Introdução ao funcionamento do Bitcoin

Paulo Matias[1]

[1]Departamento de Física e Ciência Interdisciplinar
Instituto de Física de São Carlos
Universidade de São Paulo

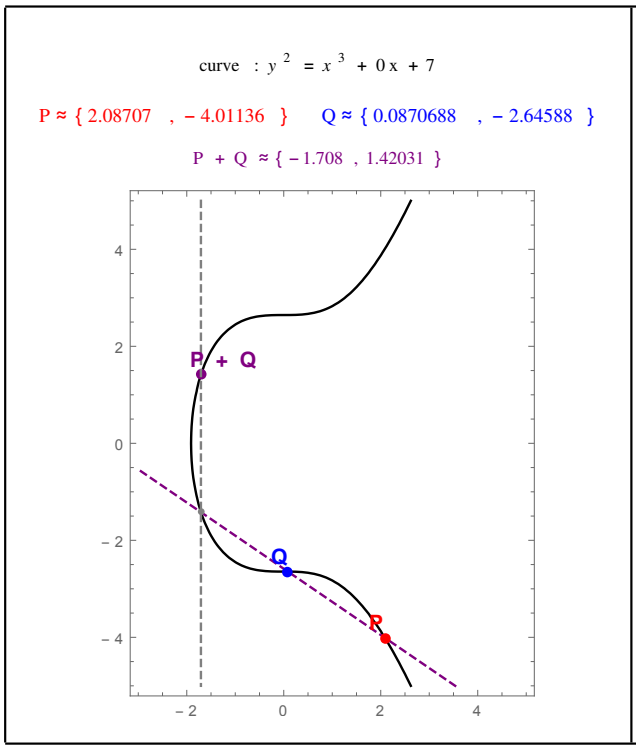
17ª Semcomp

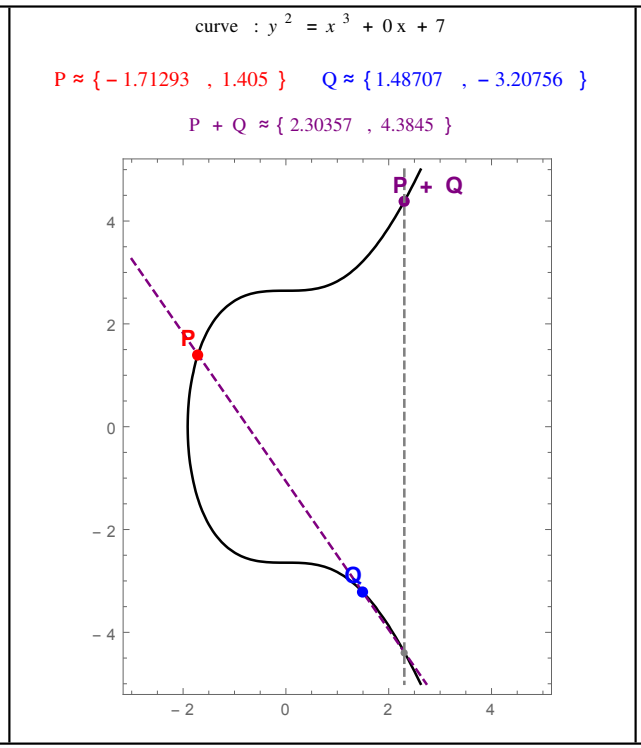
* Para criar uma moeda digital, é necessário:
  ** Uma forma de **somente o proprietário** de um certo valor **autorizar** sua transferência $\Rightarrow$ **assinaturas digitais**.
  ** Uma forma de **impedir** que um certo valor seja gasto **duas vezes**.
    *** Moedas de jogos $\Rightarrow$ autoridade central.
    *** Bitcoin $\Rightarrow$ **blockchain**.

