

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM MATEMÁTICA

Luiz Carlos da Silva Leão

Uma introdução ao estudo de bitcoins e blockchains

Rio de Janeiro
2019

Luiz Carlos da Silva Leão

Uma introdução ao estudo de bitcoins e blockchains

Trabalho de Conclusão de Curso apresentado ao Programa de Pós-graduação em Matemática PROF-MAT da UNIRIO, como requisito para a obtenção do grau de MESTRE em Matemática.

Orientador: Silas Fantin
Doutor em Matemática - USP

Rio de Janeiro
2019

Catálogo informatizada pelo(a) autor(a)

L953 Leão, Luiz Carlos da Silva
Uma introdução ao estudo de bitcoins e
blockchains / Luiz Carlos da Silva Leão. -- Rio de
Janeiro, 2019.
118

Orientador: Silas Fantin.
Dissertação (Mestrado) - Universidade Federal do
Estado do Rio de Janeiro, Programa de Pós-Graduação
em Matemática, 2019.

1. Bitcoin. 2. Blockchain. 3. Hash. I. Fantin,
Silas, orient. II. Título.

Luiz Carlos da Silva Leão

Uma introdução ao estudo de bitcoins e blockchains

Trabalho de Conclusão de Curso apresentada ao Programa de Pós-graduação em Matemática PROF-MAT da UNIRIO, como requisito para a obtenção do grau de MESTRE em Matemática.

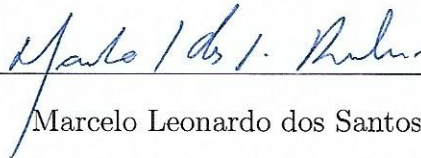
Aprovado em 24 de Outubro de 2019

BANCA EXAMINADORA



Silas Fantin

Doutor em Matemática - USP



Marcelo Leonardo dos Santos Rainha

Doutor em Engenharia Nuclear - UFRJ



Sergio José Xavier de Mendonça

Doutor em Matemática - IMPA

Dedico este trabalho à minha mãe, que não está mais entre nós, mas continua sendo minha maior força na vida. Sua lembrança me inspira e me faz persistir.

Resumo

Esta dissertação de mestrado apresenta uma introdução aos elementos de Bitcoins e de Blockchains. Primeiramente introduz-se como se dá o funcionamento do Blockchain e os seus fundamentos básicos. Depois se revisa a criptografia até chegar às funções de hash criptográficas. Destaca-se a criptografia por chaves públicas e os elementos de simetria criptográfica. A seguir, estuda-se transações, formação de blocos e de endereços Bitcoin. É feita uma sugestão de didática sobre blockchain para educação. São apresentadas as funções hash RIPEMD-160 e SHA-256, esta última com um procedimento de cálculo. Por fim, contextualiza-se Blockchain e educação superior e é feita uma sugestão para estudos posteriores.

Palavras-chaves: Bitcoin, Blockchain, Hash.

Abstract

This master thesis presents an introduction to Bitcoin and Blockchain elements. Firstly, we introduce how Blockchain works and its fundamentals. Then the cryptography is reviewed until it reaches the cryptographic hash functions. Public keys cryptography and the elements of symmetric cryptography are highlighted. Bitcoin transactions, block formation, and addressing system are studied. A suggestion of blockchain didactics for education is made. Are presented the Bitcoin Address composing and the cryptographic hash functions used in the Bitcoin RIPEMD-160 and SHA-256, this one with a calculation procedure. Finally, Blockchain is contextualized with college education and it is made a suggestion for further study.

Keywords: Bitcoin, Blockchain, Hash

Agradecimentos

Agradeço primeiramente a Deus.

À minha esposa amada, que precisou suportar tantos momentos de ausência para que esse trabalho fosse realizado.

À minha família, pois sem eles eu nada seria.

Ao professor Silas Fantin, pela orientação zelosa e por não deixar acreditar que seria possível.

Aos demais professores do programa PROFMAT na UNIRIO, pela formação.

Aos meus colegas de turma, por todo apoio quando precisei.

Aos meus gestores do emprego atual, por autorizar conciliar os horários para cursar as disciplinas.

À Sociedade Brasileira de Matemática por manter o programa.

A todos que, de alguma forma, contribuíram para a realização deste trabalho.

“Eu não acredito que teremos um bom dinheiro de novo antes de tirá-lo das mãos do governo, isto é, não podemos tirá-los violentamente das mãos do governo, tudo o que podemos fazer é por algum caminho indireto introduzir algo que eles não podem parar.”

F.A. Hayek, 1984

Sumário

Lista de Figuras	8
Lista de Tabelas	15
Introdução	16
Terminologia	19
Funcionamento do Blockchain	22
Criptografia	26
Criptografia de chave pública	32
Criptografia assimétrica no blockchain	35
Funções Hash	36
Funções Hash	36
Usando aritmética modular em função hash	39
As funções JHA e JHA-1	43
A função hash criptográfica RIPEMD-160	45
A função hash criptográfica SHA-256	46
Blockchains	48
Transações	48
Pool de transações	50
Assinaturas digitais	51
Transação de Bitcoin na vida real	53
Blocos	62

Ponto de partida: um livro	64
Proof-of-work	66
Endereço Bitcoin	70
Blockchain	72
Proposta Didática	75
Administrando uma escola secundária de maneira distribuída	76
Blockchain para o ensino médio distribuído	78
Uma introdução ao sistema distribuído	80
A corrida pela prova de trabalho	82
Construindo um blockchain	85
Apresentando Chaves Públicas e Privadas	89
Considerações finais	92
Conclusão	93
Apêndice	94
Referências Bibliográficas	114

Lista de Figuras

1	Caixa eletrônico de Bitcoin em Milwaukee, Wisconsin	17
2	Rede Peer-to-peer	20
3	Cadeia de blocos de hash	22
4	Semelhança entre a página de um livro caixa do início do Século XX e um bloco na blockchain	23
5	Grafo de um Esquema Ponzi	25
6	Claude Elwood Shannon	26
7	Horst Feistel	27
8	Martin Hellman	28
9	Whitfield Diffie	29
10	Alan Konheim	29
11	Neal Koblitz	30
12	Victor Miller	30
13	Ilustração esquemática da criptografia simétrica	32
14	Ilustração esquemática da criptografia assimétrica	33
15	Ilustração da função hash	36
16	Exemplo cálculo de hash	37
17	Cálculo de hash abreviado	38
18	Problema do Estacionamento e função hash	42
19	Planilha de cálculo de uma rodada da função SHA-256	47
20	Diagrama de Transações	49
21	Criação de uma assinatura digital	51
22	Usando uma assinatura digital	52

23	Identificando uma fraude	53
24	O novo endereço bitcoin da Alice	54
25	A tela de envio de bitcoin	56
26	Cadeia de Transações	58
27	Transação de Alice para o Bob	60
28	Blocos	63
29	Páginas do livro	65
30	Dificuldade Mineração	69
31	Conversão para Endereço Bitcoin	70
32	Blockchain	72
33	Professor como autoridade central	76
34	Banco como autoridade central	77
35	Transação	78
36	Transação de bitcoin	78
37	Veterano	79
38	Um bloco	79
39	Conjunto de memória	81
40	Bob classifica Alice	81
41	Mineração	82
42	Caça ao Tesouro	83
43	Tesouro descoberto	84
44	Histórico do jardim de infância ao 1 ^o Ano	85
45	Histórico do jardim de infância ao 1 ^o Ano com funções hash	86
46	Nova tentativa	87
47	Tentativa de Alice de substituir sua nota anterior	87
48	Chance de uma transação voltar no dia seguinte	88

49	Armários	89
50	Transação com hashing	90
51	Armários com chaves	91
52	Operação de soma em uma curva elíptica	94
53	Método ElGamal para Criptografia de Curvas Elípticas	98
54	A Curva Koblitz	101
55	Tabela-verdade da negação (\sim)	103
56	Tabela-verdade do E (\wedge)	103
57	Tabela-verdade do OU (\vee)	103
58	Tabela-verdade do OU-Exclusivo (\oplus)	103
59	O cálculo da função Maj	104
60	O cálculo da primeira função Rotate	104
61	O cálculo da função Ch	105
62	O cálculo da segunda função Rotate	106
63	O cálculo de $T1$	108
64	O último passo do cálculo da função hash SHA-256	108
65	A função de hash RIPEMD-160	109
66	Metades esquerda e direita do RIPEMD-160	111

Lista de Tabelas

1	Nonces para resolver um quebra-cabeça de hash	68
2	Conceitos técnicos do blockchain e seus propósitos	74
3	Hexadecimal da raiz quadrada da parte fracionária dos primeiros 8 primos	102
4	Tabela-verdade da função <i>Maj</i>	104
5	Tabela-verdade da função XOR	105
6	Tabela-verdade da função <i>Ch</i>	106
7	Deslocamento <i>s</i> para o lado direito	112
8	Deslocamento <i>s</i> para o lado esquerdo	113

Introdução

Atualmente o uso da moeda está se tornando cada vez menos comum. As despesas estão sendo pagas com cartão de crédito emitido por um banco, mas ao voltarmos um pouco no tempo, verificaremos que o cartão de crédito marcou uma revolução financeira em termos de segurança das transações e de liquidez. Agora assistimos à uma nova revolução financeira causada pela chegada de um novo tipo de dinheiro, a criptomoeda, que é uma nova geração de recurso financeiro criado através de tecnologia que utiliza algoritmos criptografados.

Nos últimos anos, várias novas criptomoedas foram criadas. Neste trabalho pretendemos apresentar uma introdução sobre essa nova tecnologia, abordando fundamentalmente seus aspectos criptográficos.

Para apresentar como funciona a criptografia da tecnologia Bitcoin devemos esclarecer alguns dos seguintes questionamentos (que serão abordados ao longo deste trabalho):

- O que é o Bitcoin?
- Como armazenar os bitcoins ¹?
- Como funcionam as transações de bitcoin?
- Como funciona a mineração de bitcoins?
- O que é e como funciona a tecnologia Blockchain?
- O que é um ledger (livro razão) distribuído?

O Bitcoin é uma das criptomoedas mais conhecidas. Blockchain (ou cadeia de blocos) é a nova tecnologia na qual o Bitcoin é baseado. Schwab, 2016 [46] afirma que a tecnologia Bitcoin logo dará origem a inúmeros outros blockchains.

¹Segundo Ulrich, 2014 [57] quando se refere ao sistema, à rede ou ao projeto Bitcoin, usa-se sempre inicial maiúscula. No entanto, quando se fizer referência às unidades monetárias bitcoins, utiliza-se a palavra em caixa baixa.

As moedas digitais que recebem o nome de criptomoedas são aquelas que são consideradas seguras por causa da criptografia. As criptomoedas não são controladas por autoridades centralizadas. Contudo, as implicações desta tecnologia são tão vastas, que alguns Bancos Centrais tentam criar suas próprias criptomoedas [11], [24], [28], mas as produzidas segundo Hollins, 2018 [26] não são consideradas criptomoedas oficialmente, tendo em vista que o dinheiro ainda é centralizado.

A descentralização tem o intuito de permitir que todos tenham o controle, enquanto, com a centralização, todo o controle está com os bancos centrais e isso implica que em termos da quantidade de dinheiro criado ou seu valor, nenhuma pessoa possui controle. Assim, os bancos centrais podem estipular o valor de moedas tradicionais, mediante a impressão de mais moeda.

O Bitcoin tem tido bastante publicidade. Grande exposição de informações nos noticiários, redes sociais e instituições financeiras podem ser a causa. Como aumentaram os níveis de alfabetização, tanto financeira como digital da população, a aceitação das criptomoedas também cresceu.

Há agora um envolvimento crescente de empresas e instituições com as criptomoedas. Novas aplicações são incorporadas e já existem caixas eletrônicos para permitir que transações de criptomoedas sejam realizadas. Vide Figura 1:



Figura 1: Caixa eletrônico de Bitcoin em Milwaukee, Wisconsin Disponível em [60]

Em seguida há uma descrição sucinta do que será apresentado em cada capítulo:

No Capítulo Terminologia será abordada a terminologia dos principais conceitos que permeiam o funcionamento de blockchains. Introduziremos os conceitos de rede Peer-to-peer, blocos, cadeias, transações e mineração de bitcoins.

No Capítulo Criptografia é feita uma contextualização da criptografia, passando pelos pesquisadores precursores como Claude Shannon e Horst Feistel até Martin Hellman e Alan Konheim. Falaremos de Neal Koblitz e Victor Miller, ambos pesquisadores da criptografia por curvas elípticas. Até chegarmos ao Satoshi Nakamoto, autor do artigo fundamental do Bitcoin. São introduzidos os conceitos de criptografia de chave pública e criptografia assimétrica em blockchains.

No Capítulo Funções Hash é apresentada uma introdução às funções de hash, seus princípios, hashing com aritmética modular e funções hash criptográficas. É apresentado um exemplo de cálculo de hash. Devido a sua facilidade de entendimento e aprendizagem, falaremos das funções hash JHA e JHA-1. Também será falado sobre duplo hashing. Serão introduzidas as funções hash SHA-256 e RIPEMD-160.

Já no Capítulo Blockchains os conceitos de transações, pool de transações, blocos, criação de assinaturas serão aprofundados. Será apresentada uma analogia de um livro com um blockchain. Também será visto como se dá uma transação real de bitcoins. Será apresentada uma breve introdução técnica da criação de Endereços Bitcoin. Ao final, será apresentada uma tabela contendo um resumo dos conceitos abordados.

No Capítulo Proposta Didática é feita uma sugestão de didática sobre bitcoins e blockchains para estudantes do ensino médio com analogias dos termos e conceitos que estão presentes nas escolas: alunos, professores, boletins, diretores, testes, turmas.

Por fim, no Apêndice serão introduzidos os aspectos matemáticos da criptografia de Curvas Elípticas e das funções hash criptográficas SHA-256 e RIPEMD-160, ambas componentes criptográficas do blockchain do Bitcoin. Na função hash SHA-256, será mostrado em detalhes um algoritmo de cálculo assim como as tabelas-verdade dos passos sugeridos.

Terminologia

Neste capítulo introduzimos a terminologia que será utilizada no decorrer deste trabalho, ou seja, serão analisados alguns dos fundamentos tecnológicos que tornam possível o funcionamento das criptomoedas, em especial do Bitcoin. No Capítulo Blockchains haverá um maior aprofundamento teórico dessas definições.

Definição 1.1 (Blockchain): Blockchain, ou cadeia de blocos, é um registro público, que existe apenas em meio digital e que armazena todas as transações das criptomoedas. Estas informações não estão armazenadas em um único computador central.

A ideia por trás do blockchain é substituir instituições centralizadas. Seu objetivo foi que se pudesse criar um caminho para que estranhos tivessem confiança entre si nas transações, sem a necessidade de um banco ou de um governo como mediador.

Basicamente, todas as aplicações de blockchains baseiam-se no conceito de descentralização. Por isso que, em vez da rigidez na tomada de decisões de uma autoridade central, o blockchain pretende devolver o poder regulatório para os indivíduos, isto é, em vez de confiar em uma instituição importante, o blockchain gera confiança através do **consenso**.

Em termos simples, o blockchain é o livro razão ou livro diário que contém todas as operações que se realizaram na história do Bitcoin. Como não existe um governo central ou banco de dados único, o livro se situa em uma rede composta por cada computador que executa o software de Bitcoin e todos trabalham juntos para construir a rede. Tudo isso acontece ao vivo, de forma transparente, e qualquer um pode ver o tráfego enquanto está acontecendo utilizando o método **Peer-to-peer** para distribuir as informações entre todos os usuários. Segundo Hollins, 2018 [26] este nível de transparência é quase inédito no sistema financeiro.

Definição 1.2 (Peer-to-peer): **Peer-to-peer** ou P2P (tradução livre: ponto-a-ponto) é um método de troca de dados (informações, arquivos, documentos, filmes, jogos, etc.) entre dois ou mais usuários.

A principal característica é que o P2P estabelece uma conexão direta entre computadores conectados à internet, sem a necessidade de um serviço intermediário ou de um servidor central. Portanto, a base do blockchain é a rede **Peer-to-peer**. Tradicionalmente, quando se pensa em confiança, se pensa em instituições como intermediárias.

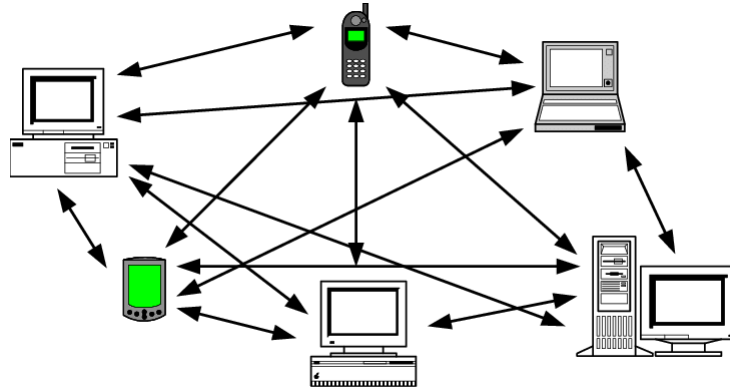


Figura 2: Rede Peer-to-peer. Disponível em: [55]

Exemplo 1.1 Uma transferência bancária de R\$ 100,00 reais de João no banco A para Maria no banco B, funciona da seguinte forma:

1. João, faz a transferência no banco A;
2. O banco A cobraria um percentual como comissão para processar a transação;
3. O banco A verificaria se tem R\$100 na conta do João;
4. O banco A perguntaria ao banco B se a conta de Maria é válida e se está aberta para depósitos;
5. O banco A atualizaria seu livro de contabilidade para subtrair R\$100 da conta de João;
6. O banco B atualizaria o seu livro de contabilidade para adicionar R\$100 na conta de Maria.

Por outro lado, uma rede **Peer-to-peer** não necessita de um intermediário, mas usa o **ledger** distribuído ou livro de contabilidade distribuído para processar as transações. Todos os computadores que fazem parte da rede mantêm uma cópia do livro, e as transações são adicionadas sistematicamente no livro contábil. É bastante difícil alterar o livro de contabilidade porque isso exigiria alterar a cópia do livro em milhares

de computadores na rede **Peer-to-peer**. A mesma transferência do Exemplo 1.1 em uma rede **Peer-to-peer** com um **ledger** distribuído funcionaria conforme Exemplo 1.2 a seguir:

Exemplo 1.2 Uma transferência de R\$ 100 reais em rede **Peer-to-peer** com um **ledger** distribuído de João para Maria, funciona da seguinte forma:

1. João envia a solicitação de transferência para a rede;
2. Em seguida, os computadores mais próximos a João na rede comprovam que João tem criptomoeda suficiente em sua conta e que a conta receptora de Maria é válida;
3. Uma vez que verificam a transação, transmitem a transação para todos os computadores próximos a João e Maria na rede;
4. Por sua vez, esses computadores voltam a confirmar a transação e a transmitem, o que gera um efeito cascata até que a transação seja adicionada a todos os livros da rede **Peer-to-peer**.

Os computadores na rede **Peer-to-peer** são tanto usuários como verificadores. As transações do Blockchain podem não ter custo, e o efeito cascata de verificar transações significa que uma transação pode ser processada em minutos ou horas, em vez de dias. Com base nestes benefícios, muitas vezes, o Blockchain é apresentado como o início do fim das instituições financeiras tradicionais (Hollins, 2018) [26].

Bitcoin foi criado como um meio de proporcionar uma nova forma de realizar pagamentos e transações online, de forma descentralizada e não operada pelo governo. Apesar de não ser a única criptomoeda existente ², durante os últimos anos tem mantido sistematicamente o seu domínio como a criptomoeda mais popular.

Uma das razões pela qual o Bitcoin tem mantido sua liderança é porque implementou o primeiro blockchain da história e o interesse inicial nesta tecnologia fez com que esta criptomoeda tivesse a maior base de usuários além de uma seleção de desenvolvedores experientes para construir o sistema.

²Apodaca, 2017 [4] referencia as demais criptomoedas como Altcoins. Ele diz que elas tentam preencher algumas lacunas do Bitcoin. Ele cita as seguintes Altcoins: Ethereum, Litecoin, Dash, Monero e Zcash, Peercoin e Namecoin

A longa história do Bitcoin também lhe dá alguma legitimidade. No decorrer de uma década, o Bitcoin tem enfrentado e superado muitos obstáculos e desafios técnicos e é seu comprovado histórico de segurança que o torna a criptomoeda mais segura para os iniciantes de acordo com Hollins, 2018 [26].

Funcionamento do Blockchain

Em termos simples, o blockchain utiliza uma combinação de criptografia e um livro de contabilidade pública (ledger) para criar confiança entre as partes, mantendo a privacidade. A compreensão da mecânica de como isso funciona é um pouco mais difícil, mas, a fim de entender a tecnologia blockchain, é preciso aprofundar alguns detalhes técnicos. Os fundamentos do blockchain são sugeridos por seu próprio nome: "block chain" ou cadeia de blocos.

Definição 1.3 (Bloco): o bloco é uma lista de todas as transações de um determinado período que contém toda a informação processada na rede nos últimos minutos.

Definição 1.4 (Cadeia): É uma união de blocos onde cada bloco está associado ao bloco anterior mediante o uso de algoritmos criptográficos (funções hash).

A rede cria um bloco de cada vez e os algoritmos criptográficos são quebra-cabeças complexos que os computadores devem calcular e, várias vezes, computadores velozes demoram algum tempo para resolver. Uma vez resolvido, é criado o bloco usando mais algoritmos de criptografia, o que é praticamente impossível de modificar segundo Hollins, 2018 [26]. A cadeia se torna cada vez mais longa com o tempo. A Figura 3 ilustra uma cadeia de blocos de hash:

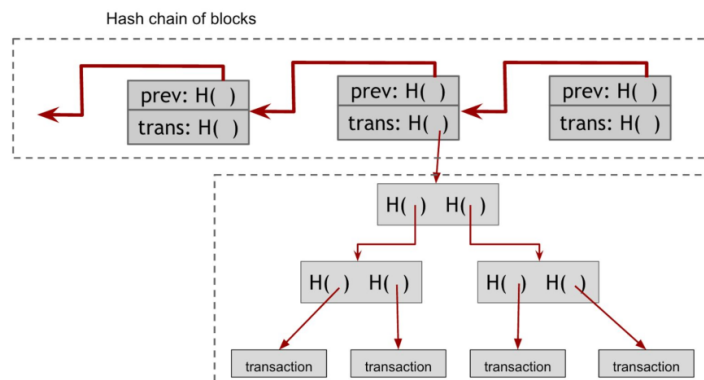


Figura 3: Cadeia de blocos de hash. Disponível em [37]

A parte fundamental da Blockchain é o **ledger**, lugar onde se armazena as informações sobre as contas da rede. O livro dentro do Blockchain é o que substitui o livro de contabilidade em um banco ou outra instituição e, para uma criptomoeda, este livro de contabilidade consiste em números de conta, transações e saldos.

O **ledger** do blockchain é distribuído em toda a rede **Peer-to-peer**, por isso que cada nó (computador) da rede mantém sua própria cópia do livro e atualiza quando alguém apresenta uma nova transação. Este "livro de contabilidade compartilhado" é o que permite a descentralização.

DATE	ITEMS	Debits	DATE	ITEMS	CREDITS
Nov 12	Cash from E. H. Allen	169.70	Nov 13	Drift to Barton	59.75
13	" " " " " "	173.50	" "	" "	77.50
		343.20			33.20
13	Bal.	27.74	20	Drift to Barton	166.60
20	Cash from Payroll	15.20	20	Bal.	21.60
		42.64			42.64
20	Bal.	26.60	27	Drift to Barton	154.35
27	Cash from Payroll	100.10	27	Bal.	21.10
		36.60			36.60
27	Bal.	21.10	Dec 4	Drift to Barton	146.20
Dec 4	Cash from Payroll	12.60	" "	" "	1.10
" "	" " " " " "	20			36.30
		36.30			21.60
Dec 4	Bal.	21.60	11	Drift to Barton	123.60
11	Cash from Payroll	11.70			33.30
		33.30			21.60
11	Bal.	21.60	18	Drift to Barton	121.05
18	Cash from Payroll	20	18	Bal.	31.60
		233.40	18	Bal.	150.75
		20.75	25	Drift to Barton	181.30

TxHash	Block	Age	From	To	Value
0x978f4b1cab4e73...	6057332	36 secs ago	0x78af16ac3023...	0x6fc220c5c21...	1.995 Ether
0x3aa89473d21e06...	6057332	36 secs ago	0x78af16ac3023...	0xa6544093b3fcb...	1.14926148 Eth
0x27c37c5d0c5c3...	6057330	46 secs ago	0x78af16ac3023...	0xf8b12ee860c923...	11.995 Ether
0x890c604b8b8b7f...	6057328	1 min ago	0x42fed1874c29df...	0x78af16ac3023...	13.851225 Ether
0x2cd90dc08f1a7...	6057327	1 min ago	0x78af16ac3023...	0x667983ec439c39...	2.3979269 Ether
0x82d445757dc2b7...	6057326	2 mins ago	0x78af16ac3023...	0x00a8800a6fc7a...	1.99500013 Eth
0x729c201e0677b...	6057320	2 mins ago	0x78af16ac3023...	0x25102ee4d04dc...	12 Ether
0x9037693111713...	6057318	3 mins ago	0x78af16ac3023...	0x7c95776c25877b...	14.49 Ether
0x959502c84a718...	6057318	3 mins ago	0x78af16ac3023...	0xac132655749ee5...	5.999997 Ether
0xd726830bc28e...	6057317	3 mins ago	0x708bc45430671...	0x78af16ac3023...	5.62075721 Eth
0xd2e6e728472c0...	6057314	3 mins ago	0x78af16ac3023...	0x74497711007b...	23.9572689 Eth
0x27e2ac90319c...	6057312	3 mins ago	0x0e0a0296c26c25...	0x78af16ac3023...	2.28458171 Eth
0xd508a2fe18da52...	6057312	3 mins ago	0x35cead474e4311...	0x78af16ac3023...	1.15364088 Eth
0x091a774934e0a...	6057312	3 mins ago	0x78af16ac3023...	0xd77748e2e49f521...	2.00068862 Eth
0x1d8f7c573b398...	6057312	3 mins ago	0x78af16ac3023...	0x1e086ed1442df...	0.31924 Ether

Figura 4: Semelhança entre a página de um livro caixa do início do Século XX e um bloco na blockchain. Disponível em [12]

Em vez do banco manter uma cópia oficial das transações, todo mundo pode manter a sua própria cópia do livro de contabilidade, as transações são verificadas por consenso. Ou seja, se alguém tentasse enganar o sistema e, de alguma forma conseguisse modificar o livro de contabilidade que está no computador, aumentando o saldo de bitcoins para, em seguida, fazer uma transação, também deveria "hackear" 51% dos computadores da rede Bitcoin para modificar os livros de contabilidade e, desta forma, a maioria dos usuários da rede estaria de acordo validando a transação.

Isto seria uma missão complexa, mas se imaginar que, a cada 10 minutos o livro de contabilidade de todo mundo sofre alteração com novos códigos de criptografia, na prática, teria menos de 10 minutos para hackear o livro de contabilidade e 51% da rede, o que é impossível segundo Hollins, 2018 [26].

Cabe destacar que qualquer um pode ser parte da rede Bitcoin e ter uma cópia do blockchain tão somente baixando-o para um computador.

Definição 1.5 (Transação com bitcoins): Para uma transação com bitcoins são requeridas três partes de uma informação:

- **Entrada:** O remetente indica a rede onde conseguiu seu bitcoin;
- **Quantidade:** Informar a quantidade de bitcoins a ser enviado;
- **Endereço de Saída:** Endereço de bitcoin do destinatário em que se deve depositar o bitcoin.

No Capítulo Blockchains será aprofundado como se dão as transações de bitcoins.

Definição 1.6 (Mineração de bitcoins): A mineração de bitcoins é o processo de investir na capacidade computacional para processar transações, garantir a segurança da rede e fazer com que todos os participantes da rede **Peer-to-peer** estejam sincronizados. Pode-se dizer que o blockchain é centro de dados do Bitcoin, embora este centro de dados tenha sido projetado para ser totalmente descentralizado, com mineradores operando em todos os países e sem que ninguém tenha o controle absoluto sobre a rede.

A mineração de bitcoins oferece uma recompensa para os mineradores em troca de seus serviços, que são necessários para que a rede funcione de forma segura. O processo é chamado de "mineração" como analogia a mineração do ouro, já que também é o mecanismo temporário utilizado para emitir novos bitcoins, até chegar ao limite de 21 milhões de bitcoins em circulação (Nakamoto, 2009) [36].

Cabe destacar que a emissão finita de moedas é uma estratégia para evitar a inflação e, desta maneira, evitar a perda de valor com o passar do tempo. O bitcoin não é real. Não há bitcoins físicos e não há bitcoins em um disco rígido em parte alguma. Não se pode apontar para um objeto físico, um arquivo digital, ou um pedaço de código e dizer: "Isso é um bitcoin." Em vez disso, toda a rede Bitcoin é apenas uma série de registros de transações e cada transação da história do Bitcoin está no ledger (livro-razão) do blockchain, por isso que se alguém quiser provar que tem 20 bitcoins, a única maneira de fazer isso é apontando as transações em que recebeu esses 20 bitcoins.

