

A diagram illustrating the architecture of a blockchain. It features several blocks, each represented as a rounded rectangle with a blue border and a yellow base. Each block contains a header with a key icon and a hexadecimal hash (e.g., 0x77a6b34f, 0xf9017a34, 0x37a1e556, 0x10e6c7a9, 0xaf013c45, 0x13a5fc78). The main body of each block is filled with binary code (0s and 1s). Dashed blue arrows connect the blocks in a sequence, indicating the flow of data and the linking of blocks. The background is a dark blue grid with binary code and circuit-like patterns.

IMD0293

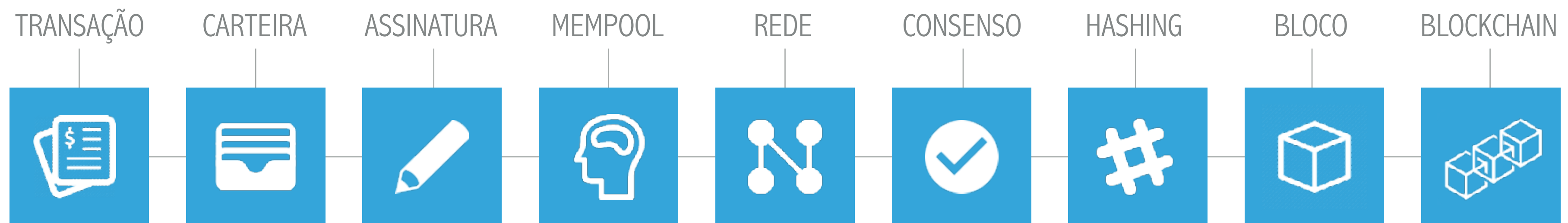
# ARQUITETURA DE UM BLOCKCHAIN

HASHING



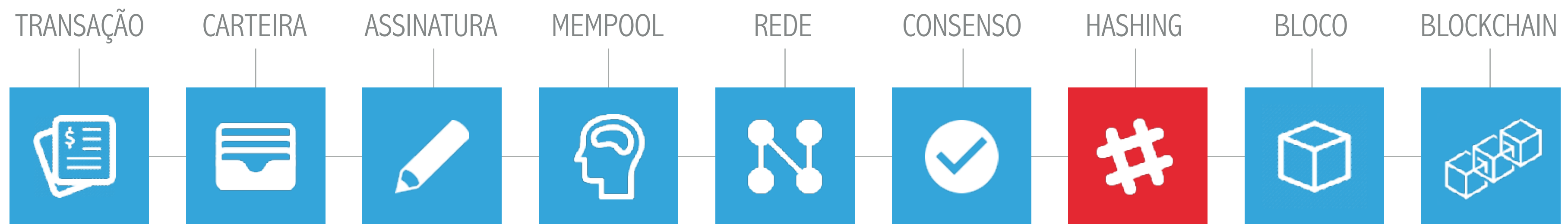
## ARQUITETURA DE UM **BLOCKCHAIN**

---



## ARQUITETURA DE UM **BLOCKCHAIN**

---



# FUNÇÕES CRIPTOGRÁFICAS DE HASH

Como garantir confiança em um ambiente *trustless*?

