Pontifícia Universidade Católica
do Rio de Janeiro

Raoni do Nascimento Gonzaga

Bitcoin: uma introdução à matemática das transações

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Matemática, do Departamento de Matemática da PUC-Rio.

Orientador: Prof. Sinésio Pesco

Rio de Janeiro
Julho de 2021

Pontifícia Universidade Católica
do Rio de Janeiro

Raoni do Nascimento Gonzaga

Bitcoin: uma introdução à matemática das transações

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Matemática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof. Sinésio Pesco
Orientador
Departamento de Matemática – PUC-Rio

Profª Renata Rosa
Departamento de Matemática - PUC-Rio

Prof Marcos Craizer
Departamento de Matemática - PUC-Rio

Profª Miriam del Milagro Abdón
Instituto de Matematica e Estatística - UFF-RJ

Rio de Janeiro, 16 de Julho de 2021

Raoni do Nascimento Gonzaga

Graduado em Engenharia Mecânica pelo Instituto Militar de Engenharia (IME). Professor da rede particular de ensino e de projetos sociais de preparação para concursos. Engenheiro de Equipamentos na indústria de Óleo e Gás.

Ficha Catalográfica

Gonzaga, Raoni do Nascimento

Bitcoin: uma introdução à matemática das transações / Raoni do Nascimento Gonzaga; orientador: Sinésio Pesco. – 2021.

65 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática, 2021.

Inclui bibliografia

1. Matemática – Teses. 2. Criptomedas. 3. Bitcoin. 4. ECDSA. I. Pesco, Sinésio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. III. Título.

CDD: 510

À minha filha Valentina, jóia das nossas vidas, que você realize os seus sonhos com seu esforço e amor.

Agradecimentos

Gostaria de agradecer à minha esposa Karlla Fernanda pela paciência nessa jornada pandêmica que possibilitou a minha dedicação para elaboração desse trabalho, foram só nós dois e nossa filha Valentina em isolamento por mais de um ano. Obrigado pelo seu amor durante esse período que foi diariamente demonstrado. Fomos esteio um do outro em todos os momentos.

Agradeço à minha mãe Sônia pelo seu exemplo e empenho em criar a mim e meus irmãos e por sempre me incentivar a estudar.

Aos professores Alexandre e Paulo, por serem meus primeiros professores de matemática e por alimentarem minha vontade de aprender.

Ao eterno Malba Tahan pelas estórias de Beremiz que me fizeram admirar a matemática.

Ao Colégio Pedro II por me apresentar uma educação pública de excelência.

Agradeço ao meu Orientador, Prof. Sinésio Pesco sempre disponível e motivador, que me ajudou muito no direcionamento desse trabalho. Gostaria de agradecer ao Programa de Mestrado Profissional em Rede Nacional em Matemática, à SBM e à PUC-Rio por permitirem a realização desse curso.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Gonzaga, Raoni do Nascimento; Pesco, Sinésio. **Bitcoin: uma introdução à matemática das transações**. Rio de Janeiro, 2021. 65p. Dissertação de Mestrado – Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

O conceito de moedas descentralizadas vem sendo amplamente disseminado com o advento das criptomoedas, dentre as quais tem destaque o Bitcoin. O objetivo deste trabalho é apresentar as etapas de uma transação de Bitcoin, explorando os conceitos matemáticos como Curvas elípticas e suas aplicações sobre a geração de chaves públicas nas transações de Bitcoin chamando a atenção para as características que conferem segurança, em particular, a aplicação de criptografia por meio do algoritmo ECDSA.

Palavras-chave

Criptomedas; Bitcoin; ECDSA.

Abstract

Gonzaga, Raoni do Nascimento; Pesco, Sinésio (Advisor). **Bitcoin: an introduction to Mathematics of transactions**. Rio de Janeiro, 2021. 65p. Dissertação de Mestrado – Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

The concept of decentralized currencies has been widely disseminated with the advent of cryptocurrencies, among which Bitcoin stands out. The objective of this work is to present the steps of a Bitcoin transaction, exploring mathematical concepts such as elliptical curves and their applications on the generation of public keys in Bitcoin transactions, drawing attention to the characteristics that provide security, in particular, the application of encryption through the ECDSA algorithm.

Keywords

Cryptocurrencies; Bitcoin; ECDSA.

Sumário

Lista de figuras

Lista de tabelas

Lista de Abreviaturas

ECDSA – Elliptic Curve Digital Signature Algorithm

SHA – Secure Hash Algorithm

NSA – Natioanl Secure Agency

PROFMAT – Progama de Mestrado Profissional em Matemática em Rede Nacional

UTXO – Unspent Transaction Output

SCGE – Standards for Efficient Cryptography Group

P2PKH – Pay to Public Key Hash

UBUNTU: eu sou porque nós somos

Nelson Mandela, *Longo caminho para a liberdade.*

1
Introdução

A economia global influencia todas as nossas vidas e atividades, são muitos números, índices, moedas e taxas. Termos como "geração de renda", "taxa básica de juros", "fundos de investimento"e, mais recentemente até a possibilidade de pagamento por PIX ocupam diariamente o noticiário e, grande parte das vezes, os cidadãos não têm conhecimento básico para concatenar essas ideias e entender relações de causa e efeito entre os números apresentados e sua realidade.

No conteúdo programático da Educação Básica de Ensino Fundamental e Médio temos pouca oportunidade de trabalhar a educação financeira para além da matemática financeira básica, pensando nesse ponto e na crescente digitalização de grande parte de nossa realidade, inclusive a econômica, que decidimos lançar luz sobre um tema intrigante e motivante como o das criptomoedas, nesse trabalho representadas pelo Bitcoin.

O Bitcoin vem se consolidando ao longo da última década como um ativo digital, reconhecido pelo mercado e chamando atenção pela sua alta volatilidade. No início do ano de 2021, em poucos meses a moeda chegou a acumular variações de mais de 400%, por exemplo. Quanto mais ganha notoriedade, mais dúvidas são geradas em seu entorno: do que se trata? É confiável? Como se dão as transações?

A intenção desse texto é de tornar o leitor capaz de reconhecer os principais termos desse universo e entender o quanto há de matemática na implementação dessa criptomoeda, em particular, entender como a matemática confere segurança às transações através da Criptografia de Curvas Elípticas no corpo dos Inteiros Módulo p.

Durante a pesquisa bibliográfica realizada nesse trabalho, foi constatado que ainda há muito a ser publicado em Lingua Portuguesa sobre esse tema, os textos não são muitos e, frequentemente serão citados textos em inglês. O artigo [1] e o livro [2] apresentam um bom tutorial sobre os conceitos relacionados ao tema que são trabalhados no capítulo 02 desse texto. Os trabalhos [3; 4] abordam a criptografia com emprego de curvas elípticas, tanto em $\mathbb{R}$ quanto em $\mathbb{Z}_p$. Já em [5] o autor traz o conceito das funções hash criptográficas e elementos de blockchain, inclusive um exemplo de aplicação em ambiente educacional.

A série de artigos [6; 7; 8; 9], fornece boas referências teóricas sobre o estabelecimento de uma criptomoeda, em particular do Bitcoin, já a série de artigos [10; 11] exibe implementações interessantes de operações em curvas elípticas e do ECDSA (Elliptic Curve Digital Signature Algorithm).

Por se tratar de um tema dos mais recentes, não tratado corriqueiramente em aula, procurou-se ao longo do texto ponderar formalismo e linguagem didática. O texto se destina aos estudantes dos períodos iniciais do Ensino Superior, entretanto um aluno concluinte do Ensino Médio é capaz de compreender o texto em quase sua totalidade. Adicionalmente, esse trabalho tem como público professores de matemática que desejem contextualizar o estudo de aritmética, criptografia e teoria dos números.

1.1 Contextualização

O prefixo "cripto" da palavra criptomoeda vem do grego e traz a noção de algo que está escondido, o que nos remete à ideia de manter sigilo sobre algo, essa ideia acompanha a humanidade ao longo de toda a sua evolução, todos nós desejamos sigilo em algum grau para assuntos pessoais. Por exemplo, se perguntarmos em uma enquete se os participantes gostariam de ter o seu sigilo telefônico quebrado, provavelmente ouviríamos, de grande parte dos entrevistados, um NÃO como resposta, o mesmo ocorreria para o sigilo bancário, provavelmente ninguém aceitaria tornar públicas todas as suas operações bancárias e saldo de contas.

A solução que a sociedade moderna desenvolveu de preservar a privacidade dos indivíduos ao longo do tempo foi amplamente influenciada pela criptografia, sem contar com as históricas aplicações militares, em particular nas duas grandes guerras mundiais do século passado, que desenvolveram bastante esse campo do conhecimento. A criptografia está muito presente em nossas vidas, desde as comunicações militares até as mensagens trocadas através de aplicativos como "WhatsApp", que aliás utliza a mesma técnica criptográfica [12] que o protocolo Bitcoin: a criptografia de curvas elípticas.

