

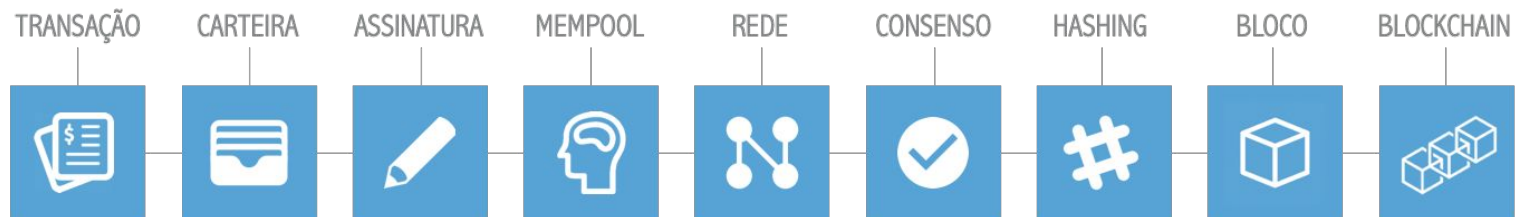
IMD0913

# ARQUITETURA DE UM BLOCKCHAIN

## ASSINATURAS DIGITAIS

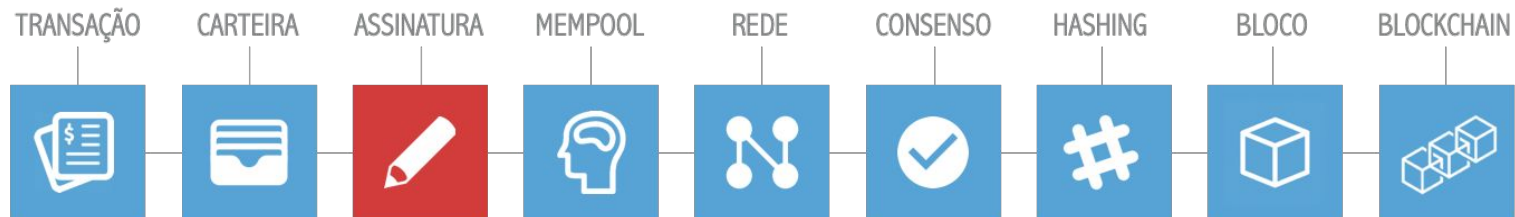
## ARQUITETURA DE UM **BLOCKCHAIN**

---



## ARQUITETURA DE UM **BLOCKCHAIN**

---



# Assinando transações

Como confiar em uma transação? **Assinaturas digitais!**

Lembrando que Bitcoin usa o modelo UTXO

Transações mapeiam entradas para saídas

Transações contém assinaturas dos proprietários dos fundos

Gastar bitcoins é **resgatar saídas de transações anteriores**

## Assinatura digital

Estabelece a **prova de propriedade** para cada transação do blockchain.

# Assinaturas digitais

Uma **assinatura digital** é algo que você pode usar para mostrar que você conhece a chave privada conectada a uma chave pública, sem precisar revelar a chave privada real.

Portanto, se alguém perguntar se você tem a chave privada de uma chave pública (ou endereço) específica, você pode fornecer uma assinatura digital para **provar** isso.

# Como funcionam assinaturas digitais?

O remetente gera uma par de chaves privada e pública (endereço)

O remetente envia a mensagem com a assinatura, envia sua chave pública, a assinatura e a mensagem para a rede

Nós que receberem checam através de um algoritmo de verificação que a mensagem foi assinada pelo remetente, que só pode ser feita pelo detentor da chave privada da chave pública que foi enviada

No contexto de transações:

Assinar uma transação auxilia na prova de propriedade (***proof-of-ownership***) e na não adulteração destas transações