Neste ponto, alguém pode pensar: "mas os Bitcoins têm que sair originalmente de algum lugar." É importante lembrar que os mineradores criam os bitcoins em troca de recompensas, o que continuam fazendo até atingir o limite de 21 milhões de bitcoins.

Portanto, o conceito fundamental a ser compreendido aqui é o seguinte: O histórico de transações é a moeda.

Alguns críticos argumentam que o Bitcoin é um esquema Ponzi ³. Franco, 2014 [23] afirma que não é. Em um esquema Ponzi existe um operador central que paga retornos aos investidores atuais com novas entradas de capital.

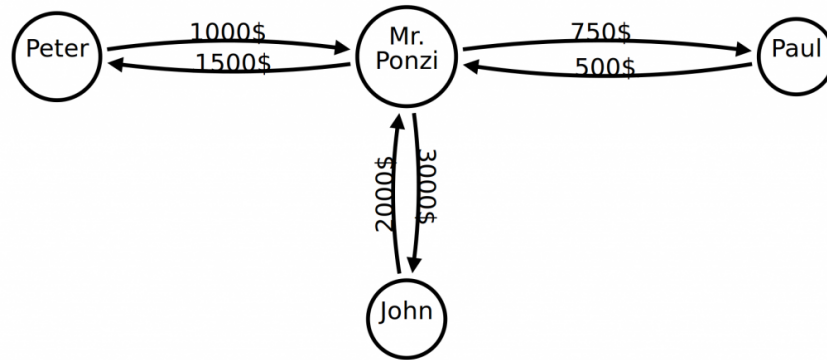


Figura 5: Grafo de um Esquema Ponzi, Disponível em [32]

Em primeiro lugar, no Bitcoin não existe um operador central que possa lucrar com a desalocação de fundos. Em segundo lugar, não há mecanismo para desviar fundos de novos investimentos para pagar retorno. Os únicos fundos reconhecidos no protocolo Bitcoin são bitcoins, a moeda. Transferências de bitcoins são iniciadas pelos usuários à sua vontade: o protocolo não pode desviar fundos de um usuário para outro. Em terceiro lugar, um novo investimento em Bitcoin é sempre correspondido com um desinvestimento. Os investidores que investem dinheiro em bitcoins geralmente operam através de troca onde eles compram os bitcoins de outro investidor que está vendendo seus investimentos. Simplesmente não há novos investimentos para bitcoins: a quantidade de moeda que entrou em bitcoins corresponde exatamente a quantidade que saiu de bitcoins.

³A pirâmide financeira se assemelha a um esquema Ponzi: ela também se mantém enquanto novos usuários entrarem no esquema

Criptografia

Neste capítulo será feita uma contextualização da criptografia, passando pelos seus precursores. Falaremos dos pesquisadores da criptografia por curvas elípticas. Até chegarmos ao autor do artigo fundamental do Bitcoin. Será introduzido o conceito de criptografia de chave pública da qual os modelos mais utilizados são a criptografia simétrica e assimétrica. Falaremos também da criptografia assimétrica no blockchain.

Em 1941, Claude Elwood Shannon (Figura 6) que viveu entre 1916 a 2001, entrou para a equipe da AT&T Bell Telephones no estado de Nova Jérsei, EUA. Ele fora contratado como pesquisador matemático, tendo permanecido naquela instituição até o ano de 1972. Em 1948, Shannon publicou artigo intitulado “A Mathematical Theory of Communication”, Shannon, 1948 [49] que é considerado como artigo fundador da teoria da troca de informações.

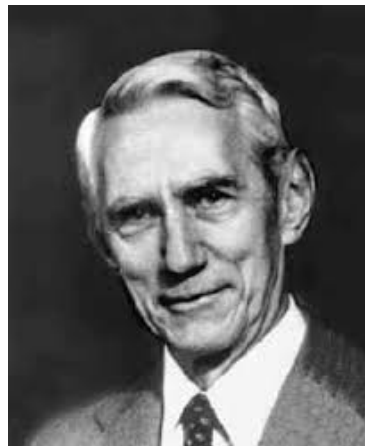


Figura 6: Claude Elwood Shannon. Disponível em [27]

Em outubro de 1949, Shannon publicou outro artigo intitulado “Communication Theory of Secrecy Systems” [48] o qual é geralmente designado como transformador da criptografia de arte para ciência. As contribuições de Shannon são consideradas como as precursoras da moderna criptografia. Horst Feistel (1915 – 1990) (Figura 7) foi um criptógrafo nascido na Alemanha que imigrou para os Estados Unidos e em 1968 juntou-se ao centro de pesquisa IBM T.J. Watson em Yorktown Heights, Nova Iorque.

De lá, ele liderou o laboratório o qual foi considerado o único grupo de pesquisa não-governamental de criptografia nos Estados Unidos e provavelmente no mundo à época. Feistel juntou-se à IBM num momento propício a tempo de resolver o problema de segurança de transações eletrônicas em caixas eletrônicos para o sistema de caixas eletrônicos do banco Lloyd's.

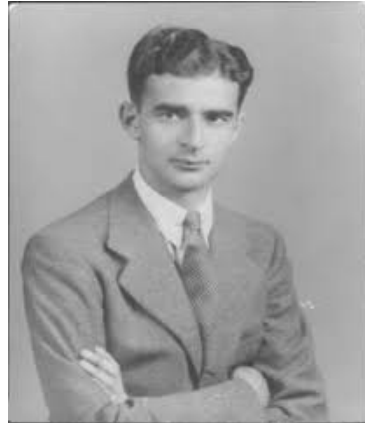


Figura 7: Horst Feistel. Disponível em [6]

Feistel desenvolveu um sistema criptográfico de cifras em blocos para proteger os dados de um sistema de caixa eletrônico remoto. Isto não apenas representou um paradigma para os sistemas de encriptação mas também principalmente trouxe a encriptação de uma ciência militar pouco conhecida para o cotidiano à época estimulando assim a pesquisa em criptografia e a competição por criar sistemas criptográficos.

Feistel tornou conhecido o Data Encryption Standard (DES) o qual foi largamente utilizada por mais de 30 anos. O padrão de encriptação de dados (DES) utiliza uma técnica de encriptação simétrica pela qual a mesma chave é aplicada nos dois lados (por quem envia e por quem recebe), conhecida como a chave secreta. Em 2001, o DES fora substituído pelo AES (Advanced Encryption Standard) ou padrão de encriptação avançado. Outros famosos algoritmos de encriptação são: Triple-DES, CDMF (Commercial Data Masking Facility), IDEA (International Data Encryption Algorithm), RC2, RC4, RC5, RC6, MARS, Blowfish e Rijndael. Em 1973, Feistel também publicou um artigo pioneiro na revista Scientific American intitulado “Cryptography and Computer Privacy” [19] ou Criptografia e privacidade computacional (tradução do autor) que é considerado a primeira informação para o público em geral sobre criptografia e privacidade.

“... seria surpreendente se a criptografia, o meio tradicional de assegurar confidencialidade na comunicação, pudesse não fornecer a privacidade para a comunidade de usuários de dados bancários.” - Feistel, 1973 (tradução livre)

Depois do trabalho de Feistel seu artigo na revista *Scientific American*, muitos pesquisadores se tornaram interessados no campo de pesquisa da criptografia e alguns deles decidiram investigar em como resolver o problema fundamental: “como mandar uma chave simétrica para o outro lado através de um canal não confiável?”

No fim de 1968 iniciou-se um trabalho onde Martin Hellman (Figura 8) encontrou membros do programa de pesquisas criptográficas, incluindo Horst Feistel e Alan G. Konheim (Figura 10), outro criptógrafo.

“Eu me lembro de várias conversas com Feistel. Naquele ano as minhas interações com o Horst foram de grande representatividade em minha mudança posterior para a criptografia” - Hellman, 1968

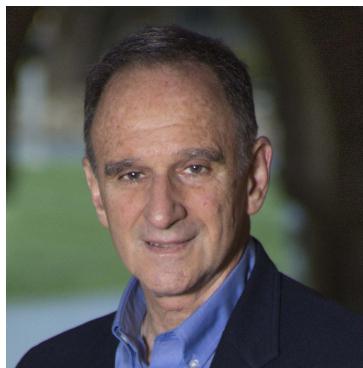


Figura 8: Martin Hellman. Disponível em [39]

Uma vez tendo almoçado com Feistel, que não apenas conversou sobre sistemas clássicos criptográficos como também sobre problemas que pareciam sem solução mas que na verdade podiam ser resolvidos. Então Hellman decidiu resolver o problema da troca de chaves. Na sequência ele foi para o MIT e então para a faculdade de engenharia elétrica de Stanford.

Whitfield Diffie (Figura 9) era um matemático da universidade de Stanford que começou a estudar criptografia. No verão de 1974, Diffie foi para a IBM e conversou com Alan Konheim sobre os desafios no campo da criptografia. Konheim disse algo muito importante para Diffie:

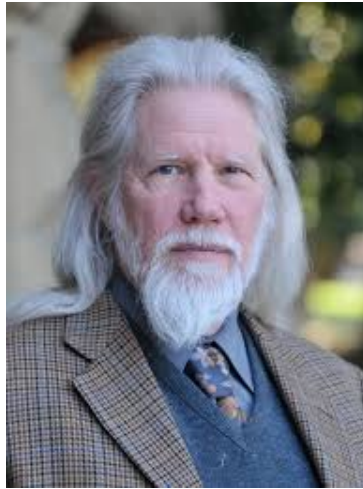


Figura 9: Whitfield Diffie. Disponível em [22]

“Um velho amigo meu, chamado Martin Hellman, trabalhou aqui até um tempo atrás, e agora ele se encontra em Stanford. E duas pessoas podem trabalhar num problema muito melhor que uma e então quando você voltar para Stanford, você deve procurar por ele.” - Alan Konheim



Figura 10: Alan Konheim. Disponível em [58]

No final de 74, dada a sugestão de Konheim, Diffie visitou Hellman em Stanford. De uma reunião de meia hora planejada para o início da tarde entre Diffie e Hellman acabou se tornando uma longa sessão de discussão seguida do jantar até a noite, sendo apenas o começo das discussões conjuntas. Nas palavras de Hellman:

“Foi uma epifania leve, encontrar uma alma gêmea intelectual.” Martin Hellman

A utilização de curvas elípticas em criptografia foi proposta de modo independente [30] [35] por Neal Koblitz (Figura 11) e Victor Miller (Figura 12) em 1985, servindo como uma eficiente forma de implementação de um sistema de chave pública. De acordo com seus desenvolvedores, a criptografia de curvas elípticas pode ser mais rápida e utilizar chaves mais curtas para proporcionar o mesmo nível de segurança de métodos mais tradicionais, como o RSA. O protocolo RSA⁴ foi criado na década de setenta pelos pesquisadores Ronald Rivest, Adi Shamir e Leonard Adleman (as iniciais dão nome ao algoritmo).



Figura 11: Neal Koblitz. Disponível em [40]



Figura 12: Victor Miller. Disponível em [21]

Torres, 2007 [56] afirma que Koblitz e Miller utilizaram as curvas elípticas na criptografia como uma forma implementação de um sistema de chave pública em algumas aplicações já existentes.

⁴para mais detalhes sobre o protocolo RSA verificar a dissertação do PROFMAT Sistema de Criptografia RSA de Corrêa, 2013 [38]

O surgimento do Bitcoin é acreditado à publicação de um artigo em 2008 intitulado "Bitcoin: A Peer-to-Peer Electronic Cash System" [36] ("Bitcoin: Um Sistema de Dinheiro Eletrônico Ponto-a-Ponto" em tradução livre), escrito por um autor desconhecido (ou um grupo de pesquisadores) sob o pseudônimo de Satoshi Nakamoto. Nakamoto combinou várias das invenções anteriores tais como b-money [13] e HashCash [5] para criar um sistema de dinheiro eletrônico completamente descentralizado que não dependesse de uma autoridade central para a emissão de moeda ou para a liquidação e validação de transações.

A principal inovação foi usar um sistema de computação distribuído (chamado algoritmo de "proof-of-work"⁵ ou "prova de trabalho") para conduzir uma "eleição" global a cada 10 minutos, permitindo à rede descentralizada chegar em um consenso sobre o estado das transações. Isto resolve de forma elegante o problema de gasto duplo, onde uma única unidade de moeda poderia ser gasta duas vezes.

Antes do Bitcoin, o problema de gasto duplo era uma fraqueza do dinheiro digital, e sua solução envolvia a transmissão e verificação de todas as transações através de uma entidade central. A rede Bitcoin surgiu em 2009, baseada em uma implementação de referência publicada por Nakamoto [36] e desde então revisada por outros programadores. A computação distribuída que proporciona segurança e robustez ao Bitcoin cresceu exponencialmente e agora excede a capacidade combinada de processamento dos principais supercomputadores do mundo. Em 2014, o valor de mercado de bitcoins era estimado entre 5 e 10 bilhões de dólares americanos, dependendo da taxa de câmbio entre o bitcoin e o dólar. A maior transação processada até 2014 pela rede foi de US\$ 150 milhões, transmitida instantaneamente e processada sem nenhuma taxa.

Satoshi Nakamoto afastou-se do público em abril de 2011, deixando a responsabilidade pelo desenvolvimento do código e da rede nas mãos de um grupo de voluntários. A identidade da pessoa ou pessoas por trás do Bitcoin ainda é desconhecida. No entanto, nem Satoshi Nakamoto nem qualquer outra pessoa exerce controle sobre o sistema Bitcoin, que opera baseado em princípios matemáticos totalmente transparentes. A invenção em si é revolucionária e já criou um novo campo de estudos nas áreas da computação distribuída, economia e econometria.

⁵O algoritmo proof-of-work é apresentado matematicamente no Capítulo Blockchains

A criptografia é frequentemente considerada complicada e de difícil entendimento. Por isso, aqui focamos em fazer uma introdução a esse assunto, discutindo o suficiente para entender o conceito de segurança do blockchain.

Segundo Correia, 2013 [15] atualmente a criptografia consiste em uma série de fórmulas matemáticas, em que se utiliza um segredo (chamado de chave) para cifrar e decifrar as mensagens. Este segredo pode ser o mesmo para as duas operações (criptografia simétrica) ou pode haver segredos diferentes, um para cifrá-la e outro para decifrá-la (criptografia assimétrica).

Criptografia de chave pública

Criptografia é área de estudo que engloba as diversas estratégias de encriptação e decriptação de informações, sendo encriptação a transformação de um texto claro (texto original) em texto cifrado (encriptado), e decriptação a recuperação do texto claro (texto original) a partir do texto cifrado.

No modelo de criptografia dita simétrica, utiliza-se apenas uma chave, chamada de chave secreta. A chave secreta é fornecida, juntamente com o texto claro (texto original), a um algoritmo de encriptação, que gerará como saída um arquivo encriptado (que será diferente conforme a utilização de chaves secretas também diferentes). Chama-se algoritmo de decriptação o código computacional a ser utilizado para reverter o processo de encriptação, ou seja, a obtenção do texto claro (texto original) a partir do texto encriptado, desde que fornecida a chave secreta utilizada no processo de encriptação.

A criptografia simétrica é um método em que uma chave idêntica é usada tanto para codificar quanto para decodificar os dados. A segurança na utilização da criptografia simétrica depende de algoritmos fortes (que não sejam vulneráveis a criptoanálise) e que as chaves secretas envolvidas nos processos de encriptação e decriptação sejam armazenadas e compartilhadas pelos usuários autorizados de maneira confidencial. Vide Figura 13 a seguir:

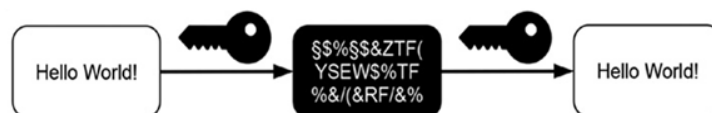


Figura 13: Ilustração esquemática da criptografia simétrica. Adaptado de [16]

Por outro lado, a criptografia de chave pública, foco desse tópico, é assimétrica, pois utiliza duas chaves distintas, porém relacionadas (chave pública e chave privada), o que a distingue da criptografia simétrica, que usa apenas uma mesma chave, chamada de chave secreta, para suas operações de encriptar e decriptar. A utilização de chaves separadas tem vantagens em relação à utilização de uma única chave, principalmente no tocante aos princípios da confidencialidade, da autenticidade e de estratégias de armazenamento e compartilhamento de chaves.

A criptografia assimétrica sempre utiliza duas chaves complementares. Todavia o texto cifrado criado com uma dessas chaves só pode ser descriptografado com a outra chave, e vice-versa.

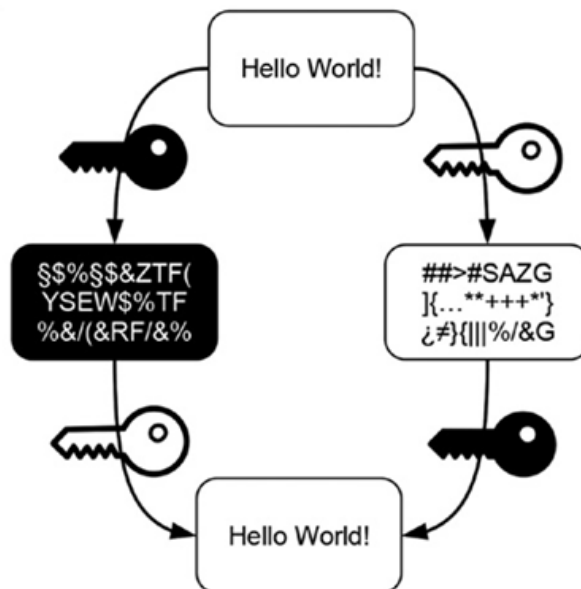


Figura 14: Ilustração esquemática da criptografia assimétrica. Adaptado de [16]

Os primeiros passos a serem executados em qualquer aplicação de criptografia assimétrica são: criar um par de chaves complementares usando um software de criptografia; dar o nome de chave pública a uma delas; dar o nome de chave privada à outra; manter a chave privada consigo e dar a sua chave pública a todos os demais. Dois passos principais são necessários para usar a criptografia assimétrica na vida real: criação e distribuição das chaves e uso das chaves. A seguir estão listadas algumas terminologias relacionadas à criptografia de chave pública:

- **Chaves assimétricas:** são chaves que se complementam em operações relativas à codificação (encriptação) e decodificação (decriptação) de informações, e também à criação e validação de assinaturas digitais;

- **Certificado de chave pública:** é um documento digital, assinado pela chave privada de uma Autoridade de Certificação, que associa o nome de um usuário a uma chave pública. O certificado sinaliza que o usuário possui domínio sobre a chave privada relacionada;
- **Algoritmo criptográfico de chave pública (assimétrica):** é o código computacional que utiliza as chaves pública e privada em operações relacionadas à criptografia, com a propriedade de ser inexequível computacionalmente a obtenção da chave privada a partir da chave pública;
- **Infraestrutura de chave pública (PKI):** é uma infraestrutura composta por processos, políticas, serviços de rede, entre outros itens, utilizada para a administração de certificados digitais e chaves públicas e privadas relacionadas.

Em se tratando de confidencialidade, a criptografia de chave pública (assimétrica) funciona com a encriptação da informação usando-se a chave pública do usuário que receberá a informação. Esse mesmo usuário, assim que recebê-la, fará uso de sua chave privada para decriptar a informação para dela fazer uso. Por exemplo: se Ana deseja enviar uma mensagem sigilosa (confidencial) a Bruno, Ana codificará (encriptará) a mensagem usando a chave pública de Bruno, e Bruno, por sua vez, usará sua chave privada para decriptar (decodificar) a mesma mensagem.

A autenticação é outro princípio da Segurança da Informação que é garantido pela criptografia de chave pública. Se um usuário deseja enviar conteúdo assinado digitalmente a outrem, deverá, para tanto, utilizar-se de sua chave privada aplicada à informação original. O usuário de destino deverá, por sua vez, utilizar a chave pública do usuário de origem para validar o arquivo. Ressalve-se que somente o usuário de origem poderia ter gerado o arquivo assinado digitalmente, haja vista a utilização de sua chave privada. Esse processo respeita o princípio da autenticidade e também o da irretratabilidade (o usuário que gerou o arquivo assinado digitalmente não poderá negar a autoria em relação ao mesmo).

Como já aludido, a criptografia assimétrica colabora com a segurança no que diz respeito à disponibilidade das chaves: a chave pública fica disponível a qualquer interessado, e a chave privada fica disponível apenas ao seu proprietário. Em contrapartida, a criptografia simétrica usa apenas uma chave, sendo que esta tem que ser de conhecimento tanto do emissor como do receptor. E ainda, por conta disso, essa chave deverá ser,

em algum momento, compartilhada entre esses usuários. Porém, o compartilhamento da chave simétrica entre usuários gera insegurança, uma vez que um terceiro não autorizado pode interceptá-la durante seu envio, trânsito ou recebimento. Um dos maiores problemas relacionados à utilização de chaves criptográficas diz respeito ao tráfego de informação em um canal de comunicação inseguro, como a Internet.

Por conta disso, algoritmos criptográficos foram criados para que uma sessão de comunicação segura possa ser gerada e utilizada. Um desses algoritmos é o Diffie-Hellman, cujo título é oriundo dos nomes de seus criadores (Whitfield Diffie [22] e Martin Hellman [39]), com publicação original sobre sua utilização remontando ao ano de 1976. O algoritmo (de troca de chaves) Diffie-Hellman permite que dois ou mais usuários estabeleçam uma sessão de comunicação criptografada em um canal inseguro por meio da criação de uma chave secreta de sessão (em uma espécie de criptografia simétrica aplicada à segurança do meio de comunicação). O algoritmo Diffie-Hellman é utilizado apenas para troca de chave, ou seja, não permite a criptografia de informações e nem assinatura digital.

A Criptografia de Curvas Elípticas é um algoritmo que permite a troca de chave e também as operações de criptografia de encriptação e decriptação de informações, além de permitir a utilização de assinatura digital. Baseia-se na estrutura algébrica de curvas elípticas sobre corpos finitos por meio da equação $y^2 = x^3 + ax + b$ com $4a^3 + 27b^2 \neq 0$. Uma breve introdução sobre curvas elípticas é feita no Apêndice .

Criptografia assimétrica no blockchain

A criptografia assimétrica, além das funções de hash, é intensamente utilizada pelo blockchain. No blockchain a criptografia assimétrica é fundamental para identificar usuários e proteger suas propriedades. Criptografia assimétrica é utilizada no blockchain para alcançar dois objetivos:

- **identificar contas:** as contas de usuários são chaves de criptografia públicas;
- **autorizar transações:** o proprietário da conta que transfere a posse cria um texto cifrado com a chave privada correspondente. Esse texto cifrado pode ser conferido utilizando a chave pública.

Funções Hash

Neste capítulo iremos abordar o conceito de funções hash. As funções hash são parte fundamental do arcabouço técnico de formação do blockchain. Iremos primeiramente efetuar uma introdução das funções hash elementares, em seguida funções hash de aritmética modular para finalmente apresentar as funções hash utilizadas no protocolo Bitcoin: RIPEMD-160 e SHA-256.

Funções Hash

Assim como as impressões digitais são capazes de identificar unicamente os seres humanos e são utilizadas para investigar crimes, identificar infratores etc. seu equivalente digital é o que denominamos de funções hash, que é um conceito para identificar dados. No sistema ponto-a-ponto distribuído, lidaremos com um volume enorme de dados de transação - como resultado - será necessário identificá-los unicamente e compará-los de maneira mais rápida e fácil possível. O blockchain utiliza o conceito de hashing criptográfico que serve para identificar os dados de transação e possivelmente, qualquer tipo de dados unicamente, por meio de suas impressões digitais. As funções hash são algoritmos que transformam qualquer tipo de dado em um número de tamanho fixo, independentemente do tamanho dos dados de entrada.

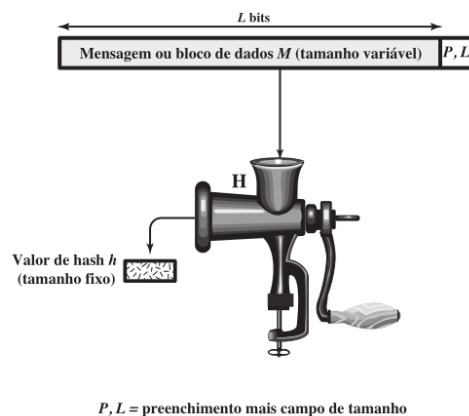


Figura 15: Ilustração da função hash - Adaptado de [53]

Um grupo importante de funções hash se chama funções de hash criptográficas, que criam impressões digitais para qualquer tipo de dado. As funções de hash criptográficas têm as seguintes propriedades:

- **fornecem valores de hash para qualquer tipo de dado rapidamente:** essa propriedade é uma combinação de duas propriedades - primeiro, a função de hash é capaz de calcular valores de hash para todos os tipos de dados, segundo, ela faz seus cálculos rapidamente;
- **são determinísticas:** a função de hash produz valores de hash idênticos para dados de entrada idênticos;
- **são pseudoaleatórias:** ser pseudoaleatória significa que o valor de hash devolvido por uma função de hash muda de forma imprevisível quando os dados de entrada são alterados;
- **são funções unidirecionais:** uma função unidirecional não possibilita que seus valores sejam rastreados com base na saída;
- **são resistentes à colisão:** dizemos que uma função de hash é resistente à colisão se for muito difícil encontrar duas ou mais porções distintas de dados para as quais ela gere valor idêntico de hash. Uma colisão de hash é o equivalente digital de ter duas pessoas com impressões digitais idênticas.

A seguir o leitor é convidado a experimentar seu próprio cálculo de hash a partir do seguinte endereço eletrônico: <http://www.blockchain-basics.com/HashFunctions.html> Ao abrir a página web em seu navegador de internet, você verá uma caixa de entrada e outra de saída, conforme mostra a Figura 16 a seguir:

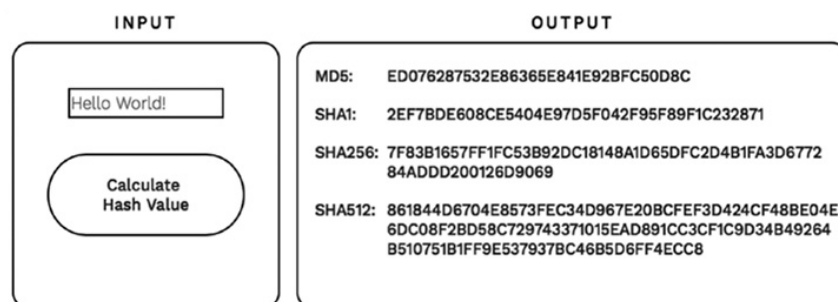


Figura 16: Exemplo cálculo de hash. Disponível em [16]

Digite o texto Hello World! na caixa de entrada à esquerda e clique no botão cujo rótulo é “Calculate Hash Value” (Calcular o valor de hash), localizado abaixo da caixa de texto. Certifique-se de ter digitado exatamente Hello World! na caixa de entrada; caso contrário, um resultado diferente daquele mostrado na Figura 16 será obtido.

Ao clicar no botão, como resultado, a caixa de saída à direita apresentará o valor de hash do texto de entrada, calculado com quatro funções de hash distintas. Os valores de hash, em geral, são considerados como números de hash, pois usam não só os dígitos de 0 a 9, mas também as letras de A a F, que representam os valores de 10 a 15, para expressar valores numéricos. Esses números são chamados de hexadecimais. Observe que os valores de hash diferem por causa dos diferentes detalhes de implementação das funções de hash que os geram.

Valores de hash criptográficos são bem longos e, desse modo, para o olho humano, são difíceis de ler ou comparar. Entretanto, durante este passo, compararemos diferentes maneiras de gerar dados de hashing, e essa tarefa envolverá a leitura e a comparação de seus valores. Será usada uma versão abreviada do valor de hash criptográfico SHA256 no restante deste passo. É possível reproduzir todos os valores de hash usando a ferramenta disponibilizada no seguinte endereço: www.blockchain-basics.com/Hashing.html

Ao acessar esse site com o navegador de internet, é visto uma caixa de entrada para textos simples e um botão com uma seta que aponta para uma caixa de saída, conforme mostra a Figura 17 a seguir:

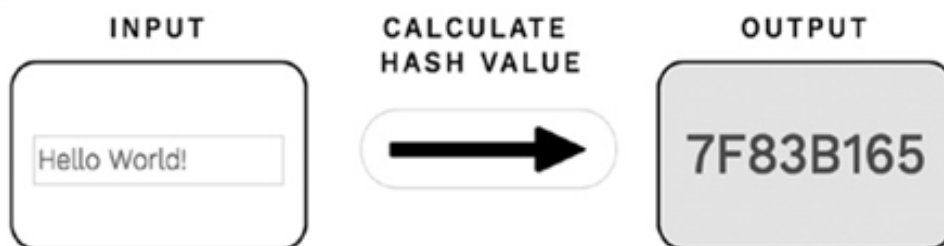


Figura 17: Cálculo de hash abreviado. Disponível em [16]

Ao clicar no botão com a seta, a caixa de saída apresentará o valor de hash abreviado para o texto fornecido na caixa de entrada.