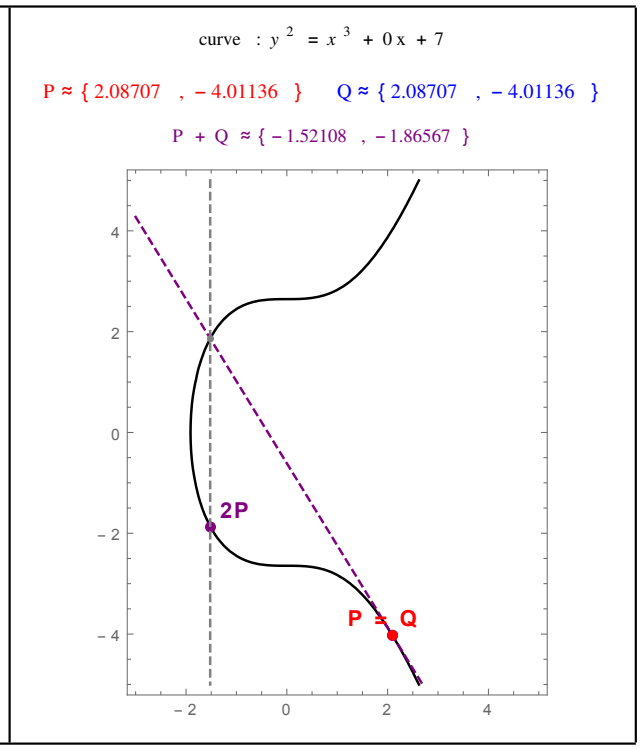
* Assinatura digital $\Rightarrow$ **curva elíptica secp256k1**
  ** Equação: $y^2 = x^3 + 7$
  ** Definida no corpo de Galois $\mathbb{Z}_{2^{256}-2^{32}-2^9-2^8-2^7-2^6-2^4-1}$
  ** Assustou? Segure-se na cadeira e você poderá **ganhar** o equivalente a **R$40** em bitcoins!
* **Blockchain**
  ** Função de via única (hash) **SHA256** utilizada para gerar uma **prova de trabalho** de uma cadeia de **registros de transações**.
  ** Inovação do Bitcoin!
  ** Essa parte é um pouco mais simples de entender.

- **Chave privada**: só o proprietário do dinheiro conhece.
  Trata-se de um grande número inteiro d_A
  que deve ser mantido em **sigilo**.
- **Chave pública**: como o nome diz, é pública, todo mundo conhece.
  Trata-se de um par de números inteiros Q_A.
- **Assinatura**: dado um conjunto de inteiros (r, s, e),
  deve ser possível verificar, conhecendo **somente** Q_A,
  que r e s foram calculados a partir do valor e por alguém
  que conhecia o valor de d_A.
- Como isso é possível?

- **Propriedade**: uma reta diagonal sempre corta 3 pontos da curva
- Como 2 pontos definem uma reta, dados 2 pontos é possível determinar um terceiro (que também corte a curva).
- Chamaremos essa operação de **"soma"**
  (apesar de não ser uma soma das coordenadas!)







Com um pouco de Geometria Analítica, é possível calcular essa operação algebricamente:

$$R = P + Q \quad \Rightarrow \quad \begin{pmatrix} r_x \\ r_y \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} + \begin{pmatrix} q_x \\ q_y \end{pmatrix}$$

Se $P \neq Q$ então:

