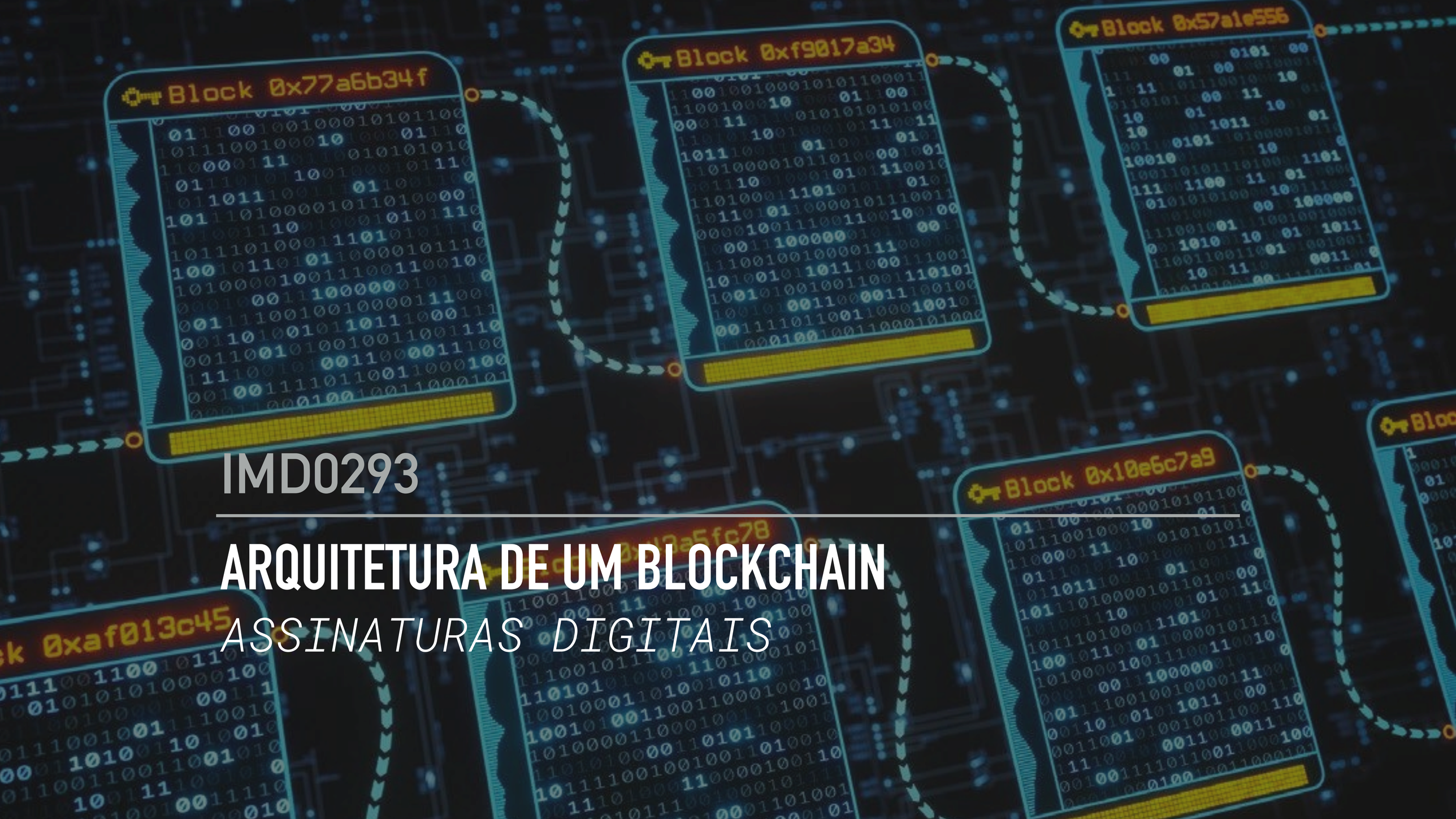
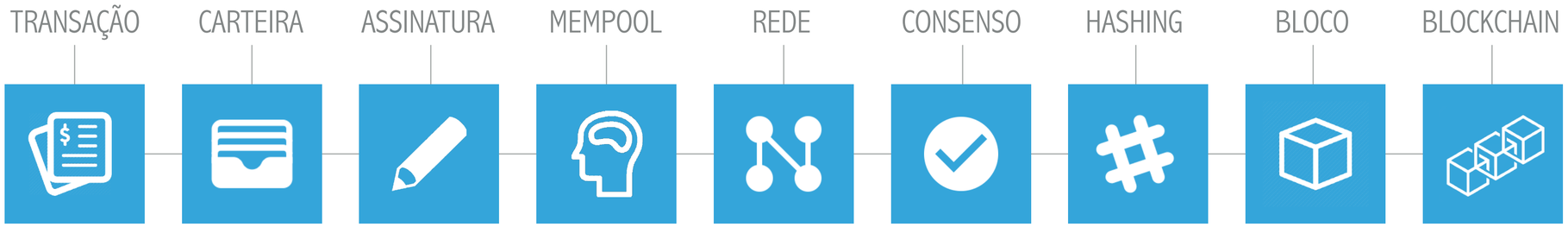