A grande quebra de paradigma associada ao Bitcoin é a descentralização da moeda, isso pode soar estranho, pois estamos habituados a sistemas econômicos centralizados, com uma autoridade como um Banco Central que é responsável por consolidar e validar as operações bancárias executadas, entretanto esses sistemas estão sensívelmente ligados à política econômica de cada país, que pode tomar decisões que afetem a taxa básica de juros do sistema financeiro ou até mesmo cause impacto no valor da moeda no mercado internacional por exemplo.

Já no Bitcoin, não há essa autoridade central, mas sim uma rede internacional que guarda toda a informação das transações de forma segura em milhares de nós diferentes de modo que em qualquer lugar da Terra se tem acesso a essa informação, que é disposta em uma cadeia de blocos denominada blockchain. Assim, a essência dessa inciativa é a liberdade sob a qual foi idealizada, sem depender diretamente de um governo ou instituição que possa influenciar diretamente sua política monetária.

1.2 Objetivo

O objetivo desse trabalho é apresentar uma introdução à matemática sobre a qual se baseiam as aplicações que permitem o protocolo Bitcoin operar com segurança. Para isso vamos trabalhar os conceitos fundamentais que compõem esse sistema no capítulo 02, o desenvolvimento matemático propriamente dito se dará mais intensamente no capítulo 03, onde será apresentada a teoria que subisidia a aplicação do algoritmo de assinatura digital (ECDSA) que garante a segurança nas transações entre endereços Bitcoins, tema do capítulo 04.

1.3 Breve Histórico

A idealização do Bitcoin se apoiou em trabalhos anteriores de Cientistas da Computação como Ralph C. Merkle, Whitfield Diffie e Martin Hellman que desenvolveram em 1976 um relevante trabalho no campo da criptografia assimétrica [13]. Bem como David Chaum e Adam Back, que nas décadas de 1980 e 1990, respectivamente, desenvolveram o e-cash e o hashcash, que abriram caminho para tornar a mineração de Bitcoin factível [13].

O conceito de Bitcoin foi proposto pela primeira vez por Satoshi Nakamoto em 2008 quando da publicação do artigo, "Bitcoin: A Peer-to-Peer Electronic Cash System"[14] por meio de um fórum online que disponibilizava uma lista de correio eletrônico de profissionais da área de criptografia [4]. Satoshi Nakamoto é um pseudônimo e até hoje não se sabe exatamente quem ele (ou ela) é (ou são, pois pode ser um grupo de pessoas) de fato. O protocolo Bitcoin foi introduzido pela primeira vez em 2009 como software de código aberto e se tornou a primeira criptomoeda descentralizada do mundo. De acordo com o artigo de Nakamoto, Bitcoin é essencialmente uma versão "peer to peer"de dinheiro eletrônico que permitiria que pagamentos online fossem enviados diretamente de uma parte para outra sem passar por uma instituição financeira.

Entretanto, foi em 22 de Maio de 2010, que o Bitcoin passou a ter valor comercial de fato, quando foram compradas duas pizzas em Jacksonville, Florida [13], por 10.000,00 BTC, valor equivalente a R$ 1,8 Bilhões de Reais em junho de 2021! Ao longo da década de de 2010, o Bitcoin passou de menos de 1 centavo de real para o pico em 64 mil reais em 2017, fechando o ano de 2019 um pouco abaixo dos 30 mil reais (Fig 1.1) .

Figura 1.1: Cotação do Bitcoin em Reais no período de Nov/2015 a Jun/2021

Fonte:https://g.co/finance/BTC-BRL

Nesse percurso a criptomoeda foi ganhando notoriedade e milhões de usuários foram adicionados à rede em todo o mundo, as pessoas foram se informando e desfazendo seus mitos em relação a esse mercado. Já recentemente, com o advento da pandemia de COVID-19 que se alastrou internacionalmente em 2020, as características do Bitcoin como independência e descentralização chamaram a atenção dos investidores frente ao cenário de grandes incertezas que a pandemia gerou no mercado internacional e a procura pela criptomoeda disparou ainda fomentada pela entrada de grandes instituições financeiras que passaram a estudar a viabilidade de aceitar pagamentos em Bitcoins.

2
Conceitos fundamentais

Nesse capítulo vamos descrever o ambiente onde se dão as operações de bitcoin, uma série de definições serão necessárias para entender as etapas das transações que serão apresentadas nos capítulos posteriores.

2.1
O que é Blockchain?

Essa palavra está sempre presente nas discussões sobre tecnologia, sempre sucitando curiosidade e um pouco de surpresa, por ser algo simples por um lado, mas hermético por outro. Literalmente a palavra blockchain pode ser traduzida como "cadeia de blocos" e, de fato não passa disso. Os blocos aqui podem ser entendidos como um conjunto de informações, entretanto há uma característica nessa estrutura que é a forma de interligar um bloco com o bloco anterior, isso acontece de maneira que um dado bloco, carrega consigo informações de seu antecessor e, caso se altere qualquer dado desse, aquele também se altera.

Por conta desse modo de conexão entre os blocos não se consegue alterar o histórico dessa sequência de informações, pois nesse caso, todos os blocos sucessores do que recebeu a modificação, também se modificariam. Essa propriedade é crucial para conferir confiabilidade a essa estrutura de dados. No escopo desse texto, a blockchain da que falamos é a que armazena os dados das transações de uma criptomoeda. Segundo [15], uma criptomoeda é um meio de troca, podendo ser centralizado ou descentralizado que se utiliza da tecnologia de blockchain e da criptografia para assegurar a validade das transações e a criação de novas unidades da moeda.

Uma das principais criptomoedas e a mais famosa de todas é o Bitcoin, cuja origem misteriosa em 2008 representou uma grande inovação por representar um sistema monetário descentralizado, alheio às decisões político-econômicas locais como o sistema bancário convencional.

Mas como garantir a veracidade das transações em uma rede descentralizada, sem um agente central como um banco para validar se as operações apresentam inconsistências financeiras? Caso não exista um consenso de que,

de fato eu disponho de R$ 20,00 na minha conta, não serei capaz de comprar uma pizza de R$10,00, por exemplo.

A maneira encontrada por Satoshi Nakamoto [14] utiliza a tecnologia blockchain para atingir esse consenso sobre a validade das transações, primeiro por garantir a possibilidade de contabilizar o saldo de todos os participantes da rede e depois por implementar uma segurança capaz de checar a veracidade de origem e destino da transação.

A blockchain pode ser vista como se fossem as folhas consecutivas de um livro caixa, no qual estão registradas todas as transações de todos os usuários da rede, desde a primeira emissão de bitcoin. Entretanto esse livro seria público, imutável e digital pois qualquer usuário pode consultá-lo a qualquer momento pela rede bitcoin, entretanto não pode alterar nenhum registro.

2.1.1 Funções Hash criptográficas

Mas como é feita essa conexão entre os blocos? Isso só é possível porque a Blockchain utiliza a função Hash criptográfica SHA-256. Essa função recebe uma entrada de tamanho aleatório (uma sequência de caracteres, um documento de texto, uma imagem ou até mesmo um vídeo) e a converte, através de transformações matemáticas combinadas, em uma sequência de saída de tamanho fixo de 256 bits escrita na base hexadecimal que conecta ao bloco atual toda a informação contida no bloco anterior.

No caso da blockchain do bitcoin, a informação de entrada são as transações que compõem o bloco, o hash do bloco anterior e mais algumas informações que vamos tratar adiante.

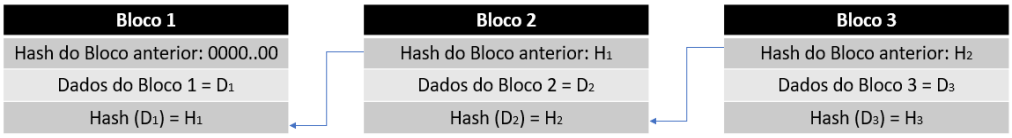


Figura 2.1: Encadeamento de blocos em uma blockchain

A função hash criptográfica SHA-256 cifra o conteúdo de um bloco de maneira que a partir do hash obtido não se consegue chegar na mensagem original. Essa função foi desenvolvida pela Agência Nacional de Segurança (NSA) dos EUA e seu nome vem de *Secure Hash Algoritm*. Em vários sites, como https://emn178.github.io/online-tools/sha256.html, podemos "rodar" essa função para visualizar como são expostos os resultados, por exemplo se tivermos como entrada a palavra PROFMAT ou caso a entrada seja 1 $googol = 10^{100}$:

•Hash(PROFMAT) =
d3ad00e5a62a28ba5cc3b1fa394a1c50ea44906317fd299860fb037750bb2b93

•Hash(10^{100}) =
$4de7cee3e8159d7d618855575d7bebf35ed07fc3556779f4fe8726e818741836$

Vamos ver as propriedades dessa função abaixo [16]:

1. Admite entradas de qualquer tamanho;

2. Produz uma saída de tamanho fixo de 256 bits;

3. Tempo de implementação é razoavelmente curto, mais tecnicamente esse tempo é linear em relação ao tamanho da entrada em bits;

4. Inexistência de função inversa: As funções hash criptográficas são especiais por serem unidirecionais, ou seja, não existe uma função inversa capaz de retornar a respectiva entrada a partir de uma dada saída.