Funções criptográficas de *hash*

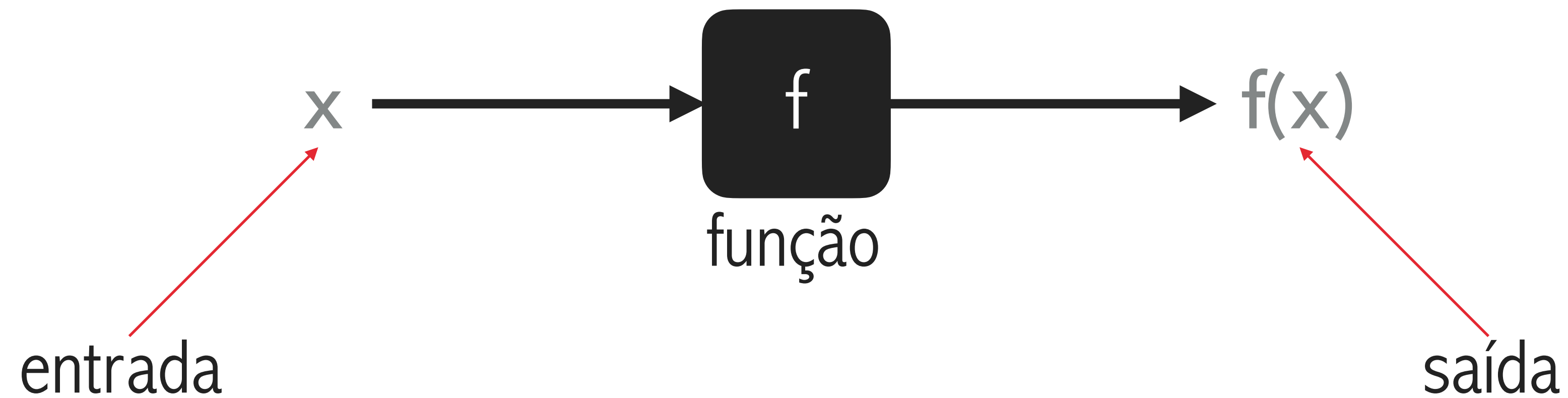
## Hash

Um *fingerprint* digital para informações.

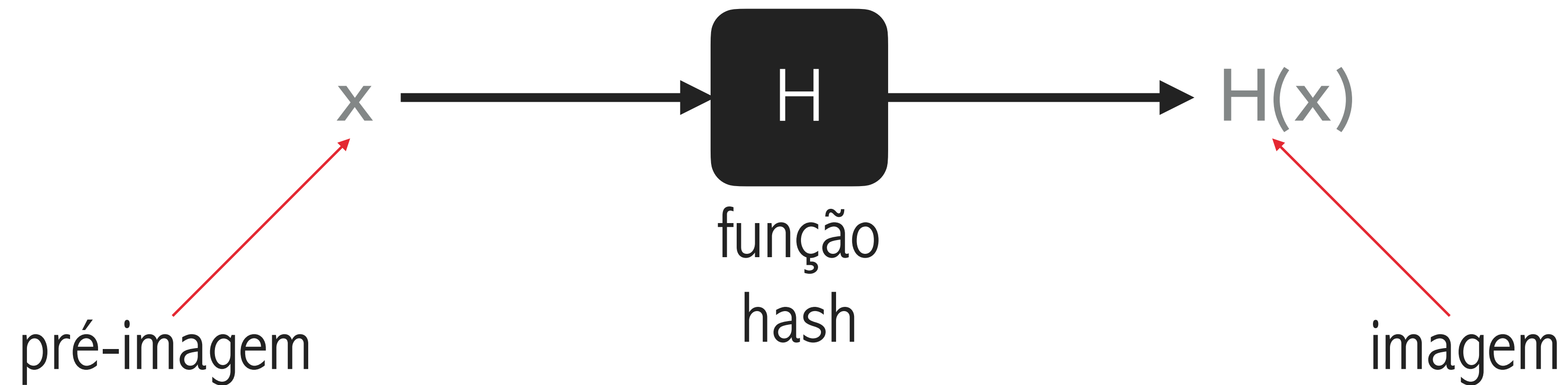
## Função Hash

Mapeiam um conjunto de dados em um valor de *hash* único.