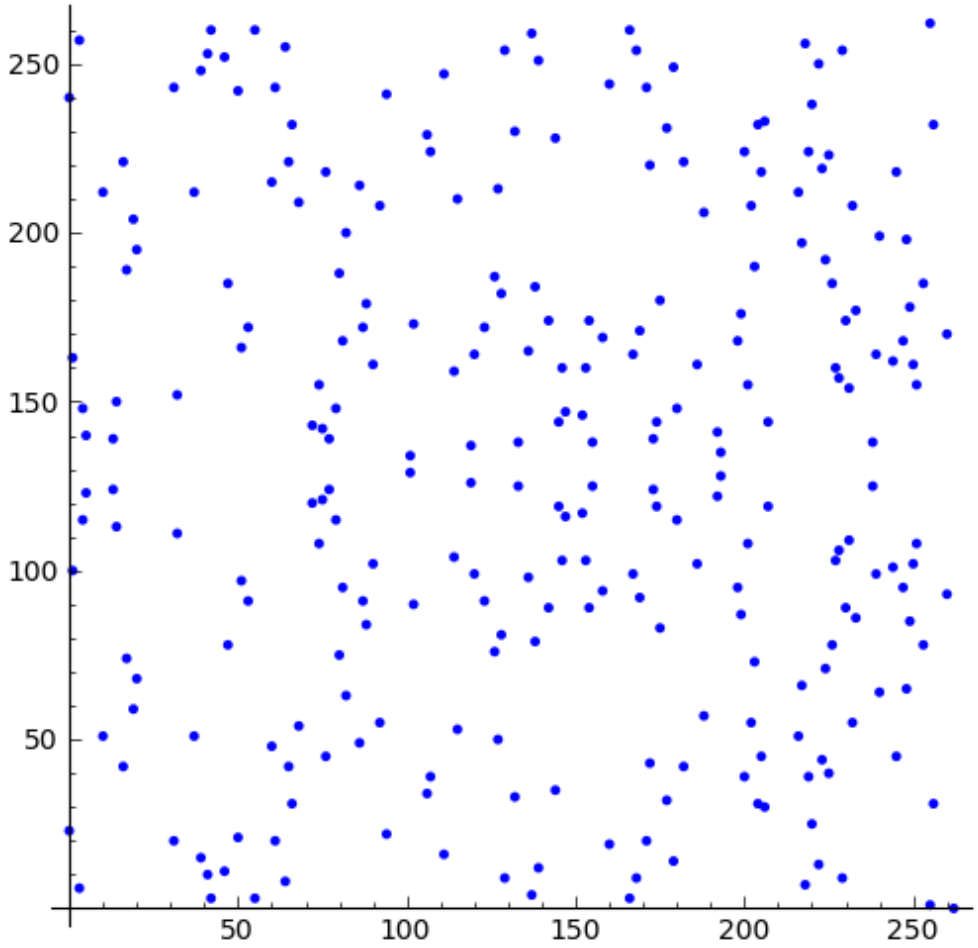
$$r_x = s^2 - p_x - q_x \quad \text{e} \quad r_y = s \cdot (p_x - r_x) - p_y \quad \text{onde} \quad s = \frac{p_y - q_y}{p_x - q_x}$$

Se $P = Q$ (ou seja, $R = 2P$), então:

$$r_x = s^2 - 2p_x \quad \text{e} \quad r_y = s \cdot (p_x - r_x) - p_y \quad \text{onde} \quad s = \frac{3p_x^2 + a}{2p_y}$$