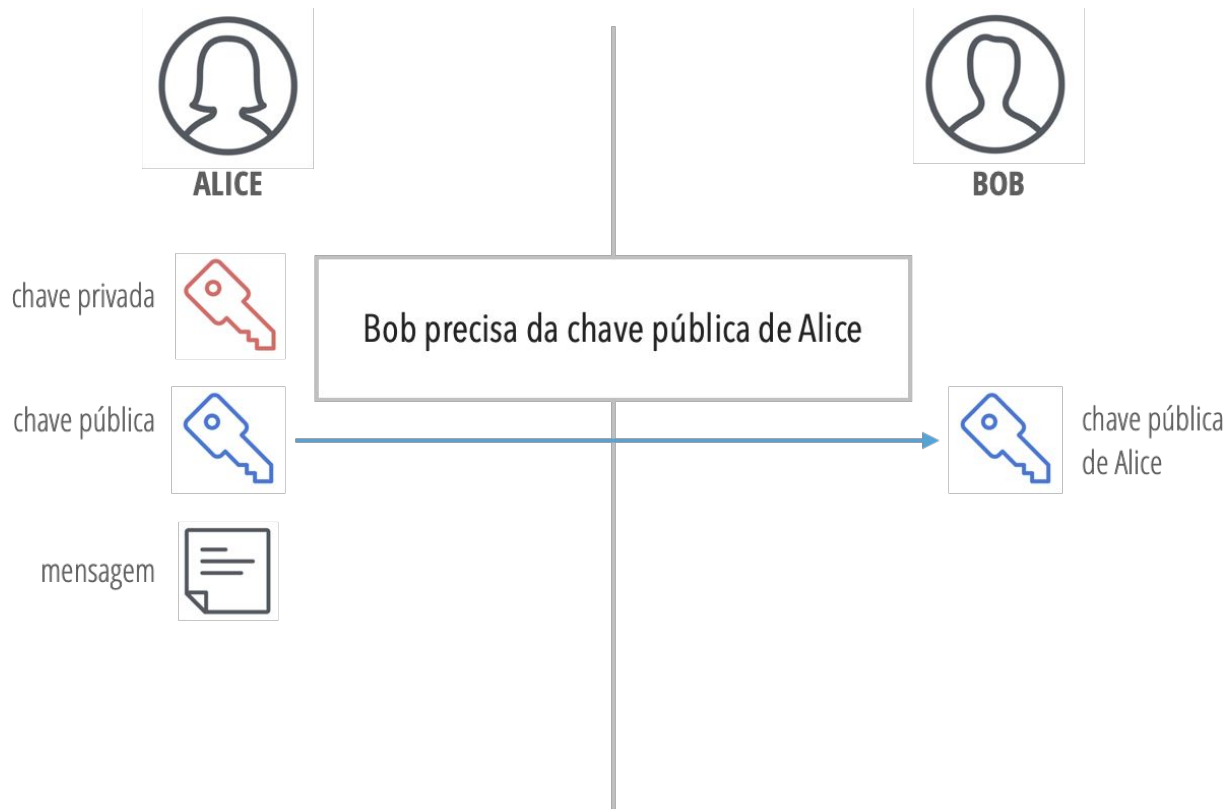
Um UTXO só pode ser usado como entrada de uma transação (resgatar bitcoins) caso a prova de propriedade seja apresentada corretamente