# FUNÇÕES CRIPTOGRÁFICAS DE HASH

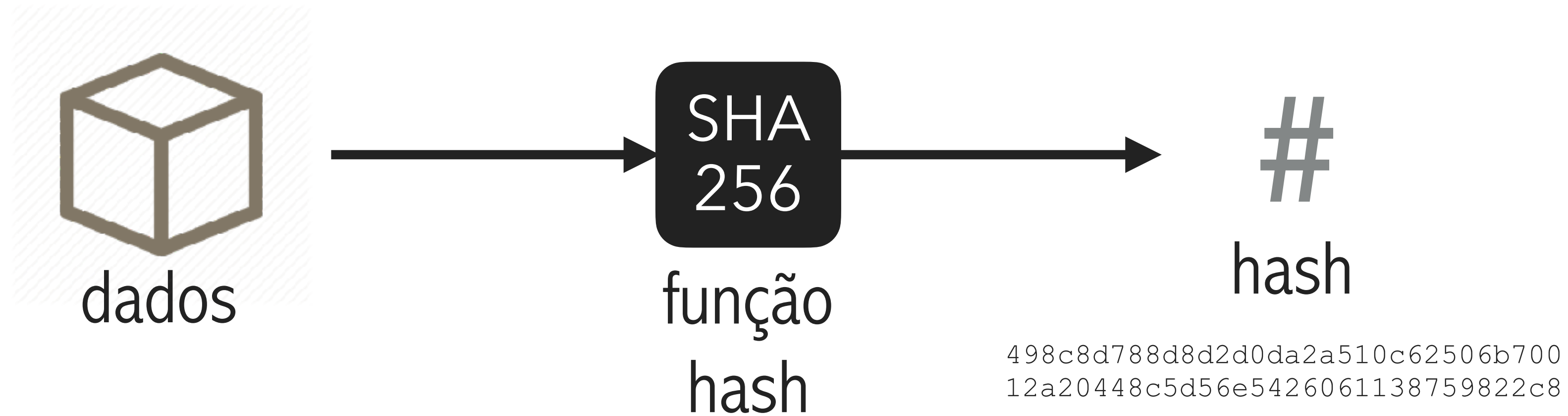


# FUNÇÕES CRIPTOGRÁFICAS DE HASH

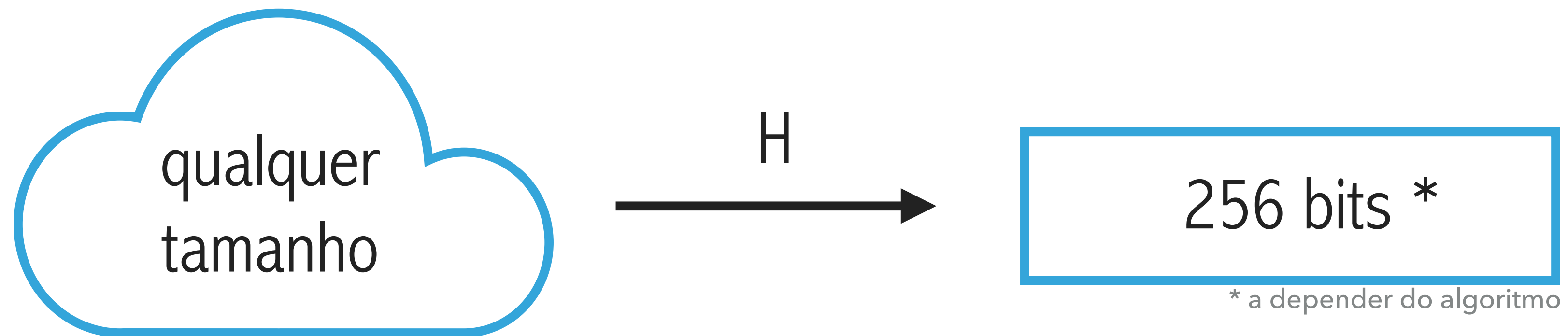




# FUNÇÕES CRIPTOGRÁFICAS DE HASH

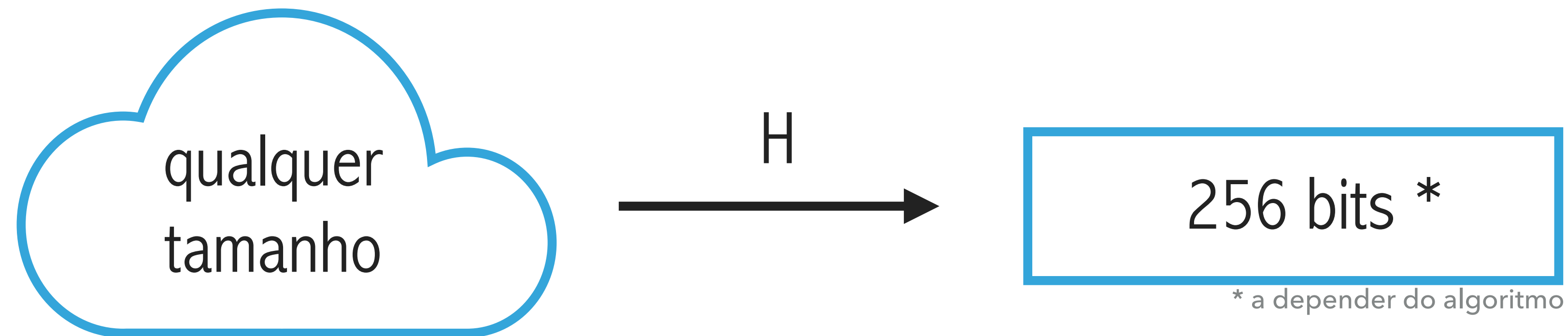


# FUNÇÕES CRIPTOGRÁFICAS DE HASH





# FUNÇÕES CRIPTOGRÁFICAS DE HASH



SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('x') = 2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881

SHA256('universidade federal do rio grande do norte') = 401016725141b697a7154f48fbae01d488dcff8bf871e15ff80e1f48bbaa1961

...

SHA256(1GB de dados) = 341d78ff8e1c6b38cbe90cbf44ceca4d6cdebef21036587860abfcd92570a4e5

# FUNÇÕES CRIPTOGRÁFICAS DE HASH

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f

...

SHA256('satoshi') = da2876b3eb31edb4436fa4650673fc6f01f90de2f1793c4ec332b2387b09726f



## FUNÇÕES HASH

- ▶ Uma função *hash* tem três propriedades especiais:
- ▶ Resistência a pré-imagem
- ▶ Resistência a segunda pré-imagem