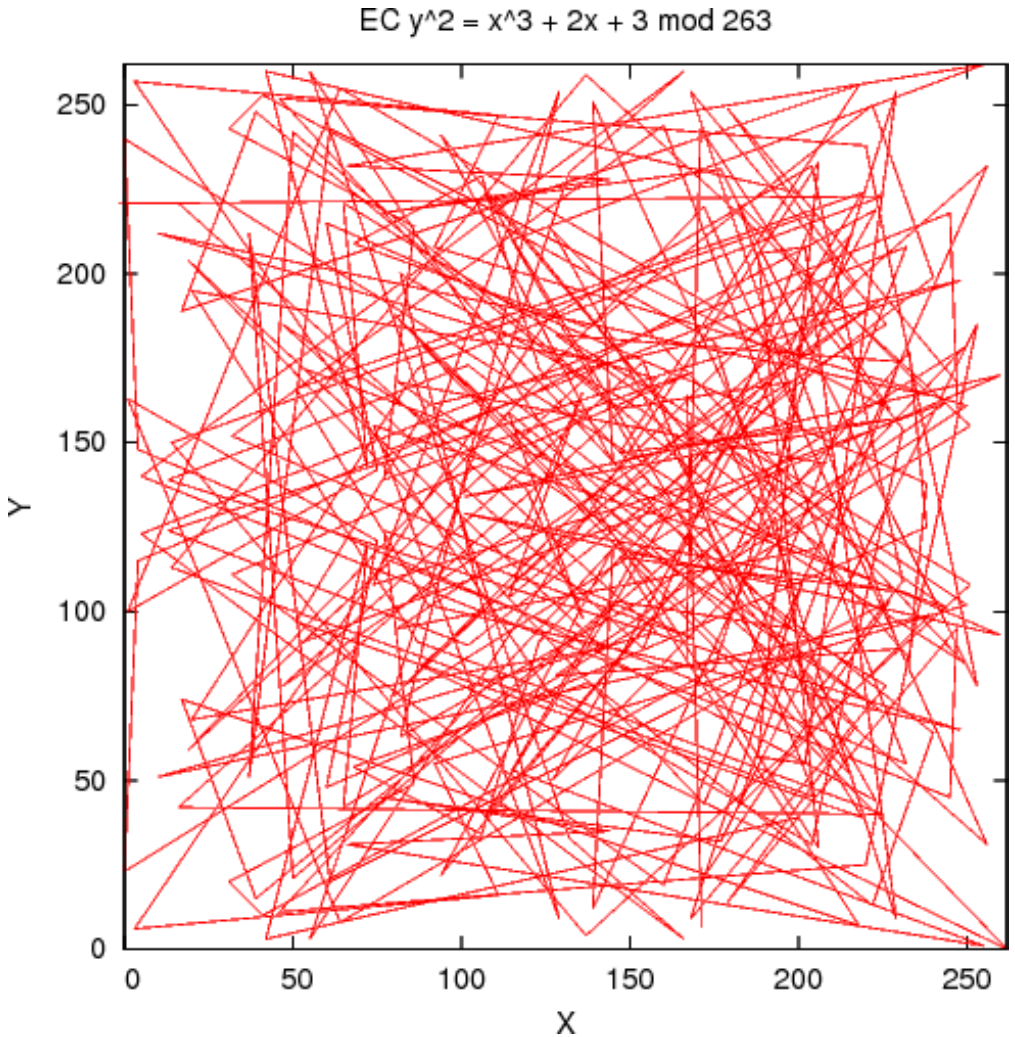
* E se em vez de fazer essas operações em números reais, fizermos em **inteiros**?
* E mais, e se **limitarmos** um **valor máximo** para esses inteiros (tomando o resto da divisão com um número **primo** após cada operação)
  ** **Propriedade**: o resto da divisão pode ser calculado a qualquer momento e quantas vezes se queira, desde que o valor final da operação fique abaixo do valor máximo.
  ** Experimente:
    *** $(2+4+3+1) \bmod 5 = (2+4+3) \bmod 5 + 1 \bmod 5 =$
    $= ((2+4) \bmod 5 + (3+1) \bmod 5) \bmod 5 = 0$
    *** $(3 \cdot (4+3)) \bmod 5 = ((3 \bmod 5) \cdot ((4+3) \bmod 5)) \bmod 5 = 1$

O que era uma curva torna-se uma **nuvem de pontos**.



* A **"multiplicação"** de um número d_A por um ponto G (somar G consigo mesmo d_A vezes: $G + G + \cdots + G$) é **rápida de calcular** em um computador.
  ** Usando a decomposição de d_A em potências de 2.
  Se $d_A = 11$, então decompomos $d_A = 1 + 2 + 8$,
  então $d_A \cdot G = G + 2G + 2^3 G$,
  onde $2^i G$ é calculado dobrando G consecutivamente.
* A **"divisão"** de um ponto $Q_A = d_A \cdot G$ pelo ponto G (para recuperar o valor de d_A) é extremamente lenta.
* **Chave pública**: Q_A. **Chave privada**: d_A.
  Sendo que G é um valor público, conhecido e fixo para um certo algoritmo de assinatura.

Múltiplos consecutivos de G geram pontos bastante **espalhados** pelo plano. Para um certo G, é rápido calcular o número de pontos n diferentes que podem ser gerados a partir de múltiplos de G. Escolhe-se um G tal que n seja **grande** (próximo de 2^{256}).