# Como funcionam assinaturas digitais?

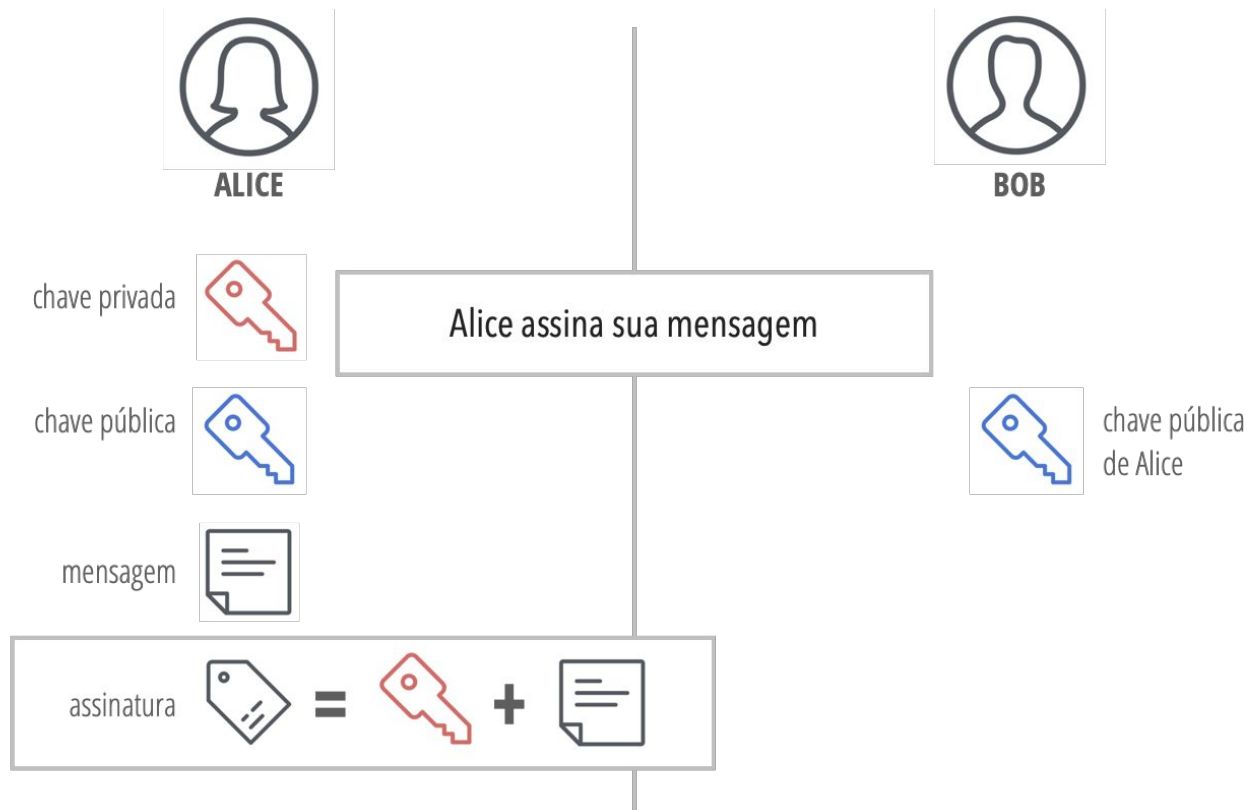




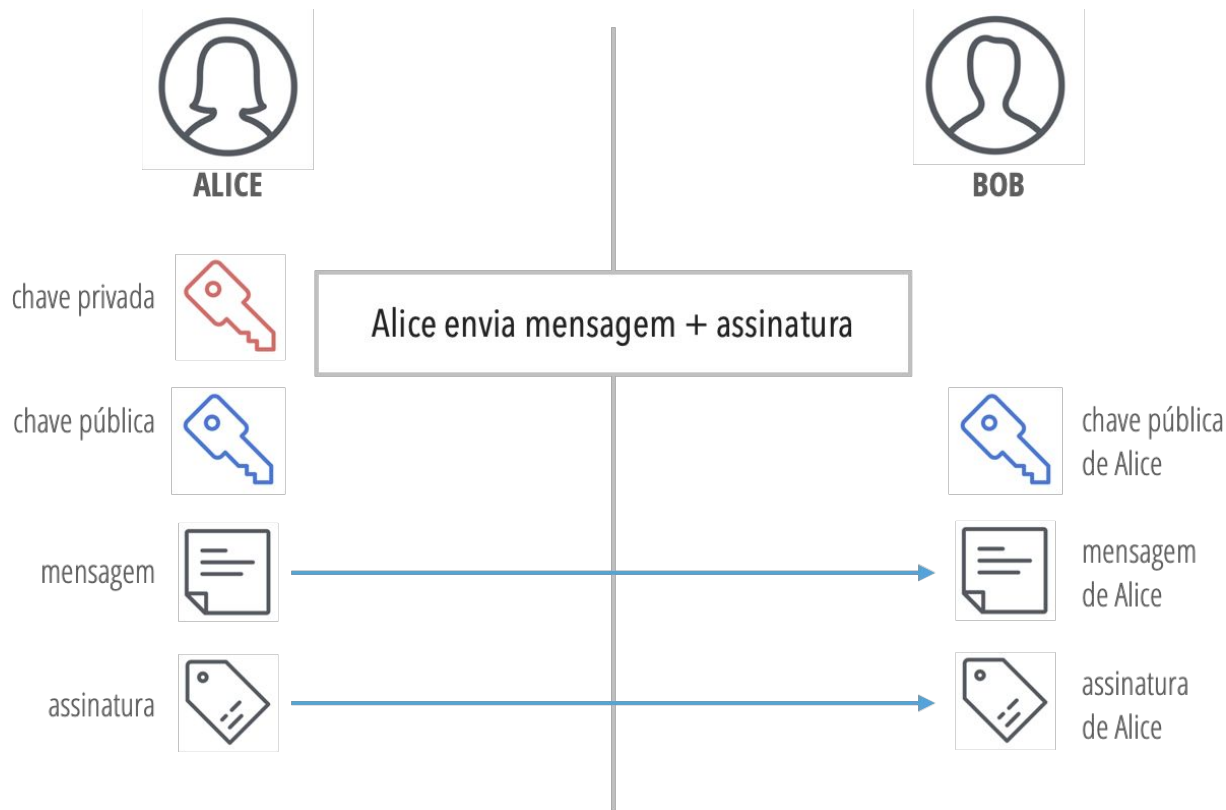
# Como funcionam assinaturas digitais?