5. Efeito avalanche: Caso a entrada seja alterada minimamente, como adicionando um ponto ao final dos caracteres ou uma letra passar de maiúscula para minúscula, o resultado da função muda drasticamente e não guarda relação alguma com o Hash anterior como vemos abaixo:

   $Hash(Criptografia) =$
   $06daa551e10e768c77f7f95a008851c50851e604ed8acbc89039b62d2add2903$
   $Hash(criptografia) =$
   $d93449f3e5b4bc1fb096a29c2fe7cb71b2694f1436f738741c35950fdb36fbaf$

6. Resistência à colisão, aqui definimos colisão como a situação na qual duas entradas distintas geram a mesma saída. Uma função Hash é dita resistente à colisão se é inviável de encontrar dois valores x e y tais que $x \neq y$ e $Hash(x) = Hash(y)$.

   Dizemos inviável e não impossível, na verdade, sabemos com certeza que existem colisões e podemos provar isso por meio de um simples argumento de contagem. O conjunto de todas as entradas para a função hash contém todas as entradas de todos os tamanhos, mas o conjunto de todas as saídas contém apenas saídas de um comprimento fixo específico. Logo, haverá necessariamente mais de uma entrada possível para uma dada saída.

   Entretanto, até o presente momento ninguém foi capaz de encontrar uma colisão nessa função, em teoria, se escolhermos $2^{256} + 1$ entradas, pelo

menos duas dessas terão a mesma saída, pois o número de entradas é maior do que o número de saídas possíveis. Acontece que para calcular apenas os hash de 2^{128} entradas (quantidade muito menor) com um PC comercial, levaria-se mais do que 10^{27} anos [16].

Assim, assumimos que a função é resistente à colisão e para $x \neq y$ temos $Hash(x) \neq Hash(y)$, o que implica no fato de se obtermos saídas iguais então as entradas são iguais, isso permite uma série de aplicações dessa função.

Quando um novo bloco de transações é adicionado à blockchain, ele é transmitido para cada nó da rede bitcoin, que são milhões de ususários em todo o mundo formando uma rede distribuída *peer to peer*. Para que a rede como um todo reconheça a validade desse bloco, todos os nós tem que receber a mesma informação e gerar um consenso em torno dessa informação.

Entretanto caso algum agente invasor mude o valor de alguma transação, o hash desse bloco se alterará assim como o hash de todos os blocos após esse também e essa informação quando restransmitida à rede vai gerar uma inconsistência, pois alguns nós vão receber os dados corretos enquanto outros os dados adulterados. Quando essa situação de falta de consenso ocorre, a rede consegue identificar que houve uma alteração e corrigi-la, garantindo assim que as informações das transações armazenadas na blockchain formem um registro imutável, confiável e auditável por qualquer nó da rede.

Representando os blocos com mais detalhes, podemos dividi-los em duas partes: Cabeçalho e Transações. No cabeçalho está o Hash do bloco anterior, informações de data e horário em que o bloco foi criado (Timestamp), um parâmetro chamado nonce e outro chamado de dificuldade, ambos estão relacionados diretamente com o processo de mineração que veremos nas seções posteriores do texto. Já no conteúdo do bloco estão as transações propriamente ditas, com as informações de valores, endereços de origem e endereços de destino de cada uma dessas operações.

Para adicionar um bloco na blockchain é necessário que seja satisfeita uma condição chamada de Prova de trabalho (Proof of Work) que consiste em se exibir que o resultado da função hash aplicada ao bloco obedece à uma restrição que requer um grande esforço computacional para ser atendida [1]. Essa restrição adotada na prática pelo protocolo da rede Bitcoin é encontrar um bloco cujo hash resultante possua os primeiros n bits iguais a zero, onde n depende da dificuldade de mineração determinada pelo sistema.

Na Fig 2.2 é ilustrado de forma simplificada como se atingir a prova de trabalho através da variação do parâmetro nonce até que o hash resultante do

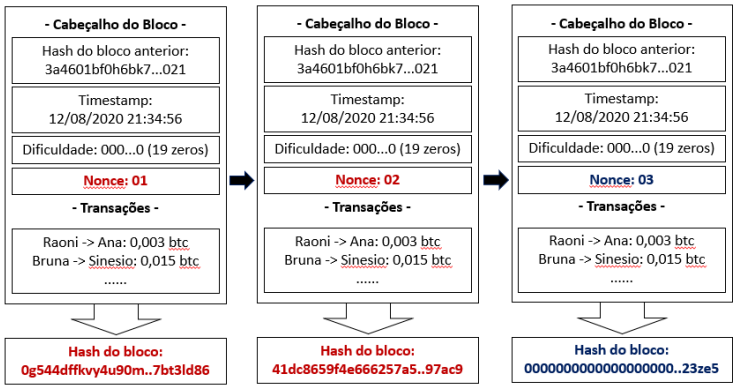


Figura 2.2: Variação do parâmetro Nonce em um determinado bloco

bloco atinja a dificuldade desejada. No exemplo bastaram apenas 03 valores diferentes de nonce para já se atingir a restrição imposta. Entretanto na prática, são necessárias mais do que milhões de tentativas até que se obtenha um hash de saída com os 19 primeiros bits iguais a zero e esse bloco então seja adicionado à blockchain. Por conta dessa dificuldade de encontrar esses parâmetros, esse processo foi comparado à atividade de mineração.

O leitor pode estar se perguntando o motivo pelo qual alguém estaria disposto a executar esse processo de mineração, que consome esse grande esforço computacional para obter a prova de trabalho, em troca de quê? A resposta é Bitcoins! A cada bloco adicionado à Blockchain com sucesso, o minerador recebe uma quantia em Bitcoins, mas afinal o que é Bitcoin? Vamos entender melhor na próxima sessão.

2.2 O que é Bitcoin?

Diferente de uma moeda fiduciária, como o Real ou o Dolar, controladas por um Banco Central que regula a sua emissão, o Bitcoin é uma moeda virtual criptograficamente segura que circula em um sistema de pagamento eletrônico descentralizado ponto a ponto (Peer to Peer).

Vamos comparar isso com exemplos do dia a dia: PayPal, Cartão de crédito ou PIX, que são sistemas de pagamento com os quais grande parte dos brasileiros está familiarizada. O PayPal tem mais paralelos com o Bitcoin [17], pois ambos envolvem transações que ocorrem online e com dinheiro digital em vez de Reais. Como o Bitcoin é descentralizado, ele não pode depender de uma única entidade para controlar a moeda; em vez disso, o Bitcoin depende da criptografia para gerar novas moedas em circulação e para validar transações. Ao contrário das moedas tradicionais, bitcoins são totalmente virtuais, o que significa que não há moeda palpável. As moedas estão implícitas em transações que transferem valor entre remetente e destinatário [2].

Pagamentos efetuados através de uma empresa de cartões de crédito convencional, se baseiam no fato de o usuário ter estabelecido uma relação de confiança com essa empresa e ter a delegado a responsabilidade de validar os seus dados e efetuar a transação corretamente. Entretanto em criptomoedas como o Bitcoin não há a necessidade de uma autoridade intermediária para essa validação, que é realizada através da tecnologia de blockchain.

A adoção do bitcoin como forma de pagamento vem crescendo por conta dos benefícios que oferecidos em relação às formas de pagamento tradicionais. Um usuário pode efetuar um pagamento para um receptor em qualquer lugar do mundo a qualquer momento, sem a necessidade de uma instituição intermediária, o que leva a menores taxas de transações, maior controle, e mais privacidade [1]. É importante notar que muitos dos benefícios oferecidos pelas criptomoedas em relação às instituições financeiras tradicionais, como descentralização e maior privacidade, são possibilitados pela utilização da tecnologia de blockchains, explorada na seção anterior.

2.2.1 Emissão de Bitcoins

Um questionamento natural quando somos apresentados ao Bitcoin é como eles são criados. Os Bitcoins não são criados à revelia de algum usuário da rede ou "do nada". No trabalho que deu origem à essa moeda [14], Satoshi Nakamoto definiu bem as regras sob as quais haveria emissão dessa criptomoeda. O protocolo escrito por ele estabalece toda a política monetária de como serão gerados os Bitcoins e como serão as transações. Apesar de o código ser aberto para melhorias, a lógica inicial permanece inalterada.

Para adquirir Bitcoins precisamos comprar de quem tem ou então minerar, que consiste em realizar os cálculos de hash necessários para satisfazer a prova de trabalho e com isso, adicionar blocos de transações à blockchain. Assim, todo bitcoin negociado hoje em dia veio de um minerador que foi remunerado pela rede bitcoin para "fazer rodar"a blockchain.

Satoshi Nakamoto estabeleceu que a cada 10 minutos um bloco seria adicionado à blockchain, assim o protocolo bitcoin inclui algoritmos integrados que regulam a dificuldade de mineração na rede.