Usando aritmética modular em função hash

A seguir faremos um exemplo em que usamos a aritmética modular para apresentar uma função hash elementar em uma situação do cotidiano que descreveremos abaixo. Este exemplo foi extraído e adaptado de Rosen, 2011 [44].

O computador central de uma companhia mantém registros para cada um de seus clientes. Como os locais de memória podem ser atribuídos para que os registros dos clientes possam ser recuperados rapidamente? A solução para esse problema é usar uma função de hash adequadamente escolhida.

Registros são identificados usando uma chave, que identifica de forma exclusiva os registros de cada cliente. Por exemplo, registros de clientes são frequentemente identificados usando o número do Registro Geral (RG) do cliente como a chave. Uma função de hash h atribui o local de memória $h(k)$ ao registro que tem k como sua chave. Na prática, muitas funções de hash diferentes são usadas.

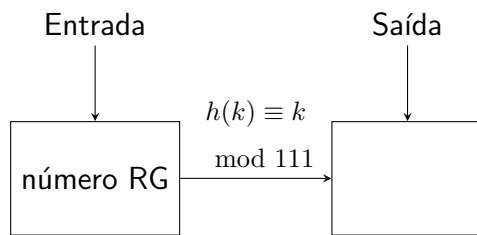
Uma das mais comuns é a função de hash modular: $h(k) = k \bmod m$ onde m é o número de locais de memória disponíveis.

Lembremos que as funções de hash devem ser facilmente avaliadas para que os arquivos possam ser localizados rapidamente. A função de hash $h(k) = k \bmod m$ atende a este requisito; para encontrar $h(k)$, necessita-se computar o resto da divisão quando k é dividido por m . Além disso, as funções de hash devem ser tais que, todos os locais de memória são possíveis. A função $h(k) = k \bmod m$ também satisfaz esta propriedade.

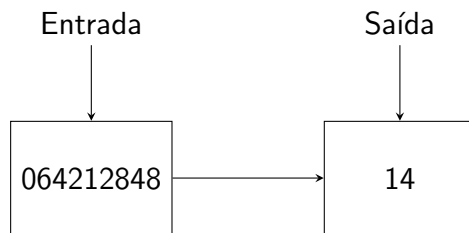
A seguir, três exemplos extraídos de Rosen, 2011 [44]:

Exemplo 3.1: Encontre os locais de memória atribuídos pela função de hash $h(k) \equiv k \bmod 111$ aos registros de clientes com os números de RG 064212848 e 037149212.

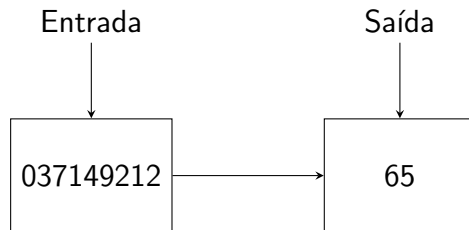
Solução:



O registro do cliente com o número de RG 064212848 é atribuído a localização de memória 14, visto que $h(064212848) \equiv 064212848 \pmod{111} \implies 14 \equiv h(k)$



Da mesma forma, uma vez que $h(037149212) \equiv 037149212 \pmod{111} \equiv 65$, o registro do cliente com número de RG 037149212 é atribuído à memória de localização 65.



Como uma função de hash não é um-para-um, mais de um arquivo pode ser atribuído a um local de memória. Quando isto acontece, dizemos que ocorre uma **colisão**.

Uma maneira de resolver a colisão é atribuir o primeiro local livre seguindo a localização da memória ocupada atribuída pela função de hash.

Exemplo 3.2: Depois de fazer as atribuições de registros para as posições no Exemplo 4.1, atribui-se posição para o registro do cliente com o número de RG 107405723.

Solução: Verifica-se que a função de hash $h(k) \equiv k \pmod{111}$ mapeia o RG número 107405723 para a localização 14, tendo em vista que: $h(107405723) \equiv 107405723 \pmod{111} \equiv 14$. Caso a posição já esteja ocupada, localizar-se-á à primeira posição livre. No entanto, este local já está ocupado (pelo arquivo do cliente com o RG número 064212848). Mas, tendo em vista que a localização da posição 15, o primeiro local após a posição 14, é livre, atribuiu-se o registro do cliente com o número de RG 107405723 para este local (15).

Exemplo 3.3: Quais posições são atribuídos pela função de hash $h(k) \equiv k \pmod{97}$ aos registros de clientes com os seguintes números de RG?

- (a) 034567981
- (b) 183211232
- (c) 220195744
- (d) 987255335

Solução: Solicita-se simplesmente calcular $k \pmod{97}$ para cada valor de k . Faz-se isso dividindo os RGs dados por 97 e pegando os restos, que pode ser encontrado multiplicando o resto decimal por 97, ou subtraindo 97 vezes o quociente de k .

- (a) $034567981 \pmod{97} \equiv 91$
- (b) $183211232 \pmod{97} \equiv 57$
- (c) $220195744 \pmod{97} \equiv 21$
- (d) $987255335 \pmod{97} \equiv 5$

Outra maneira de resolver colisões em hash é usar duplo hashing. Utiliza-se uma função de hash inicial

$$h(k) = k \pmod{p}$$

onde p é primo. Também se usa uma segunda função de hash

$$g(k) = (k + 1) \pmod{(p - 2)}$$

Quando ocorre uma colisão, usa-se uma sequência

$$h(k, i) = (h(k) + i \times g(k)) \pmod{p}$$

Use o procedimento de duplo hashing que foi descrito acima com $p = 4969$ para atribuir locais de memória a arquivos para funcionários com números de RG: $k_1 = 509496993$, $k_2 = 546332190$

Solução: Seguindo a abordagem acima:

$$(a) \quad h(509496993) = 509496993 \pmod{4969} = 578$$

$$g(509496993) = 509496994 \pmod{4967} = 2002$$

$$h(509496993, 1) = (578 + 1 \times 2002) \pmod{4969} = 2580$$

$$(b) \quad h(546332190) = 546332190 \pmod{4969} = 578$$

$$g(546332190) = 546332191 \pmod{4967} = 1927$$

$$h(546332190, 1) = (578 + 1 \times 1927) \pmod{4969} = 2505$$

O protocolo Bitcoin também faz uso do duplo hashing conforme será visto no decorrer deste trabalho.

A seguir, outro problema extraído e adaptado de Rosen, 2011 [44] é apresentado:



Figura 18: Problema do Estacionamento e função hash. Imagem disponível em [51]

Um estacionamento possui 31 vagas para visitantes, numeradas de 0 a 30. Os visitantes recebem vagas de estacionamento usando a função de hash $h(k) = k \pmod{31}$, em que k é o número formado a partir dos quatro últimos dígitos da placa do carro do visitante.

1. Quais espaços são atribuídos pela função hash para carros com esses quatro últimos dígitos em sua licença placas: 3417, 4018, 6207, 9400, 4761, 0310;
2. Descreva um procedimento que os visitantes devem seguir para encontrar um espaço de estacionamento gratuito, quando o espaço a que estão atribuídos estiver ocupado.

Solução:

1. $h(3417) \equiv 3417 \pmod{31} \equiv 7$
 $h(4018) \equiv 4018 \pmod{31} \equiv 19$
 $h(6207) \equiv 6207 \pmod{31} \equiv 7$
 $h(9400) \equiv 9400 \pmod{31} \equiv 7$
 $h(4761) \equiv 4761 \pmod{31} \equiv 18$
 $h(0310) \equiv 0310 \pmod{31} \equiv 0$
2. Usando duplo hashing:
 $h(9400) \equiv 9400 \pmod{31} \equiv 7$
 $g(9400) \equiv (9401) \pmod{29} \equiv 5$
 $h(9400, 1) \equiv (h(9400) + 1 \times 5) \pmod{31}$
 $7 + 5 \pmod{31} \equiv 12$

As funções JHA e JHA-1

Para ilustrar as funções de hashing criptográficas e aplicações, com funções de hash que fossem fáceis para os estudantes, Holden, 2013 [25] ministrou um curso de criptografia durante o Outono de 2000. O curso foi projetado para alunos que não tinham formação em matemática ou ciência da computação. Como não havia muitos recursos para esses cursos à época, Holden [25] criou sua própria função de hash, que mais tarde a chamou de "Josh's Hash Algorithm", ou JHA. Suas metas para essa função eram:

- **Simplicidade:** Alunos com experiência limitada deviam poder fazer exercícios sem um computador num único período de aula, e
- **Segurança:** Deve ser razoavelmente seguro (resistente à pré-imagem, resistente à segunda pré-imagem, e resistente a colisões) dado o objetivo anterior.

A função de hash **JHA** pega um conjunto de caracteres de letras e espaços e gera um inteiro entre 0 e 16 de acordo à seguinte regra:

$$JHA(texto) \equiv (7 \times \#vogais - 3 \times \#consoantes + \#espaços^2) \pmod{17}$$

Por exemplo, o conjunto de caracteres $M = \text{"Olá, meu nome é Alice"}$ tem um valor de hash de:

$$JHA(M) \equiv (7 \times 10 - 3 \times 6 + 4^2) \pmod{17} \equiv 0$$

No entanto Holden, 2013 [25] afirma que JHA não é tão segura. É moderadamente segura em relação a ataques de pré-imagem, todavia não é muito segura em relação a ataques de segunda pré-imagem e de colisão. Em Janeiro de 2010, Holden, 2013 [25] mais uma vez queria ter uma função de hashing com os dois objetivos acima, então ele chegou a uma versão ligeiramente modificada do JHA, que chamou de **função de hash JHA-1**:

$$JHA - 1(M) \equiv 5^{7 \times \#vogais - 3 \times \#consoantes + \#espacos^2} \pmod{17}$$

Por exemplo, o conjunto de caracteres $M = \text{”Olá, meu nome é Alice”}$ tem um valor de hashing dado por:

$$JHA - 1(M) \equiv 5^{7 \times 10 - 3 \times 6 + 4^2} \pmod{17} \equiv 13$$

De acordo com Holden em [25] JHA-1 é ainda mais seguro em relação aos ataques de pré-imagem, uma vez que eles envolveriam resolver um (pequeno) problema de logaritmo discreto, mas realmente não é melhor em relação aos outros tipos de ataques. No entanto, isso se encaixou bem em sua meta de simplicidade.

As funções de hash são unidirecionais. Sendo assim o funcionamento dos quebra-cabeças de hash dependem essencialmente desse fato. É impossível resolver um quebra-cabeça de hash checando o atendimento do valor de hash às restrições e, na sequência, aplicando na direção inversa a função de hash (o que significa, da saída desejada para a entrada exigida). A resolução dos quebra-cabeças de hash só podem se dar por tentativa e erro, portanto muita capacidade de processamento é exigida o que consome muita energia. Em média, para encontrar a solução, há influência do nível de dificuldade diretamente na necessidade de tentativas, o que, por seu turno, acaba influenciando no tempo necessário ou nos recursos computacionais. Mais especificamente, no blockchain, o hashing é usado nos casos a seguir:

- impressão digital para os dados de transação
- armazenagem de dados de transação de modo sensível a mudanças
- forma de incorrer em custos computacionais para alterar a estrutura de dados blockchain

As duas funções hash utilizadas no protocolo Bitcoin são introduzidas a seguir.

A função hash criptográfica RIPEMD-160

David Schwartz afirma em [18] que a função hash RIPEMD-160 foi utilizada no protocolo Bitcoin porque produz hashes mais curtos. Isso permite que os endereços de Bitcoin sejam mais curtos.

De acordo com McAndrew, 2016 [34] esta é uma das mais fortes funções hash modernas. Foi desenvolvida pela equipe belga-alemã de criptógrafos Hans Dobbertin, Antoon Bosselaers e Bart Preneel [14]. "RIPEMD" significa "RIPE Message Digest" ou "RIPE Resumo da Mensagem" (tradução livre), onde "RIPE" significa "Research and Development in Advanced Communications Technologies in Europe" ou "Pesquisa e Desenvolvimento em Tecnologias Avançadas de Comunicações na Europa" (tradução livre) - um bom exemplo de abreviação recursiva. Como o próprio nome sugere, o RIPEMD-160 produz hashes de 160 bits. Uma versão anterior, RIPEMD-128, foi considerada insegura; esta nova versão não só produz hashings mais longos, como também é imune aos ataques aos quais o RIPEMD-128 é vulnerável [14].

Como muitas funções hash modernas, é um descendente de MD4 (Message Digest 4) que foi desenvolvido por Ronald Rivest em 1990 [43]. Embora a MD4 tenha se mostrado muito insegura, ela gerou uma série de descendentes, dos quais RIPEMD-160 é uma.

O pseudo-código do cálculo da função hash RIPEMD-160 pode ser verificado em [10]. A continuação do processo de cálculo da função hash RIPEMD-160 se encontra no Apêndice .

A função hash criptográfica SHA-256

SHA-256 (secure hash algorithm, FIPS 182-2) é uma função de hash criptográfica com comprimento de trabalho de 256 bits.

David Schwartz afirma em [18] que a função de hash SHA-256 é usada no protocolo Bitcoin porque o uso de um hashing de uma chave pública pelo Bitcoin pode criar fraquezas únicas devido a interações inesperadas entre a função RIPEMD-160 e o protocolo ECDSA (algoritmo de assinatura de chave pública). Sendo assim, interpor uma operação de hashing adicional e muito diferente entre o RIPEMD-160 e o ECDSA tornaria quase inconcebível que pudesse haver uma maneira de encontrar colisões de endereços que fossem significativamente mais fáceis do que a força bruta testando um grande número de chaves secretas.

Ken Shirriff em seu blog [9] resolveu fazer um algoritmo para cálculo da função SHA-256 passo-a-passo gravando um vídeo ⁶ onde demonstra uma rodada de cálculo da função (são ao todo 64 rodadas). David Rabahy [42] inspirado por Shirriff resolveu criar uma planilha ⁷ com os demais passos do cálculo da função SHA-256 baseado no algoritmo de Shirriff.

O processo consiste a grosso modo em 64 iteradas partindo de valores iniciais bem definidos, onde a partir da primeira iterada conterà a mensagem de entrada a ser calculada pela função de hashing.

Para o exemplo, o input é a palavra "Hello World!" conforme Figura 16 cuja hashing SHA-256:

7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284add200126d9069

Mais detalhes desse processo de cálculo da função você pode verificar no Apêndice . A Figura 19 é a primeira iterada dos parâmetros.

⁶O vídeo está disponível em: <https://www.youtube.com/watch?v=y3dqhixzGVo>

⁷Uma cópia da planilha de Rabahy pode ser acessada em: https://docs.google.com/spreadsheets/d/1y0en0INC8qoCMLkWYasUsNbkMGyyDC_u1bYIzM03wt0/

Blockchains

Segundo Antonopoulos, 2017 [3] as transações são a parte mais importante do sistema Bitcoin. Todo o restante no Bitcoin é projetado para garantir que as transações possam ser criadas, propagadas na rede, validadas e adicionadas ao registro global do blockchain. As transações são estruturas de dados que codificam a transferência de valor entre os participantes no sistema Bitcoin. Cada transação é uma entrada pública no blockchain do Bitcoin, o registro geral de dupla-entrada.

Transações

Para as **transações** serem autorizadas, o blockchain deve garantir que somente o proprietário legítimo possa transferir sua propriedade para outras contas. O sistema ponto a ponto em consideração está aberto a todos. Sendo assim, qualquer um pode criar transações e submetê-las ao sistema. Os dados de transação são a base para descrever e deixar claro de quem é a posse. Somente o proprietário legítimo de uma conta deve ser capaz de transferir o direito de propriedade ou de posse associado à sua conta para outra conta. O desafio do blockchain é manter o sistema aberto, ao mesmo tempo em que restringe a transferência de posse ao proprietário legítimo.

Garantir que somente o proprietário legítimo transfira a posse consiste em utilizar uma medida de segurança digital equivalente às assinaturas feitas à mão e que sirvam ao mesmo propósito: identificar uma conta, declarar a concordância de seu proprietário com o conteúdo dos dados da transação específicos e aprovar a sua execução, permitindo que os dados sejam adicionados ao histórico de dados de transação.

Em termos simples, uma transação informa para a rede que o dono de uma quantidade de bitcoins autorizou a transferência de alguns destes bitcoins para outro dono. O novo dono agora pode gastar esses bitcoins ao criar uma nova transação que autoriza a transferência para um outro dono, e assim por diante, em uma cadeia de posse de bitcoins.

As transações são como linhas em um "registro contábil" (ledger) de dupla entrada. Em termos simples, cada transação contém um ou mais "inputs" (entradas), que são débitos em uma conta bitcoin. No outro lado da transação, existem um ou mais "outputs" (saídas) que são créditos adicionados em uma conta bitcoin. A soma dos inputs e outputs (débitos e créditos) não necessariamente resultam na mesma quantia. Ao invés disso, os outputs são um pouco maiores do que os inputs, e essa diferença se dá devido à "taxa de transação", que é um pequeno pagamento coletado pelo minerador que inclui a transação no registro contábil do Bitcoin (o blockchain). Uma transação Bitcoin é mostrada como uma entrada no registro contábil em transação como um registro contábil de entrada-dupla.

A transação também contém uma prova de posse para cada quantia de bitcoins (inputs) que é transferida, na forma de uma assinatura digital assinada pelo dono, que pode ser validada por qualquer pessoa, de maneira independente. Usando a terminologia do Bitcoin, "gastar" é assinar uma transação que transfere um valor (de uma transação prévia) para um novo dono, o qual é identificado através de um endereço Bitcoin ⁸.

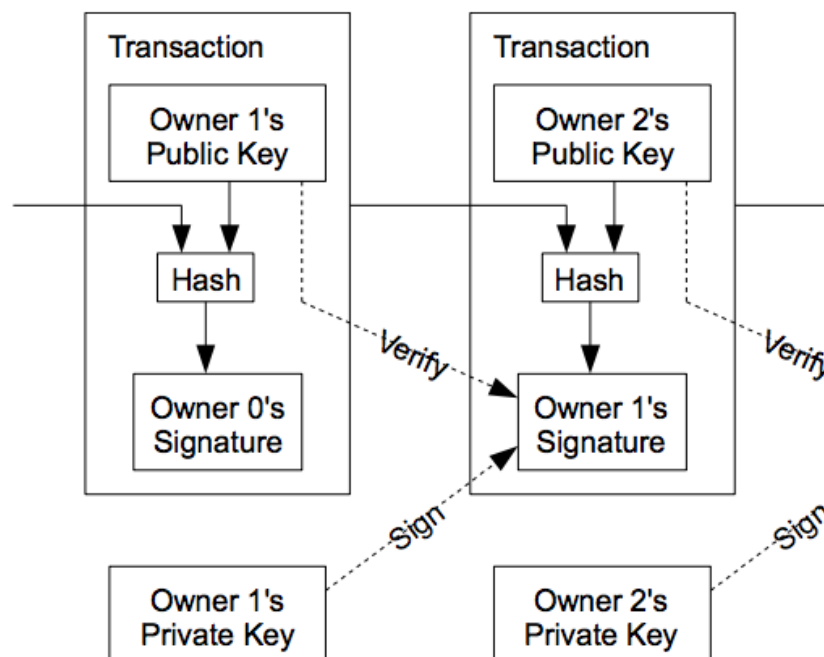


Figura 20: Diagrama de Transações. Disponível em: [36]

⁸A regra de formação de endereços Bitcoin são apresentados ao final deste capítulo

Pool de transações

Quase todo nó na rede bitcoin mantém uma lista temporária de transações não-confirmadas chamada de **mempool**, pool de memória ou pool de transações. Os nós usam esse pool para manter um acompanhamento das transações que são conhecidas pela rede mas que ainda não foram incluídas no blockchain. Por exemplo, um nó contendo uma carteira de usuário utilizará um pool de transação para acompanhar os pagamentos para essa carteira que foram recebidos na rede, mas que ainda não foram confirmados. As transações são recebidas e verificadas, sendo adicionadas ao pool de transações e transmitidas aos nós vizinhos para serem propagadas para a rede.

Algumas implementações de nós também mantêm um pool separado de transações órfãs. Caso um input de transação referir-se a uma transação que ainda não é conhecida, como um pai desconhecido, a transação órfã será armazenada temporariamente no pool órfão até que a transação pai chegue. Quando uma transação é adicionada ao pool de transações, verifica-se o pool órfão para quaisquer órfãos que sejam referenciados para esses outputs de transação (seus filhos).

Quaisquer órfãos correspondentes são então validados. Se válidos, eles são removidos do pool órfão e adicionados ao pool de transação, completando a cadeia que iniciou com a transação pai. Na presença de uma transação recém-adicionada, que não é mais uma órfã, o processo é repetido recursivamente em busca de quaisquer outros descendentes, até que não se encontre mais nenhum descendente.

Através desse processo, a chegada de uma transação pai desencadeia uma reconstrução em cascata de uma cadeia completa de transações interdependentes ao reunir os órfãos com seus pais ao longo de toda a cadeia. Tanto o pool de transações quanto o pool de órfãs (quando implementado) são armazenados na memória local e não são salvos em um armazenamento persistente; ao invés disso, eles são populados dinamicamente a partir das mensagens de rede que chegam. Quando um nó é iniciado, ambas os pools são esvaziados e são gradualmente populados com as novas transações recebidas na rede.

Assinaturas digitais

Suponha que alguém queira enviar uma saudação Hello World! de modo autorizado para o mundo. Com essa finalidade, ele cria uma mensagem contendo a saudação e uma assinatura digital correspondente. Todo o processo de assinar dados digitalmente está representado na Figura 21:

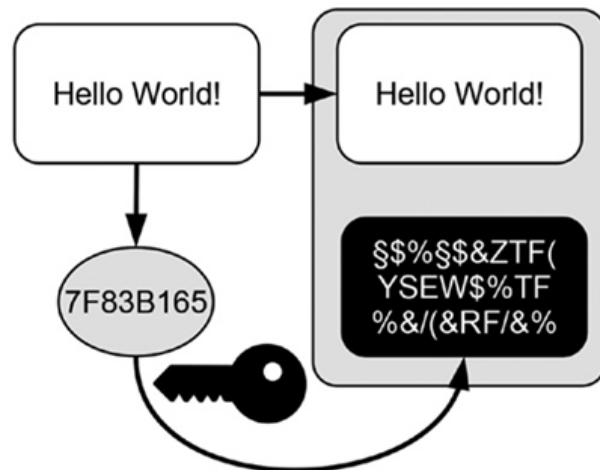


Figura 21: Criação de uma assinatura digital. Adaptado de [16]

Esse processo começa com a caixa branca na parte superior à esquerda da Figura 21 contendo a saudação. Devemos criar o valor de hash da saudação, que é $7F83B165$ e criptografar essa informação com a chave privada. O texto cifrado do valor de hash da saudação (a caixa preta com letras brancas) é a assinatura digital da saudação. Dois aspectos a tornam ela única: em primeiro lugar, pode ser associada unicamente, pois foi criada com uma chave privada única. Em segundo, ela é única no que concerne ao texto da saudação porque está baseada na impressão digital dela. Tanto a saudação quanto a assinatura digital são reunidas em um arquivo (a caixa cinza), que é a mensagem digitalmente assinada para o mundo.

A seguir, faremos a verificação dos dados usando a assinatura digital conforme Figura 22:

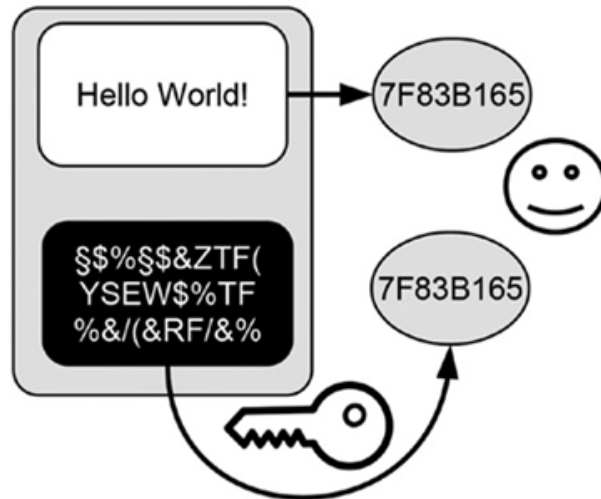


Figura 22: Usando uma assinatura digital. Adaptado de [16]

A mensagem, isto é, a saudação em conjunto com a assinatura digital, é enviada ao mundo todo. Qualquer um pode comprovar a autorização dessa mensagem utilizando a chave pública. O processo de verificar a mensagem usando a assinatura digital está na Figura 22.

O processo é iniciado pela saudação. Inicialmente, o próprio receptor da mensagem calcula o valor de hash da saudação, produzindo o valor $7F83B165$. Então ele descriptografa o texto cifrado associado (a assinatura digital) com a chave pública. Ao proceder, gera o valor $7F83B165$, que é o valor de hash daquela versão da saudação que se gostaria de enviar ao mundo. A comparação entre os dois valores de hash encerra a verificação. Como ambos são idênticos, o receptor conclui, corretamente, que, em primeiro lugar, a mensagem foi assinada, pois ele foi capaz de descriptografar a assinatura com a chave pública, e, em segundo, que o texto da saudação que se encontra na mensagem é realmente aquela que queria enviar, pois o texto cifrado descriptografado é idêntico ao valor de hash da saudação na mensagem.

A Figura 23 mostra como a assinatura digital identifica uma saudação forjada. Esta mesma figura também mostra a mensagem que chegou na caixa de correspondência de um amigo. Observemos a mudança no texto da saudação. Alguém malicioso substituiu o ponto de exclamação por um ponto de interrogação e, desse modo, alterou todo a saudação. Essa não é a forma como se queria fazer a saudação. A assinatura digital mostrará a todos que a mensagem foi alterada contra a vontade do emissor da mensagem.

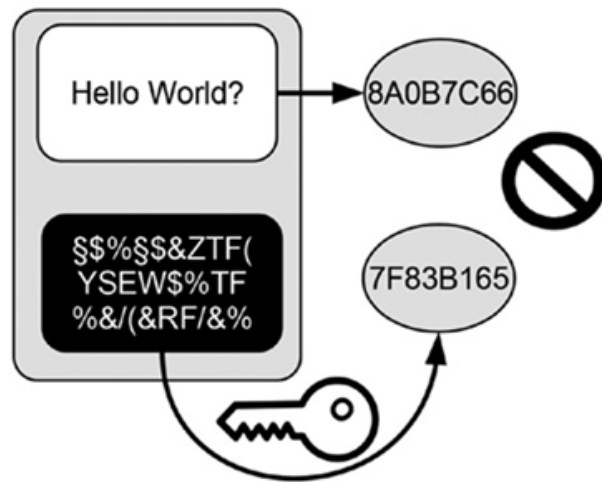


Figura 23: Identificando uma fraude. Adaptado de [16]

Inicialmente, o receptor da mensagem criará o valor de hash da saudação, produzindo o valor $8F0B7C66$. Então ele descriptografará a assinatura digital usando a chave pública. Fazer isso produz o valor $7F83B165$, que é o valor de hash da versão da saudação que realmente se gostaria de enviar ao mundo. Uma comparação entre os dois valores de hash revela que eles não são idênticos. Claramente, isso mostra que a saudação na mensagem não é a mesma. Assim, todos concluirão que a mensagem não foi autorizada e, desse modo, ninguém se responsabilizará pelo seu conteúdo. As assinaturas digitais de dados de transação são uma combinação das informações a seguir:

- valores de hash criptográficos dos dados de transação;
- texto cifrado que pode ser associado à chave privada correspondente de uma conta.

Transação de Bitcoins na vida real

Bitcoin é uma tecnologia usada para representar dinheiro, que é fundamentalmente uma linguagem para a troca de valor entre pessoas. Vamos conhecer histórias adaptadas de Antonopoulos, 2017 [3] de pessoas que estão usando bitcoins e alguns dos usos mais comuns da moeda e do protocolo. Iremos utilizar essas histórias para ilustrar os usos do dinheiro digital na vida real e como eles se tornaram possíveis por meio das várias tecnologias que são partes do Bitcoin.

Alice mora na área norte da baía da Califórnia. Ela ouviu falar sobre o Bitcoin através dos seus amigos e quer começar a usá-lo. Iremos acompanhar a história de como

ela aprende a respeito do Bitcoin, adquire algumas moedas e então gasta alguns de seus bitcoins para comprar uma xícara de café no Café do Bob em Palo Alto, Califórnia. Esta história irá nos apresentar ao software, às casas de câmbio e transações básicas desde a perspectiva de um consumidor do varejo.