IMD0293

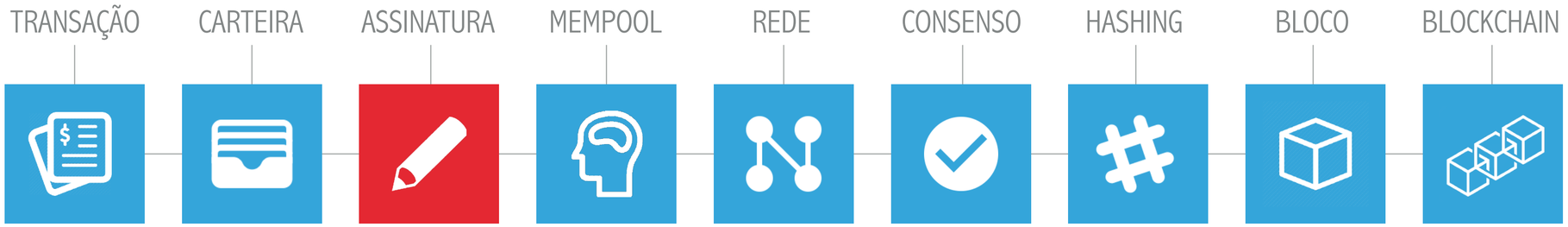
ARQUITETURA DE UM BLOCKCHAIN

ASSINATURAS DIGITAIS

ARQUITETURA DE UM **BLOCKCHAIN**



ARQUITETURA DE UM **BLOCKCHAIN**



ASSINANDO UMA TRANSAÇÃO