- ▶ Resistência a colisão

f15a19ceb84f422917a79243c124336747  
53b4dbcf2b965765163fa409f9845066a2  
498c8d788d8d2d0da2a510c62506b70012  
88bf13017cbb390c85f4295a326d8f0def  
bb361264ad2d6ebafea5dc4ae714bfe309  
e025e6d946bf1dc5364c593f921c595391  
74af15487e45b67fdffe3921e4a93bffa6  
39c4b50e02424c5112cd67fa98ff7d0101  
7120f5dd3d5c8c69f53f3178778b36ed2a  
edcdf591c55201148c100bced1812e83ab  
1b631c8e942202de0c681232aff3e54c33  
3f80ad796a2644d54ef054620fb2b1a0ed  
7d69c86fde1cc84cf135ab2bf53e885ec7  
c808374e4954ae3ba59f4af9eef1df31f2  
8ed1ef6d1f98b410a86eaa4465764bb8b2  
4569e8c8cfd14eb2325de3255904a2cb0a  
4cc7d658552478874103d01b7298d6a4bf  
b3cf352b09933ec1ea05d946079688b425  
9782656e9bc25e2b2d32e6c81b3d0e90bd  
3fc700496fde534f56f550cb4a1cc20489

# FUNÇÕES HASH

Resistência a pré-imagem

Dado  $H(x)$ , é computacionalmente difícil/  
custoso determinar  $x$

Analogia a impressão digital:

De quem é essa impressão digital?