A dificuldade do problema que os mineradores devem resolver é ajustada dinamicamente para que, em média, algum deles encontre uma resposta correta a cada 10 minutos, independentemente de quantos mineradores estão trabalhando no problema a qualquer momento.

Outras criptomoedas trabalham com tempo entre blocos diferentes, como é o caso do Ethereum com apenas 15 segundos entre os blocos. Esse parâmetro

depende de diversos fatores desde a aplicação da criptomoeda até quantidade de mineradores.

O protocolo bitcoin também possui uma programação que reduz pela metade a taxa na qual novos bitcoins são criados (e a consequente remuneração dos mineradores) a cada ciclo de 4 anos, essa redução recebe o nome de "Halving", no primeiro ciclo 50 BTC eram emitidos por bloco minerado, isso foi reduzido para 25 BTC/bloco em 2012 e novamente para 12,5 BTC/bloco em 2016. Mais recentemente, na metade de 2020, se iniciou um novo ciclo com 6,25 BTC/bloco. Assim o número total de bitcoins que serão criados é fixo e de aproximadamente 21 milhões de moedas. A diminuição da recompensa estende a vida do sistema ao impedir que todo o suprimento de moedas seja emitido em um curto período de tempo, o que acabaria com a motivação para a criação de novos blocos válidos [1].

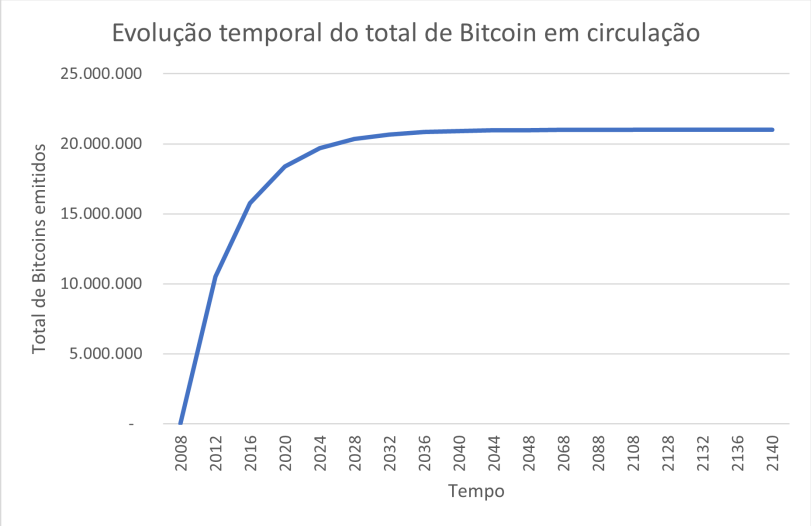


Figura 2.3: Evolução da Quantidade de Bitcoin em circulação

O resultado dessa maneira de emitir a criptomoeda é que o número de bitcoins em circulação segue a curva da Fig 2.3 que chega aproximadamente a 21 milhões até o ano 2140. Devido à diminuição da taxa de emissão do bitcoin, a longo prazo, a moeda é deflacionária, pois o bitcoin não pode ser inflado pela “impressão” de dinheiro novo além da taxa de emissão esperada, como acontece com as moedas fiduciárias. Maiores detalhes sobre esse tema podem ser encontrados em [16].

Mas o que acontece quando as 21 milhões de unidades do bitcoin forem emitidas? A blockchain do Bitcoin continuará operando normalmente assim como hoje e, já sem as recompensas por bloco, mineradores serão remunerados pelas taxas de transação que serão pagas pelos usuários, entretanto isso levará mais de um século para acontecer!

2.3
Entendendo a dificuldade da mineração

Agora vamos fazer um pouco de conta, o objetivo aqui é mostrar numericamente a dificuldade da mineração. Pegaremos por exemplo, o último bloco adicionado à blockchain do bitcoin no momento em que esse texto é escrito, o bloco é o de número 685448 e seu hash é escrito abaixo:

$$\underbrace{0000000000000000000}_{19 zeros}2c37de19d48c2d15dfd528d95c33ebb0dd81f8c6e30a8$$

- Como a função hash tem saída de 64 dígitos na base hexadecimal, o total de saídas diferentes é $16 \times 16 \times \ldots \times 16 = 16^{64} \approx 1,1579.10^{77}$;
- Entretanto o número de saídas que satisfazem a condição de se iniciarem com 19 zeros é de $16^{64-19} \approx 1,532.10^{54}$;
- Assim, a probabilidade da saída da função hash apresentar 19 zeros seguidos é dada por aproximadamente $\frac{1,532.10^{54}}{1,1579.10^{77}} = 1,32.10^{-23}$;
- Mas sabemos que as saídas são calculadas a partir da variação do parâmetro nonce, que é um número de 32 bits, portanto, a quantidade máxima de escolhas para esse parâmetro é de 2^{32}, assim a probabilidade de ao menos uma dessas escolhas resultar em uma saída que se inicie com 19 zeros é dada por: $(2^{32}).(1,32.10^{-23}) = 5,68.10^{-14}$ que ainda é um número muito pequeno.

Um parâmetro importante quando falamos de mineração é a capacidade de processamento de saídas da função hash, que é medido em hash/segundos. Atualmente existem hardwares especializados para mineração capazes de atingir altas taxas de processamento. Para aumentar a probabilidade de conseguirem realizar a prova de trabalho, a maioria dos mineradores se unem em chamadas mining pools, compartilhando suas máquinas para processamento em paralelo, assim, quando conseguem adicionar um bloco à rede, a recompensa é dividida entre todos os mineradores.

2.4
Transações de Bitcoin

Agora vamos entender um pouco como o sistema bitcoin estrutura suas transações [18]. Suponha que eu tenha recebido bitcoins de 03 amigos da seguinte maneira:

- Bruno -> Raoni: 0,15 BTC;
- Ana -> Raoni: 0,04 BTC;
- Sinesio -> Raoni: 0,003 BTC

Cada uma dessas operações gera uma UTXO (Unspent Transaction Output) que pode ser traduzida como saída de transação não gasta. Assim

eu tenho 03 UTXOs que posso utilizar para negociar como bem entender. Caso queira adquirir um smartphone de 0,01 BTC, por exemplo, posso utilizar a UTXO que recebi de Ana como a entrada gerando duas outras UTXOs: uma de 0,01 para a loja de celulares e uma outra de 0,03 BTC de volta para mim, como ilustrada na Fig 2.4 abaixo. Isso mesmo, a UTXO de origem tem que ser "consumida" totalmente, caso o valor dela seja superior do que o necessário na transação é gerada uma UTXO adicional com o "troco" de volta para o detentor dos bitcoins.

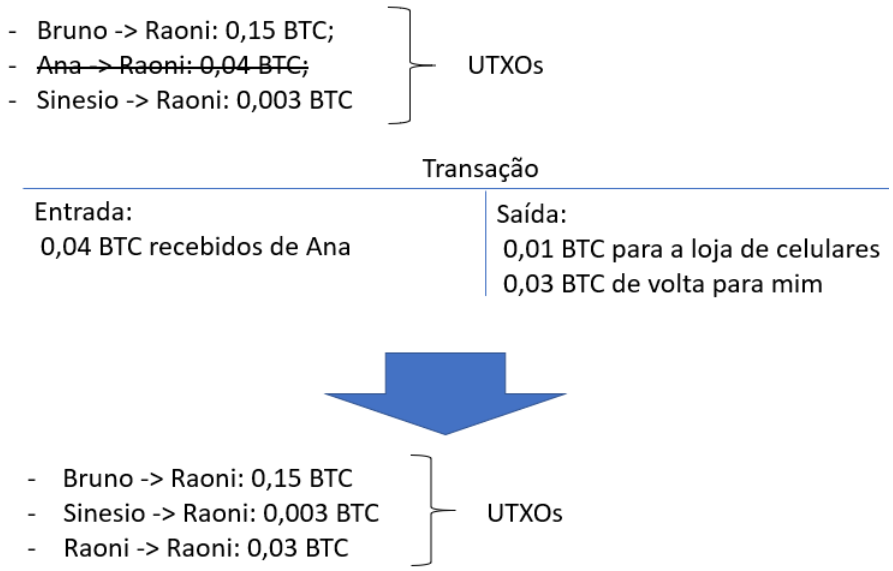


Figura 2.4: Exemplo de transação e geração de UTXO de "troco"

Na realidade, essa transação após a sua geração vai para o mempool, que é o repositório de todas as transações, a partir do qual se coletam as que irão compor o próximo bloco da blockchain. Existe uma ordem de prioridade nessa coleta e as transações que pagam melhores taxas para o minerador são as que entram nos blocos primeiro, assim, na saída da transação acima haveria ainda mais uma UTXO destinada à remuneração do minerador.