- 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

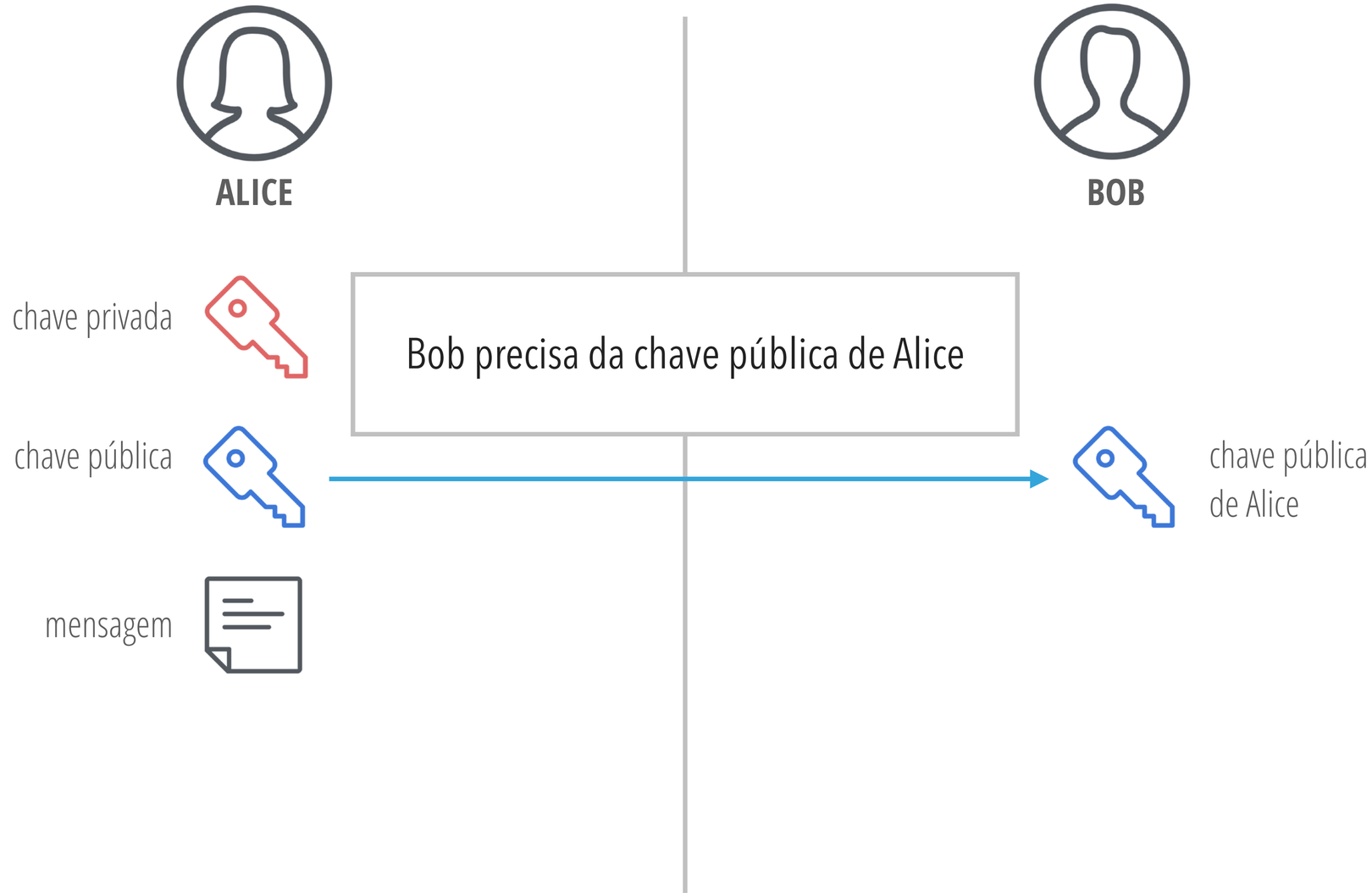
Estabelece a prova de propriedade para cada transação do blockchain.

COMO FUNCIONA?