# FUNÇÕES HASH

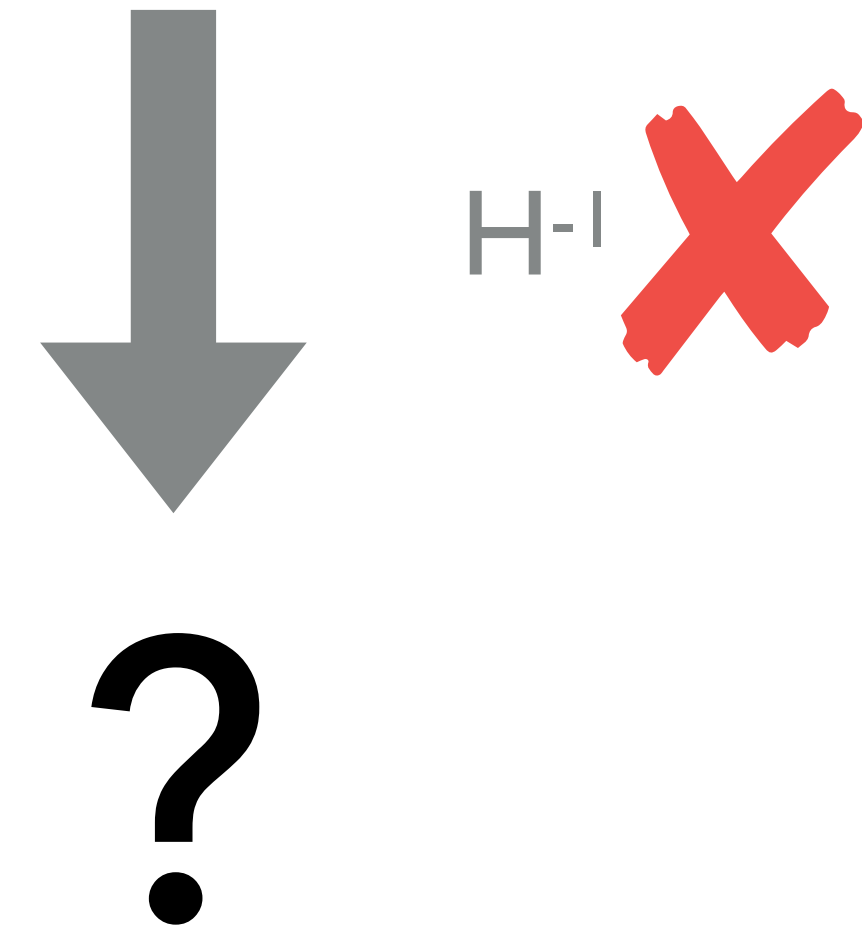
## Resistência a pré-imagem

Dado  $H(x)$ , é computacionalmente difícil/  
custoso determinar  $x$

Analogia a impressão digital:

De quem é essa impressão digital?

da2876b3eb31edb4436fa4650673fc6f  
01f90de2f1793c4ec332b2387b09726f



# FUNÇÕES HASH

## Resistência a segunda pré-imagem

Dado  $x$ , é computacionalmente difícil/custoso encontrar algum valor  $x'$  em que:

$$H(x) == H(x')$$

Analogia a impressão digital:

Você pode achar alguém com a mesma impressão digital que você?





# FUNÇÕES HASH

## Resistência a segunda pré-imagem

Dado  $x$ , é computacionalmente difícil/  
custoso encontrar algum valor  $x'$  em que:

$$H(x) == H(x')$$

## Analogia a impressão digital:

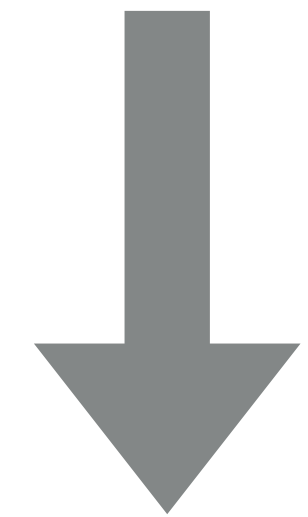
Você pode achar alguém com a mesma  
impressão digital que você?

satoshi



da2876b3eb31edb4436fa4650673fc6f  
01f90de2f1793c4ec332b2387b09726f

?



da2876b3eb31edb4436fa4650673fc6f  
01f90de2f1793c4ec332b2387b09726f

# FUNÇÕES HASH

## Resistência a colisão

É computacionalmente difícil/custoso encontrar **x** e **y** em que:

$$H(x) == H(y)$$

Analogia a impressão digital:

Você pode encontrar duas pessoas quaisquer com a mesma impressão digital?