Continuando a explorar as negociações possíveis, uma outra situação é quando preciso de mais bitcoins do que há em uma dada UTXO, por exemplo, se meu objetivo fosse comprar um carro de luxo de 0,18 BTC. Nesse caso teria que utilizar como entrada as UTXOs recebidas de Bruno e de mim mesmo para obter o valor necessário, pois nenhuma sozinha atingira o saldo suficiente para a transação ser válida, conforme ilustrado na Fig 2.5

Após a relização dessas duas compras, só me restaria a UTXO gerada pelo depósito de Sinesio, no valor de 0,003 BTC, mas como a rede consegue saber o saldo de cada usuário? A resposta está no aplicativo da Carteira (Wallet em inglês) que todo usuário utiliza para saber o seu saldo, esse script percorre

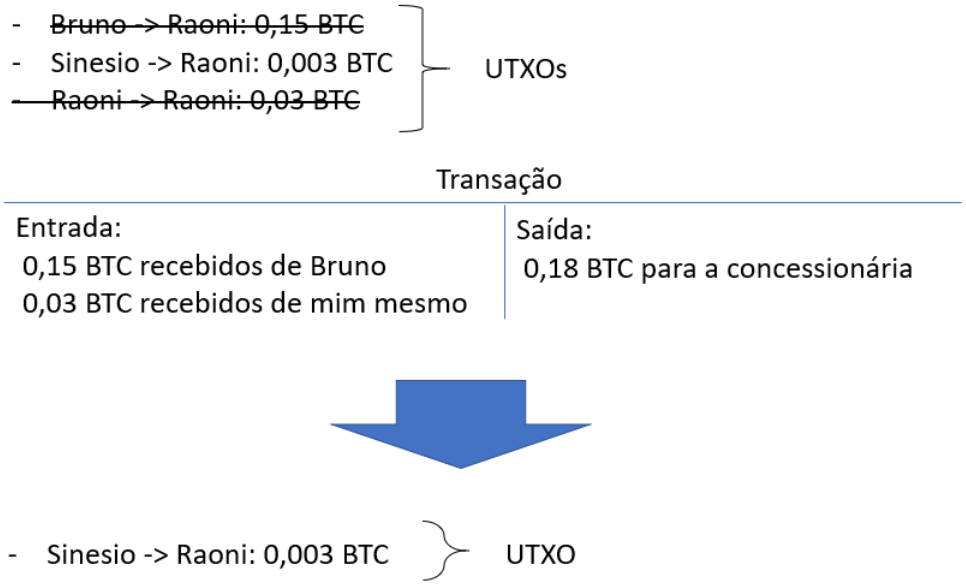


Figura 2.5: Exemplo de transação com consumo de mais de uma UTXO de entrada

toda a blockchain totalizando todas as UTXOs feitas cujo endereço de saída é o meu. Assim, na minha carteira estaria sendo exibida apenas a UTXO de Sinesio para mim, pois o aplicativo identificaria que as UTXOs que recebi anteriormente de Bruno, Ana e de mim mesmo, foram gastas.

A carteira Bitcoin é equivalente a uma carteira física, onde estão de fato as suas moedas. Esse software de carteira cria uma chave pública e uma chave privada para o usuário. A chave privada pode ser considerada uma senha do banco que permite que você acesse seus bitcoins e autorize pagamentos, devendo ser mantida em segredo. Enquanto que a chave pública pode ser considerada um número de conta bancária.

A chave pública é derivada matematicamente da chave privada usando a multiplicação de pontos em curvas elípticas. Entretanto as transações se dão entre endereços, que são obtidos a partir de saídas de funções hash criptográficas aplicadas às chaves públicas dos usuários. Achou complicado? Não se preocupe, pois vamos construir um caminho para entender essas etapas nos Capítulos 3 e 4.

As propriedades matemáticas das curvas elípticas e das funções hash criptográficas significam que é computacionalmente impossível descobrir-se a chave privada de um usuário a partir de seu endereço bitcoin. Cada endereço tem seu próprio saldo de bitcoins e as transações são essencialmente a transferência de bitcoins entre esses endereços. Pronto, agora que temos uma noção básica do ambiente em que as transações ocorrem, vamos explorar a matemática sobre a qual são baseadas.

3
Fundamentos matemáticos das transações

Nesse capítulo serão apresentadas as bases matemáticas sobre as quais a segurança das transações é construída, então serão expostas brevemente a teoria de grupos, o problema do logaritmo discreto, as curvas elípticas e como essas são utilizadas no contexto da aritmética modular para gerar chaves públicas a partir de chaves privadas.

3.1
Grupos, Anéis e Corpos

Nessa seção vamos pensar a álgebra e a aritmética de uma maneira um pouco diferente do que é abordado no Ensino Médio. Vamos abstrair para um conjunto de números diferentes dos conjunto dos Inteiros ou Reais, por exemplo e de operações diferentes de somar e multiplicar. Serão omitidas algumas demonstrações e em vários pontos do texto o formalismo característico da teoria de grupos não foi enfatizado em detrimento da didática, a motivação é entender o que conjuntos com essas operações definidas têm em comum entre si.

Vamos trabalhar com algumas definições para construirmos um entendimento sobre como essas propriedades resultam em uma robusta infraestrutura que esses conjuntos numéricos fornecem para aplicações criptográficas. Os conceitos apresentados são baseados nos livros [19; 20] e nos artigos [8; 6; 11].

3.1.1
Grupos

Um Grupo é um conjunto não vazio G associado a uma operação binária $G \times G \to G$, denotada por $*$, tal que $(G, *)$ satisfaça as seguintes propriedades:

1. *Associatividade*: sejam a, b e $c \in G \Rightarrow a * (b * c) = (a * b) * c$;
2. *Existência de elemento neutro*: $\exists e \in G$ tal que $\forall a \in G$, tem-se $a*e = e*a$;
3. *Existência de elemento inverso*: $\forall a \in G$, $\exists$ um único elemento $a^{-1} \in G$ tal que $a * a^{-1} = a^{-1} * a = e$. Dizemos que a^{-1} é o inverso de a em $(G, *)$

Em particular, se o grupo $(G, *)$ satisfaz à propriedade da *comutatividade*: a, $b \in G \Rightarrow a * b = b * a$ então o grupo é dito **abeliano**.

Dessa definição pode-se concluir que o conjunto dos números reais $\mathbb{R}$ é um grupo abeliano em relação à soma de números reais, de elemento neutro $e = 0$ e dado $a \in \mathbb{R}$ então $a^{-1} = -a$ é o seu inverso. Esse grupo é denotado por $(\mathbb{R}, +)$

Um grupo $(G, *)$ é dito finito se o conjunto G for finito e, a quantidade de elementos de G é chamada de ordem do grupo G, assim usaremos a seguinte notação: $ordem(G, *) = |G|$. Caso G seja infinito, sua ordem é infinta.

3.1.2 Subgrupos

Seja G um grupo em relação à operação $*$ e com elemento neutro e. Um subconjunto $H \subset G$ é dito Subgrupo de G se $(H, *)$ é grupo, ou seja:

1. $e \in H$;
2. $h_1 * h_2 \in H$ para todos $h_1 \in H$ e $h_2 \in H$;
3. $h^{-1} \in H$ para todo $h \in H$

A partir dessa definição pode-se verificar que $(\mathbb{Z}, +)$ e $(\mathbb{Q}, +)$ são subgrupos de $(\mathbb{R}, +)$

Um subgrupo interessante, considerando a operação de produto, é o subgrupo gerado pelas potências de um determinado elemento de um dado grupo, por exemplo:

Seja $(G, *)$ um grupo e $a \in G$. Considere o conjunto $< a >= \{a^n; n \in \mathbb{Z}\}$ de todas as potências de a defindo da seguinte maneira para cada $n \in \mathbb{Z}$:

$$a^n = \begin{cases} \underbrace{a * a * \ldots * a}_{n\ vezes}, \text{ se } n > 0 \\ e, \text{ se } n = 0 \\ \underbrace{a^{-1} * a^{-1} * \ldots * a^{-1}}_{-n\ vezes}, \text{ se } n < 0 \end{cases}$$

Pode-se provar que $(< a >, *)$ é um subgrupo de $(G, *)$, chamado de *subgrupo gerado por a*. Em particular, se a operação $(*)$ fôr a de soma usual, o leitor pode verificar que a notação se torna $< a >= \{na; n \in \mathbb{Z}\}$;

Por exemplo, considerando $\mathbb{Q}^*$ o conjunto dos Racionais não nulos, temos que o subgrupo de $(\mathbb{Q}^*, \times)$ gerado por 2 é

$$< 2 >= \{2^n; n \in \mathbb{Z}\} = \left\{\ldots, \tfrac{1}{8}, \tfrac{1}{4}, \tfrac{1}{2}, 1, 2, 4, 8, \ldots\right\}$$

Já o subgrupo de $(\mathbb{Z}, \times)$ gerado por 2 é

$$< 2 >= \{2n; n \in \mathbb{Z}\} = \{\ldots, -6, -4, -2, 0, 2, 4, 6, \ldots\}$$

Dizemos que um grupo é cíclico se , e somente se, ele pode ser gerado por um de seus elementos, que seria o elemento gerador do grupo. A partir dessa definição pode-se verificar que $(\mathbb{Z}, +)$ é um grupo cíclico infinito, gerado por 1 ou por -1.