Alice não é uma usuária técnica e só recentemente ouviu falar do Bitcoin através de um amigo. Ela começa sua jornada visitando o website oficial `bitcoin.org`, onde encontra uma ampla seleção de clientes bitcoin. Seguindo o conselho do site `bitcoin.org`, ela escolhe o cliente bitcoin compacto Multibit. Alice segue um link do site `bitcoin.org` para baixar e instalar a Multibit no computador pessoal dela.

Assim que a Alice terminar de baixar e instalar o aplicativo Multibit, ela o executa e o Multibit automaticamente cria uma carteira e um novo endereço bitcoin:

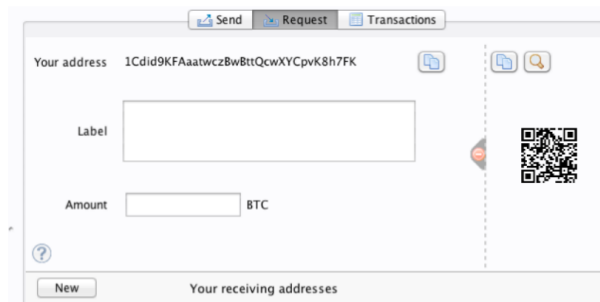


Figura 24: O novo endereço bitcoin da Alice. Disponível em [3]

A parte mais importante da tela da Figura 24 é o endereço Bitcoin da Alice. Assim como um endereço de email, Alice pode compartilhar este endereço e qualquer um pode usá-lo para mandar dinheiro diretamente à carteira dela. Mais detalhes sobre como se dá tecnicamente a formação de endereços Bitcoin está presente ao final deste capítulo. Agora, Alice está pronta para com Alice pode compartilhar este endereço e qualquer um pode usá-lo para mandar dinheiro diretamente à carteira dela. Mais detalhes socomocomo se dá tecnicamente a formação de endereços Bitcoin está presente ao final deste capítulo. Agora, Alice está pronta para começar a usar sua nova carteira Bitcoin.

Depois de criar a sua carteira Bitcoin, Alice agora está pronta para receber fundos (recursos). A carteira gera aleatoriamente uma chave privada junto com o endereço Bitcoin correspondente. Nesse ponto, o endereço Bitcoin dela ainda não é conhecido pela rede Bitcoin, nem "registrado" em qualquer parte do sistema Bitcoin. O endereço Bitcoin dela é simplesmente um número que corresponde a uma chave que ela pode usar para controlar o acesso aos fundos. Não há uma conta ou associação entre aquele endereço e uma

conta. Até o momento em que este endereço esteja referenciado como o destinatário de um valor em uma transação publicada no ledger ou registro contábil de bitcoin (o blockchain), ele é simplesmente parte da vasta quantidade de possíveis endereços considerados "válidos" em Bitcoin. A partir do momento em que esteja associado com uma transação, ele se torna parte dos endereços conhecidos na rede e a Alice poderá comprovar o saldo dela no registro público.

Alice encontrou-se em um restaurante local com um amigo, o Joe, que a apresentou ao Bitcoin, para que eles possam trocar alguns dólares e colocar bitcoins na conta dela. Ela trouxe um papel com o endereço dela e o QR code impressos conforme aparecem na carteira Bitcoin. Não há nenhuma informação que deva ser protegida, desde um ponto de vista de segurança, no endereço Bitcoin. Ele pode ser publicado em qualquer lugar sem nenhum risco de segurança à conta da Alice.

Alice quer trocar somente 10 dólares por bitcoins, para que assim ela não arrisque muito dinheiro nessa nova tecnologia. Ela dá ao Joe uma nota de \$10 e o papel impresso com seu endereço para que o Joe possa lhe mandar o montante equivalente em bitcoins.

Em seguida, Joe tem que descobrir a taxa de câmbio para que ele possa dar a quantidade certa de bitcoins à Alice.

Usando um dos aplicativos que informa a taxa de câmbio entre bitcoins e dólares, Joe determina o preço do bitcoin como aproximadamente 100 dólares por bitcoin. Nesse momento, ele deveria dar a Alice 0.10 bitcoin, também chamado de 100 millibits, em troca dos 10 dólares que ela lhe deu.

Uma vez que Joe determinou um preço justo para a troca, ele abre um aplicativo de carteira em seu celular e seleciona "enviar" bitcoin.

No campo para inserir o endereço Bitcoin, há um pequeno ícone de um QR code. Isso permite que Joe escaneie o código de barras com a câmera de seu smartphone para que ele não tenha que digitar o endereço bitcoin da Alice:

1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK

O que seria algo grande e difícil de se digitar. Joe toca no ícone do QR code e ativa a câmera para escanear o QR code da carteira impressa que a Alice trouxe consigo. O aplicativo de carteira móvel preenche o endereço Bitcoin e Joe pode verificar que o código

foi escaneado corretamente ao comparar alguns dígitos com o endereço impresso pela Alice.

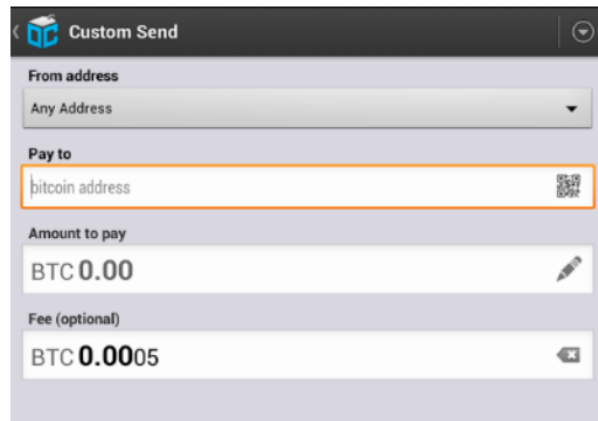


Figura 25: A tela de envio de bitcoin . Disponível em [3]

Então Joe digita o valor em bitcoins da transação, 0,10 bitcoin. Ele confere com cuidado para ter certeza de que digitou a quantia correta, pois ele está a ponto de transmitir dinheiro e qualquer erro pode custar caro. Finalmente ele aperta Enviar para transmitir a transação. A carteira móvel do Joe constrói a transação que assina 0,10 bitcoin ao endereço da Alice, gerando os fundos da carteira do Joe e assinando a transação com as chaves privadas dele. Isso informa a rede Bitcoin que o Joe autorizou uma transferência de valor de um de seus endereços para o novo endereço da Alice. À medida que a transação se transmite conforme o protocolo Peer-to-peer, ela rapidamente se propaga pela rede bitcoin. Em menos de 1 segundo, a maioria dos nós com melhor conexão na rede recebem a transação e veem o endereço da Alice pela primeira vez.

Se Alice tiver um smartphone ou um laptop com ela, também será capaz de ver a transação. O registro contábil do Bitcoin — um arquivo que não pára de crescer e que guarda cada uma das transações em bitcoin que já ocorreram desde o início — é público, o que significa que tudo que ela tem de fazer é olhar seu próprio endereço e ver se quaisquer fundos foram mandados para ele. Ela pode fazer isso facilmente no site blockchain.info, digitando o endereço dela no campo de busca. O website lhe vai mostrar uma página listando todas as transações de e para aquele endereço. Se a Alice estiver olhando essa página, vai ver uma atualização que mostra uma nova transação transferindo 0,10 bitcoin para o saldo dela logo depois do Joe apertar Enviar.

Inicialmente, o endereço da Alice vai mostrar a transação do Joe como ”Transação não Confirmada.” Isto significa que a transação já se propagou pela rede, mas ainda não foi

incluída no registro contábil de transações do Bitcoin, conhecido como o blockchain (cadeia de blocos). Para ser incluída, a transação deve ser "escolhida" por um minerador e incluída em um bloco de transações. Quando um novo bloco é criado, em aproximadamente 10 minutos, as transações dentro do bloco passam a ser aceitas como "confirmadas" pela rede e então podem ser gastas. A transação é vista instantaneamente por todos, mas só se torna "confiada" por todos quando está incluída em um novo bloco minerado.

Alice é uma nova usuária que acabou de obter seu primeiro bitcoin. Ela encontrou com seu amigo, Joe, para trocar algum dinheiro por bitcoin. A transação criada por Joe alocou 0,10 BTC na carteira de Alice. Agora, ela irá fazer sua primeira compra, um transação de varejo, comprando uma xícara de café na cafeteria do Bob, em Palo Alto, Califórnia. A cafeteria do Bob começou a aceitar recentemente pagamentos em bitcoin, ao adicionar a opção de pagamentos por bitcoin no sistema do seu ponto de vendas. Os preços na cafeteria são listados na moeda local (dólares americanos), mas no caixa, os clientes agora contam com a opção de pagar tanto em dólares quanto em bitcoin. Alice faz seu pedido - uma xícara de café - e Bob entra a transação em seu sistema de vendas. O sistema do ponto de vendas fará a conversão do preço total em dólares para bitcoins, tendo como referência a cotação do momento, e apresenta o valor final nas duas moedas, bem como um QR code contendo uma requisição de pagamento para essa transação.

Total: \$1.50 USD 0,015 BTC

Bob diz: "A conta deu 1,50 dólares, ou 15 millibits."

Alice então usa o smartphone dela para escanear o código de barras mostrado na tela do Bob. O smartphone dela mostra um pagamento de 0,0150 BTC para o Bob's Cafe e ao clicar em Enviar ela autoriza o pagamento. Dentro de alguns segundos (aproximadamente o mesmo tempo que leva uma autorização de cartão de crédito), o Bob visualiza a transação em seu caixa, completando a transação.

O pagamento da Alice para o Café do Bob usa uma transação prévia como sua entrada. Alice recebeu bitcoins do amigo dela em troca de dinheiro. Aquela transação continha um número de bitcoins "trancados" (alienados) com a chave da Alice. Sua nova transação para o Café do Bob utiliza a transação prévia como uma entrada e cria novas saídas para pagar pela xícara de café e receber o troco. As transações formam uma cadeia, onde os inputs da última transação correspondem às saídas das transações anteriores. A chave da Alice fornece a assinatura que desbloqueia estas saídas de transações prévias,

desta maneira provando à rede Bitcoin que ela é a dona dos fundos. Ela vincula seu pagamento pelo café ao endereço do Bob, desta maneira "alienando" esta saída com o requisito de que Bob produza uma assinatura, liberando essa quantidade de bitcoins para ser gasta. Isso representa a transferência de valor entre Alice e Bob. Essa cadeia de transações, do Joe para a Alice, e dela para o Bob, é ilustrada na Figura 26:

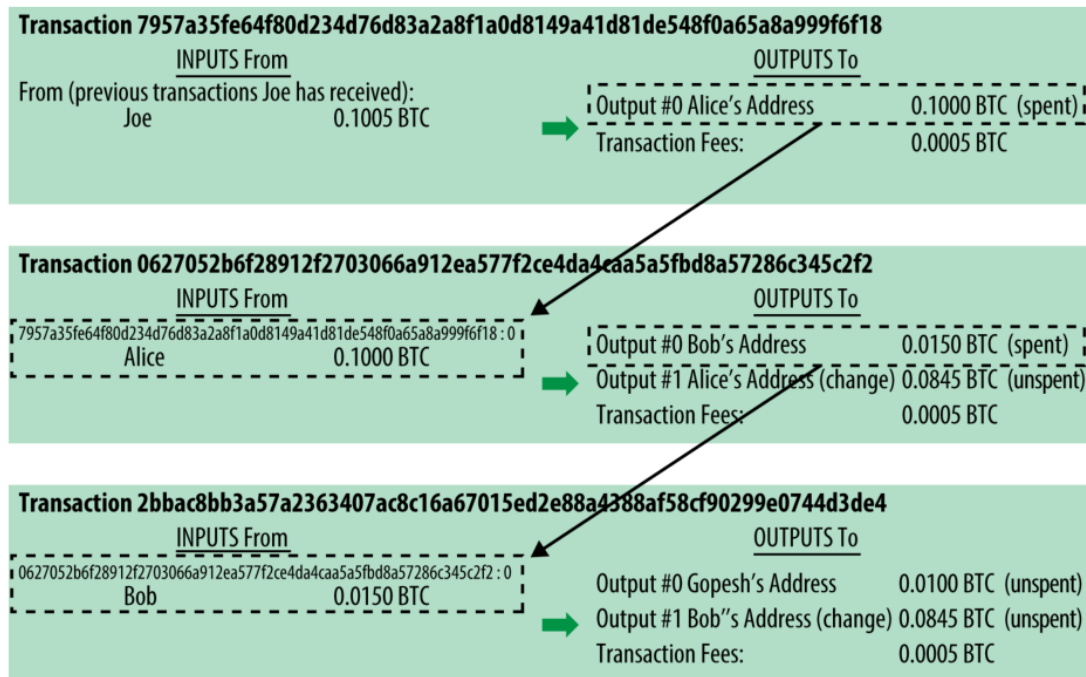


Figura 26: Cadeia de Transações. Disponível em [3]

O aplicativo de carteira contém toda a lógica para selecionar as entradas e saídas apropriadas para construir uma transação com os dados especificados pela Alice. Ela só precisa fornecer os dados de destino e uma quantia: o seu aplicativo de carteira faz todo o resto, sem que ela sequer veja os detalhes. Outro aspecto importante, é que o aplicativo de carteira também pode construir transações mesmo estando completamente offline. Da mesma maneira que você pode preencher um cheque em casa para depois depositá-lo em um envelope no banco, uma conexão com a rede bitcoin não é necessária para que uma transação seja construída e assinada. A transação só precisa ser enviada para a rede quando a pessoa quiser efetuar-la.

O aplicativo de carteira da Alice terá primeiro que achar os inputs que podem pagar pela quantia que ela quer enviar para o Bob. A maioria dos aplicativos de carteira mantém um pequeno banco de dados de "saídas de transações não gastos" que são trancados (alienados) com as próprias chaves da carteira. Logo, a carteira de Alice iria conter uma cópia da saída da transação do Joe, que foi criada na troca pelo dinheiro.

Um aplicativo de carteira de bitcoin que roda como um cliente de índice completo na verdade contém uma cópia de cada saída não gasta de todas as transações presentes na blockchain. Isso permite que a carteira construa inputs de transação, além de verificar rapidamente se as transações que chegam tem inputs corretos. No entanto, como um cliente de índice completo ocupa muito espaço de armazenamento em disco, a maioria das carteiras roda clientes "leves" que mantêm somente o registro das saídas não gastas do usuário.

A carteira de Alice contém bitcoins suficientes em uma saída não-gasta isolado para pagar pela xícara de café. Caso não contivesse, o aplicativo carteira de Alice teria que "vasculhar" uma pilha de pequenas saídas não-gastas, como se estivesse pegando as moedas em uma bolsa, até encontrar o suficiente para poder pagar o café. Em ambos os casos, pode haver uma necessidade de receber algum troco de volta, quando o aplicativo carteira cria as saídas da transação (pagamentos).

Uma saída de transação é criada na forma de um código que cria uma alienação no valor a ser transferido, de maneira que o valor só pode ser regastado se uma solução for apresentada ao código. De maneira simplificada, a saída da transação de Alice irá conter um código que diz algo como "Essa saída é pagável para aquela pessoa que conseguir apresentar uma assinatura para a chave correspondente ao endereço público de Bob". Como somente o Bob possui a carteira com as chaves correspondentes àquele endereço, somente a carteira do Bob pode apresentar a assinatura para resgatar essa saída. Alice ao fazer uma exigência de assinatura do Bob, ela está fazendo uma "alienação" ao valor da saída.

Essa transação também incluirá uma segunda saída, porque os fundos de Alice estão na forma de uma saída de 0,10 BTC, que é dinheiro demais para a transação de 0,015 BTC pela xícara de café. Alice precisará de 0,085 BTC de troco. O pagamento do troco da Alice é criado pela carteira da Alice na mesma transação que o pagamento do Bob. Essencialmente, a carteira de Alice divide seus fundos em dois pagamentos: um para o Bob, e outro de volta para si mesma. Ela pode então usar a saída do troco em uma transação no futuro, gastando-o mais tarde.

Finalmente, para que a transação seja processada pela rede em tempo hábil, o aplicativo de carteira da Alice irá adicionar uma pequena taxa. Isso não está explícito na transação: isso está implícito na diferença entre as entradas e as saídas. Se ao invés de

receber 0,085 de troco, Alice cria somente 0,0845 como uma segunda saída, haverá 0,0005 (metade de um milibitcoin) restantes. A entrada de 0,10 BTC não é totalmente gasta com as duas saídas, porque ele irá se somar até menos do que 0,10. A diferença resultante é a taxa de transação que é coletada pelo minerador como um pagamento por ter incluído a transação em um bloco e adicionar esse bloco no livro razão da blockchain.

A transação resultante pode ser vista usando um aplicativo web explorador de blockchain, como visto na Figura 27:

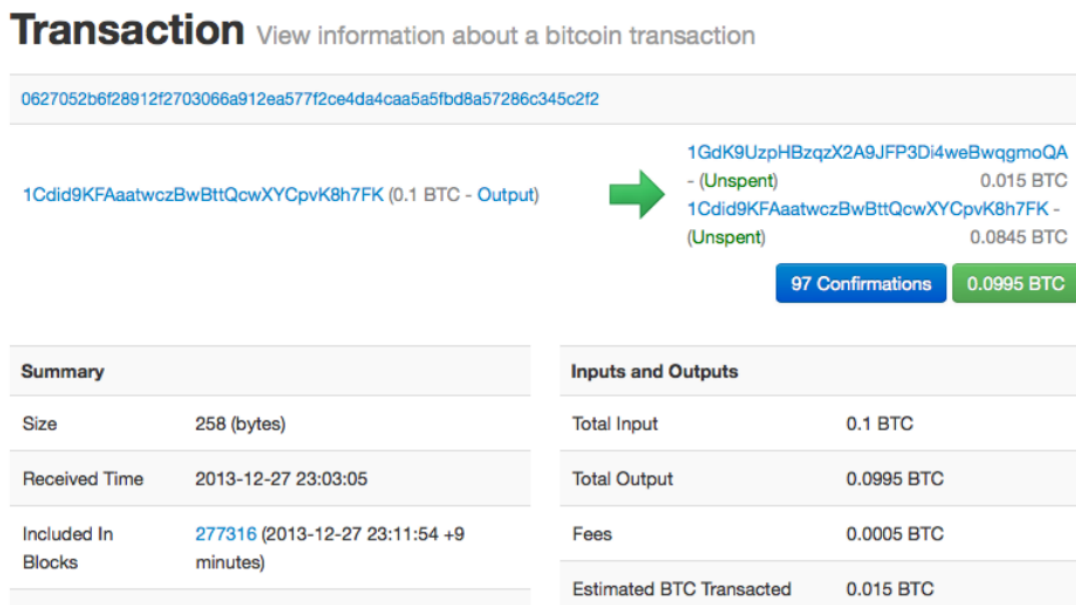


Figura 27: Transação de Alice para o Bob. Disponível em [3]

Agora a carteira da Alice pode enviar a nova transação para qualquer um dos outros clientes bitcoins se ela estiver conectada através de uma conexão de Internet: por cabo, WiFi ou móvel. A sua carteira não tem que obrigatoriamente estar conectada diretamente à carteira do Bob ou usar a conexão de internet oferecida pela cafeteria, embora essas opções também sejam possíveis. Qualquer nó (outro cliente) na rede bitcoin que receber uma transação válida que não tenha sido vista anteriormente, irá propagá-la imediatamente para outros nós com os quais está ligado. Logo, a transação rapidamente é propagada através da rede ponto-a-ponto (P2P), atingindo uma grande percentagem dos nós dentro de poucos segundos.

Se a carteira do Bob estiver diretamente conectada à carteira da Alice, o aplicativo pode ser o primeiro nó a receber a transação. Entretanto, mesmo que a carteira de Alice envie a transação através de outros nós, a transação chegará à carteira do Bob

dentro de pouco segundos. A carteira de Bob irá identificar imediatamente a transação de Alice como um pagamento porque ela contém saídas que são resgatáveis pelas chaves do Bob. A carteira de Bob também pode verificar independentemente que a transação é bem formada, utiliza entradas previamente não-gastos e contém taxas de transação suficientes para ser incluída no próximo bloco. Neste momento Bob pode esperar, com um alto grau de probabilidade, que a transação será em breve incluída em um bloco e será confirmada.

A transação foi propagada na rede bitcoin. Ela só vai tornar-se parte do livro-ração compartilhado (a blockchain) quando for verificada e incluída em um bloco, através de um processo chamado mineração.

O sistema de confiança do bitcoin é baseado em computação. As transações são agrupadas em blocos, o que requer uma enorme quantidade de processamento para prová-las, mas apenas uma pequena quantidade de processamento para verificá-las como previamente provadas. O processo de mineração do bitcoin possui dois propósitos:

- A mineração cria novos bitcoins em cada bloco, quase como um banco central imprimindo novas moedas e notas. A quantidade de bitcoin criada por bloco é fixa e diminui com o tempo.
- A mineração cria confiança ao garantir que as transações sejam confirmadas somente se poder de processamento suficiente for dedicado ao bloco que as contém. Mais blocos requerem mais processamento, o que significa maior confiança.

Uma boa maneira de descrever a mineração é a sugerida por Antonopoulos, 2017 [3]. Ele descreve mineração como um jogo de sudoku, gigantesco e competitivo, que reinicia cada vez que alguém encontra uma solução e cuja dificuldade se ajusta automaticamente, de maneira que leve cerca de 10 minutos para que uma solução seja encontrada. Imaginemos um sudoku gigantesco, com milhares de colunas e linhas de tamanho. Se alguém mostrar para outra pessoa um sudoku completo, essa pessoa pode verificar rapidamente que ele está corretamente preenchido. No entanto, se o sudoku tiver apenas alguns quadrados preenchidos e o resto estiver vazio, levará muito trabalho para resolvê-lo! A dificuldade do sudoku pode ser ajustada ao mudar o seu tamanho (mais ou menos linhas ou colunas), mas o sudoku ainda pode ser verificado de maneira rápida, mesmo que ele seja muito grande. O "quebra-cabeças" usado no bitcoin é baseado em um hash criptográfico, que exhibe características semelhantes: ele é assimetricamente difícil de resolver, mas fácil de verificar, e sua dificuldade pode ser ajustada.

Blocos

Acumular e somar todas as transações atuais é o primeiro passo na construção de um bloco para o blockchain. Quando um usuário cria uma transação, esta é transmitida para toda a rede e, em seguida, o computador de um membro da rede captura essa transação e revisa, para se certificar de que ela é válida. O computador da rede que tem essa função é conhecido como "verificador". Dado que as moedas no blockchain não são mais do que uma série de transações, o primeiro passo para confirmar uma transação consiste em olhar de onde o remetente diz que originalmente recebeu o dinheiro. O verificador deve rever a história do blockchain para encontrar o bloco e a transação em que o emissor recebeu o dinheiro originalmente.

- Se a transação for confirmada, então a transação é válida, e terá que ser confirmado o endereço da parte receptora, para, em seguida, seja adicionada esta transação ao livro;
- Se a transação de entrada não for válida, por exemplo, se alguém está enviando 1 bitcoin a um usuário, mas no blockchain não há registro de que alguma vez que esse alguém tenha recebido bitcoins, esta transação é considerada inválida, será excluída e não será incluída no livro razão

Uma vez que todas as transações nesse bloco forem verificadas, é o momento de adicioná-las à cadeia. O exemplo a seguir é simples, e mostra onde as transações são listadas uma após a outra:

[Entrada] [Quantidade] [Endereço de saída], [Entrada] [Quantidade] [Endereço de saída], [Entrada] [Quantidade] [Endereço de saída], [Entrada] [Quantidade]...

Em seguida, o verificador aplica uma técnica criptográfica chamada de função de hash ou função de hashing para cada uma das transações. Em sua definição mais básica, o hashing (impressão digital do bloco) pega uma cadeia de caracteres e gera outra cadeia de caracteres. Por isso, quando se alimenta um algoritmo de hash com a entrada, a quantidade e o endereço de saída, o resultado será uma cadeia de caracteres únicos para essa transação, como essa:

aba128d3931e54ce63a69d8c2c1c705ea9f39ca950df13655d92db662515eacf

(Esse é um hashing de uma transação real do Blockchain)

Assim, o hash é utilizado para normalizar os dados, enquanto fornece uma segurança que permite revisar que os dados não foram manipulados.

Por exemplo, se alguém tentar mudar uma transação no blockchain, teriam que fazer novamente o hashing da transação, e seria completamente diferente, o que ficaria em evidência, que foi manipulado.

Para que o blockchain seja ainda mais difícil de manipular, são realizadas várias etapas de hash. Isso significa que o hash de uma transação é combinado com um hash de outra transação, e um novo hash é feito gerando um código menor. Essa combinação das transações é conhecida como árvore de Merkle ⁹ e assegura que todas as transações no bloco sejam válidas enquanto utiliza menos memória a longo prazo.

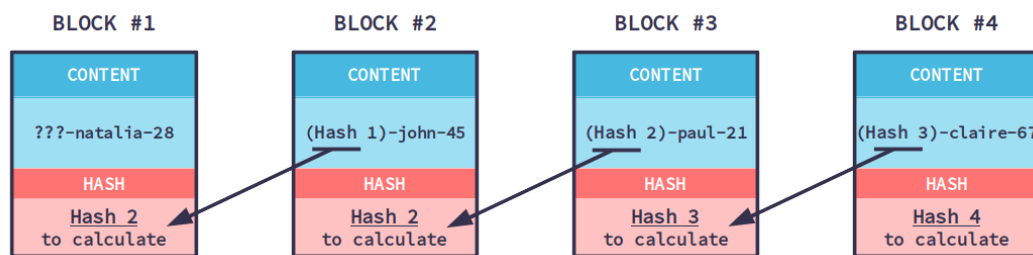


Figura 28: Blocos. Disponível em: [2]

O último elemento adicionado a um bloco é um selo de tempo e a vinculação aos blocos anteriores da cadeia, utilizando novamente um hash que, de alguma maneira, insere o conteúdo de blocos anteriores no conteúdo do novo bloco.

O hashing toma uma entrada, não importa quão grande ou pequena seja, e a transforma em uma cadeia de caracteres. Se alguém alterar a entrada, mesmo que minimamente, a saída mudará completamente. Desta forma, se o conteúdo de um antigo bloco já foi vinculado a um novo bloco e depois alguém altera o bloco anterior, por menor que seja a alteração, mudará todo o hashing do bloco.

Desta forma, uma vez que um bloco foi concluído e, quanto mais tempo passa, mais difícil se torna alterar, com sucesso, uma transação sem que seja detectado. Esta é a razão por que o hash é o núcleo de segurança do blockchain e permite que o livro razão seja público e seguro ao mesmo tempo.

⁹Segundo Judmayer et al. Árvores de Merkle são árvores binárias em que os nós das folhas são marcados com os valores que precisam ser autenticados e cada um nó folha é marcado com o hash das marcações ou valores de seus nós filhos

No entanto, é preciso dizer que o hashing, por si só, não é tão difícil de resolver. A maioria dos computadores podem criar um hash de um blockchain em poucos segundos, de forma que, com o objetivo de garantir a segurança, é necessário introduzir um outro nível de dificuldade para a criação de um novo bloco. Este nível de dificuldade é chamado de "proof-of-work".

O objetivo do blockchain é manter todo o histórico de dados de transação ordenado. O desafio consiste em armazenar todos os dados das transações realizadas, preservando a ordem em que ocorreram, e que possibilite detectar, de modo rápido e fácil, qualquer alteração feita nos dados. Detectar modificações rapidamente é importante, pois constitui a base para evitar que o histórico de transações seja manipulado ou forjado.

A ideia é criar uma biblioteca de dados de transação para manter um catálogo ordenado que preserve a ordem em que as transações foram adicionadas à biblioteca. Para detectar qualquer mudança feita, seja no catálogo de ordenação ou nos dados de transações individuais, os dados devem ser armazenados de modo sensível a mudanças, usando referências de hash.

Esta seção explica como transformar um livro em uma pequena biblioteca com um catálogo de ordenação, que, na verdade, será uma versão simplificada da estrutura de dados blockchain.

Ponto de partida: um livro

Durante muitos séculos, as informações escritas eram preservadas em rolos de pergaminho de difícil manipulação. Atualmente, estamos acostumados a ter informações escritas preservadas em códices: conjuntos de páginas numeradas unidas por uma lombada, que chamamos de livros. Pelo fato de serem tão comuns, aceitamos a inovação que os livros representam, sem questioná-la. Algumas de suas propriedades importantes incluem:

- **Armazenamento de conteúdo:** os livros armazenam conteúdo em suas páginas.
- **Ordenação:** as frases nas páginas, assim como as páginas do livro, são ordenadas.
- **Conexão entre as páginas:** as páginas estão fisicamente conectadas por meio da lombada do livro, e logicamente conectadas pelo conteúdo e pelos números das

páginas.

Como consequência dessas propriedades, podemos navegar pelos livros para a frente e para trás, virando as páginas, ou podemos pular diretamente para páginas específicas utilizando seus números. Vamos ver o que poderíamos fazer se mudássemos algumas dessas propriedades. Vamos resumir o que conseguimos com esse exemplo (Figura 29).