Assinatura: Escolhe-se um número k aleatório entre 0 e n. Esse k é mantido em segredo. Calcula-se $R = k \cdot G$. A partir da coordenada r_x do ponto R, calcula-se:

$$\boxed{s = \frac{e + d_A \cdot r_x}{k} \mod n}$$

Onde e é o número a ser assinado. Lembrando que d_A é a chave privada, e que n é um número fixo para o algoritmo.
Envia-se r_x e s juntamente com e para o destinatário.
Verificação: Não é necessário d_A para validar a assinatura, apenas a chave pública Q_A. Calcula-se:

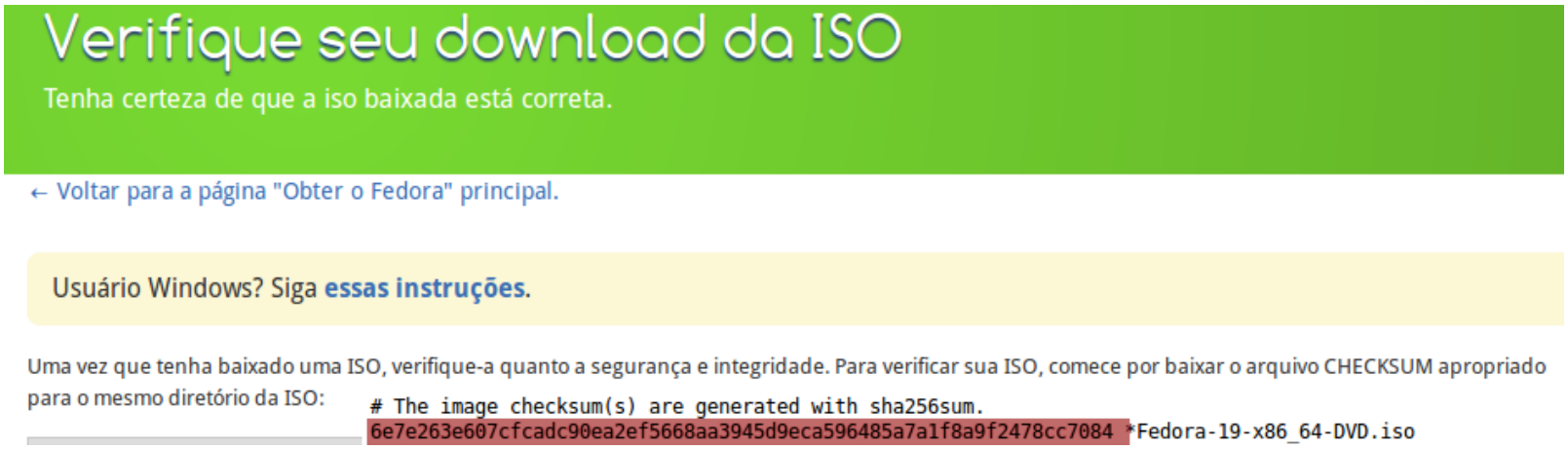
$$\boxed{P = e \cdot s^{-1} \cdot G + r_x \cdot s^{-1} Q_A}$$

A assinatura é válida se a coordenada p_x de P for igual a r_x, pois, lembrando que $Q_A = d_A \cdot G$:

$$P = s^{-1} \cdot (e + r_x \cdot d_A) \cdot G = \left(\frac{e + d_A \cdot r_x}{k} \right) \cdot (e + r_x \cdot d_A) \cdot G = k \cdot G = R$$

- Fixe o conceito de chave **pública** e chave **privada**. Uma assinatura pode ser gerada com uma chave privada e apenas a chave pública é necessária para validá-la.
- Anote este link e separe algumas horas, algum dia, para ler uma explicação mais detalhada: http://www.johannes-bauer.com/compsci/ecc/
- **Contextualizando**: um **endereço Bitcoin** é o identificador (hash) de uma **chave pública**. Ele é proprietário de uma certa quantidade de bitcoins. Para transferi-las, ele utiliza sua chave privada para assinar uma transação.