No campo dos grupos finitos, Lagrange conseguiu relacionar as ordens de um subgrupo e de um grupo da seguinte maneira: Se G é um grupo finito e H é subgrupo de G, então $|H|$ divide $|G|$, ou seja, a ordem de H divide a ordem de G.

3.1.3 Classes Residuais

Seja $m \in \mathbb{Z}$ tal que $m > 1$. Podemos repartir o conjunto dos números inteiros $\mathbb{Z}$ em subconjuntos, onde cada um deles é formado por todos os números inteiros que possuem o mesmo resto quando divididos por m.

Se a, b e m são inteiros ($m > 1$), dizemos que a é congruente a b módulo m se a e b deixam o mesmo resto quando divididos por m. Denotamos como $a \equiv b \bmod m$

Dessa maneira, teremos as seguinte partição de $\mathbb{Z}$:

$$\begin{aligned}
[0] &= \{x \in \mathbb{Z}; x \equiv 0 \bmod m\}, \\
[1] &= \{x \in \mathbb{Z}; x \equiv 1 \bmod m\}, \\
&\vdots \\
[m-1] &= \{x \in \mathbb{Z}; x \equiv m-1 \bmod m\}.
\end{aligned}$$

O último termo escrito foi $[m-1]$ uma vez que $[m] = [0]$, $[m+1] = [1], \ldots$ Assim, o conjunto

$$[a] = \{x \in \mathbb{Z}; x \equiv a \bmod m\}$$

é chamado de *classe residual módulo* m do elemento a de $\mathbb{Z}$.

O conjunto de todas as classes residuais módulo m é representado por $\mathbb{Z}_m$. Assim

$$\mathbb{Z}_m = \{[0], [1], \ldots, [m-1]\}$$

Como resultado direto, no caso de $m = 2$, qualquer inteiro par é um representante da classe residual [0] e qualquer inteiro ímpar é representante da classe residual [1].

As classes residuais gozam das seguintes propriedades:

1. $[a] = [b]$ se e somente se $a \equiv b \bmod m$;

2. Se $[a] \cap [b] \neq \varnothing$ então $[a] = [b]$;

3. $\bigcup_{a \in \mathbb{N}} [a] = \mathbb{Z}$.

Uma característica relevante das classes residuais resulta da propriedade 1 acima, a partir dela se transforma a congruência $a \equiv b \bmod m$ na igualdade $[a] = [b]$

Em $\mathbb{Z}_m$ podemos definir as seguintes operações:

Adição em $\mathbb{Z}_m$: $[a] \oplus_m [b] = [a + b]$

Multiplicação em $\mathbb{Z}_m$: $[a] \otimes_m [b] = [a.b]$

Essas operações trazem consigo as seguintes propriedades:

Propriedades da Adição em $\mathbb{Z}_m$: para todos $[a], [b], [c] \in \mathbb{Z}_m$, temos:

- *Associatividade*: $([a] \oplus_m [b]) \oplus_m [c] = [a] \oplus_m ([b] \oplus_m [c])$;
- *Comutatividade*: $[a] \oplus_m [b] = [b] \oplus_m [a]$;
- *Existência de elemento neutro da adição (zero)*: $[a] \oplus_m [0] = [a]$, $\forall [a] \in \mathbb{Z}_m$;
- *Existência de inverso (simétrico)*: $[a] \oplus_m [-a] = 0$.

Propriedades da Multiplicação em $\mathbb{Z}_m$: para todos $[a], [b], [c] \in \mathbb{Z}_m$, temos:

- *Associatividade*: $([a] \otimes_m [b]) \otimes_m [c] = [a] \otimes_m ([b] \otimes_m [c])$;
- *Comutatividade*: $[a] \otimes_m [b] = [b] \otimes_m [a]$;
- *Existência de elemento neutro do produto (1)*: $[a] \otimes_m [1] = [a]$, $\forall [a] \in \mathbb{Z}_m$;
- *Distributividade*: $[a] \otimes_m ([b] \oplus_m [c]) = [a] \otimes_m [b] \oplus_m [a] \otimes_m [c]$.

Todo conjunto munido de uma operação de soma e de uma operação de multiplicação que satisfaz a essas propriedades é chamado de **anel**. Em particular, o fato de existir elemento neutro para multiplicação e de satisfazer a propriedade de comutatividade qualifica esse anel como **anel comutativo com unidade** Assim, $(\mathbb{Z}_m, \oplus_m, \otimes_m)$ é o anel das classes residuais módulo m ou *anel dos inteiros módulo m.*

Um elemento $[a] \in \mathbb{Z}_m$ será dito invertível se existir um único $[b] \in \mathbb{Z}_m$ tal que $[a] \otimes_m [b] = [1]$, assim $[a]^{-1} = [b]$ e $[b]$ é dito inverso de $[a]$.

Como exemplo dessas operações, vamos calcular as tabelas da adição (3.1) e da multiplicação (3.2) em $\mathbb{Z}_3 = \{[0], [1], [2]\}$

Utilizando a definição de elementos invertíveis, na Tabela 3.2 podemos constatar que apenas o elemento nulo [0] não admite inverso, os demais [1] e [2] são inversíveis.

Agora vamos estender nossa procura de elementos invertíveis às tabelas de multiplicação em $\mathbb{Z}_4 = \{[0], [1], [2], [3]\}$

$\oplus_3$	[0]	[1]	[2]
[0]	[0]	[1]	[2]
[1]	[1]	[2]	[0]
[2]	[2]	[0]	[1]

Tabela 3.1: Adição em $\mathbb{Z}_3$

$\otimes_3$	[0]	[1]	[2]
[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]
[2]	[0]	[2]	[1]

Tabela 3.2: Multiplicação em $\mathbb{Z}_3$

$\oplus_4$	[0]	[1]	[2]	[3]
[0]	[0]	[1]	[2]	[3]
[1]	[1]	[2]	[3]	[0]
[2]	[2]	[3]	[0]	[1]
[3]	[3]	[0]	[1]	[2]

Tabela 3.3: Adição em $\mathbb{Z}_4$

$\otimes_4$	[0]	[1]	[2]	[3]
[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]
[2]	[0]	[2]	[0]	[2]
[3]	[0]	[3]	[2]	[1]

Tabela 3.4: Multiplicação em $\mathbb{Z}_4$

Analisando a Tabela 3.4 conclui-se que apenas [1] e [3] são invertíveis. Nota-se também que a multiplicação de elementos diferentes do elemento nulo podem resultar em [0], como é o caso de $[2] \otimes_4 [2] = [0]$. Esse fato implica na seguinte definição[19]:

Um elemento $a \neq 0$ de um anel A é chamado de *divisor de zero* se existir $b \neq 0$ em A tal que $ab = 0$. Pela definição, esses elementos jamais serão invertíveis, pois, como exemplo, se a fosse invertível existiria a' tal que $aa' = 1$ o que conduz ao seguinte absurdo:

$$0 = a'0 = a'(ab) = (a'a)b = 1b = b$$

Vamos construir também as tabelas de adição e multiplicação em $\mathbb{Z}_5 = \{[0], [1], [2], [3], [4]\}$ e tirar algumas conclusões.

$\oplus_5$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[1]	[2]	[3]	[4]
[1]	[1]	[2]	[3]	[4]	[0]
[2]	[2]	[3]	[4]	[0]	[1]
[3]	[3]	[4]	[0]	[1]	[2]
[4]	[4]	[0]	[1]	[2]	[3]

Tabela 3.5: Adição em $\mathbb{Z}_5$

$\otimes_5$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]
[2]	[0]	[2]	[4]	[1]	[3]
[3]	[0]	[3]	[1]	[4]	[2]
[4]	[0]	[4]	[3]	[2]	[1]

Tabela 3.6: Multiplicação em $\mathbb{Z}_5$

Observando as Tabelas 3.2 e 3.6 de $\mathbb{Z}_3$ e $\mathbb{Z}_5$, respectivamente se nota que todo elemento distinto de [0] é invertível, entretanto isso não é regra para todos $\mathbb{Z}_m$, vimos que em $\mathbb{Z}_4$ [2] é divisor de zero, logo não é invertível.

Um anel comutativo com unidade em que todo elemento não nulo possui um inverso multiplicativo é chamado de **corpo**, assim $\mathbb{Z}_3$ e $\mathbb{Z}_5$ com as operações de adição e multiplicação definidas nesse texto são corpos, porém $\mathbb{Z}_4$ não é.

Proposição 1: Um elemento $[a] \in \mathbb{Z}_m$ é invertível se, e somente se, $mdc(a, m) = 1$.

Demonstração: Se [a] é invertível, então existe $[b] \in \mathbb{Z}_m$ tal que $[1] = [a] \otimes_m [b] = [a.b]$. Logo $a.b \equiv 1 \bmod m$, ou seja , existe um inteiro t tal que $a.b + t.m = 1$ e, consequentemente $mcd(a, m) = 1$.

Reciprocamente, se $mdc(a, m) = 1$, existem inteiros b e t tais que $a.b + m.t = 1$ e, consequentemente $[1] = [a.b + m.t] = [a.b] \oplus_m [m.t] = [a] \otimes_m [b] \oplus_m [0] = [a] \otimes_m [b]$. Assim $[a]$ é invertível.