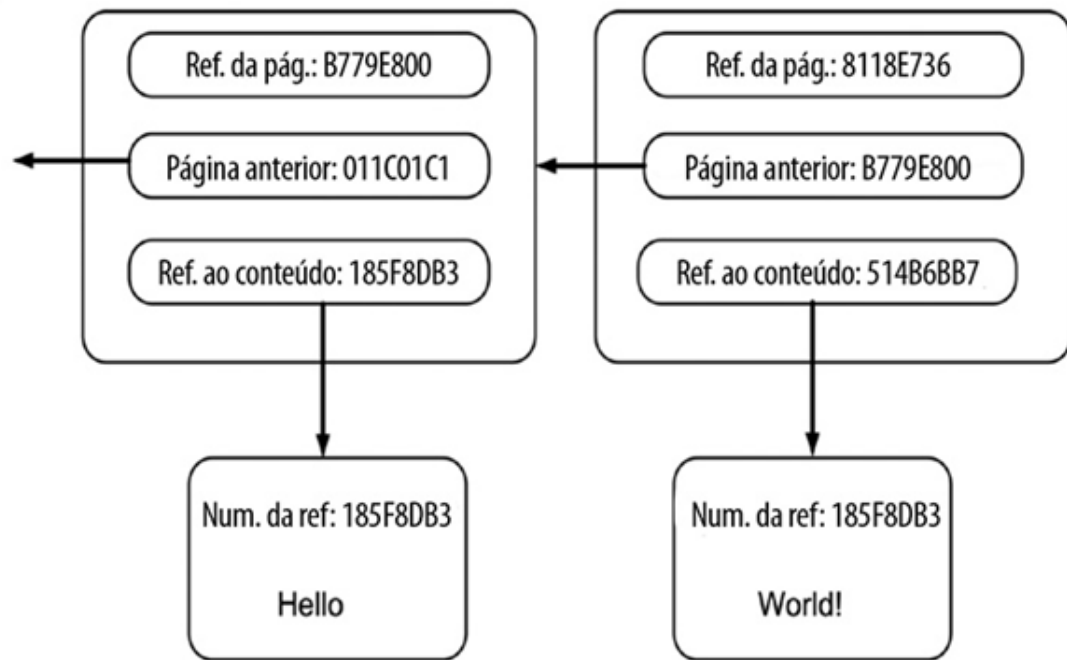


Figura 29: Páginas do livro. Adaptado de [16]

Transformamos um livro clássico em duas pilhas fisicamente não ordenadas de páginas soltas, ligadas por números de referência únicos. Uma pilha de páginas tem o conteúdo, enquanto a outra mantém a ordenação. Por questões de simplicidade, chamaremos a última pilha de páginas de catálogo de ordenação. Cada página do catálogo de ordenação contém o número de referência para a sua página anterior e o número de referência para a página de conteúdo correspondente. Como resultado, separamos as informações de ordenação das informações de armazenamento, e a localização lógica (a ordem) da localização física das páginas. Pelo fato de termos usado valores de hash como números de referência, qualquer um pode verificar se esses valores estão corretos apenas os recalculando. Como as páginas do catálogo de ordenação não estão mais fixas em uma lombada, somente poderemos navegar por ele de trás para a frente, página a página, seguindo os números de referência de página que apontam para a página anterior.

Proof-of-work

No contexto do blockchain, os quebra-cabeças de hash são chamados de prova de trabalho (proof of work), tendo em vista que a solução prova que alguém (o minerador) fez o trabalho necessário para resolvê-lo.

Proof-of-work (Prova de trabalho) é o que protege o Bitcoin e, em um nível profundo, permite a descentralização do Bitcoin. Encontrar uma prova de trabalho dá ao minerador o direito de colocar o bloco anexado ao blockchain. A prova de trabalho é difícil mas ao mesmo tempo objetiva: qualquer um pode ser um minerador se assim o desejar.

A prova de trabalho é chamada de "mineração" pelo seguinte motivo (Song, 2019) [52]: como na mineração física, há algo que os mineiros estão procurando. Uma operação típica de mineração de ouro processa 45 toneladas de sujeira e rocha antes de acumular 1 onça de ouro. Isto é porque o ouro é muito raro. No entanto, uma vez encontrado o ouro, é muito fácil verificar se o ouro é real. Existem testes químicos, pedras de toque e outras maneiras sendo relativamente barato de dizer se o material encontrado é ouro.

Segundo Dwork e Naor [17] a implementação mais simples da ideia do proof-of-work é basear a dificuldade em extrair raízes quadradas módulo um primo p . Não há atalho conhecido para esta função.

- **Índice** Um primo p de comprimento dependendo do parâmetro de diferença;
- **Definição de f_p** O domínio de f_p é Z_p . $f_p(x) = \sqrt{x} \pmod p$.
- **Verificação** Dados x, y verifique que $y^2 \equiv x \pmod p$

A etapa de verificação requer apenas uma multiplicação. Por outro lado, não é conhecido nenhum método de extrair raízes quadradas módulo primo que exija menos do que $\log p$ multiplicações. Assim, quanto maior o comprimento de p , maior a diferença entre o tempo necessário para avaliar f_p e o tempo necessário para a verificação.

De acordo com Judmayer et al [29] o requisito básico para uma prova de trabalho deve ser que seja difícil para produzir, mas fácil de verificar.

Recapitulando, para Campello, 2018 [12] transação é a anotação de que uma

quantidade de recursos saiu de uma origem para um destino, ou na linguagem da blockchain, saiu de uma carteira para outra carteira. Afirma ainda que blocos nada mais são do que um conjunto de transações, além de um cabeçalho contendo algumas informações. O bloco é a página com anotações no topo (cabeçalho) e as transações são as linhas dessa página, com cada linha contendo, no mínimo, as seguintes informações: um identificador único (id); carteira de origem; carteira de destino; valor transferido.

As informações que estão no cabeçalho do bloco geralmente são: o número do bloco (feito o número da página, por exemplo); a data e a hora que o bloco foi criado (minerado); um hash gerado a partir das transações daquele bloco; um hash gerado a partir do cabeçalho do bloco anterior e o nonce.

O nonce é um número inteiro que prova que alguém trabalhou muito para poder encontrá-lo. É a chave, o resultado final, de um desafio computacional promovido pelo protocolo da blockchain. Segundo Song, 2019 [52] Nonce significa "number used only once" ou "número usado apenas uma vez" ou n-once (uma) vez. Esse número é o que é alterado pelos mineiros ao procurar prova de trabalho. O protocolo da blockchain entrega aos membros da rede, os nós, o seguinte desafio: "Concatenar um inteiro ao final do hash gerado a partir das transações e gere um novo hash a partir dessa nova string. Caso o hash gerado a partir da junção do hash oriundo das transações mais esse número inteiro não tenha 20 zeros a sua frente, vá incrementando esse número inteiro até achar um novo hash com 20 zeros na frente."

Existe um desafio computacional envolvido na obtenção do nonce, a chave para o quebra-cabeças do protocolo da blockchain.

Será considerado agora um quebra-cabeça de hash real para ilustrar o seu funcionamento. Vimos que o valor de hash abreviado de Hello World! é 7F83B165. Mas qual dado combinado com Hello World! produziria um valor de hash abreviado com três zeros na frente? Portanto, este é o quebra-cabeça de hash: encontre o nonce que, combinado com Hello World!, produza um valor de hash abreviado com três zeros na frente. Experimentaremos alguns nonces. A Tabela 1 mostra o nonce, o texto cujo hash será gerado e o valor de hash abreviado resultante. Como se pode ver, o nonce 614 resolve o quebra-cabeça de hash, o que implica que, se começar com um nonce 0 e for incrementado sequencialmente, seriam necessários 615 passos para encontrar a solução. Se a restrição fosse encontrar um valor de hash com 1 (hum) zero na frente, esse quebra-

cabeça teria sido resolvido depois de quatro passos, pois Hello World!3 gera um valor de hash com 1 (hum) zero na frente.

Nonce	Texto cujo hash sera gerado	Saida
0	Hello World!0	4EE4B774
1	Hello World!1	3345B9A3
2	Hello World!2	72040842
3	Hello World!3	02307D5F
	...	
613	Hello World!613	E861901E
614	Hello World!614	00068A3C
615	Hello World!615	5EB7483F

Tabela 1: Nonces para resolver um quebra-cabeça de hash

É possível fazer o teste em www.blockchain-basics.com/HashPuzzle.html. Parte essencial do quebra-cabeça de hash é exigir que o valor de hash atenda a uma determinada restrição. Assim sendo, a sua descrição e a restrição não são arbitrárias. Os quebra-cabeças de hash utilizam restrição padronizada, de tal forma que outros são desafiados por computadores com tais quebra-cabeças. As restrições em geral são chamadas de dificuldade ou nível de dificuldade em se tratando de quebra-cabeças de hash. Um número natural expressa a dificuldade e se refere ao total de zeros na frente que o hash deve ter de valor. Portanto, dificuldade igual a 1 (hum) implica que o valor de hash deve ter (pelo menos) um zero na frente, enquanto uma dificuldade igual a 10 (dez) quer dizer que o valor de hash deve ter no mínimo 10 zeros na frente. Mais complicado será o quebra-cabeça de hash e mais zeros na frente são necessários quanto maior o nível de dificuldade. Mais tempo ou capacidade de processamento serão necessários para resolvê-lo quanto mais complicado o quebra-cabeça.

A métrica de dificuldade mostrada na Figura 30, ao longo de dois anos é medida como a razão da dificuldade atual pela dificuldade mínima (a dificuldade do primeiro bloco). Após achar o fatídico nonce, o nó sortudo informa aos demais que conseguiu resolver o desafio primeiro que os demais e, para provar o seu trabalho, informa o número do nonce e o conjunto de transações as quais serão inclusas nesse bloco. Os demais nós facilmente atestam o nonce e o bloco enfim é fechado (minerado), partindo para o próximo bloco. Antes disso são creditadas na carteira do nó sortudo algumas criptomoedas, visto

que o vencedor minerou e através de seu esforço computacional conseguiu achar o nonce e fechar um bloco.

Lembrando que no cabeçalho do próximo bloco existe um hash gerado a partir do cabeçalho do anterior o que inclui tanto o nonce quanto o hash do cabeçalho anterior ao anterior. Das transações é gerado um hash que é incluso no cabeçalho do próprio bloco junto com o nonce. O hash do cabeçalho, por sua vez, compõe o cabeçalho do bloco em seguida, formando uma cadeia de blocos: a blockchain. Desta forma temos uma sequência de blocos encadeados (blockchain) garantido o estado de todos os blocos visto que caso se altere qualquer mínima informação numa transação, será necessário gerar outro hash da transação e ir atrás de outro nonce. E se não for do bloco imediatamente anterior, o pretenso hacker teria que fazer isso em todos os blocos posteriores. Mesmo se ele tivesse poder computacional pra isso, o comportamento anômalo dele na rede já seria notado e teria sido desconectado pelos nós saudáveis, tudo isso graças a outra grande premissa da blockchain: o **consenso**.



Figura 30: Métrica de dificuldade de mineração do Bitcoin, ao longo de dois anos. Disponível em [3]

Neste momento o leitor é encorajado a visitar no Apêndice os detalhes matemáticos sobre o aprofundamento dos cálculos de funções hash criptográficas do protocolo bitcoin (SHA-256 e RIPEMD-160) bem como da soma de pontos das Curvas Elípticas.

Endereço Bitcoin

A seguir está uma breve visão geral de como funciona a geração de endereços Bitcoin, explicação disponível no endereço eletrônico [59] resumido pelo diagrama a seguir (Figura 31):

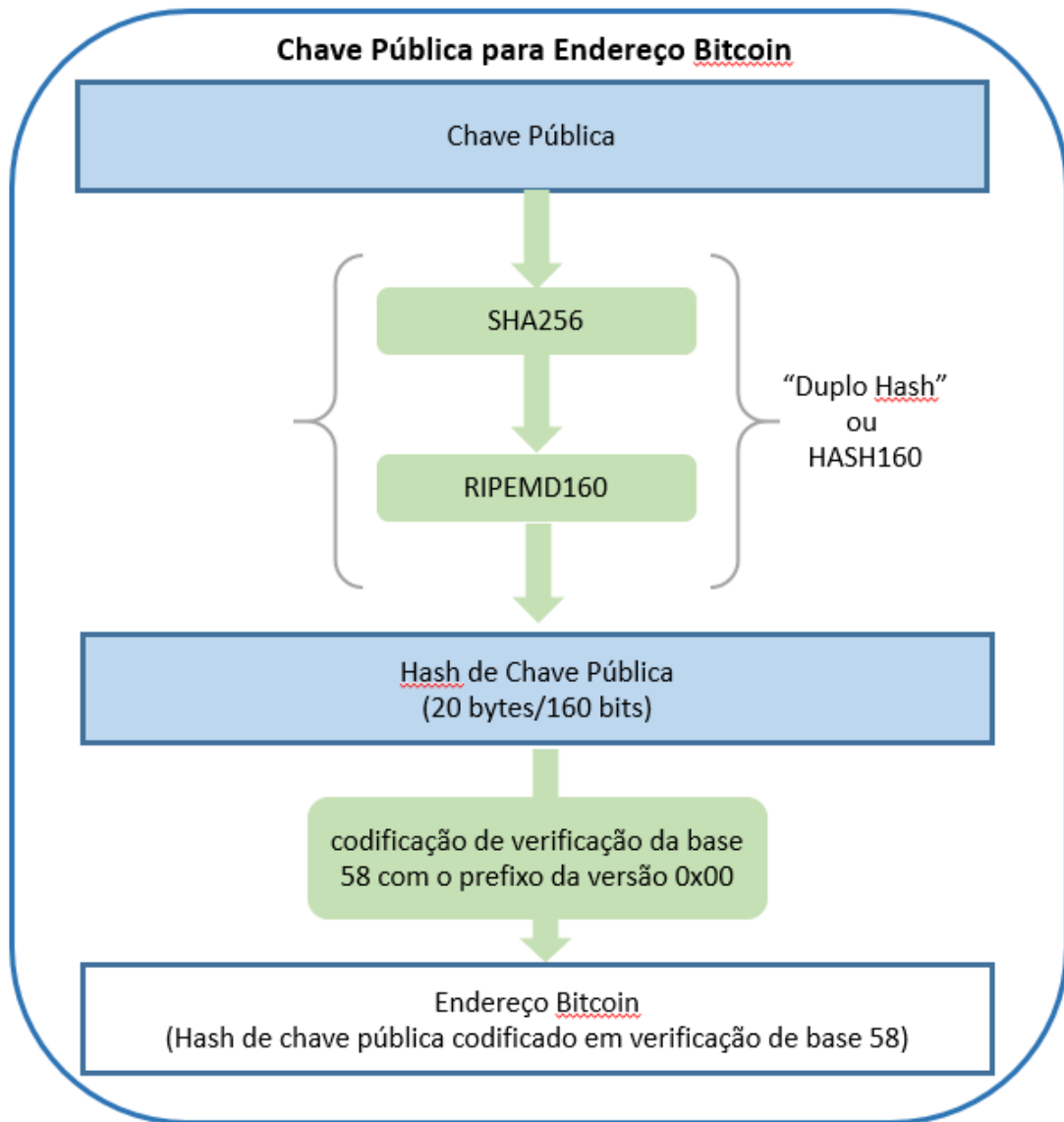


Figura 31: Chave pública para endereço bitcoin: conversão de uma chave pública em um endereço bitcoin. Adaptado de [3]

Os próximos passos da criação de endereços bitcois foram apresentados no seguinte artigo [59]:

1. Ter uma chave privada do tipo ECDSA.
2. Pegar a chave pública correspondente gerada a partir da chave privada gerada (33 bytes, 1 byte 0x02 (y-coord é par) e 32 bytes correspondentes à coordenada X)
3. Executar a função SHA-256 na chave pública do item 2
4. Executar a função hash RIPEMD-160 no resultado de SHA-256 do item 3
5. Adicionar byte de versão na frente do hash RIPEMD-160 (0x00 para rede principal)
6. Executar o hash SHA-256 no resultado estendido do RIPEMD-160
7. Executar o hash SHA-256 no resultado do hash SHA-256 anterior
8. Pegar os primeiros 4 bytes do segundo hash SHA-256. Este é o endereço checksum.
9. Adicionar os 4 bytes de soma de verificação do estágio 7 no final do hash RIPEMD-160 estendido do estágio 4. Esse é o Endereço Bitcoin binário de 25 bytes.
10. Converter o resultado de uma cadeia de bytes em um caracter base58 usando a codificação Base58Check.

No endereço eletrônico <https://gobittest.appspot.com/Address> é possível testar os passos acima da geração de Endereços Bitcoin.

Blockchain

O blockchain é um sistema ponto a ponto puramente distribuído para gerenciamento de posses. É constituído de computadores individuais que mantêm a própria versão de um livro-razão imutável no qual é armazenado o histórico completo dos dados de transação. Desse modo, os computadores individuais são equivalentes às testemunhas que podem afirmar se determinada transação ocorreu de acordo com suas próprias memórias. Um sistema ponto a ponto puramente distribuído não tem nenhum ponto central de coordenação ou controle. Portanto, não há nenhum componente central que dissemine informações a todos os computadores que compõem o sistema. A comunicação entre os nós que compõem o sistema ponto a ponto distribuído tem os três propósitos a seguir:

- manter as conexões existentes ativas;
- estabelecer novas conexões;
- distribuir novas informações.

Sua finalidade é fazer o gerenciamento de posses. Assim, o terceiro tipo de comunicação tem como foco a adição de novos dados de transação e de novos blocos à estrutura de dados blockchain. É importante compreender que os conceitos principais que constituem o blockchain dependem de outros conceitos e tecnologias. Compreender o blockchain exige, no mínimo, uma apreciação desses conceitos também.

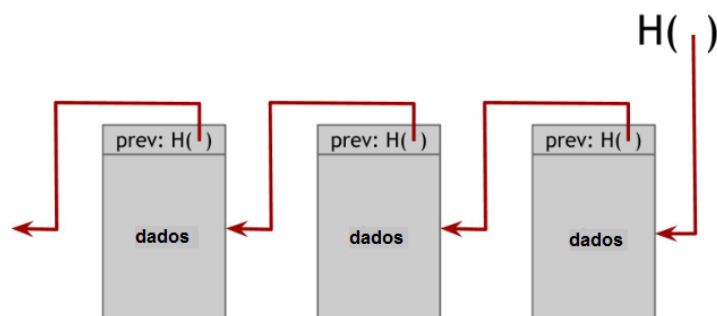


Figura 32: Uma Blockchain é uma lista vinculada criada com ponteiros de hash. Adaptado de [37]

O blockchain é um sistema Peer-to-peer aberto a qualquer um. Quem quiser pode se conectar e submeter novos dados de transação ao sistema ou contribuir com recursos computacionais.

A exclusividade é uma característica constituinte da propriedade privada. O direito de transferir a posse para outra conta deve estar restrito ao dono da conta que cede a posse. Assim, Drescher, 2017 [16] afirma que o desafio do blockchain é proteger a propriedade atribuída às contas sem restringir a arquitetura aberta do sistema distribuído.

Segundo Swan, 2015 [54] os diferentes tipos de atividades em potencial do blockchain são divididas em três categorias: Blockchain 1.0, 2.0 e 3.0.

- Blockchain 1.0 é moeda, a implantação de criptomoedas em aplicativos relacionados a dinheiro, como transferência de moeda, remessa, e sistemas de pagamento digital;
- Blockchain 2.0 são contratos, toda a lista econômica, mercado e aplicativos financeiros usando o blockchain mais extensos do que transações simples em dinheiro: ações, títulos, futuros, empréstimos, hipotecas, títulos, propriedades e contratos inteligentes;
- Blockchain 3.0 são aplicativos blockchain além da moeda, finanças e mercados - particularmente nas áreas de governo, saúde, ciência, literatura, cultura, arte e também de educação.

A Tabela 2 resume as tecnologias que compõem o blockchain em um nível mais detalhado:

Conceito	Propósito
Dados de transação	Descrever a transferência da posse
Histórico de transações	Apresentar o estado atual da posse
Valor de hash criptográfico	Identificar qualquer tipo de dado unicamente
Criptografia assimétrica	Criptografar e descriptografar dados
Assinatura digital	Declarar concordância com o conteúdo dos dados de transação
Referência de hash	Uma referência que se torna inválida assim que os dados referenciados são alterados
Estruturas de dados sensíveis a mudanças	Armazenar dados de modo que qualquer manipulação se torne evidente de imediato
Quebra-cabeça de hash	Impor uma tarefa custosa do ponto de vista de processamento
Estrutura de dados blockchain	Armazenar dados de transação de modo sensível a mudanças e manter a sua ordem
Imutabilidade	Tornar impossível a alteração do histórico de dados de transação
Rede ponto a ponto distribuída	Compartilhar o histórico de transações entre todos os nós de uma rede
Transmissão de mensagens	Garantir que todos os nós do sistema recebam todas as informações em algum momento
Algoritmo de blockchain	Garantir que somente dados de transação válidos sejam adicionados à estrutura de dados blockchain
Consenso distribuído	Garantir que todos os nós do sistema utilizem um histórico de dados de transação idêntico
Pagamento	Dar um incentivo aos nós por manter a integridade

Tabela 2: Conceitos técnicos do blockchain e seus propósitos. Adaptado de [16]

Proposta Didática

Neste capítulo, apresentaremos uma proposta sugerida inicialmente por Kononenko, 2018 [31] e adaptada para o nosso contexto cuja didática visa a facilitar o entendimento do blockchain a alunos da educação de ensino médio. Serão apresentadas analogias lúdicas utilizando conceitos do próprio ambiente escolar dos alunos.

Quem já tentou aprender o básico do blockchain percebe que fica técnico rapidamente sendo apresentado de cara a conceitos como:

- “Livro distribuído”
- ”Hash criptográfico”
- ”Assinaturas digitais”

Embora o aluno certamente possa perseverar durante a pesquisa inicial, ele precisa entender uma série de novos conceitos técnicos antes de entender todo o sistema. É tão difícil porque Bitcoin (e blockchain) são baseados em um paradigma distribuído e descentralizado. Estamos acostumados a autoridades centralizadas e confiáveis, como bancos, prestadores de serviços de saúde e corporações.

Cada uma dessas instituições possui sistemas complicados para manter a alta qualidade. A fim de manter esses mesmos padrões para produtos vitais sem a autoridade centralizada os alunos precisam de novas regras complicadas para manter os sistemas descentralizados também. Então, a didática proposta é criar uma nova escola chamada “Escola Secundária Distribuída” que operará usando os princípios do blockchain. Vamos criar uma nova maneira de classificar as tarefas de uma aula de matemática usando um sistema distribuído. Os alunos poderão manter o sistema de notas sem a ajuda de professores.

Administrando uma escola secundária de maneira distribuída

Primeiramente, a proposta será direcionada para os alunos calouros no ensino médio (os alunos do 1º ano) que estão fazendo uma aula de matemática. Para passar na aula, o aluno precisa obter uma pontuação suficiente nas tarefas de casa, testes e provas. Existem 30 alunos no total da turma, mas o professor poderá adaptar para o total de alunos de sua classe. Tudo isso é gerenciado por uma autoridade centralizada - o **professor**. Esta pessoa avalia todas as tarefas, entrega os boletins a cada trimestre e garante que ninguém esteja trapaceando nos testes, o que arruinaria a integridade da turma.

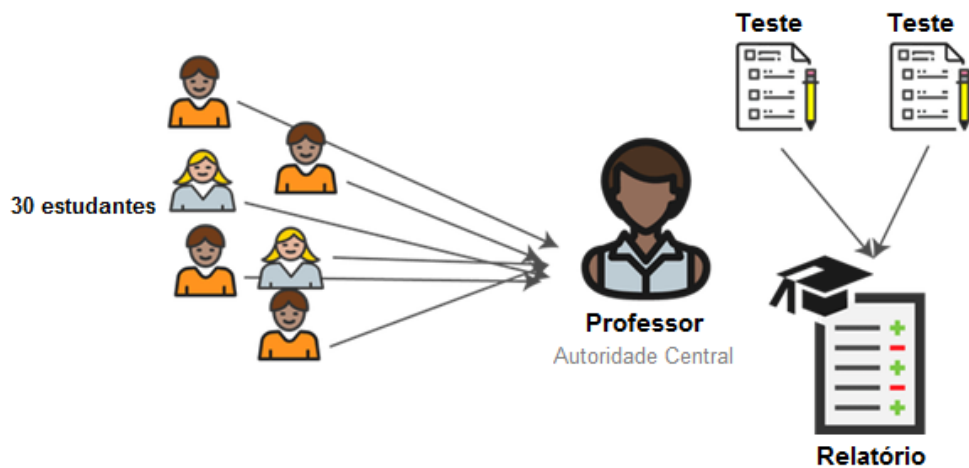


Figura 33: Professor como autoridade central. Adaptado de [31]

Embora este seja o sistema que os alunos estão acostumados, na verdade tem algumas falhas importantes:

- **Ineficiente** os alunos demoram a se lembrar do que aconteceu quando recebem 30 testes de um professor. O professor levará bastante tempo para corrigi-los porque leva algum tempo para corrigir 30 testes.
- **Arriscado** os alunos provavelmente já viram um professor que perde o teste de um avaliado. Ou aquele que mantém seus testes em um arquivo em um quarto, que pega fogo um dia e, de repente, os testes estão corrompidos. Ambos acontecem e os professores lidam com tantas tarefas que pode haver algum erro.
- **Corruptível** E se o aluno for o encrenqueiro da classe? Quando o professor se senta

para corrigir seu teste, ele pode dar uma olhada no nome na parte superior do teste e tornar-se instantaneamente parcial ao classificar o teste. Isso pode ser intencional ou não intencional.

- **Custoso** Todo o tempo gasto em provas pode não estar adicionando valor aos alunos ou à sociedade. Esta também é provavelmente a parte menos preferida do trabalho pelo professor. Eles provavelmente se tornaram professores para que pudessem ajudar os alunos a aprender, não somente para aplicar testes.

Podemos ver problemas semelhantes com outros sistemas centralizados. Por exemplo, o setor bancário por meio de práticas arriscadas passou pela crise financeira de 2008 que exigiram um resgate massivo de recursos.

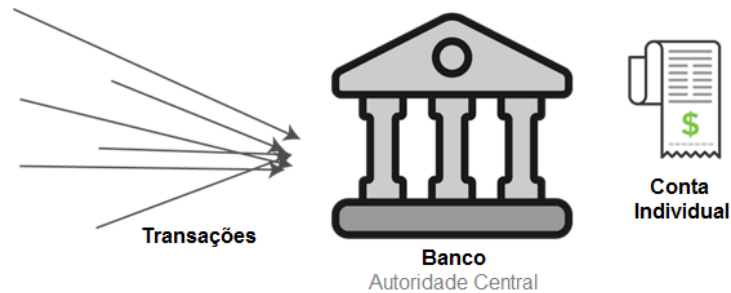


Figura 34: Banco como autoridade central. Adaptado de [31]

A esta altura, os alunos podem estar se perguntando: como vamos resolver todos esses problemas, removendo a influência do professor, a única pessoa com mais experiência neste sistema? Como impediríamos que isso se tornasse uma anarquia? É aí que entra o conceito do blockchain. Antes de entrar na maneira específica, usaremos o blockchain para criar uma nova maneira para a Escola Secundária Distribuída funcionar, onde os alunos devem saber que cada blockchain tem regras específicas que são instituídas pelo seu criador. No exemplo do Bitcoin, seria “Satoshi Nakamoto” [36], que escreveu o livro branco original e criou as regras (algoritmos) que lhe permitiram operar sem intervenção humana. No nosso exemplo didático, teremos um **diretor** que mudou as regras.

Blockchain para o ensino médio distribuído

Enquanto um professor classifica testes e gerencia notas em particular, um blockchain torna todas as transações públicas. Portanto, blockchain não dependerá de nenhuma autoridade central, além da pessoa que a criou. Isso significa que, na Escola Secundária Distribuída, os alunos do 1^a ano vão classificar os testes uns dos outros. Digamos que é o dia do teste e o período da aula é de uma hora. Os alunos empilham seus testes em cima da mesa do professor quando terminam. Mas, em vez de levar todos os testes para casa, o professor mistura todos eles em uma grande pilha e pede a cada aluno que faça um teste aleatório e o classifique com uma chave de resposta.

Isso é conhecido como uma **transação**. Esta é a unidade fundamental que compõe um blockchain. Digamos que um aluno, Bob, deu a outra aluna, Alice, uma nota de 84. Nesse caso, Bob é o remetente e Alice é a destinatária.