Antes de prosseguir para a explicação sobre transações, precisamos responder à pergunta: **o que é um hash?**



Propósito similar ao dígito verificador nos números de CPF. Porém números de CPF são lineares!

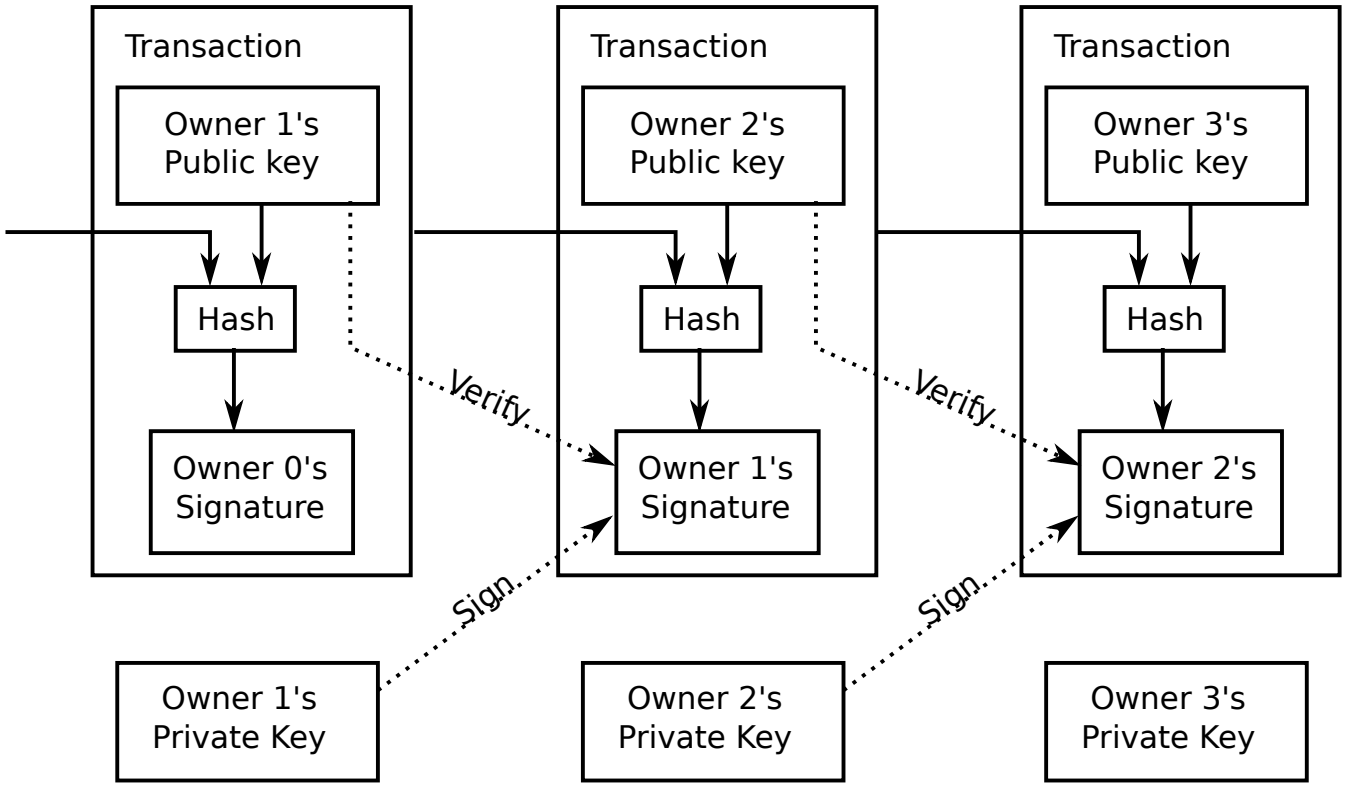
$$\mathrm{CPF}(A+B) = \mathrm{CPF}(A) + \mathrm{CPF}(B)$$

```
  123.123.123-87      712.836.511-73      297.173.619-91
+ 321.321.321-78    + 221.152.476-12    + 768.172.923-28
----------------    ----------------    ----------------
  444.444.444-44      933.988.987-85      944.235.431-09

onde:               onde:               onde:
(8 + 7) mod 11 = 4  (7 + 1) mod 11 = 8  (9 + 2) mod 11 = 10
(7 + 8) mod 11 = 4  (3 + 2) mod 11 = 5  (1 + 8) mod 11 =  9
```