# FUNÇÕES HASH

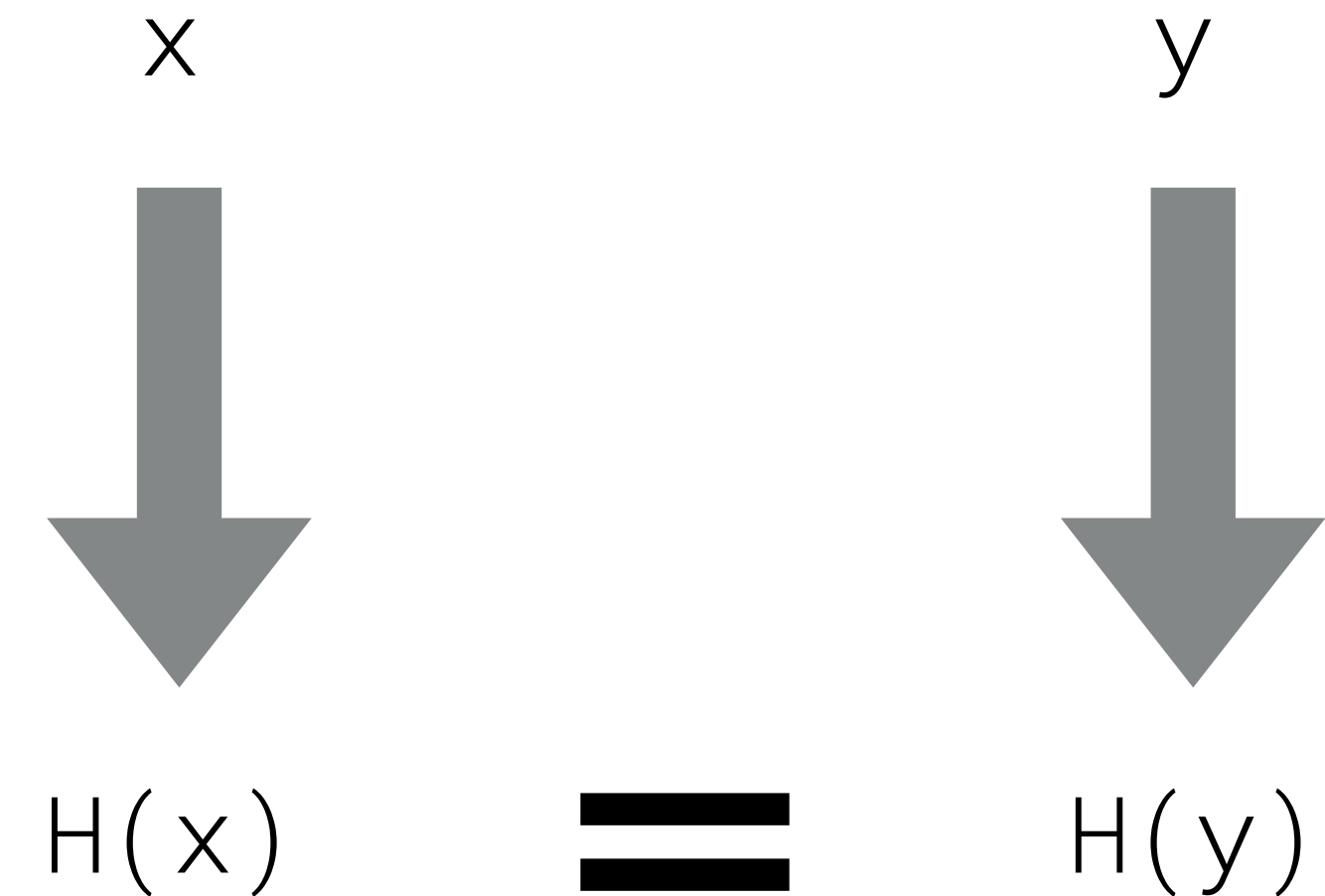
## Resistência a colisão

É computacionalmente difícil/custoso encontrar **x** e **y** em que:

$$H(x) == H(y)$$

Analogia a impressão digital:

Você pode encontrar duas pessoas quaisquer com a mesma impressão digital?