Corolário: $\mathbb{Z}_m$ é um corpo se, e somente se, m é primo.

Demonstração: Suponha por absurdo que $\mathbb{Z}_m$ é um corpo e m não é primo, então $m = m_1.m_2$ com $1 < m_1 < m$ e $1 < m_2 < m$. Logo, $[0] = [m] = [m_1].[m_2]$ com $[m_1] \neq 0$ e $[m_2] \neq 0$ o que é absurdo, pois um dos fatores seria um divisor de zero e portanto não invertível.

Reciprocamente, supondo m primo. Como $mdc(i, m) = 1$ para $i = 1, \ldots, m - 1$ pela proposição anterior, tem-se que $[1], [2], \ldots, [m - 1]$ são invertíveis. Logo $\mathbb{Z}_m$ é um corpo.

Analogamente ao que vimos nos grupos, a ordem de um corpo finito é definida como sendo a quantidade de elementos do corpo, assim o corpo $\mathbb{Z}_p$ com p primo é dado por $\mathbb{Z}_p = \{[0], [1], \ldots, [p - 1]\}$ e tem ordem p.

A característica de um corpo $\mathbb{K}$, denotada por $car(\mathbb{K})$ consiste no menor inteiro positivo m tal que, $m.1 = \underbrace{1 + 1 + \ldots + 1}_{mvezes} = 0$ (aqui as notações de soma e produto seriam aquelas definidas junto à definição do corpo $\mathbb{K}$), caso esse inteiro exista. Se tal m não existir, a característica do corpo é definida como zero.

Como exemplo, em $\mathbb{Z}_5$:

$$[1] \oplus_5 [1] \oplus_5 [1] \oplus_5 [1] \oplus_5 [1] = [0]$$

Assim a característica de $\mathbb{Z}_5$ é 5, generalizando pode-se demonstrar que a característica de $\mathbb{Z}_p$ é p com p primo.

3.2 Problema do Logaritmo Discreto

Nessa seção vamos apresentar um problema que está associado diretamente à criptografia, a dificuldade de resolvê-lo garante a segurança da troca de chaves, mesmo se a mensagem for interceptada por um agente externo durante a transmissão.

O Problema do Logaritmo Discreto é defindo por STINSON[21] da seguinte maneira: Sejam $(G;\times)$ um grupo multiplicativo, cíclico e finito e $\alpha, \beta \in G$. Pretendemos encontrar um inteiro x tal que

$$\alpha^x = \beta$$

O inteiro x denotado por $\log_\alpha \beta$ é chamado de logaritmo discreto de β. Seja p primo e seja $a \in \mathbb{Z}$ com $a \not\equiv 0 \bmod p$. Suponhamos que para cada inteiro b, com $b \not\equiv 0 \bmod p$, exista um inteiro x tal que

$$a^x \equiv b \bmod p$$

O Problema do Logaritmo Discreto consiste em encontrar o inteiro x para cada b. Como exemplo podemos fazer $p = 11$, $a = 3$ e variar b para encontrar o inteiro x correspondente

$$3^1 \equiv 3 \equiv 3 \bmod 11 \Rightarrow b = 3 \text{ e } x = 1$$
$$3^2 \equiv 9 \equiv 9 \bmod 11 \Rightarrow b = 9 \text{ e } x = 2$$
$$3^3 \equiv 27 \equiv 5 \bmod 11 \Rightarrow b = 5 \text{ e } x = 3$$
$$3^4 \equiv 81 \equiv 4 \bmod 11 \Rightarrow b = 4 \text{ e } x = 4$$
$$3^5 \equiv 243 \equiv 1 \bmod 11 \Rightarrow b = 1 \text{ e } x = 5$$
$$3^6 \equiv 729 \equiv 3 \bmod 11 \Rightarrow b = 3 \text{ e } x = 6$$
$$3^7 \equiv 2187 \equiv 9 \bmod 11 \Rightarrow b = 9 \text{ e } x = 7$$

Analisando os resultados, vê-se um comportamento cíclico, assim $b \in \{3, 9, 5, 4, 1\}$, escolhendo $b = 9$ a solução é da forma $x = 2 + 5k$ com $k \in \mathbb{Z}$, entretanto se fazemos $b = 7$, por exemplo, consequentemente $3^x \equiv 7 \bmod 11$ não tem solução.

Para garantir que o problema do logaritmo discreto tem solução temos que garantir que existe um inteiro x para cada valor de b. Para evitarmos infinitas soluções como no caso acima, vamos restringir x ao corpo $\mathbb{Z}_p^*$, pois $b \not\equiv 0 \bmod p$. Em [3] é apresentada a seguinte proposição:

Proposição 2: Dado um inteiro fixo $a \neq 0$, o Problema do Logaritmo Discreto $a^x \equiv b \bmod p$ possui solução em $\mathbb{Z}_p$ para qualquer $b \in \mathrm{Z}_p$, com $b \not\equiv 0 \bmod p$, se, e somente se, a é um gerador do grupo multiplicativo $\mathbb{Z}_p^*$.

Demonstração: Considerando que $a^x \equiv b \bmod p$ possui solução em $\mathbb{Z}_p$ para todo $b \in \mathrm{Z}_p$. Isto significa que a é um gerador de $\mathbb{Z}_p^*$ pois qualquer que seja $b \in \mathbb{Z}_p$, existe um $x \in \mathbb{Z}_p$ tal que $a^x \equiv b \bmod p$. Reciprocamente, considerando que a é um gerador de $\mathbb{Z}_p^*$, vemos que cada um dos elementos de $\mathbb{Z}_p$ é congruente a alguma potência de a, assim, para todo $b \in \mathbb{Z}_p$ existe $x \in \mathbb{Z}_p$ tal que $a^x \equiv b \bmod p$, logo o problema do logaritmos discreto possui solução.

Exemplo 1: Resolver o seguinte problema do logaritmo discreto $3^x \equiv 5 \bmod 7$

Vamos verificar se 3 é um gerador de $\mathbb{Z}_7$:

$$3^1 \equiv 3 \equiv 3 \bmod 7$$
$$3^2 \equiv 9 \equiv 2 \bmod 7$$
$$3^3 \equiv 27 \equiv 6 \bmod 7$$
$$3^4 \equiv 81 \equiv 4 \bmod 7$$
$$3^5 \equiv 243 \equiv 5 \bmod 7$$
$$3^6 \equiv 729 \equiv 1 \bmod 7$$
$$3^7 \equiv 2187 \equiv 3 \bmod 7$$

De fato, 3 é um gerador de $\mathbb{Z}_7$, logo pela proposição 2, o problema possui solução e como $3^5 \equiv 5 \bmod 7$, conclui-se que $x = 5$.

Analisando o problema do logaritmo discreto podemos ver que é relativamente fácil , dado $x \in \mathbb{N}$ calcular o resto da divisão de a^x por p, entretanto o caminho inverso não é trivial, conhecendo esse resto, é difícil determinar qual é o expoente x. Nesse exemplo foi possível construir uma tabela para verificar os restos, mas nem sempre isso é possível. Em [21] é afirmado que escolhendo p com 150 algarismos ou mais, o problema passa a ser intratável até mesmo por computador, pois requer um tempo de processamento muito extenso para ser utilizado na prática.

3.3 Troca de Chaves - Sistema DHM

A essência de um método criptográfico reside em seu protocolo de comunicação cifrada, como serão trocadas as informações entre dois correspondentes de forma que um agente externo que intercepte a mensagem não a compreenda? Esse problema foi solucionado pelo trio de americanos Withfield Driffie, Martin Hellman e Ralph Merkle da seguinte maneira [19]:

João e Maria querem trocar entre si uma chave secreta em um meio de comunicação inseguro,

1. Inicialmente, Maria e João escolhem um primo p suficientemente grande e um inteiro a tal que $0 < a < p$ e a seja um gerador de $\mathbb{Z}_p$. Estes valores p e a são divulgados ao público.

2. Maria escolhe um inteiro m, $1 \leqslant m \leqslant p-2$, que não será divulgado.

3. João escolhe um inteiro j, $1 \leqslant j \leqslant p-2$ que também não será divulgado.

4. Maria calcula um inteiro $M \equiv a^m \bmod p$ e o envia para João.

5. João calcula um inteiro $J \equiv a^j \bmod p$ e o envia para Maria.

6. Maria calcula uma chave $K_m \equiv J^m \bmod p$

$$K_m \equiv (a^j)^m \equiv a^{jm} \bmod p$$

7. João, procede de maneira análoga e calcula uma chave $K_j \equiv M^j \bmod p$

$$K_j \equiv (a^m)^j \equiv a^{jm} \bmod p$$

8. Dessa maneira João e Maria compartilham secretamente a chave $K_m \equiv K_j \equiv K \bmod p$

Diante desse protocolo de troca de chaves, caso ocorra uma interceptação por um agente externo, entre os passos 4 e 5 o espião que já conhece a e p saberá também qual é o valor de M, assim o problema recai em calcular m, que nada mais é do que o logaritmo discreto de $M \equiv a^m \bmod p$. Por outro lado, se a interceptação ocorre entre os passos 6 e 7, por exemplo, o intruso conhecerá K_m, J e p, caindo no problema do logaritmo discreto novamente para calcular m. Assim a segurança do método de troca de chaves é garantida pela dificuldade de resolver esse problema.