Uma função hash H segura jamais poderia ter essa propriedade. Uma pequena mudança na entrada deve corresponder a uma grande mudança na saída. A saída deve parecer aleatória caso a entrada não seja conhecida. Mais que isso — obviamente existem hashs repetidos para certos pares de entradas $M_1 \neq M_2$, pois a entrada tem geralmente muito mais bits que a saída, mas nenhum destes pares deve ser conhecido, ou seja, $H(M_1) \neq H(M_2)$ para $M_1 \neq M_2$ para todo hash já calculado no mundo.

* Um bitcoin não existe como uma entidade isolada. Uma transação sempre se refere a uma transação anterior.
* A saída (quantidade de bitcoins recebida) de uma transação pode ser usada como entrada (bitcoins a serem transferidos) em outra transação.



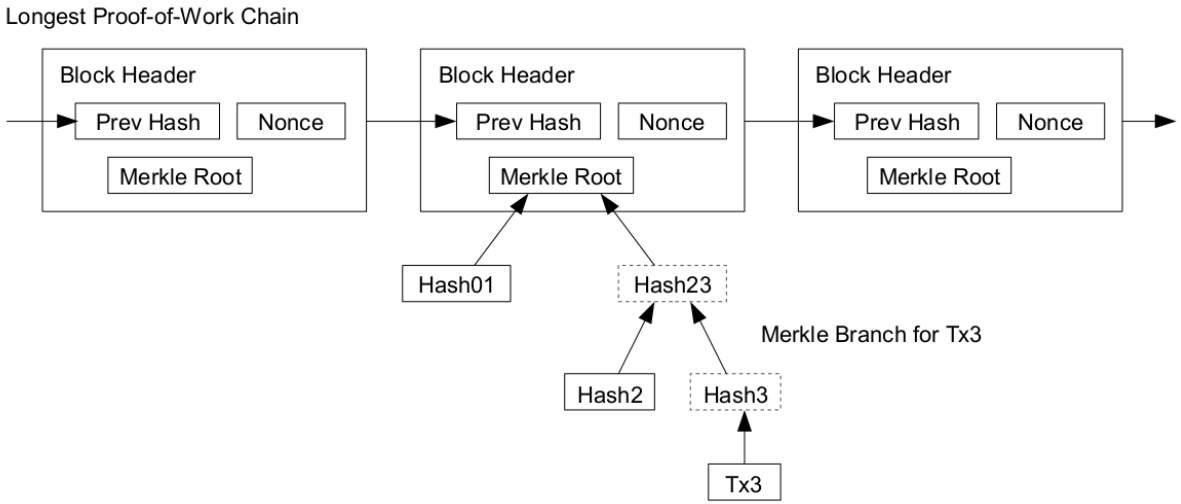
Uma transação é composta basicamente de:

* Lista de entradas:
  ** Identificador (hash SHA256 duplo) de uma transação preexistente.
  ** Número da saída dessa transação que deve ser utilizada.
  ** Assinatura provando que o usuário é dono do dinheiro contido naquela saída (scriptSig).
* Lista de saídas:
  ** Valor da saída em satoshis (10^{-8} bitcoins).
  ** Identificador da chave pública que passa a ser dona desta saída (scriptPubKey).

O valor de e utilizado no algoritmo de curva elíptica para cada assinatura corresponde ao hash de uma versão incompleta da transação, que ainda não contém assinaturas em suas entradas.