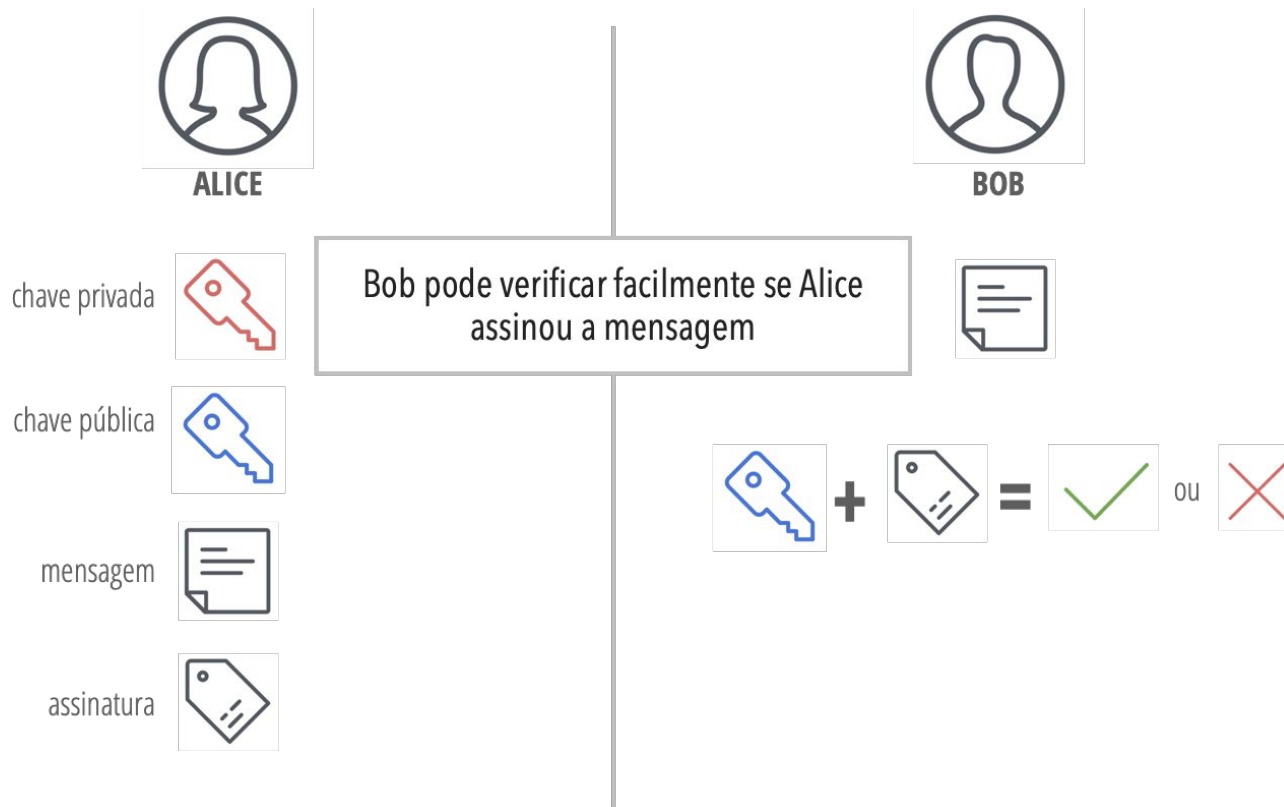
# Como funcionam assinaturas digitais?