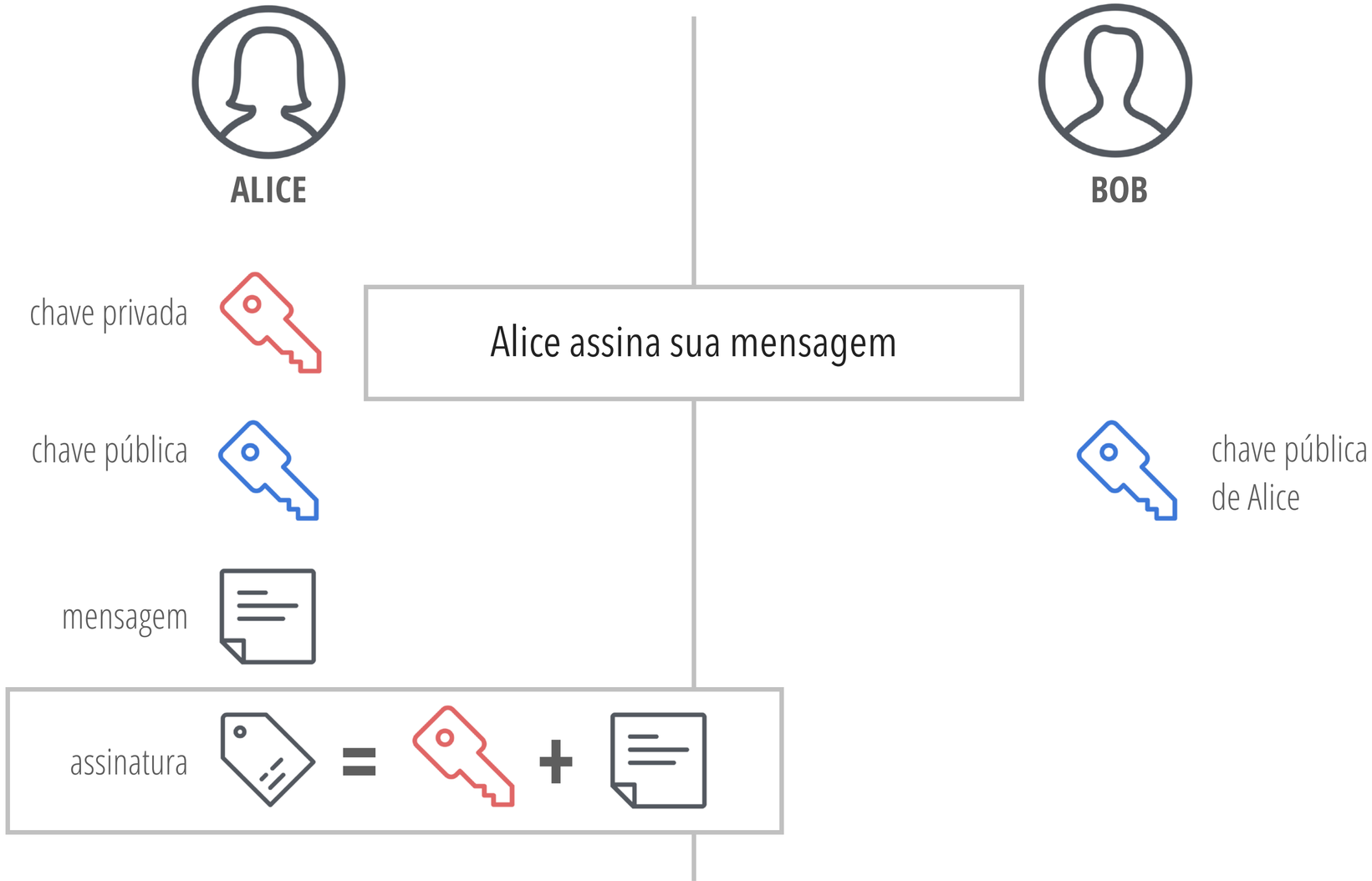
- O remetente gera uma par de chaves privada e pública (endereço)
- O remetente envia a mensagem com a assinatura, envia sua chave pública, a assinatura e a mensagem para a rede
- Nós que receberem checam através de um algoritmo de verificação que a mensagem foi assinada pelo remetente, que só pode ser 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 (resgar bitcoins) caso seja a prova de propriedade seja apresentada corretamente

ESQUEMA DE ASSINATURA DIGITAL (DSS)