- OK, mas **de onde surgiram** as primeiras transações, se os bitcoins não existem como entidade isolada?
- **E se** um usuário **gastar** os mesmos bitcoins **duas vezes**, ou seja, usar a mesma saída como entrada em duas transações diferentes? Qual dessas duas transações é válida? **Quem** se torna o **dono** dos bitcoins?

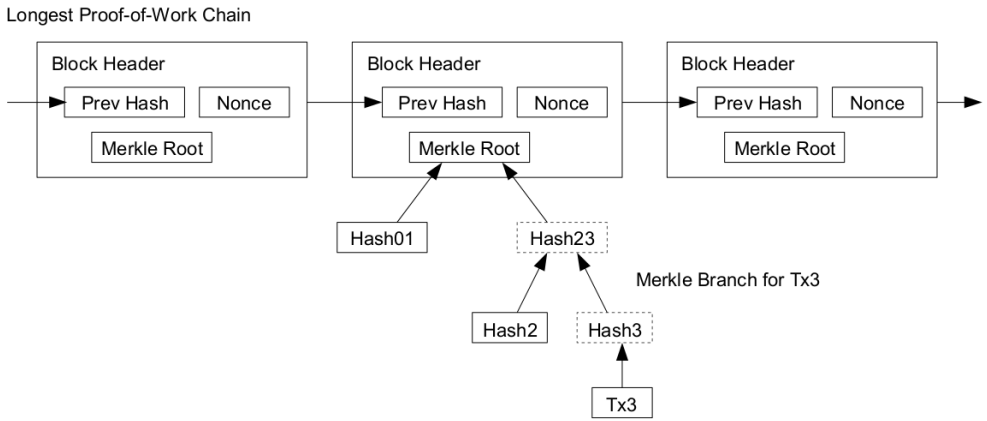
* **Bloco**: é um conjunto de transações.
  ** Precisa de muito trabalho computacional para ser gerado $\Rightarrow$ 10 min de processamento de toda a rede Bitcoin no mundo.
* Cadeia de blocos (**blockchain**): conjunto de todo o histórico da rede, onde um bloco sempre contém o hash do anterior.



* O software aceita como válida a cadeia na qual tenha sido depreendido o maior esforço computacional.
* Portanto, quanto mais antiga uma transação, mais difícil de retirá-la do histórico.

- A ideia do blockchain é imitar o livro de registros de transferência de imóveis em um cartório. **O registro é público**, assim como o livro de um cartório.
  - O Bitcoin está longe de ser anônimo como alguns dizem.
  - Pesquisas recentes pretendem resolver este problema: Zerocoin.
- Uma vez no blockchain depois de gerados 6 blocos, o receptor do dinheiro pode se sentir seguro de que ele não será gasto novamente pelo dono original.
  - Metade da rede teria que conspirar conjuntamente, ao longo de 1 hora, para mudar o histórico.

* Para ser válido, o valor numérico do hash do bloco deve ser menor que um limiar de esforço ajustado a cada 2016 blocos ($\approx$ 14 dias).
* Existe um "campo livre" no bloco (nonce) que pode ser modificado para alterar o hash final.



* As máquinas participantes da rede Bitcoin estão continuamente tentando gerar blocos com um hash considerado válido $\Rightarrow$ **mineração**.
* A primeira máquina a conseguir gerar um bloco válido pode definir o destinatário de uma transação de prêmio (atualmente 25 BTC, reduzido pela metade a cada 4 anos).

O blockchain garante que o receptor de uma quantia não vá mais perdê-la depois de consolidados alguns blocos, porém

Lição de casa: o que acontece se ainda assim forem geradas múltiplas transações que utilizem como entrada uma mesma saída?

- Anote a seguinte **chave privada**, proprietária do equivalente a **R$40** em bitcoins:
  "5KAh7DRoZAc6RL8huL5dLthmkBPEx4CEV2gUUK3WF3MP5CArGEm"
- Instale o **Electrum**. Vá no menu Wallet→Private keys→Import.
- Não esqueça de transferir imediatamente esse valor para outro endereço Bitcoin em sua posse. (Por que?)