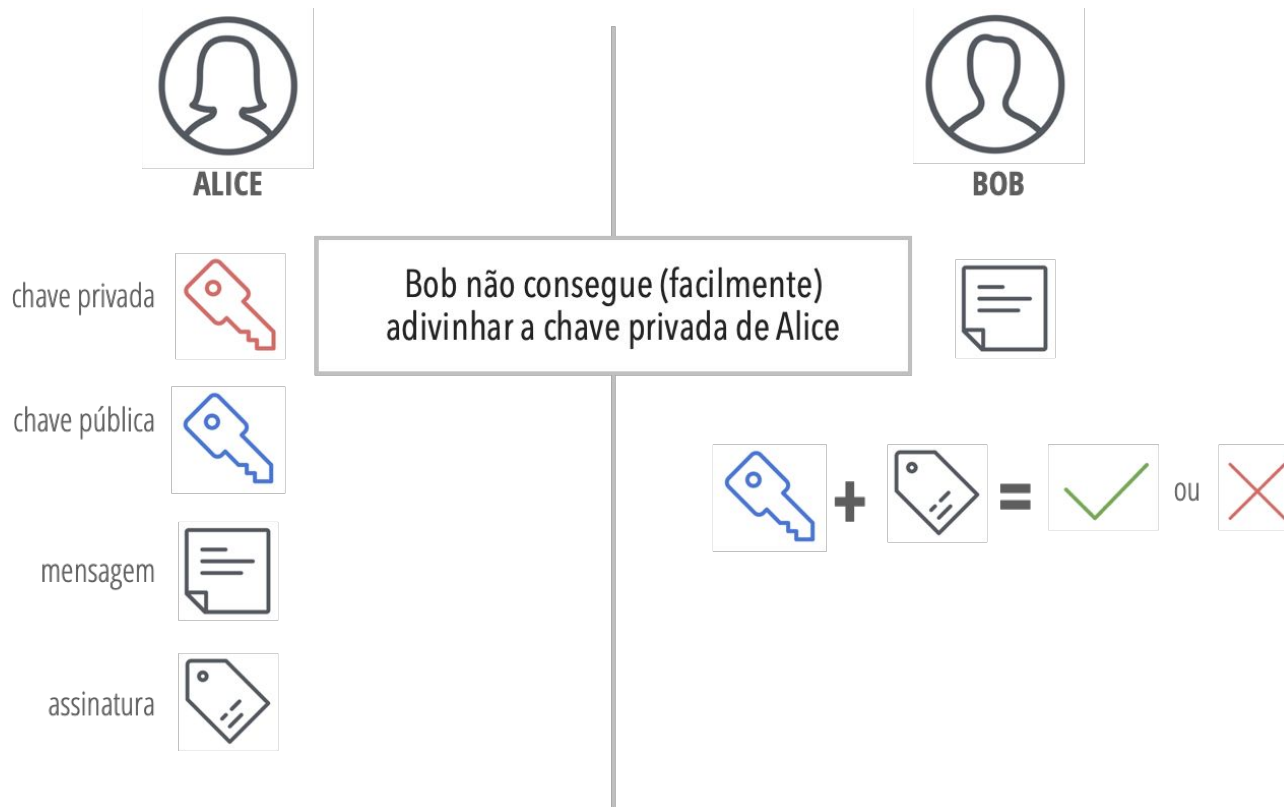
# Como funcionam assinaturas digitais?



