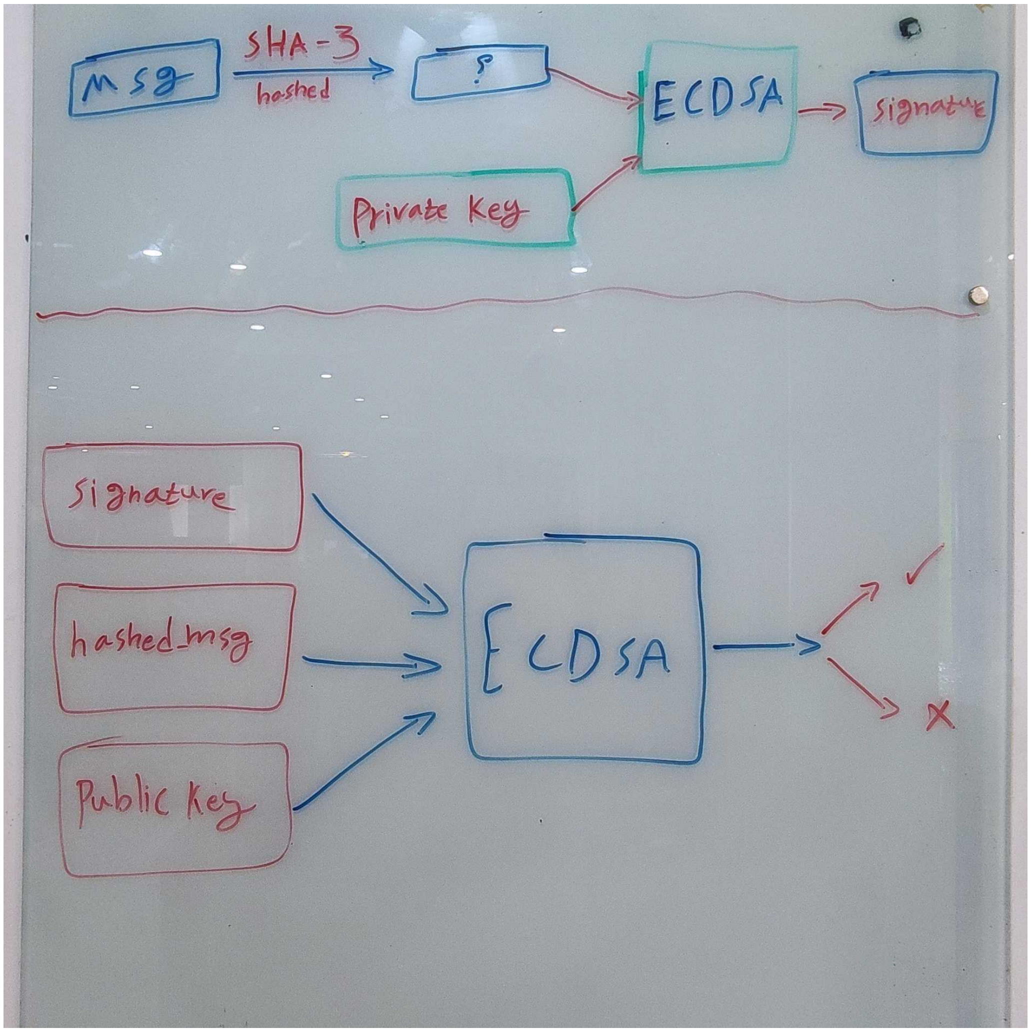
0xd5e099c71b797516c10ed0f0d895f429c2781142

Note: 0x in coding indicates that the number/string is written in hex.

How an Ethereum Blockchain make a Signature :

This process is performed using the ECDSA algorithm, first the message is hashed using the SHA-3 algorithm, then it is given to this algorithm as an input with a private key, the algorithm uses mathematical operations etc. a digital signature is generated And displays as output, then to confirm the accuracy of the message and digital signature, it is enough to give the hashed message of the message along with the signature and the public key of the message signer to the algorithm to determine whether the signature and message are true or false.

Refrence



Solidity and NodeJs

Link to Document