# FUNÇÕES HASH

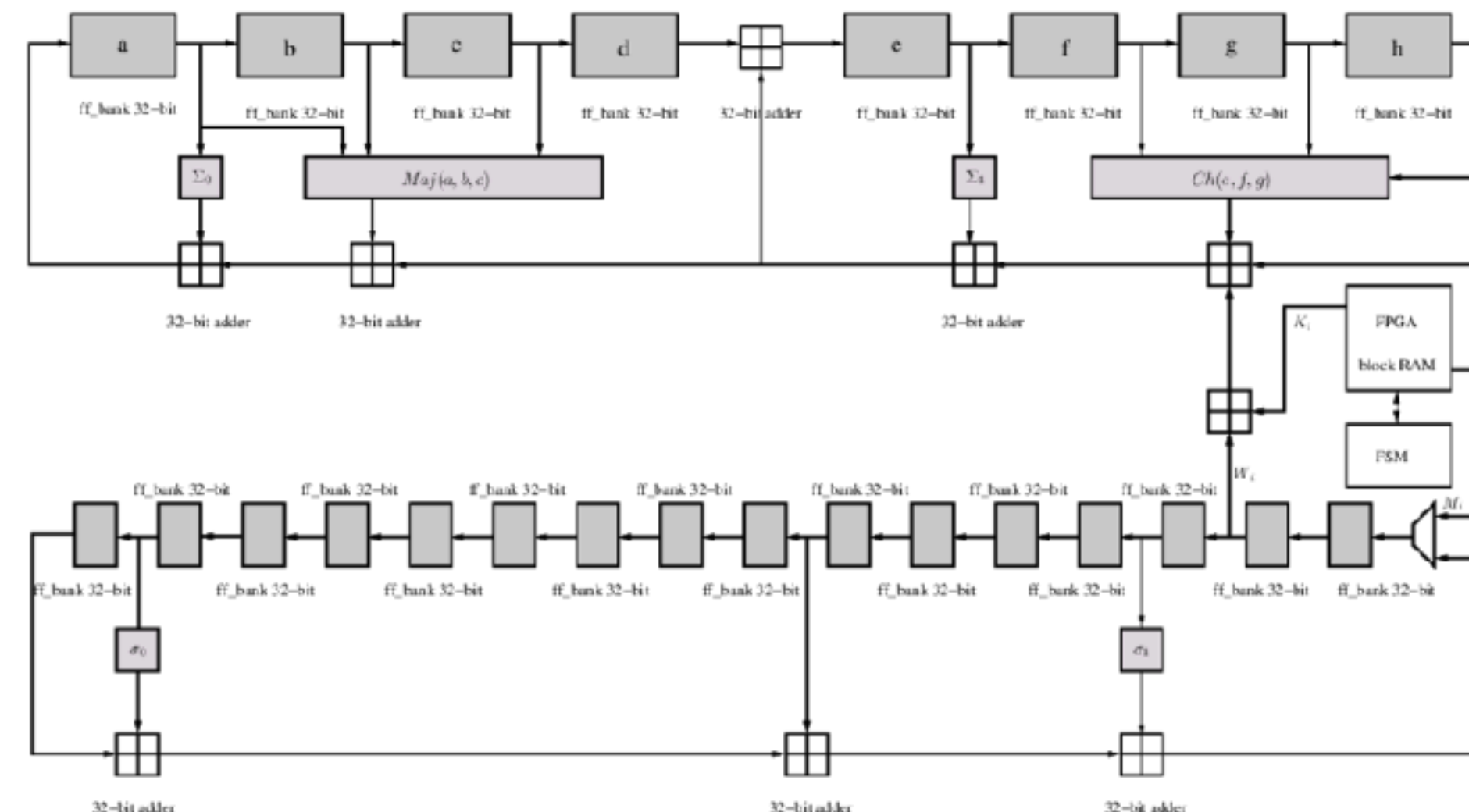
- ▶ **Efeito Avalanche:** uma pequena mudança na entrada produz uma mudança pseudo-aleatória na saída
- ▶ Normalmente uma mudança significativa em relação a entrada anterior
- ▶ Previne jogo de “quente ou frio” com entradas para prever saídas

```
Eu sou o Satoshi Nakamoto 1 => f15a19ceb84f422917a79243c1243367477b446898f11197dba47f72c8fadbe9
Eu sou o Satoshi Nakamoto 2 => 53b4dbcf2b965765163fa409f9845066a22128ed67d9c66c64f521e22edb6941
Eu sou o Satoshi Nakamoto 3 => 498c8d788d8d2d0da2a510c62506b70012a20448c5d56e5426061138759822c8
Eu sou o Satoshi Nakamoto 4 => 88bf13017cbb390c85f4295a326d8f0def7c2d4ded140ae8354d85029dbc9e77
Eu sou o Satoshi Nakamoto 5 => bb361264ad2d6ebafea5dc4ae714bfe309d882370deb9a7d63d3e70bd3d78c42
Eu sou o Satoshi Nakamoto 6 => e025e6d946bf1dc5364c593f921c59539186d312042d8c1b072b3b06b26f1620
Eu sou o Satoshi Nakamoto 7 => 74af15487e45b67fdffe3921e4a93bffa6914597f65660c496df498690c04437
Eu sou o Satoshi Nakamoto 8 => 39c4b50e02424c5112cd67fa98ff7d0101d4c43d5bb49e38c66a2470cd668aa8
Eu sou o Satoshi Nakamoto 9 => 7120f5dd3d5c8c69f53f3178778b36ed2a5ae5d84505e3f880aea92a869202cb
Eu sou o Satoshi Nakamoto 10 => edcdf591c55201148c100bced1812e83ab26e26a476b2a710952c54caaed1a3a
Eu sou o Satoshi Nakamoto 11 => 1b631c8e942202de0c681232aff3e54c33cde3e22fa305043fd68f1d2665f861
Eu sou o Satoshi Nakamoto 12 => 3f80ad796a2644d54ef054620fb2b1a0ed9ed22956e98cf72b676a8c7bb2d7b3
Eu sou o Satoshi Nakamoto 13 => 7d69c86fde1cc84cf135ab2bf53e885ec708c7148ffa2eed2c4c0e7d11fd6239
Eu sou o Satoshi Nakamoto 14 => c808374e4954ae3ba59f4af9eef1df31f2b8abaa38588e480d8ac0a9b90e83e2
Eu sou o Satoshi Nakamoto 15 => 8ed1ef6d1f98b410a86eaa4465764bb8b218a40e4a28f8a76c5deadb5ba05f42
Eu sou o Satoshi Nakamoto 16 => 4569e8c8cfd14eb2325de3255904a2cb0a182612c9d5cfa4dedf5c509ea7ab1d
Eu sou o Satoshi Nakamoto 17 => 4cc7d658552478874103d01b7298d6a4bf3b040e970176b45ff8e4051e9648f0
Eu sou o Satoshi Nakamoto 18 => b3cf352b09933ec1ea05d946079688b425dcf7a20629106afbd5952cde77b6a7
Eu sou o Satoshi Nakamoto 19 => 9782656e9bc25e2b2d32e6c81b3d0e90bd062f2db068b6cc14e64b51f3a5f706
Eu sou o Satoshi Nakamoto 20 => 3fc700496fde534f56f550cb4a1cc2048997e3fc82ee9e182a6c62626b2253e8
```