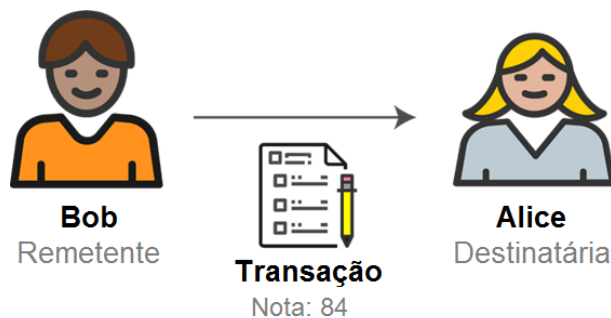


Figura 35: Transação. Adaptado de [31]

Em termos de Bitcoin, isso não seria aleatório - o aluno sabe para onde está enviando dinheiro.

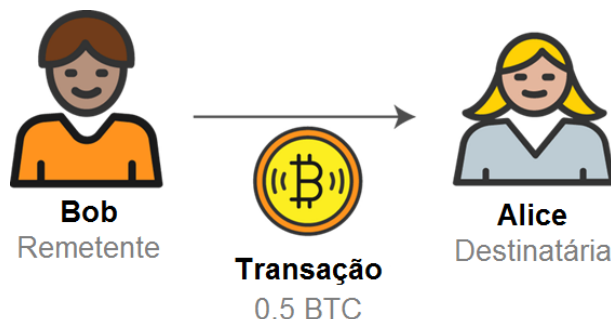


Figura 36: Transação de bitcoin. Adaptado de [31]

Até agora, resolvemos os problemas de velocidade e custo. Os professores não precisam mais dedicar tempo à avaliação e cada aluno pode avaliar um outro teste

rapidamente. Mas ainda há um enorme potencial para fraudes. Isso é bem próximo da anarquia. É preciso haver uma rede de pessoas responsáveis que mantenha todos os participantes honestos. É aqui que entram as políticas do diretor. O diretor controla a única coisa com a qual todos se importam - o sistema de classificação. Na Escola Secundária Distribuída, o diretor decide permitir que os mais antigos (alunos do 3º ano) administrem esse sistema blockchain em troca de uma recompensa. Se um veterano fizer uma revisão de 20 desses testes em 1 dia, ele poderá participar de uma competição para obter 10 pontos adicionados a um de seus próprios testes.

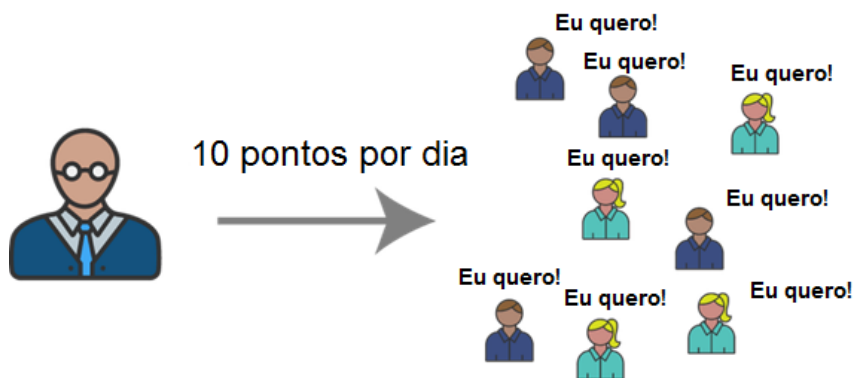


Figura 37: Veterano. Adaptado de [31]

Esse conjunto de 20 transações é conhecido como um **bloco** e, eventualmente, mostraremos como todos os blocos funcionam juntos para formar um **blockchain**.

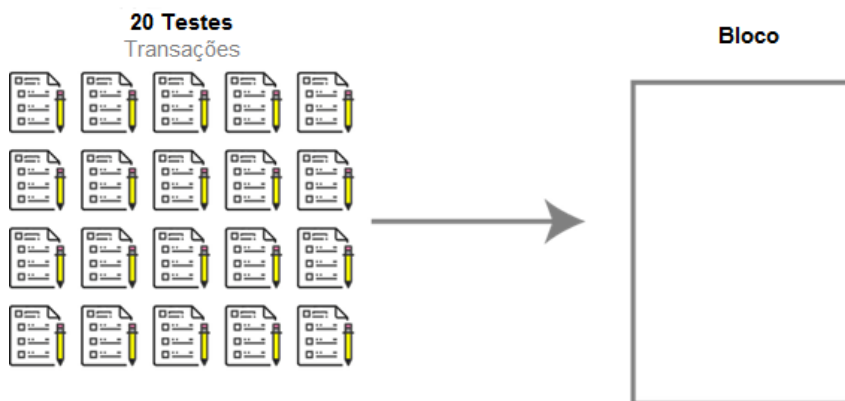


Figura 38: Um bloco. Adaptado de [31]

A esta altura, um aluno pode perguntar: "Por que só os mais antigos podem fazer isso? E por que deve ser uma competição?" Deve ser mais antigo porque o diretor precisa de participantes que possam lidar com a carga de trabalho dos testes de classificação todos os dias, se assim o desejarem. Se o sistema ficar lento, ninguém teria seus testes validados e registrados, o que seria uma falha completa. E deve ser uma competição

para que o sistema de pontos não seja completamente desvalorizado. Imagine se todos os alunos revisassem seus 20 testes e recebessem 10 pontos em seu próprio teste? A inflação seria desenfreada, semelhante à forma como a inflação de moeda aumenta quando o governo imprime mais dinheiro. Deve haver uma competição por um número escasso de pontos. O diretor não está obrigando nenhum aluno mais antigo a participar, mas há um forte incentivo para que eles o façam.

Uma introdução ao sistema distribuído

Agora sabemos como um teste é classificado (uma transação) e os incentivos para que os mais antigos mantenham o sistema com integridade - eles obtêm mais pontos revendo e validando mais testes. Mas ainda estamos perdendo toda a infraestrutura distribuída de como esse trabalho realmente é feito. Digamos que 10 alunos mais antigos levaram sua oferta ao diretor. Eles querem fazer parte desta competição para ganhar mais pontos em seu próprio teste. Outro grupo de 10 mais antigos decide se voluntariar para ajudar a manter o sistema, mas não participa da competição. Isto está fora do suporte para o sistema distribuído e parte do espírito do movimento em direção à classificação de código aberto.

Cada um desses alunos mais antigos é um nó completo na rede - eles se comunicam em tempo real sobre novas transações e blocos. Os 10 mais antigos que decidiram participar da competição são chamados de **mineradores**. Eles constroem blocos com transações disponíveis no **banco de dados**, o reservatório de transações não confirmadas. Quando um aluno, como Bob, termina de classificar um teste, o aluno transmite uma transação não validada para a rede de mais antigos. Cada nó completo compartilha com todos os outros, como um boato. Torna-se parte do mempool (conjunto de memória).

Todo nó deve **validar** a transação primeiro, em outras palavras, determinar se foi possível ou não. Neste exemplo, validação pode significar confirmar que o avaliador realmente classificou o teste corretamente, digitando todas as respostas finais na sua calculadora. Após a validação, cada minerador tem a oportunidade de construir seu próprio bloco a partir de 20 testes ou transações. Nos dias de teste, 30 novas transações devem ser adicionadas à rede, pois são 30 alunos da turma. Um aluno pode perguntar: "Como os mineradores escolhem as transações para adicionar ao seu bloco?"

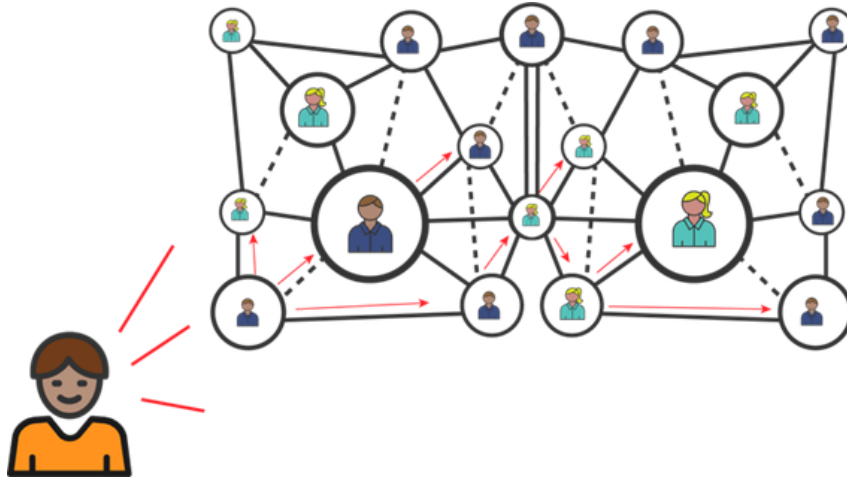


Figura 39: Conjunto de memória. Disponível em [31]

A resposta é uma **taxa de transação**. Cada remetente deve anexar uma taxa de transação à sua transação para compensar os mineradores pelo seu trabalho. Assim, os mineradores geralmente optam por colocar todas as transações com as taxas mais altas em seus blocos imediatamente. Como isso opera na oferta e na demanda, eles podem incluir as transações com taxas mais baixas nos dias em que há menos testes para validar. No nosso exemplo de escola, esta taxa de transação pode ser um ponto fora do teste do remetente para doar ao minerador. Não sairia do teste de Alice. Em Bitcoin, seria uma pequena fração de um Bitcoin, como 0,000003 BTC. O remetente paga a taxa, pois é a maneira mais fácil de lidar com a logística.

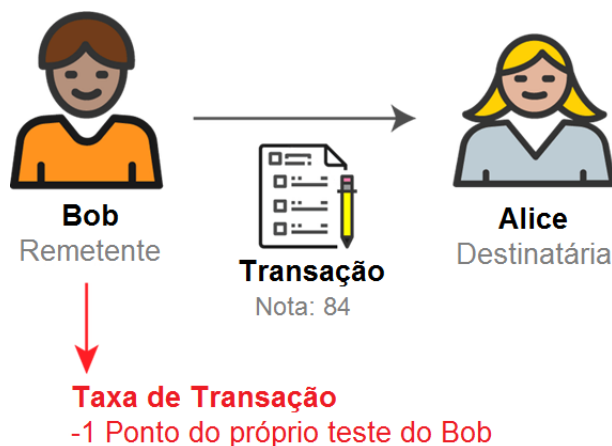


Figura 40: Bob classifica Alice. Adaptado de [31]

Neste ponto, cada minerador tem seu bloco de 20 transações validadas que gostaria de adicionar ao blockchain. Agora, é hora de a competição ver qual dos 10 mineradores terá seu bloco aceito e receber os pontos do diretor.

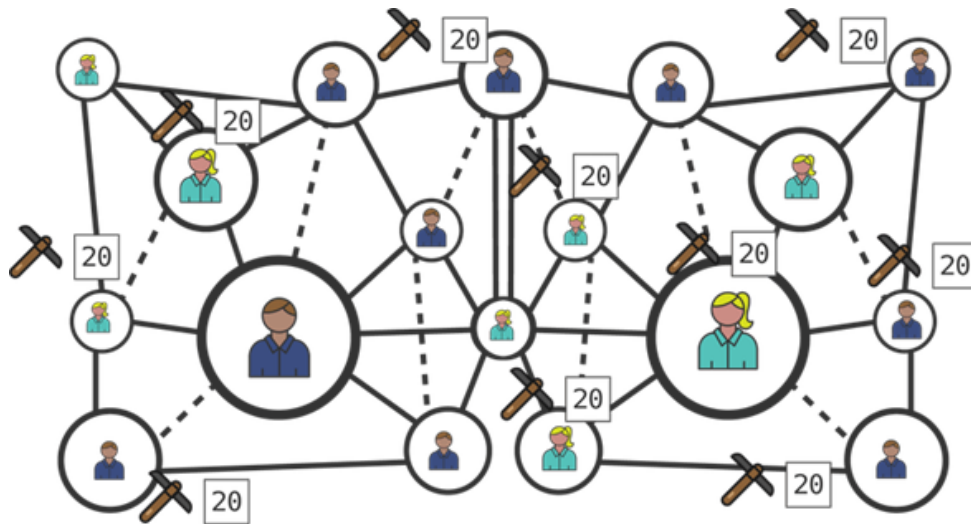


Figura 41: Mineração. Disponível em [31]

Neste ponto os alunos estão começando a ver a quantidade de redundância ou trabalho repetido. Cada bloco proposto terá muitas transações (testes) em comum. Essa é uma medida de segurança necessária para executar um sistema distribuído. Se todos os nós estiverem validando transações separadamente, isso dificulta muito a trapaça do sistema.

A corrida pela prova de trabalho

Agora, vamos solicitar aos alunos que imaginem que depois de todo esse trabalho para criar um bloco de 20 transações, o diretor então compartilhou um problema de matemática de nível 12 para cada minerador, e a pessoa que resolveu o problema primeiro recebeu todos os pontos e teve seu bloqueio confirmado. Isso significaria que os “ricos ficam mais ricos” e distorceria os incentivos para todo o sistema. Todos os dias, os melhores alunos de matemática teriam uma excelente chance de vencer a competição, e o resto dos mais antigos teria pouca ou nenhuma chance. Logo, os mineradores / mais antigos deixariam de participar, já que nunca receberiam pontos.

Então, em vez disso, nosso diretor vai organizar uma caçada na escola todas as noites. Importante, a caça ao tesouro não tem nada a ver com a capacidade matemática de um minerador. Isso encoraja a todos a continuarem a minerar. O diretor esconderá um troféu em algum lugar da escola. Os alunos devem correr até encontrarem e gritarem para que o resto dos alunos em toda a escola possa confirmar que eles o encontraram e ir

para casa. Como esse diretor tem algumas habilidades mágicas de previsão, ele esconde o troféu em um lugar perfeito, de modo que levará **uma hora para encontrar todas as noites**.

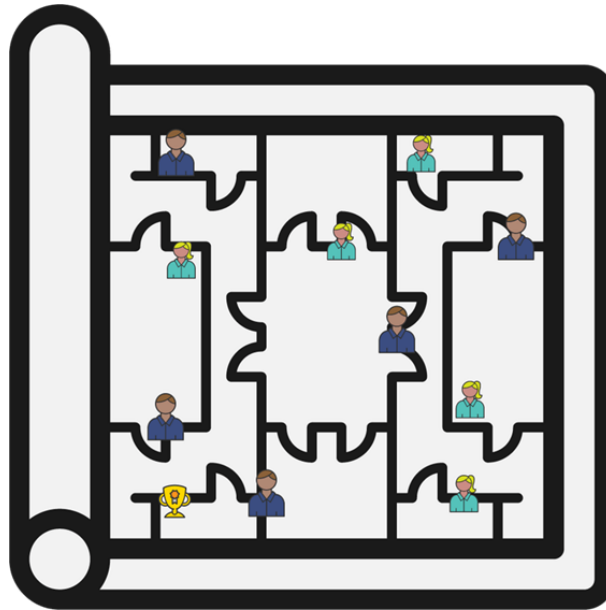


Figura 42: Caça ao Tesouro. Disponível em [31]

Esse processo não deve estar relacionado com a validação de teste, para que possa nivelar o campo de jogo. Isso é conhecido como “prova de trabalho” no Bitcoin. É um algoritmo difícil de resolver, mas fácil para os outros nós confirmarem depois de resolvido. Cada minerador de Bitcoin deve adivinhar números até escolher o correto que resolve o quebra-cabeça. Em Bitcoin, um novo bloco é confirmado a cada 10 minutos, em média.

O algoritmo também fica gradualmente mais difícil com o tempo, à medida que mais mineradores se juntam à rede. Se houver mais mineradores, isso significa que haverá mais suposições, então o desafio deve ficar mais difícil se o Bitcoin quiser continuar confirmando um bloco a cada 10 minutos.

Este exemplo demonstra como o Bitcoin (e nossa aula didática com exemplo escolar) força cada minerador a competir com o resto da rede. Uma vez que um minerador resolve o quebra-cabeça, eles compartilham sua resposta com o resto da rede, o que pode ser rapidamente confirmado. Depois que os nós chegarem a um consenso, ou mais de 50% concordarem que o bloco está confirmado, ele pode ser adicionado ao blockchain.

Isso motivou alguns mineradores a formarem **grêmios**. Em nossa aula com exemplo escolar, isso significa que alguns dos alunos concordariam em dividir os pontos

quando um deles encontrasse o troféu. Simplesmente aumenta a probabilidade de que o primeiro a encontrar o troféu seja um membro de sua equipe.

Em Bitcoin, o poder de computação total que está trabalhando na solução desse “quebra-cabeça prova de trabalho” é chamado de **taxa de hash**. As maiores agremiações de Bitcoin controlam cerca de 10% da taxa de hash, o que ainda dá ao resto dos mineradores uma boa chance de resolver o quebra-cabeça. Se um grêmio contivesse 50% da taxa de hash, haveria menos incentivo para os outros continuarem a minerar.

Quando um bloco é confirmado, o minerador recebe o prêmio (10 pontos em um teste) e todas as taxas de transação das transações são confirmadas. As transações que não faziam parte do bloco retornam ao conjunto de memória para serem incluídas em um bloco futuro.

~ 1 Hora depois

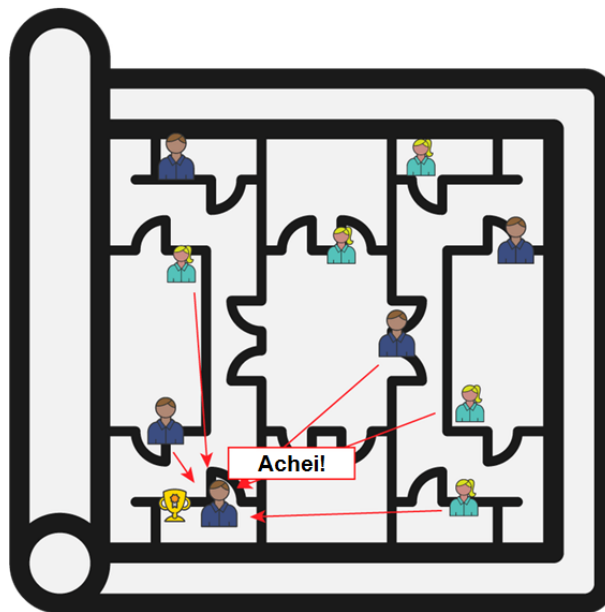


Figura 43: Tesouro descoberto. Disponível em [31]

Construindo um blockchain

Até agora, os alunos conheceram a maioria das etapas que incluem adicionar mais um bloco ao blockchain. Mas, os alunos não conhecem ainda como construir uma blockchain em si. Um blockchain tem uma estrutura simples de três níveis. Uma série de transações compõem um bloco. E uma série de blocos compõem um blockchain. Embora você possa dividir um blockchain em partes com base no tempo, normalmente, cada nó individual (mais antigo) manterá o histórico completo do blockchain.

Em nosso exemplo do ensino médio, estamos estudando uma turma do 1º ano. Assim, a história completa da turma pode ser de todas as notas de todos os alunos de toda a turma, do jardim de infância até hoje. Como estamos adicionando blocos em um intervalo de 1 dia e há aproximadamente 180 dias no ano letivo, isso significa que o blockchain contém cerca de 1700 blocos.

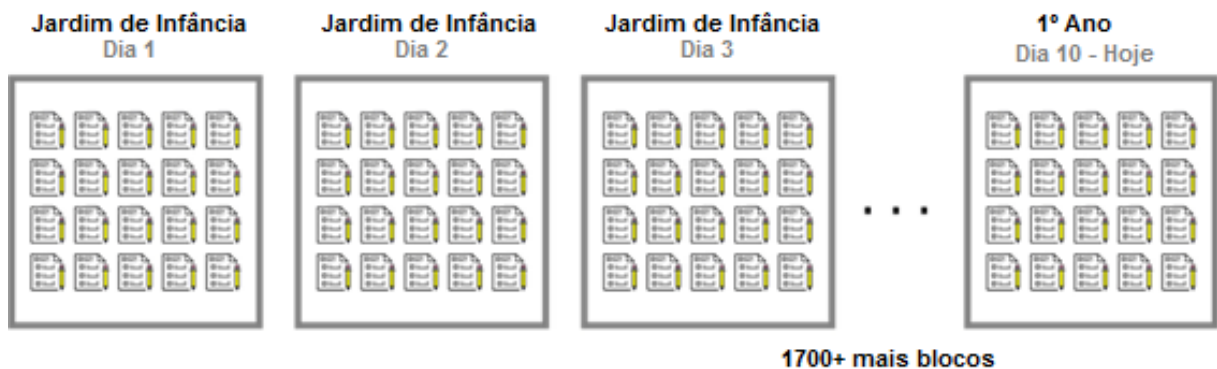


Figura 44: Histórico do jardim de infância ao 1º Ano. Adaptado de [31]

Cada bloco possui uma identificação única, que devido a uma “função hash”, depende da identificação do bloco anterior. Isso é o que protege o blockchain - não existe uma substituição de bloco ou um histórico de reescrita, porque ele mudará a identificação de cada bloco subsequente. Como o nosso exemplo educacional para a aula usa intervalos de um dia, um aluno pode pensar: “Ah, deve ser fácil criar uma identificação única para cada bloco, já que cada data ocorre apenas uma vez!”. Mas, isso introduziria uma vulnerabilidade de segurança. Se um minerador fosse capaz de introduzir um novo bloco em algum lugar no meio da corrente, ele não quebraria o padrão! O minerador trapaceiro poderia facilmente replicar a identificação do bloco, e nenhum dos seguintes blocos mudaria seu valor, já que as datas seguem um padrão confiável que pode ser facilmente replicado.

A função hash torna muito difícil imitar e substituir os blocos.

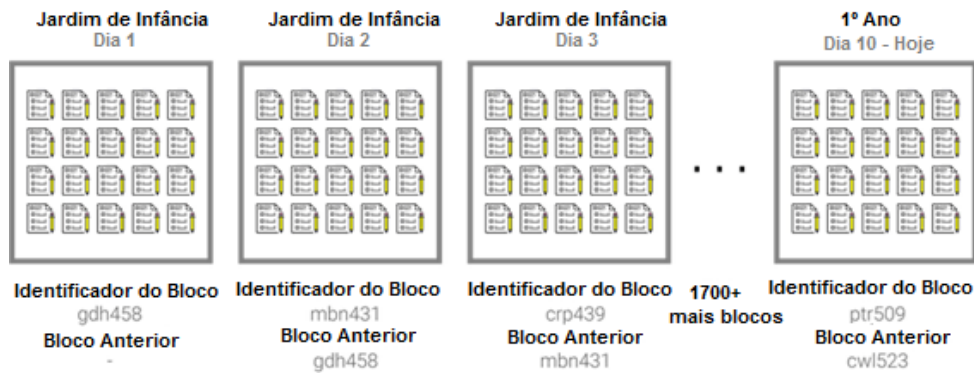


Figura 45: Histórico do jardim de infância ao 1^o Ano com funções hash. Adaptado de [31]

Até agora, parece que todo aluno do 3^a ano pode ver toda a história da série de cada aluno do 1^a ano e não é isso que queremos. Precisamos de algum princípio de privacidade. Mas, no lado positivo, o sistema distribuído permite que cada indivíduo mais antigo garanta a validade dos testes classificados à medida que eles circulam pela rede.

Este sistema de pedidos é **relativo**, e não absoluto. A ordem dos blocos é muito mais importante do que os tempos em que foram adicionados à cadeia. Os selos de tempo, como discutimos acima, são fáceis de copiar e imitar.

Vamos dar um exemplo do que é conhecido como um **ataque de gasto duplo**.

Digamos que uma de suas colegas faça um teste de matemática na segunda-feira e saiba que ela foi mal. Um de seus colegas classifica essa versão do teste e depois a transmite para os nós, como de costume. Seu colega de classe estuda como louco naquela noite e depois aparece no dia seguinte para fazer o mesmo teste com outra turma. Como a professora não estava prestando muita atenção, ela consegue convencer a professora de que ela não estava presente no dia anterior. Lembre-se de que o professor não tem nenhum papel nos testes de classificação, portanto, o professor não pode fazer referência rápida aos exames do dia anterior. Ela tem permissão para fazer o teste novamente e o envia ao resto da turma.

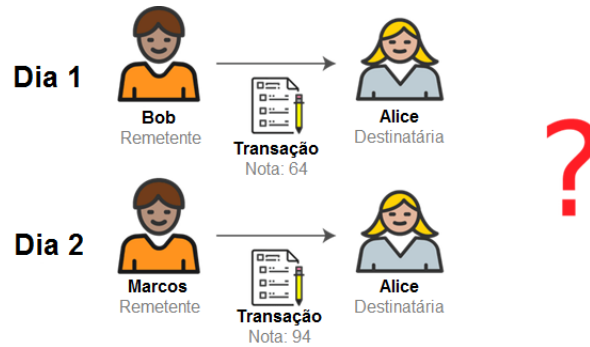


Figura 46: Nova tentativa. Adaptado de [31]

Aqui estão as últimas opções para blocos do blockchain, com a tentativa de Alice de substituir sua nota anterior. Alice agora precisará se tornar um minerador na rede e participar da caçada. Ela é agora a 11^a mineradora dessa rede. A regra é que a “cadeia mais longa ganha”. Isso significa que no dia 11, o resto da rede pode estar trabalhando na adição de um novo bloco com o último conjunto de transações. Mas Alice estará trabalhando em “bifurcação” da cadeia, e adicionando um novo conjunto de transações para o dia 10 com 19 transações em comum, e sua nova pontuação no teste como uma substituição para a pontuação do teste antigo.

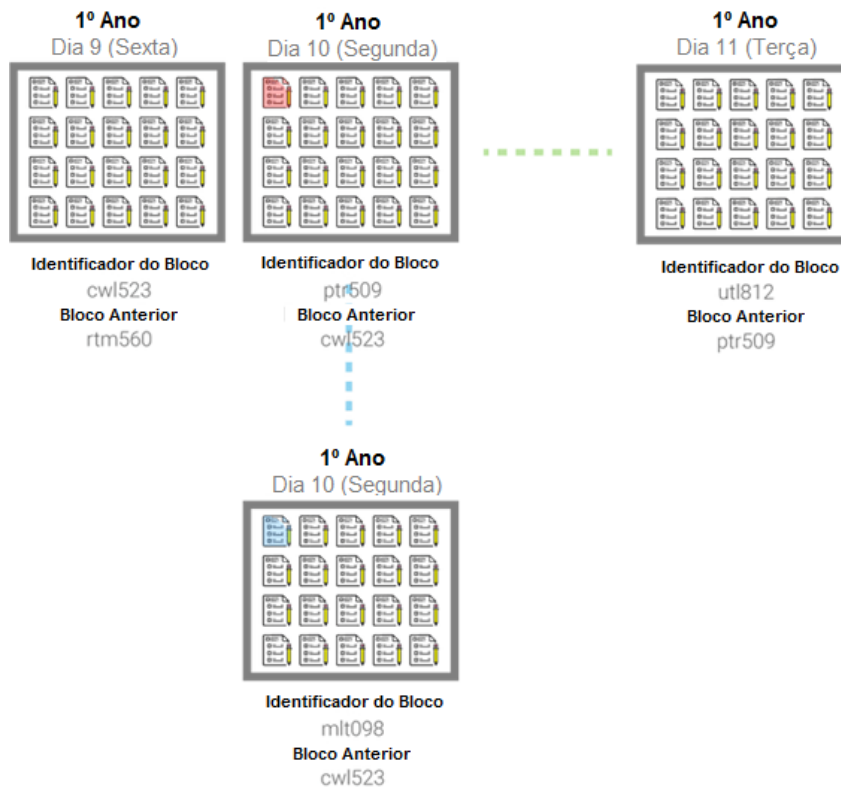


Figura 47: Tentativa de Alice de substituir sua nota anterior. Adaptado de [31]

Bifurcação significa que ela está tentando construir uma nova cadeia mais longa, ao contrário da cadeia que o resto da rede assume ser a mais longa. Se ela puder vencer a caçada naquele dia e depois voltar no dia seguinte e vencê-la novamente, ela terá a maior corrente. Isso faz parte do valor do sistema de “prova de trabalho”. Como Alice é um dos 11 mineradores da rede, ela tem aproximadamente 1% de chance de resolver dois blocos seguidos. Há 99% de chance de ela colocar todo esse trabalho apenas para não receber nada. o que não é um grande incentivo. É também por isso que as identificações de bloco e as identificações de bloco anteriores são um esquema de rotulagem melhor que uma data específica. Se Alice ganhar a corrida no dia em que ela secretamente fizer o teste uma segunda vez, todos os novos testes daquele dia ainda serão armazenados no blockchain. Eles só precisam esperar mais um dia.