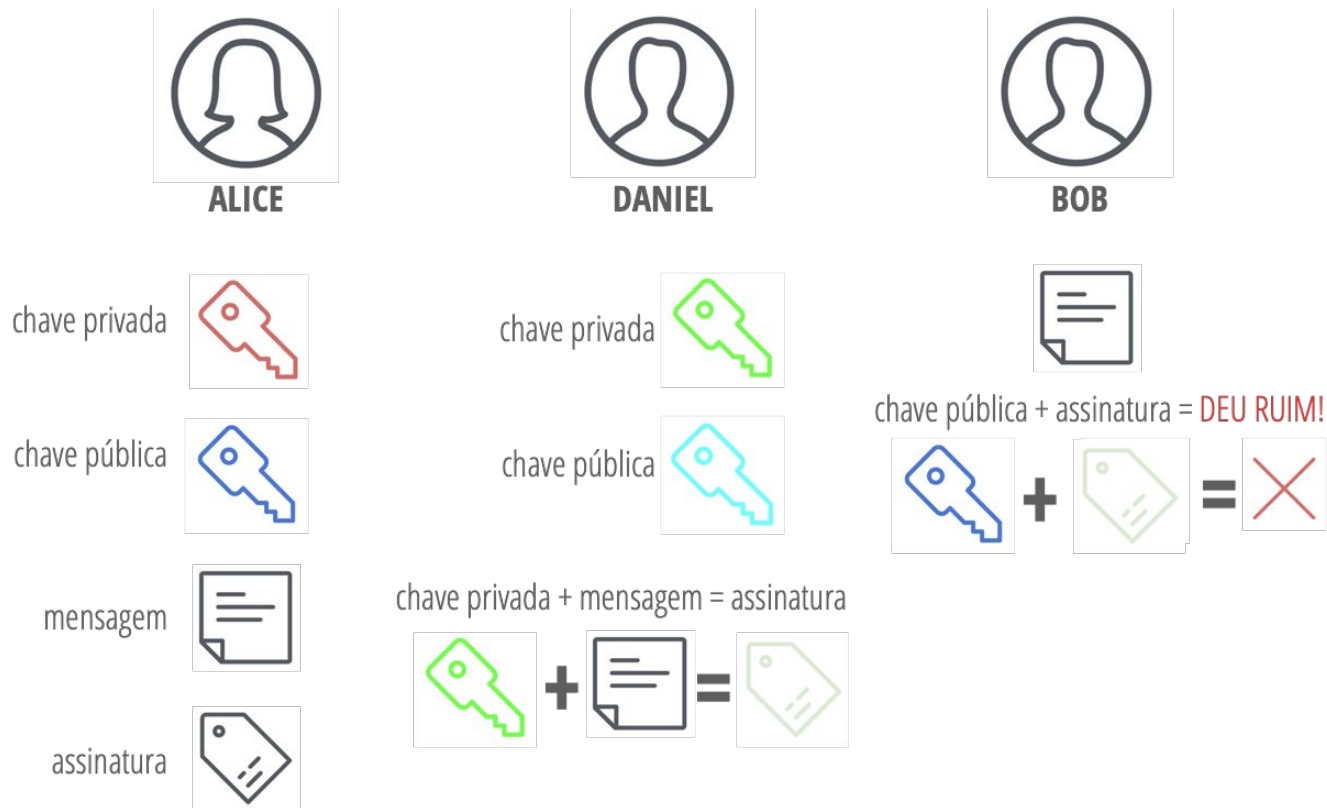
# Como funcionam assinaturas digitais?