# SHA-256

- ▶ **SHA-256**: Uma função criptográfica *hash* desenvolvida pela NSA
- ▶ Bitcoin usa **SHA-256<sup>2</sup>** (SHA-256 ao quadrado), isso significa que:

$$H(x) = \text{SHA256}(\text{SHA256}(x))$$



# ONDE FUNÇÕES HASH SÃO UTILIZADAS NO BITCOIN?

▶ IDs das transações

6a47de6fb4ce6f351ca0b0e7ec3eb39dddbffd1756c70bf11ac654f54f58edaa

▶ IDs dos blocos

00000000000000000000000087ffafde405cafe4370827d199751f49ef039863447f7

▶ Mineração

▶ Merkle root

9a1c92df49c0f79d3bf03bbbc33b3cbf97caa2fc7c036ca96ad111907fed34ea

▶ Endereços

19iqYbeATe4RxghQZJnYVFU4mjUUu76EA6

▶ Assinaturas digitais

7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8



# HASHING – DEMO

<https://andersbrownworth.com/blockchain/hash>

# HASHING EM PYTHON

import **hashlib**

<https://docs.python.org/3/library/hashlib.html>

hashlib.sha256(\*).hexdigest()

```
import hashlib  
  
print(hashlib.sha256('Blockchain'.encode()).hexdigest())  
  
# 625da44e4eaf58d61cf048d168aa6f5e492dea166d8bb54ec06c30de07db57e1
```