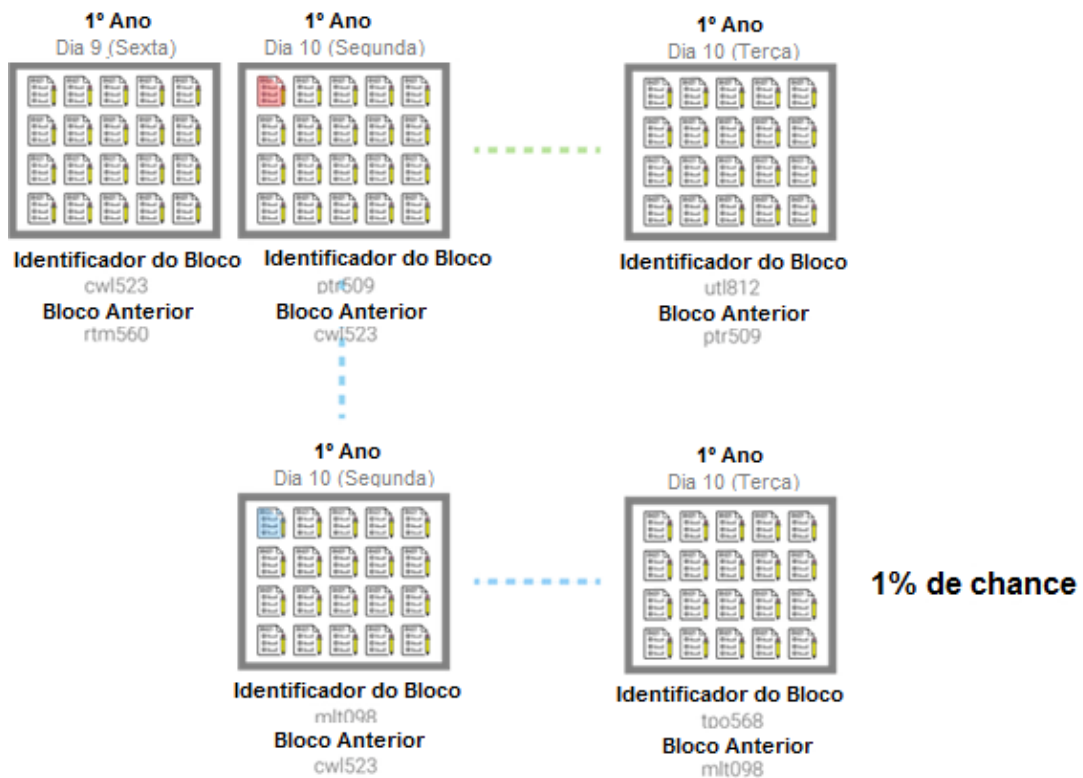


Figura 48: Chance de uma transação voltar no dia seguinte. Adaptado de [31]

Apresentando Chaves Públicas e Privadas

Até agora, cobrimos todos os mecanismos que permitirão aos alunos da escola secundária Distribuída gerenciar suas próprias notas. Estamos perdendo apenas uma coisa importante: privacidade. Neste momento, as notas de cada aluno são expostas para sempre no blockchain. Se isso fosse uma moeda, seria fácil descobrir quanto dinheiro cada pessoa tinha. Não é isso que queremos. Ao mesmo tempo, a transparência é uma ótima maneira de manter as pessoas individualmente responsáveis por classificação injusta e outras práticas fraudulentas. É por isso que o Bitcoin usa um sistema criptográfico com chaves públicas e privadas. No ensino médio, os alunos provavelmente estão acostumado com os armários que ocupam os corredores.

No Bitcoin há um número (essencialmente) ilimitado de combinações de chaves privadas-públicas. Então, em vez disso, imagine que as paredes desta escola secundária estão alinhadas como as pequenas caixas de correio que você vê em um prédio de apartamentos.



Figura 49: Armários. Disponível em [31]

E eles cobrem todas as paredes desta escola. E, como há um número total ilimitado de armários, cada aluno na escola pode possuir um número ilimitado de armários. Em termos matemáticos:

$$\text{Ilimitado} / 30 \text{ alunos} = \text{Ilimitado}$$

Por uma questão de simplicidade, vamos supor que cada aluno receba uma caixa de correio para cada série que eles precisavam para fazer os testes. Se um aluno está no 1º ano, isso significa que ele está usando o 1º vestiário. Vamos voltar para a nossa transação, onde Bob classifica Alice. Os nós completos, os mais antigos, devem primeiro avaliar se Bob está qualificado para receber os testes de matemática do 1º ano. Bob precisa provar a si mesmo. Aqui está uma questão - se Bob anunciar alegremente à

rede que ele classificou a tarefa de Alice, ele corre o risco de expor Alice. E se ela tiver uma nota negativa? Ela não quer que o mundo inteiro saiba disso para sempre! Então, ele deve transmitir enquanto mantém os dois anônimos. Ele pode deslizar aleatoriamente uma nota para um dos nós ... assim como a maioria das fofocas começam!

Então o nó completo compartilharia essa fofoca com o resto da rede. É aqui que entram nossas **chaves públicas**. Quando Bob envia uma nota sobre seu teste para a rede, ele está realmente dizendo:

“Meu endereço de caixa de correio atual é 126900trl.

Para provar que eu estava lá no dia do teste, aqui está a chave de resposta que o professor me deu para classificar este teste específico. (**assinatura digital**) A fim de provar que eu sou de fato um aluno do nono ano em aula de álgebra, aqui estão as notas dos exames finais da aula de matemática a cada ano, do 1^o ao 8^o ano do ensino fundamental, e a chave de resposta para cada um desses testes. (**cadeia de transação**)

”Eu vou estar entregando o teste para a caixa de correio 856734pok”



Figura 50: Transação com hashing. Adaptado de [31]

Isso está levantando duas questões principais:

1. O remetente é a pessoa que ele afirma ser?
2. O remetente está qualificado para ser o remetente (classificar o teste)?

Para responder a primeira pergunta, o Bitcoin usa uma **assinatura digital**. A assinatura digital é exclusiva para todas as transações e é formada com um hash da identificação da transação e da chave privada. Neste caso, isso é como a chave de teste - o aluno só pode possuí-la se estivesse lá no dia de teste específico e o professor a entregasse a ele.

Para a segunda pergunta, lembre-se que no Bitcoin, não há conceito de “conta” ou “saldo de conta”. Se houvesse, Bob poderia apenas compartilhar um número de identificação que provasse que ele estava qualificado.

Para provar que essa chave pública específica (a chave pública de Bob) tem aprovação suficiente, ele deve compartilhar um histórico de teste que todo nó completo possa validar. Dessa forma, todos podem validar o preenchimento do primeiro ao oitavo ano do ensino fundamental. Bob também deve fornecer a chave de resposta para cada um desses testes para provar que ele estava na sala naquele momento. Isso é chamado de **cadeia de transação**. Depois que a transação de Bob for **validada** e incluída em um bloco que foi **confirmado**, ele poderá fazer o teste na caixa de correio de Alice sem conhecimento público.

Como você percebeu na transação acima, Bob teve que acessar os testes dos últimos 8 anos! Este sistema de armários só permite que Bob acesse seus testes. Bob tem um conjunto de 8 chaves privadas. Toda vez que ele começou um novo ano, ele abriu outro armário e colocou suas notas daquele ano no armário.



Figura 51: Armários com chaves. Disponível em [31]

Outros podem passar seus últimos resultados de teste em seu armário, mas só ele pode recuperar os resultados. O protocolo Bitcoin permite que os alunos criem muitas combinações de chaves públicas / privadas dentro de sua carteira. Isso melhora a segurança. O aluno nunca vai querer entregar suas chaves privadas, que são a única maneira de acessar os Bitcoins que foram transferidos para ele. Ao contrário de um banco tradicional, não há ninguém a quem recorrer se o aluno esquecer ou perder uma chave privada. O Bitcoin será bloqueado.

Considerações finais

Nesse ponto, encorajamos a fazer uma revisão das analogias. Assim sendo, temos:

- Testes (Transações)
- Respostas (assinaturas digitais)
- Turma do 1^a ano (remetentes e destinatários)
- Alunos do 3^o ano (nós completos)
- Diretor (criador de blockchain)
- Armários com chaves (chaves pública / privada)
- **Sem** professores (autoridade centralizada)
- **Não** há boletins (contas / saldos de contas)

Nessa proposta didática, os alunos puderam compreender que para administrar um sistema (seja ele educacional ou mesmo bancário) a confiança em uma autoridade central desempenha um papel muito importante.

Para devolver esse controle a pessoas individuais (alunos), deve haver uma grande quantidade de redundância dos nós para evitar fraudes, bem como protocolos de segurança cuidadosos (como as assinaturas digitais, as funções hash e as chaves públicas e privadas) para evitar que hackers se infiltrem no sistema.

Constata-se que realmente é possível criar uma proposta didática com exemplos lúdicos de objetos do ambiente escolar e de sala de aula com a finalidade de gerar analogias com os termos do Blockchain, visando facilitar o entendimento para os alunos e até mesmo para os professores.

Conclusão

Esse trabalho objetivou apresentar uma introdução aos fundamentos de bitcoins e blockchains. Para tanto, introduzimos o que é o Bitcoin e apresentamos as definições dos principais elementos que compõem o blockchain, bem como os princípios básicos da criptografia de hash, começando por umas das mais elementares: a hash de aritmética modular até chegar às funções hash do protocolo Bitcoin (Apêndice). Introduzimos também a criptografia por curvas elípticas.

O potencial dos bitcoins e do blockchain são enormes. Seja como nova forma de troca de mercadorias, seja como local de registro de contratos, é notável como ainda há o que ser explorado na tecnologia. A educação superior também pode fazer uso da tecnologia, uma vez que, assim como o protocolo da internet nasceu no ambiente acadêmico, uma criptomoeda para as universidades poderia contribuir para o desenvolvimento da ciência e da pesquisa, ou como forma de fomento, ou como forma de pagamento único das compras feitas pelas universidades.

No que tange ao ensino básico, com o crescimento cada vez maior de aceitação de pagamentos por bitcoins e o surgimento das "fintechs" é notável como é importante que os alunos do ensino básico detenham conhecimento mínimo do tema para que possam não só discorrer na sociedade que os cercam como também entender perfeitamente o funcionamento desse novo meio de pagamento.

Santos, 2018 [45] informa que soluções de blockchain aplicadas à educação estão presentes em instituições renomadas, como no Massachusetts Institute of Technology e na Holberton School, em San Francisco (EUA), onde o armazenamento e a entrega de diplomas emitidos eletronicamente são feitos utilizando-se a blockchain pública da Bitcoin, por meio de pagamento de um pequeno valor a cada diploma gerado, como medida para evitar falsificação desses documentos.

Por fim, tendo em vista a computação quântica ser uma ameaça real às criptomoedas, Aggarwal et al [1], sugerimos como proposta de pesquisas futuras, o estudo de blockchains com funções hash e proof-of-works alternativas às do protocolo Bitcoin.

Apêndice

Curvas Elípticas

Segundo Martins, 2018 [33] no campo da criptografia, as curvas elípticas utilizadas são definidas sobre corpos finitos. Um corpo é um conjunto com duas operações, normalmente soma e multiplicação, que satisfazem propriedades usuais das mesmas operações com números reais. A soma deve ser comutativa, associativa, possuir elemento neutro e elemento simétrico. Já a multiplicação deve ser comutativa, associativa, distributiva, possuir elemento neutro, e todo elemento não-nulo deve possuir um inverso.

Segundo Correia, 2013 [15] o que torna curvas elípticas particularmente interessantes do ponto de vista algébrico/aritmético é o fato de que toda curva elíptica é um grupo abeliano. Isso quer dizer que podemos “SOMAR” dois pontos de uma curva elíptica E , obtendo um terceiro ponto $R = P + Q$ de E , e esta operação goza das propriedades: comutatividade, existência de elemento neutro, existência do elemento inverso e associatividade.

Esta lei de grupo possui a seguinte descrição geométrica, conhecida popularmente como “lei da corda-tangente”:

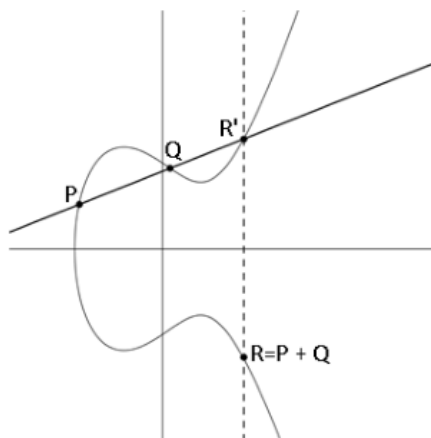


Figura 52: Operação de soma em uma curva elíptica. Disponível em [15]

LEI DA CORDA-TANGENTE: A soma de dois pontos P e Q é a reflexão R , pelo eixo- x , do terceiro ponto R' de interseção da curva elíptica E com a reta que liga P e Q .

Para somarmos os pontos P e Q da curva, traçamos a reta PQ e marcamos o ponto R' , outra interseção dessa reta com a curva. Pelo ponto R' traçamos uma reta perpendicular ao eixo horizontal. A interseção dessa reta com a curva E é o ponto $R = P + Q$. Observe que o ponto R é a reflexão do ponto R' em relação ao eixo horizontal.

Teorema de Hasse-Weil Se E é uma curva elíptica sobre \mathbb{Z}_p então:

$$p + 1 - 2\sqrt{p} \leq \#(\mathbb{Z}_p) \leq p + 1 + 2\sqrt{p}$$

Algoritmo de Soma de Pontos na Curva Elíptica: Sejam $E : y^2 = x^3 + ax + b$ com $4A^3 + 27B^2 \neq 0$ e P_1 e P_2 pontos da curva elíptica E .

1. $P_1 = 0 \implies P_1 + P_2 = P_2$
2. $P_2 = 0 \implies P_1 + P_2 = P_1$
3. Caso contrário, $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$
4. Se $x_1 = x_2$ e $y_1 = -y_2 \implies P_1 + P_2 = 0$
5. Caso contrário, defina λ por $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ se $P_1 \neq P_2$ ou $\lambda = \frac{3x_1^2 + a}{2y_1}$ se $P_1 = P_2$

Então $P_3 = P_1 + P_2 = (x_3, y_3)$ onde $x_3 = \lambda^2 - x_1 - x_2$ e $y_3 = \lambda(x_1 - x_3) - y_1$

Calculando Pontos em Curvas Elípticas

Exemplo: Considere a curva elíptica dada pela equação $E : y^2 = x^3 + 2x + 7$ sobre \mathbb{Z}_{13}

Podemos encontrar os pontos de $E(\mathbb{Z}_{13})$ substituindo cada $x \in \mathbb{Z}_{13}$ no polinômio $x^3 + 2x + 7$ e decidir se este valor é um quadrado ou não.

$$\begin{cases} \lambda = \frac{y_2 - y_1}{x_2 - x_1} \text{ se } P_1 \neq P_2 \\ \lambda = \frac{3x_1^2 + A}{2y_1} \text{ se } P_1 = P_2 \end{cases}$$

Então $P_3 = P_1 + P_2 = (x_3, y_3)$ onde:

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

x	y^2
0	0
1	1
2	4
3	9
4	3
5	12
6	10
7	10
8	12
9	3
10	9
11	4
12	1

x	$x^3 + 2x + 7$
0	7
1	10
2	6
3	1
4	1
5	12
6	1
7	0
8	2
9	0
10	0
11	8
12	4

(x, y)	y^2	$x^3 + 2x + 7$
(1,6),(1,7)	10	10
(3,1),(3,12)	1	1
(4,1),(4,12)	1	1
(6,1),(6,12)	1	1
(5,5),(5,8)	12	12
(7,0)	0	0
(9,0)	0	0
(10,0)	0	0
(12,2),(12,11)	4	4

Segue da última tabela que $E(\mathbb{Z}_{13}) = 16$ onde

$$E(\mathbb{Z}_{13}) = (1, 6), (1, 7), (3, 1), (4, 1), (6, 1), (3, 12), (4, 12), (6, 12) \\ \cup (5, 5), (5, 8), (7, 0), (9, 0), (10, 0), (12, 2), (12, 11) \cup 0$$

Agora, usando o Algoritmo de Soma de Pontos na Curva Elíptica, podemos observar que:

$$\text{Se } P_1 = (5, 8) \text{ e } P_2 = 0 \text{ então } P_1 + P_2 = (5, 8)$$

Se $P_1 = (3, 1)$ e $P_2 = (3, 12)$ então $P_1 + P_2 = 0$. **De fato**, $(3, 1)$ e $(3, 12)$ são simétricos em relação ao eixo horizontal pois $12 \equiv -1 \pmod{12}$ e conseqüentemente $(3, 12) = (3, -1)$

$$\text{Se } P_1 = (5, 8) \text{ e } P_2 = (12, 11) \text{ então}$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{11 - 8}{12 - 5} = \frac{3}{7}$$

Parece que temos um problema, pois estamos trabalhando módulo 13. Neste momento surge uma pergunta muito natural: Qual o significado de $\lambda = \frac{3}{7} \pmod{13}$? A resposta para isso é baseada no fato de que $\lambda = \frac{3}{7}$ é a solução da equação $7\lambda = 3$. Portanto, basta resolvermos a equação $7\lambda \equiv 3 \pmod{13}$

Não é difícil verificar que $\lambda = 6$ é a solução em \mathbb{Z}_{13} .

Agora podemos calcular efetivamente

- $x_3 = \lambda^2 - x_1 - x_2 = 6^2 - 12 - 5 = 19 \equiv 6(\text{mod}13)$
- $y_3 = \lambda(x_1 - x_3) - y_1 = 6(5 - 6) - 8 = -14 \equiv 12(\text{mod}13)$

Temos explicitamente as coordenadas de

$$P_3 = P_1 + P_2 = (x_3, y_3) = (6, 12)$$

Se $P_1 = P_2 = (1, 6)$, então

$$\lambda = \frac{3x_1^2 + A}{2y_1} = \frac{3 \cdot 1^2 + 2}{2 \cdot 6} = \frac{5}{12}$$

Resolvendo a equação $12\lambda \equiv 5(\text{mod}13)$, encontramos $\lambda = 8$. Segue que $P_3 = (x_3, y_3) = (10, 0)$, conforme contas a seguir:

- $x_3 = \lambda^2 - x_1 - x_2 = 8^2 - 1 - 1 = 62 \equiv 10(\text{mod}13)$
- $y_3 = \lambda(x_1 - x_3) - y_1 = 8(1 - 10) - 6 = -78 \equiv 0(\text{mod}13)$

Segue que: $P_3 = P_1 + P_1 = (x_3, y_3) = (10, 0)$

A curva $E : y^2 = x^3 + 3x + 8$ possui a tabela abaixo completa de cálculo da soma de pontos de uma curva sobre um corpo finito \mathbb{Z}_{13} . Os pontos dessa curva são os pontos que aparecem na tabela a seguir e serão somados:

(+)	0	(1, 5)	(1, 8)	(2, 3)	(2, 10)	(9, 6)	(9, 7)	(12, 2)	(12, 11)
0	0	(1, 5)	(1, 8)	(2, 3)	(2, 10)	(9, 6)	(9, 7)	(12, 2)	(12, 11)
(1, 5)	(1, 5)	(2, 10)	0	(1, 8)	(9, 7)	(2, 3)	(12, 2)	(12, 11)	(9, 6)
(1, 8)	(1, 8)	0	(2, 3)	(9, 6)	(1, 5)	(12, 11)	(2, 10)	(9, 7)	(12, 2)
(2, 3)	(2, 3)	(1, 8)	(9, 6)	(12, 11)	0	(12, 2)	(1, 5)	(2, 10)	(9, 7)
(2, 10)	(2, 10)	(9, 7)	(1, 5)	0	(12, 2)	(1, 8)	(12, 11)	(9, 6)	(2, 3)
(9, 6)	(9, 6)	(2, 3)	(12, 11)	(12, 2)	(1, 8)	(9, 7)	0	(1, 5)	(2, 10)
(9, 7)	(9, 7)	(12, 2)	(2, 10)	(1, 5)	(12, 11)	0	(9, 6)	(2, 3)	(1, 8)
(12, 2)	(12, 2)	(12, 11)	(9, 7)	(2, 10)	(9, 6)	(1, 5)	(2, 3)	(1, 8)	0
(12, 11)	(12, 11)	(9, 6)	(12, 2)	(9, 7)	(2, 3)	(2, 10)	(1, 8)	0	(1, 5)

Correia, 2013 [15] sugeriu construir um análogo ao ElGamal para o grupo de curvas elípticas sobre corpos finitos Figura 53

Joãozinho	Mônica-(parâmetros Públicos)	Serginho
	$p = 362, \quad P = (6, 730)$ $E: y^2 = x^3 + 14x + 19$	
Passo 2: Chave secreta: $n_A = 435$		
Passo 3: Joãozinho calcula $Q_A = n_A P = (932, 1204)$ e envia para Serginho		
	Q_A público	Recebe Q_A
		Passo 4: Serginho escolhe $k = 13$ e a mensagem $M = (2058, 3022)$ Calcula $R = kP = (1330, 144)$ $S = M + kQ_A = (2940, 2636)$.
		Passo 5: Serginho envia para João a seguinte mensagem (R, S)
Recebe (R, S)	(R, S) público	
Passo 6: João calcula $S - n_A R =$ $= (M + kQ_A) - n_A kP$ $= M + k(n_A P) - n_A kP$ $= M = (2058, 3022)$		

Figura 53: Método ElGamal para Criptografia de Curvas Elípticas. Disponível em [15]

É fácil construir um análogo ao ElGamal para o grupo das curvas elípticas sobre corpos finitos. Porém, teremos que supor que a mensagem que Serginho deseja enviar para Joãozinho é um ponto $M \in E(\mathbb{Z}_p)$ da curva elíptica onde o primo p e a curva elíptica E sobre \mathbb{Z}_p são previamente fixados por um canal não seguro de comunicação. Além disso, eles ainda não compartilham um ponto $P \in E(\mathbb{Z}_p)$.

ECDSA

De acordo com Rykwalder, 2014 os próprios bitcoins não são armazenados de forma centralizada ou local e, portanto, nenhuma entidade é a sua custodiante. Eles existem como registros em um livro contábil distribuído chamado Blockchain, cujas cópias são compartilhadas por uma rede voluntária de computadores conectados. "Possuir" um bitcoin simplesmente significa ter a capacidade de transferir o controle para outra pessoa, criando um registro da transferência na Blockchain. O que concede essa habilidade? O acesso a um par de chaves pública e privada do ECDSA. Rykwalder, 2014 afirma ainda que o ECDSA (Elliptic Curve Digital Signature Algorithm, ou Algoritmo de Assinatura Digital de Curvas Elípticas, em português) é um algoritmo que faz uso de curvas elípticas dentro de um corpo finito para assinar os dados emitidos, de forma que os usuários possam verificar tal autenticidade enquanto que o assinante se dedica exclusivamente à concepção de novas assinaturas.

Segundo Barros e Silva, 2018 [50] na rede Bitcoin os dados assinados são as transações entre os usuários. O ECDSA assina e verifica os dados em procedimentos separados, onde cada procedimento é um algoritmo formado por operações matemáticas. Enquanto o algoritmo do processo de assinatura utiliza chaves privadas, o algoritmo de processo de verificação utiliza chaves públicas.

Curvas de Koblitz

Segundo Bjoernsen, 2009 [8] as curvas de Koblitz são um tipo de curvas elípticas¹⁰ caracterizadas por sua construção não aleatória que permite uma computação especialmente eficiente. Isto é diferente das curvas elípticas mais comumente usadas que têm uma estrutura pseudo-aleatória onde os parâmetros são escolhidos por um algoritmo especificado [8].

As curvas de Koblitz foram utilizadas para descrever curvas anômalas binárias sobre $GF(2^k)$ a qual possui $a, b \in 0, 1$. As curvas são na forma:

$$y^2 + xy = x^3 + ax^2 + 1$$

Ou para curvas definidas sobre o campo finito F_p :

$$y^2 = x^3 + ax + b$$

com

$$4a^3 + 27b^2 \neq 0$$

As curvas de Koblitz definidas pelas Standards for Efficient Cryptography Group (SECG) são definidos sobre o campo finito F_p . Esta é uma generalização da curva de Koblitz, mas os mesmos princípios para eficiência na computação estão presentes em ambas as formas das curvas.

A Curva de Koblitz secp256k1

Bitcoin usa uma curva de Koblitz específica: secp256k1 definida pelo Standards for Efficient Cryptography Group (SECG). A curva é definida sobre o corpo finito $F_p: y^2 = x^3 + ax + b$ com $a = 0$ e $b = 7$

Chicarino et al. [20] afirma que no Bitcoin a chave privada é obtida gerando um número aleatório de 256bits, uma chave pública é obtida ao efetuar a multiplicação da chave privada por um ponto na curva conhecido como "ponto gerador". Ele é sempre o mesmo para todos os usuários do Bitcoin e é definido na especificação secp256k1. O resultado da multiplicação da chave privada pelo ponto gerador é um ponto na curva, este

¹⁰para mais detalhes sobre as Curvas Elípticas verificar a dissertação do PROFMAT Criptografia via Curvas Elípticas de Correia, 2013 [15]

ponto é a chave pública. Os nós armazenam somente as suas chaves privadas, pois ele pode a qualquer momento gerar a pública correspondente.

Martins, 2018 [33] afirma que para produzir um endereço Bitcoin, a primeira etapa a ser seguida deve ser a criação de uma chave privada k de 256 bits, que é simplesmente um número escolhido ao acaso entre 1 e 2^{256} . O método para a escolha do número é livre, mas não deve ser previsível ou repetível. Para a geração da chave pública K , o protocolo Bitcoin utiliza o padrão secp256k1, que estabelece uma curva elíptica definida sobre um corpo finito. Além da curva, o padrão estabelece um ponto gerador G , que deve ser multiplicado pela chave privada, resultando em outro ponto na curva, que é então chamado de chave pública $K = k \times G$. Apesar da relação matemática entre as chaves k e K , o par somente pode ser calculado a partir da chave privada, obtendo-se então a chave pública.

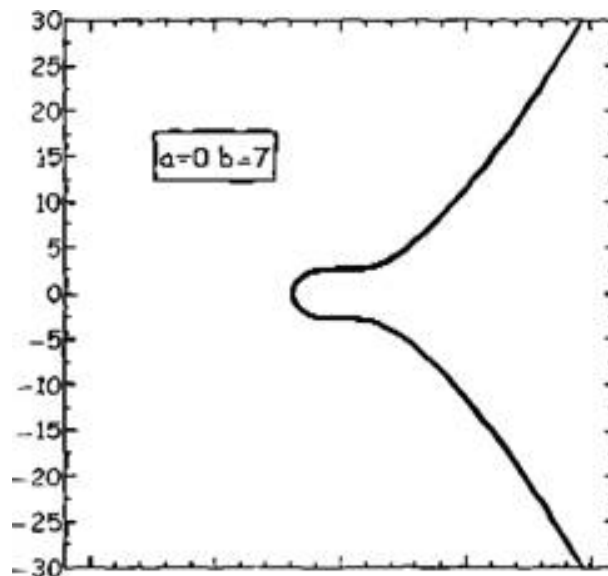


Figura 54: A Curva Koblitz. Disponível em [7]

A função hash criptográfica SHA-256

Processo de cálculo

A seguir é apresentado o processo de cálculo passo-a-passo bem como são explicados alguns passos sugeridos por Shirriff mas não que foram matematicamente demonstrados:

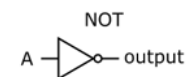
Passo 1. Primeiro, oito variáveis são definidas para seus valores iniciais, dados pelos primeiros 32 bits da parte fracionária das raízes quadradas dos primeiros 8 números primos (em hexadecimal):

$$\begin{aligned}
 H_1^{(0)} &= 6a09e667 & H_2^{(0)} &= bb67ae85 & H_3^{(0)} &= 3c6ef372 & H_4^{(0)} &= a54ff53a \\
 H_5^{(0)} &= 510e527f & H_6^{(0)} &= 9b05688c & H_7^{(0)} &= 1f83d9ab & H_8^{(0)} &= 5be0cd19
 \end{aligned}$$

<i>Primos</i>	$\sqrt{\text{Primos}}$	<i>ParteFracionaria(C)</i>	$C \times 16^8$	<i>Hexadecimal</i>
2	1,414213562	0,414213562	1779033703	6A09E667
3	1,732050808	0,732050808	3144134277	BB67AE85
5	2,236067977	0,236067977	1013904242	3C6EF372
7	2,645751311	0,645751311	2773480762	A54FF53A
11	3,316624790	0,316624790	1359893119	510E527F
13	3,605551275	0,605551275	2600822924	9B05688C
17	4,123105626	0,123105626	528734635	1F83D9AB
19	4,358898944	0,358898944	1541459225	5BE0CD19

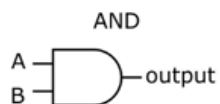
Tabela 3: Hexadecimal da raiz quadrada da parte fracionária dos primeiros 8 primos. Adaptado de [42]

Passo 2. Antes é importante entender quais saídas as tabelas-verdade informam de uma porta lógica para cada combinação de entradas Figuras 55 56 57 58.