# Como funcionam assinaturas digitais?



# Como funcionam assinaturas digitais?



# Esquema de assinatura digital

Destinatários de posse do par [mensagem, assinatura] podem verificar:

**Autenticidade:** remetente original (detentor da chave privada) autorizou essa mensagem/transação

**Não-repúdio:** remetente original (detentor da chave privada) não pode negar que autorizou essa mensagem/transação

**Integridade:** Mensagem não pode ter sido modificada após o seu envio

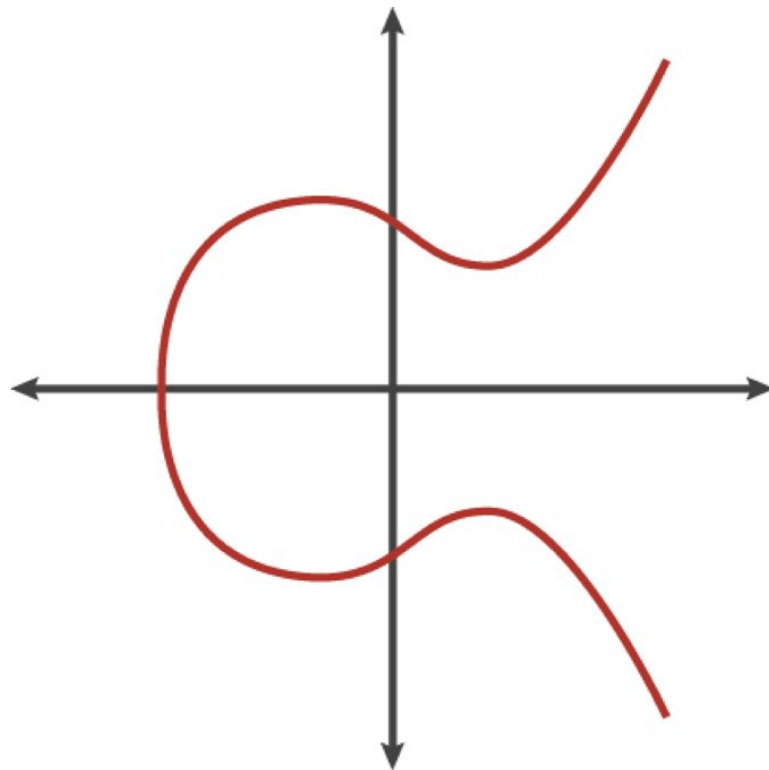
# Criptografia de curva elíptica

Bitcoin usa o algoritmo **ECDSA** para produzir o par de chaves

*Elliptic Curve Digital Signature Algorithm*

*secp256k1*

Chave pública é derivada da chave privada





# Assinaturas em Python

## /04-sign-and-verify

1. Implementar o método **sign** para retornar a assinatura de uma mensagem passada como argumento
2. Implementar o método **verifySignature** para retornar True se assinatura é válida para a mensagem e o endereço passado como parâmetro

Obs: Usar endereços Bitcoin! Gere um endereço válido em:

<https://www.bitaddress.org>

Verifique se está correto em:

<https://www.bitcoin.com/tools/verify-message/>

# Assinaturas em Python

```
1  @staticmethod
2  def sign(wifCompressedPrivKey, message):
3      # Retorna a assinatura digital da mensagem e a respectiva chave privada WIF-compressed.
4      return bitcoinlib.ecdsa_sign(message, wifCompressedPrivKey)
5
6  @staticmethod
7  def verifySignature(address, signature, message):
8      # Verifica se a assinatura é correspondente a mensagem e o endereço BTC.
9      # Você pode verificar aqui também: https://tools.bitcoin.com/verify-message/
10     return bitcoinlib.ecdsa_verify(message, signature, address)
```

## TESTE:

Mensagem: Bora assinar essa mensagem?

Endereço BTC: 19sXoSbfCQD9K66f5hwP5vLwsaRyKLPgXF

Assinatura gerada: ILh7tecVUPuvjm+N0mZPd/eeFujagpG/Ztc34dXmeTccWDzMGb1AVu5HjgIBAHj0aJB31phf7EjpS5NnqRko5Ks=

Assinatura válida para mensagem e endereço indicado? True

<https://www.verifybitcoinmessage.com/>