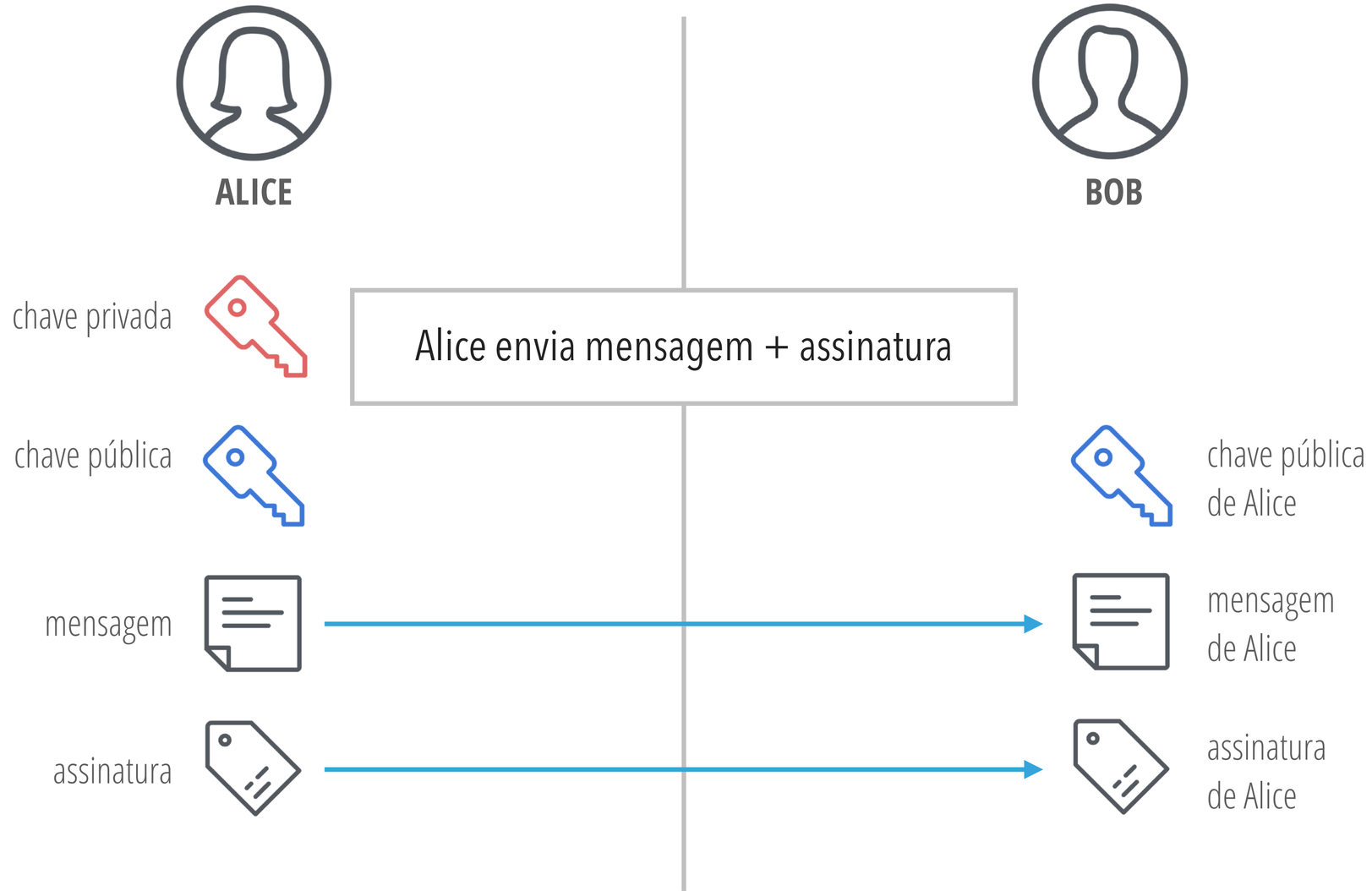
ESQUEMA DE ASSINATURA DIGITAL (DSS)



ESQUEMA DE ASSINATURA DIGITAL (DSS)



ESQUEMA DE ASSINATURA DIGITAL (DSS)



ESQUEMA DE ASSINATURA DIGITAL (DSS)

ALICE

chave privada

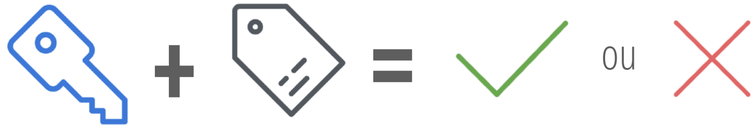
chave pública

mensagem

assinatura

Bob pode verificar facilmente se Alice assinou a mensagem

BOB



ESQUEMA DE ASSINATURA DIGITAL (DSS)

ALICE

BOB

chave privada 

chave pública 

mensagem 

assinatura 

Bob não consegue (facilmente) adivinhar a chave privada de Alice

ESQUEMA DE ASSINATURA DIGITAL (DSS)

ALICE

chave privada

chave pública

mensagem

assinatura

DANIEL

chave privada

chave pública

chave privada + mensagem = assinatura

BOB

chave pública + assinatura = DEU RUIM!