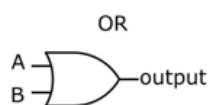
Input A	Output
0	1
1	0

Figura 55: Tabela-verdade da negação (\sim). Disponível em [47]



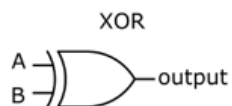
Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Figura 56: Tabela-verdade do E (\wedge). Disponível em [47]



Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Figura 57: Tabela-verdade do OU (\vee). Disponível em [47]



Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Figura 58: Tabela-verdade do OU-Exclusivo (\oplus). Disponível em [47]

A	6	a	0	9	e	6	6	7
	0110	1010	0000	1001	1110	0110	0110	0111
B	b	b	6	7	a	e	8	5
	1011	1011	0110	0111	1010	1110	1000	0101
C	3	c	6	e	f	3	7	2
	0011	1100	0110	1110	1111	0011	0111	0010
maj	0011	1100	0110	1111	1100	1100	0110	0111
	3	a	6	f	e	6	6	7

Figura 59: O cálculo da função Maj. Disponível em [9]

Passo 3. Depois, calcula-se inicialmente a função $Maj(X, Y, Z)$ ou seja, de A, B, C com o seguinte regramento: se houver mais zeros, considera-se o bit da $Maj(X, Y, Z)$ como zero e se houver mais uns, considera-se 1 (hum) Figura 59. Esta regra pode ser deduzida ao se analisar a tabela-verdade da referida função ($Maj(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$)

Tabela 4:

X	Y	Z	$X \wedge Y$	$X \wedge Z$	$(X \wedge Y) \oplus (X \wedge Z)$	$Y \wedge Z$	$(X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$
0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	1	1	0	0	0
1	1	1	1	1	0	1	1

Tabela 4: Tabela-verdade da função Maj

Passo 4. Em seguida transforma-se o número binário obtido da função $Maj(A, B, C)$ para hexadecimal gerando o resultado 3a6fe667.

Σ	c	e	2	0	6	4	7	e
	1100	1110	0010	0000	0101	1010	0111	1110
>>> 22	0010	0111	1000	1001	0011	0011	0101	0100
>>> 13	0011	0011	0011	0110	1010	1000	0000	0101
>>> 2	1101	1010	0000	0000	1001	1110	0110	0110
A	6	a	0	9	e	6	6	7
	0110	1010	0000	1001	1110	0110	0110	0111

Figura 60: O cálculo da primeira função Rotate. Disponível em [9]

Passo 5. O próximo passo (Figura 60) é aplicar a função $RotR(X, 2)$ em A que implica em rotacionar os bits duas posições à direita. Na sequência, aplica-se a função $RotR(X, 13)$ em A que implica em rotacionar os bits 13 (treze) posições à direita. Por fim, aplica-se a

função $RotR(X, 22)$ em A que implica em rotacionar os bits 22 (vinte-e-duas) posições à direita.

Passo 6. De posse de tais valores, verifica-se se a soma dos algoritmos binários são par ou ímpar. Se a soma for par, o valor resultante será 0 (zero), se for ímpar, 1 (hum). Isto é atestado matematicamente através da tabela-verdade da função \oplus (XOR) ou (ou-exclusivo) Tabela 5:

X	Y	Z	$X \oplus Y$	$X \oplus Y \oplus Z$
0	0	0	0	0
0	1	1	1	0
1	0	0	1	1
1	1	1	0	1

Tabela 5: Tabela-verdade da função XOR

Com isso, obtém-se o valor em binário de $\sum_0(X) = RotR(X, 2) \oplus RotR(X, 13) \oplus RotR(X, 22)$

Passo 7. A seguir transforma-se novamente o número em hexadecimal, cujo resultado na 1ª rodada é $ce20b47e$.

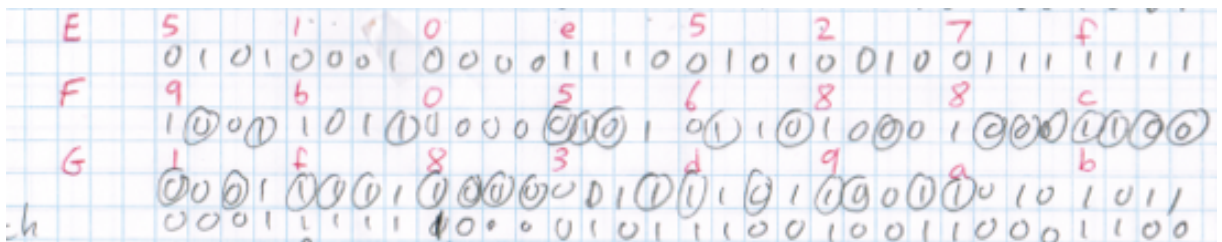


Figura 61: O cálculo da função Ch. Disponível em [9]

Passo 8. O próximo passo 61 é transformar os valores E, F e G em binários e aplicar a função $Ch(X, Y, Z)$ ou seja $Ch(E, F, G)$. Onde $Ch(X, Y, Z) = (X \wedge Y) \oplus (\bar{X} \wedge Z)$ Isso implica em para cada algarismo binário E, se for zero, circula-se o algarismo binário correspondente de G, e se for 1, circula-se o algarismo binário correspondente de F. Esta ação pode ser confirmada analisando-se tabela-verdade de $Ch(X, Y, Z)$ de acordo com a teoria presente em Shamieh, 2015 [47]:

X	Y	$(X \wedge Y)$	\bar{X}	Z	$(\bar{X} \wedge Z)$	$(X \wedge Y) \oplus (\bar{X} \wedge Z)$
0	0	0	1	1	1	1
0	1	1	1	0	1	0
1	0	1	0	1	1	0
1	1	1	0	0	0	1

Tabela 6: Tabela-verdade da função Ch

	3	5	8	7	2	7	2	6
Σ	00110	1011000	0111001	1110010	0100100	1111101	0101010	1110010
» 25	10000	11100	10100	100100	1001111	1101010	1000	0
» 11	01001	11111	10101	001000	100001	1110010	1010	0
» 6	11111	10101	00010	00011	10010	10010	0100	01
E	5	1	0	e	5	2	7	f
	0101000	1000011	1001010	1010010	0100111	1111		

Figura 62: O cálculo da segunda função Rotate. Disponível em [9]

Passo 9. O próximo passo (Figura 62) é aplicar a função $RotR(X, 6)$ em E que implica em rotacionar os bits 6 (seis) posições à direita. Na sequência, aplica-se a função $RotR(X, 11)$ em E que implica em rotacionar os bits 11 (onze) posições à direita. Por fim, aplica-se a função $RotR(X, 25)$ em E que implica em rotacionar os bits 25 (vinte-e-cinco) posições à direita.

Passo 10. O próximo passo é combinar todos os resultados efetuar o somatório (em hexadecimal), dado por:

$$T_1 = H + \sum_1 (E) + Ch(E, F, G) + K_i + W_i$$

onde W_i são 64 palavras de 32 bits contendo a mensagem inicial e as 64 palavras binárias K_i dadas pelos primeiros 32 bits das partes fracionárias das raízes cúbicas dos primeiros 64 números primos:

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

Em seguida calculamos

$$T_1 = H_0 + \sum_1 (E) + Ch(E, F, G) + K_i + W_i$$

$$T_2 = \sum_0 (A) + Maj(A, B, C)$$

Obtemos os novos parâmetros para $t = 1$, dados por, e com os novos valores calcular o novo valor de $H_j^{(t)}$ para $t = 1$:

$$\begin{aligned}
 H_1^{(1)} &= H_1^{(0)} + A_1 \\
 A_1 &= T_1 + T_2 & H_2^{(1)} &= H_2^{(0)} + B_1 \\
 B_1 &= A_0 & H_3^{(1)} &= H_3^{(0)} + C_1 \\
 C_1 &= B_0 & H_4^{(1)} &= H_4^{(0)} + D_1 \\
 D_1 &= C_0 & H_5^{(1)} &= H_5^{(0)} + E_1 \\
 E_1 &= D_0 + T_1 & H_6^{(1)} &= H_6^{(0)} + F_1 \\
 F_1 &= E_0 & H_7^{(1)} &= H_7^{(0)} + G_1 \\
 G_1 &= F_0 & H_8^{(1)} &= H_8^{(0)} + H_1 \\
 H_1 &= G_0
 \end{aligned}$$

E após calcularmos $T1$ e $T2$ para os novos parâmetros, que de maneira geral $H_j^{(t)}$:

$$\begin{aligned}
 H_1^{(t)} &= H_1^{(t-1)} + A_t & A_t &= T_1 + T_2 \\
 H_2^{(t)} &= H_2^{(t-1)} + B_t & B_t &= A_{t-1} \\
 H_3^{(t)} &= H_3^{(t-1)} + C_t & C_t &= B_{t-1} \\
 H_4^{(t)} &= H_4^{(t-1)} + D_t & D_t &= C_{t-1} \\
 H_5^{(t)} &= H_5^{(t-1)} + E_t & E_t &= D_{t-1} + T_1 \\
 H_6^{(t)} &= H_6^{(t-1)} + F_t & F_t &= E_{t-1} \\
 H_7^{(t)} &= H_7^{(t-1)} + G_t & G_t &= F_{t-1} \\
 H_8^{(t)} &= H_8^{(t-1)} + H_t & H_t &= G_{t-1}
 \end{aligned}$$

O hashing da mensagem é a concatenação das variáveis H_i^N após o último bloco ter sido processado

$$H = H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} \| H_8^{(N)}$$

	2	2	1	1	2	1	2
w	0	2	0	0	0	0	0
k	4	2	8	a	2	f	9
H	5	b	e	0	c	d	1
ch	1	f	8	5	c	9	8
Σ	3	5	8	7	2	7	2
	4	5	7	7	e	d	6

Figura 63: O cálculo de $T1$. Disponível em [9]

A	0	5	d	8	a	5	a	3	B	f	b	8	a	0	4	e	d	C	8	5	3	7	a	e	d	4	D	0	9	1	4	0	2	1	5	
+h0	6	a	0	9	e	6	6	7	+h1	b	b	6	7	a	e	8	5	+h2	3	c	6	e	f	3	7	2	+h3	a	5	4	f	f	5	3	a	
=	6	f	e	2	8	e	0	a	=									=	c	1	a	6	a	2	4	6	=	a	e	6	3	f	7	4	f	
E	4	2	1	0	3	0	e	6	F	4	6	5	4	a	0	1	0	G	4	9	5	2	3	f	5	5	H	a	4	1	f	3	2	e	7	
+h4	5	1	0	e	5	2	7	f	+h5	9	b	0	5	6	8	8	c	+h6	1	f	8	3	d	9	a	b	+h7	5	b	e	0	c	d	1	9	
=									=	e	1	5	a	0	8	9	c	=									=	0	0	0	0	0	0	0	0	0

Figura 64: O último passo do cálculo da função hash SHA-256. Disponível em [41]

A função hash criptográfica RIPEMD-160

Muitas funções de hashing usam variáveis de encadeamento. Estas têm valores iniciais especificados pelo algoritmo e são atualizados a cada rodada de acordo com o algoritmo e com os valores do bloco de mensagens atual. Os valores finais das variáveis de encadeamento formam o hashing. A maioria das funções de hash baseadas no MD4 tem apenas uma cadeia de operações através da qual as variáveis de encadeamento passam, mas o RIPEMD-160 tem duas vertentes, uma esquerda e direita, através das quais as variáveis passam independentemente, para serem unidas no final. A Figura 65 mostra uma única rodada do RIPEMD-160 no nível mais básico.

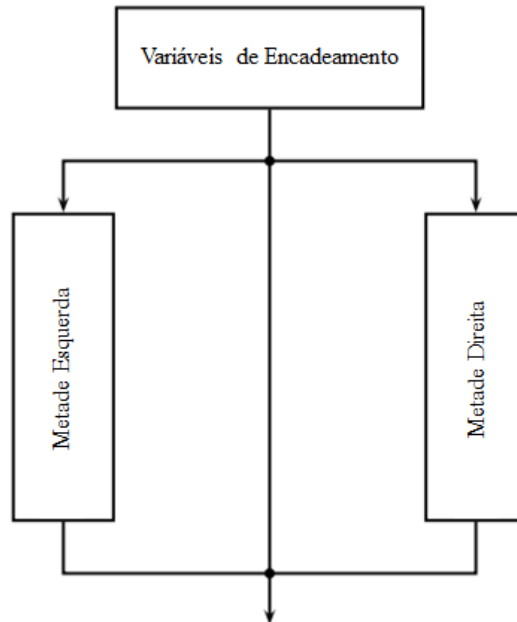


Figura 65: A função de hash RIPEMD-160. Adaptado de McAndrew, 2016 [34]

O RIPEMD-160¹¹ usa blocos de mensagem de tamanho 512 bits. Para preencher o último bloco, um único 1 é adicionado seguido de zeros suficientes para produzir 448 bits. Os últimos 64 bits contêm a representação binária do comprimento da mensagem inteira. Se a mensagem for maior que 2^{64} , somente os menos significativos 64 bits de informação serão usados (De acordo com McAndrew, 2016 [34] toda a Biblioteca do Congresso dos EUA contém cerca de 2^{64} bits de informação, então uma mensagem seria realmente muito rara)

¹¹Um pseudo-código da função pode ser verificado em: <https://homes.esat.kuleuven.be/~bosselae/ripemd/rmd160.txt>

Existem cinco variáveis de encadeamento A, B, C, D e E iniciadas como segue (em hexadecimal):

$$A = 67452301$$

$$B = EFC DAB89$$

$$C = 98BADC FE$$

$$D = 10325476$$

$$E = C3D2E1F0$$

Cada valor é apenas uma permutação de um subconjunto de todos os 16 dígitos hexadecimais. Cada uma das metades direita e esquerda consistem em cinco etapas, chamadas de rodadas, e cada rodada envolve uma função lógica. São elas:

$$f_1(x, y, z) = x \oplus y \oplus z$$

$$f_2(x, y, z) = (x \wedge y) \vee (\neg y \wedge z)$$

$$f_3(x, y, z) = (x \vee \neg y) \oplus z$$

$$f_4(x, y, z) = (x \wedge z) \vee (y \wedge \neg z)$$

$$f_5(x, y, z) = x \oplus (y \wedge \neg z)$$

Considere um único bloco de mensagens de 512 bits. Ele é dividido em 16 palavras de 32 bits ($m_0, m_1, m_2, \dots, m_{15}$) (Observe que cada uma das variáveis de encadeamento tem 32 bits de comprimento). Para cada rodada, as palavras da mensagem são permutadas de acordo com duas permutações baseadas em:

$$\rho = [7, 4, 13, 1, 10, 6, 15, 3, 12, 0, 9, 5, 2, 14, 11, 8]$$

$$\pi = [5, 14, 7, 0, 9, 2, 11, 4, 13, 6, 15, 8, 1, 10, 3, 12]$$

A segunda permutação é definida por $\pi_i = 9i + 5 \pmod{16}$. A partir dessas duas permutações, oito outras são produzidas, de modo que as metades esquerda e direita, em mais detalhes, podem ser descritas como na Figura 66, onde id é a permutação de identidade.

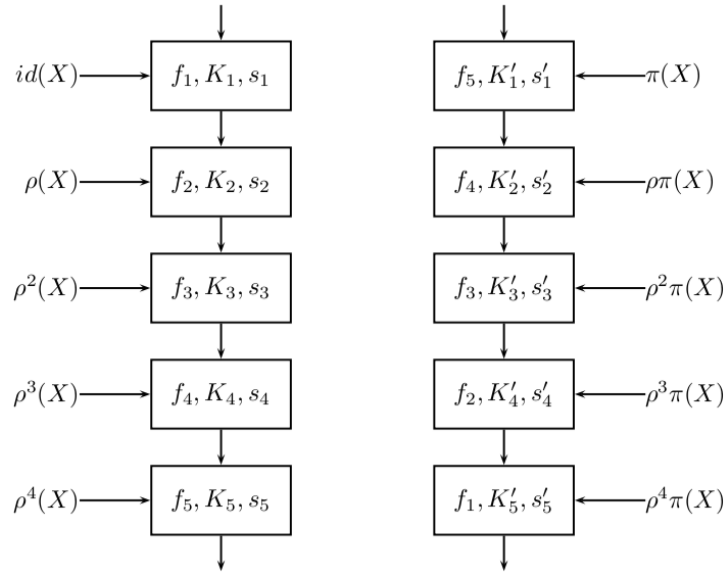


Figura 66: Metades esquerda e direita do RIPEMD-160. Adaptado de McAndrew, 2016 [34]

As constantes extras K_i e K'_i são definidas como:

$$K_1 = 00000000$$

$$K_2 = 5A827999 = [2^{30}\sqrt{2}]$$

$$K_3 = 6ED9EBA1 = [2^{30}\sqrt{3}]$$

$$K_4 = 8F1BBCDC = [2^{30}\sqrt{5}]$$

$$K_5 = A953FD4E = [2^{30}\sqrt{7}]$$

$$K'_1 = 50A28BE6 = [2^{30}\sqrt[3]{2}]$$

$$K'_2 = 5C4DD1246 = [2^{30}\sqrt[3]{3}]$$

$$K'_3 = 6D703EF3 = [2^{30}\sqrt[3]{5}]$$

$$K'_4 = 7A6D769E = [2^{30}\sqrt[3]{7}]$$

$$K'_5 = 00000000$$

Os valores s_i e s'_i são mudanças que são aplicadas a cada uma das palavras do bloco de mensagens. As entradas para cada caixa são os valores atuais das variáveis de encadeamento A, B, C, D, E e as dezesseis palavras de 32 bits do bloco de mensagens, permutadas de acordo com as definições acima. Suponha que:

$$[X_0, X_1, X_2, \dots, X_{15}]$$

são as palavras permutadas. Em seguida, em cada caixa, o seguinte pseudocódigo é implementado, onde \boxplus é o módulo de adição 2^{32} e $\lll s$ significa rotação esquerda de s bits:

```

For j = 1 to 15 {

    T ← ((A ⊕ f(B, C, D) ⊕ Xj ⊕ K) ≪≪ s) ≪≪ E

        A ← E

        E ← D

        D ← C ≪≪ 10

        C ← B

        B ← T

}

```

O valor de K , a função f e o deslocamento s são escolhidos de acordo com o esquema da Figura 66.

$j =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s'_1	11	14	15	12	5	8	7	9	11	13	14	15	6	7	9	8
s'_2	7	6	8	13	11	9	7	15	7	12	15	9	11	7	13	12
s'_3	11	13	6	7	14	9	13	15	14	8	13	6	5	12	7	5
s'_4	11	12	14	15	14	15	9	8	9	14	5	6	8	6	5	12
s'_5	9	15	5	11	6	8	13	12	5	12	13	14	11	8	5	6

Tabela 7: Deslocamento s para o lado direito

Os valores do deslocamento s para o lado direito são dados conforme a Tabela 7 e para o lado esquerdo por (Tabela 8). O requisito final é reunir os dois fios dos lados esquerdo e direito. Suponha que h_0, h_1, h_2, h_3, h_4 sejam as saídas do lado direito e A', B', C', D', E' são as saídas do lado esquerdo. Isso é feito de acordo com:

$$T = h_1 \boxplus C \boxplus D'$$

$j =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_1	8	9	9	11	13	15	15	5	7	7	8	11	14	14	12	6
s_2	9	13	15	7	12	8	9	11	7	7	12	7	6	15	13	11
s_3	9	7	15	11	8	6	6	14	12	13	5	14	13	13	7	5
s_4	15	5	8	11	14	14	6	14	6	9	12	9	12	5	15	8
s_5	8	5	12	9	12	5	14	6	8	13	6	5	15	13	11	11

Tabela 8: Deslocamento s para o lado esquerdo

$$h_1 = h_2 \boxplus D \boxplus E'$$

$$h_2 = h_3 \boxplus E \boxplus A'$$

$$h_3 = h_4 \boxplus A \boxplus B'$$

$$h_4 = h_0 \boxplus B \boxplus C'$$

$$h_0 = T$$

Referências Bibliográficas

- [1] Divesh Aggarwal, Gavin Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. Quantum attacks on bitcoin, and how to protect against them. *Ledger*, 3, 10 2017.
- [2] alikhan912. What is blockchain? <https://web.cbnt.io/article/15564831236121>. Acesso em: 07 set. 2019.
- [3] Andreas M. Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media, jul 2017.
- [4] Richard L. Apodaca. *Owning Bitcoin - The Illustrated Guide to Security, Privacy, and Possibility*. Richard L. Apodaca, 2017.
- [5] A. Back. *Hashcash*, May 1997.
- [6] Computer Science UC Santa Barbara. Feistel. <https://www.cs.ucsb.edu/~konheim/Feistel%20Talk%20-2Final.pdf>. Acesso em: 14 ago. 2019.
- [7] Conrad Barski. *Bitcoin for the Befuddled*. No Starch Press, nov 2014.
- [8] Kristian Bjoernsen. Koblitz curves and its practical uses in bitcoin security. *order (ϵ ($GF(2k)$), 2(1):7*, 2009.
- [9] Ken Shirriff's blog. A pencil-and-paper round of sha-256. <http://www.righto.com/2014/09/mining-bitcoin-with-pencil-and-paper.html>. Acesso em: 14 ago. 2019.
- [10] Antoon Bosselaers and Bart Preneel. Ripemd-160 pseudo-code. <https://homes.esat.kuleuven.be/~bosselae/ripemd/rmd160.txt>. Acesso em: 25 set. 2019.
- [11] Ryan Browne. China's central bank says it's close to releasing its own digital currency. <https://www.cnbc.com/2019/08/12/china-central-bank-close-to-releasing-digital-currency-pboc-official.html>. Acesso em: 17 ago. 2019.

- [12] Carlos Campello. Demonstrando o conceito de prova de trabalho da blockchain. https://medium.com/@carlos_49934/demonstrando-o-conceito-de-prova-de-trabalho-da-blockchain-445a1faf5f0f. Acesso em: 05 set. 2019.
- [13] Wei Dai. B-money. <http://www.weidai.com/bmoney.txt>. 1988.
- [14] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. Ripemd-160: A strengthened version of ripemd. In *Proceedings of the Third International Workshop on Fast Software Encryption*, pages 71–82, London, UK, UK, 1996. Springer-Verlag.
- [15] Sérgio dos Santos Correia Júnior. CRIPTOGRAFIA VIA CURVAS ELÍPTICAS. Master's thesis, Universidade Federal do Estado do Rio de Janeiro, Urca, 2013.
- [16] Daniel Drescher. *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Apress, mar 2017.
- [17] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '92, pages 139–147, London, UK, UK, 1993. Springer-Verlag.
- [18] Stack Exchange. Why does bitcoin use two hash functions (sha-256 and ripemd-160) to create an address? <https://bitcoin.stackexchange.com/questions/9202/why-does-bitcoin-use-two-hash-functions-sha-256-and-ripemd-160-to-create-an-ad>. Acesso em: 14 ago. 2019.
- [19] H. Feistel. *Cryptography and Computer Privacy*. Scientific American, 1973.
- [20] Emanuel Ferreira, Célio Albuquerque, Antônio Rocha, and Vanessa Rocha Leandro Chicarino. *Uso de Blockchain para Privacidade e Segurança em Internet das Coisas.*, page 51. 11 2017.
- [21] International Association for Cryptologic Research. Victor s. miller. <https://www.iacr.org/fellows/2013/miller.jpeg>. Acesso em: 20 ago. 2019.
- [22] Stanford Center for International Security and Corporation. Whitfield diffie. https://cisac.fsi.stanford.edu/people/whitfield_diffie. Acesso em: 14 ago. 2019.
- [23] Pedro Franco. *Understanding Bitcoin: Cryptography, Engineering and Economics (The Wiley Finance Series)*. Wiley, nov 2014.

- [24] Kevin Helms. Central banks worldwide testing their own digital currencies. <https://news.bitcoin.com/central-banks-testing-digital-currencies/>. Acesso em: 17 ago. 2019.
- [25] Joshua Holden. A good hash function is hard to find, and vice versa. *Cryptologia*, 37(2):107–119, 2013.
- [26] Steve Hollins. *Bitcoin para iniciantes - O guia definitivo para aprender a usar bitcoin e criptomoedas. Crie uma carteira, compre bitcoin, aprenda o que é o blockchain e a mineração de bitcoin (Portuguese Edition)*. CreateSpace Independent Publishing Platform, mar 2018.
- [27] IME-USP. Shannon. <http://ecalculo.if.usp.br/historia/imagens/Shannon.jpg>. Acesso em: 14 ago. 2019.
- [28] Claire Jones. Central bank plans to create digital currencies receive backing. <https://www.ft.com/content/428a0b20-99b0-11e9-9573-ee5cbb98ed36>. Acesso em: 17 ago. 2019.
- [29] Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl. *Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms (Synthesis Lectures on Information Security, Privacy, and Tru)*. Morgan & Claypool Publishers, 2017.
- [30] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
- [31] Kevin Kononenko. Blockchain explained by trying to pass high school math class. <https://blog.codeanalogies.com/2018/04/18/blockchain-explained-by-trying-to-pass-high-school-math-class/>. Acesso em: 20 ago. 2019.
- [32] Marco. Ponzi scheme graph model. http://abierto.altervista.org/ponzi-scheme-with-neo4j/?doing_wp_cron=1565817509.2785780429840087890625. Acesso em: 14 ago. 2019.
- [33] Thiago Fonseca Martins. Prova de existência de arquivos digitais utilizando a tecnologia blockchain do protocolo Bitcoin, 2018. Monografia (Engenharia da Computação), UFRGS (Universidade Federal do Rio Grande do Sul), Porto Alegre, Brazil.

- [34] A. McAndrew. *Introduction to Cryptography with Open-Source Software*. Discrete Mathematics and Its Applications. CRC Press, 2016.
- [35] Victor S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology, CRYPTO '85*, pages 417–426, Berlin, Heidelberg, 1986. Springer-Verlag.
- [36] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.
- [37] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA, 2016.
- [38] João Gregório Corrêa Neto. SISTEMA DE CRIPTOGRAFIA RSA. Master's thesis, Universidade Federal do Estado do Rio de Janeiro, Urca, 2013.
- [39] Stanford Department of Electrical Engineering. Martin e. hellman. <https://ee.stanford.edu/~hellman/PRphoto2016.jpg>. Acesso em: 14 ago. 2019.
- [40] University of Washington Department of Mathematics. Neal i. koblitz. https://math.washington.edu/sites/math/files/styles/portrait/public/photos/19_neal_koblitz1.jpg?itok=KIJxrlv2&c=e7a184478d0804fb5b22f1707c30c526. Acesso em: 20 ago. 2019.
- [41] Quartz. *Quartz: The Objects That Power the Global Economy*. Quartz, aug 2019.
- [42] David Rabahy. spreadsheet showing sha-256 calculations in step-by-step detail. <https://bitcointalk.org/index.php?topic=809430.0>. Acesso em: 14 ago. 2019.
- [43] Ronald L. Rivest. The md4 message digest algorithm. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '90*, pages 303–311, Berlin, Heidelberg, 1991. Springer-Verlag.
- [44] Kenneth Rosen. *Discrete Mathematics and Its Application*. Mcgraw-Hill, seventh edition, 2011.
- [45] Cleórbete Santos. Tecnologia Blockchain: Uma proposta de implementação na Universidade Federal do Tocantins. Master's thesis, UNIVERSIDADE FEDERAL DO TOCANTINS, Brasil, 2018.
- [46] Klaus Schwab. *A Quarta Revolução Industrial*. Edipro, 2016.

- [47] Cathleen Shamieh. *Electronics For Dummies*. For Dummies, jul 2015.
- [48] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, Vol 28, pp. 656–715, Oktober 1949.
- [49] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:623–656, 1948.
- [50] Diogo Silva and Janaína Barros. A matemática que estabelece o bitcoin. *Revista Diálogos*, 2:77–90, 10 2018.
- [51] smartdraw. Parking garage plan. <https://www.smartdraw.com/parking/examples/parking-garage-plan/>. Acesso em: 09 set. 2019.
- [52] Jimmy Song. *Programming Bitcoin: Learn How to Program Bitcoin from Scratch*. O’Reilly Media, mar 2019.
- [53] W. Stallings. *Cryptography and network security*. Prentice Hall, 2003.
- [54] Melanie Swan. *Blockchain: Blueprint for a New Economy*. O’Reilly Media, feb 2015.
- [55] Tine Thorn, Anders Baumann, Mikkel Fennestad, and Peter Sestoft. A distributed, value-oriented xml store, 2002.
- [56] Ibraim Silva Torres. Elliptical Curve Cryptography. Master’s thesis, Universidade do Minho, Portugal, 2007.
- [57] F. Ulrich. *Bitcoin - A Moeda Na Era Digital*. LVM EDITORA, 2014.
- [58] Department of Computer Science University of California. Alan g. konheim. <http://www.cs.ucsb.edu/~konheim/CV-AGK.pdf>. Acesso em: 14 ago. 2019.
- [59] Bitcoin Wiki. Technical background of version 1 bitcoin addresses. https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses. Acessado: 2019-07-16.
- [60] Wrosenb2. A cryptocurrency atm in milwaukee, wisconsin. https://en.wikipedia.org/wiki/Bitcoin_ATM#/media/File:CoinFlip%C2%AE_Cryptocurrency_ATM_in_Peoria,_Illinois.jpg. Acesso em: 14 ago. 2019.