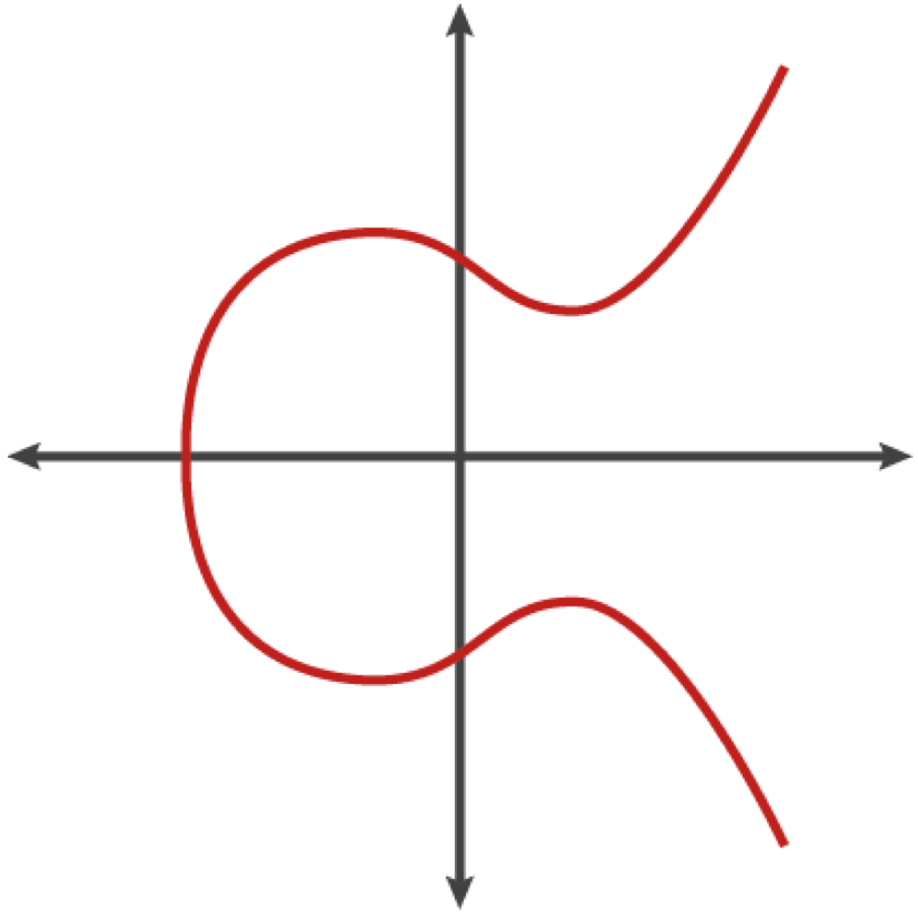
ESQUEMA DE ASSINATURA DIGITAL (DSS)

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 **ECSDA** 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://tools.bitcoin.com/verify-message/

ASSINATURAS EM PYTHON

```python
@staticmethod
def sign(wifCompressedPrivKey, message):
    # Retorna a assinatura digital da mensagem e a respectiva chave privada WIF-compressed.
    return bitcoinlib.ecdsa_sign(message, wifCompressedPrivKey)

@staticmethod
def verifySignature(address, signature, message):
    # Verifica se a assinatura é correspondente a mensagem e o endereço BTC.
    # Você pode verificar aqui também: https://tools.bitcoin.com/verify-message/
    return bitcoinlib.ecdsa_verify(message, signature, address)
```

TESTE:
Mensagem: Bora assinar essa mensagem?
Endereço BTC: 19sXoSbfcQD9K66f5hwP5vLwsaRyKLPgXF
Assinatura gerada: ILh7tecvUPuvjm+NOmZPd/eeFujagpG/Ztc34dXmeTccWDzMGb1AVu5HjgIBAHj0aJB31phf7EjpS5NnqRKo5Ks=
Assinatura válida para mensagem e endereço indicado? True

https://tools.bitcoin.com/verify-message/