3.4 Curvas Elípticas e suas operações

Vamos definir curvas elípticas como um conjunto de pontos que satisfazem a equação geral de Weierstrass[6]:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Os coeficientes a_1, a_2, a_3, a_4 e a_6 são escolhidos em um corpo $\mathbb{K}$ e assim dizemos que a curva elíptica está definda sobre um corpo $\mathbb{K}$. Esse corpo pode ser qualquer corpo finito ou não como exemplifica SILVERMAN[20] $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ ou $\mathbb{Z}_p$. Nesse texto vamos trabalhar com essas curvas definidas sobre $\mathbb{R}$ e sobre $\mathbb{Z}_p$ mas antes vamos simplificar um pouco a equação de Weierstrass. Observando o lado esquerdo podemos tentar completar quadrados da seguinte maneira:

$$y^2 + a_1xy + a_3y = (y + \lambda)^2 - \lambda^2$$

Dessa maneira anulamos o termo em y^2 e λ^2 e obtemos a igualdade:

$$2\lambda y = a_1xy + a_3y$$

Resolvendo essa equação para λ temos que $\lambda = \frac{a_1x+a_3}{2}$, entretanto estamos tratando de uma curva elíptica sobre um corpo arbitrário $\mathbb{K}$ que não pode ter $car(\mathbb{K}) = 2$, caso contrário o corpo não admitiria um inverso multiplicativo para 2. Assim, considerando $car(\mathbb{K}) \neq 2$ e fazendo uma substituição de variáveis tal que $u = (y + \lambda)$ temos a seguinte formulação para a equação de Weierstrass:

$$u^2 = (\tfrac{a_1x+a_3}{2}) = x^3 + a_2x^2 + a_4x + a_6$$

$$u^2 = \left(\tfrac{a_1x}{2}\right)^2 + \left(\tfrac{a_3}{2}\right)^2 + \left(\tfrac{a_1a_3x}{2}\right)^2 + x^3 + a_2x^2 + a_4x + a_6$$

$$u^2 = x^3 + \left(a_2 + \tfrac{a_1^2}{4}\right)x^2 + \left(a_4 + \tfrac{a_1a_3}{2}\right)x + (a_6 + \tfrac{a_3^2}{4})$$

Fazendo $a'_2 = \left(a_2 + \frac{a_1^2}{4}\right), a'_4 = \left(a_4 + \frac{a_1a_3}{2}\right)$ e $a'_6 = (a_6 + \frac{a_3^2}{4})$ a equação da curva elíptica se torna:

$$u^2 = x^3 + a'_2x^2 + a'_4x + a'_6$$

Observando dessa vez o lado direito da equação temos uma equação cúbica, na qual podemos aplicar uma substituição de variáveis para obtermos a sua forma reduzida, sem o termo quadrático, basta fazer $v = x + t$ e substituir na equação acima, vamos descobrir o valor desse t após igualarmos o coeficiente do termo quadrático a zero:

$$u^2 = (v - t)^3 + a'_2(v - t)^2 + a'_4(v - t) + a'_6$$

$$u^2 = v^3 - 3v^2t + 3vt^2 - t^3 + a'_2v^2 - 2a'_2vt + a'_2t^2 + a'_4v - a'_4t + a'_6$$

$$u^2 = v^3 + (a'_2 - 3t)v^2 + (3t^2 - 2a'_2t + a'_4)v - t^3 + a'_2t^2 - a'_4t + a_6$$

Igualando o coeficiente do termo quadrático a zero, obtemos:

$$-3t + a'_2 = 0$$

Se $car(\mathbb{K}) \neq 3$ então existe inverso multiplicativo de 3, assim podemos isolar t e obter $t = \frac{a'_2}{3}$ substituindo temos:

$$u^2 = v^3 + \left(3\left(\tfrac{a'_2}{3}\right)^2 - 2a'_2\left(\tfrac{a'_2}{3}\right) + a'_4\right)v - \left(\tfrac{a'_2}{3}\right)^3 + a'_2\left(\tfrac{a'_2}{3}\right)^2 - a'_4\left(\tfrac{a'_2}{3}\right) + a_6$$

$$u^2 = v^3 + \left(-\tfrac{(a'_2)^2}{3} + a'_4\right)v + \left(a'_6 + \tfrac{2(a'_2)^3}{27} - \tfrac{a'_2a'_4}{3}\right)$$

Sem perda de generalidade, podemos substiuir as variáveis (v, u) por (x, y) e chamar de $A = \left(-\frac{(a'_2)^2}{3} + a'_4\right)$ e $B = \left(a'_6 + \frac{2(a'_2)^3}{27} - \frac{a'_2a'_4}{3}\right)$ para obtermos (finalmente!) a Equação de Weierstrass simplificada de uma curva elíptica sobre um corpo $\mathbb{K}$, tal que $car(\mathbb{K}) \notin \{2, 3\}$:

$$E : f(x, y) = y^2 - x^3 - Ax - B = 0$$

Um dos principais motivos para o emprego de curvas elípticas em criptografia é a estrutura de grupo da qual gozam, para mostrar isso vamos apresentar argumentos geométricos e algébricos. A reta tangente à curva em um dado ponto tem um papel fundamental na construção desse raciocínio, assim, não nos interessam aqui os casos de curvas elípticas que apresentam singularidades (pontos da curva não-diferenciáveis). Exemplos de curvas singulares serão apresentados mais adiante no texto.

Analiticamente, vamos investigar as condições necessárias para que a curva não tenha singularidades. Assim, vamos calcular as condições para que as derivadas parciais da Equação simplificada de Weierstrass sejam nulas no ponto $P = (x_p, y_p)$:

- $f_x(x_p, y_p) = 0 \Rightarrow -3x_p^2 - A = 0$
- $f_y(x_p, y_p) = 0 \Rightarrow 2y_p = 0$

Aqui vemos que a condição de singularidade no ponto P nos conduz a $y_p = 0$ e $x_p = \pm\sqrt{-\frac{A}{3}}$. Substituindo na equação de Weierstrass temos:

$$y_p^2 - x_p^3 - Ax_p - B = 0$$

$$x_p(x_p^2 + A) + B = 0$$

$$\left(\pm\sqrt{-\tfrac{A}{3}}\right)\left(\tfrac{2A}{3}\right) = -B$$

$$\left(-\tfrac{A}{3}\right)\left(\tfrac{4A^2}{9}\right) = B^2$$

$$4A^3 + 27B^2 = 0$$

O termo $\Delta = 4A^3 + 27B^2$ é chamado de discriminante, dessa maneira, se $\Delta = 0$ então a curva elíptica é singular e no escopo desse trabalho iremos tratar apenas de curvas elípticas não singulares sobre corpos de características diferentes de 2 e 3. Em resumo:

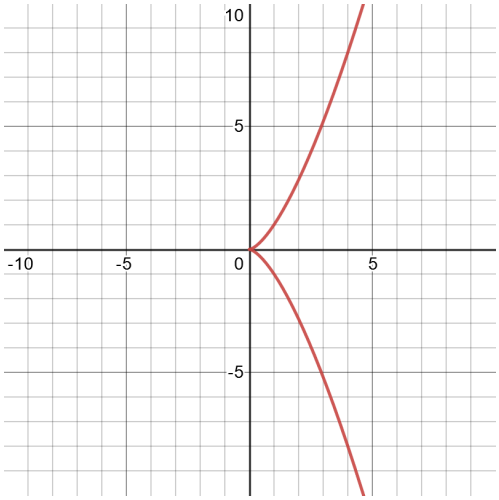
$$E = \{(x, y) \in \mathbb{K} | (y^2 = x^3 + Ax + B)\} \text{ onde } (car(\mathbb{K}) \notin \{2, 3\}) \text{ e } (\Delta \neq 0)$$

3.4.1
Caracteríticas Geométricas

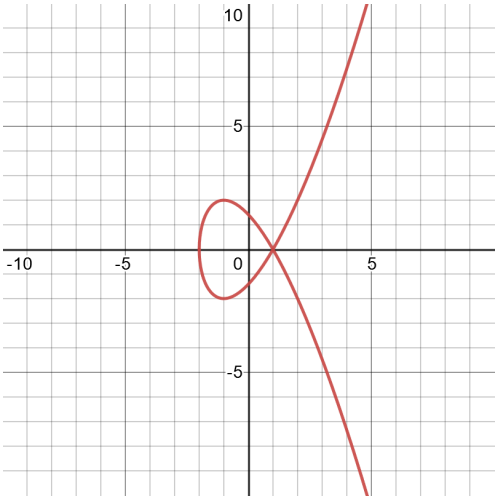
Vamos trabalhar com as curvas elípticas sobre $\mathbb{R}$ para visualizarmos algumas caracterísitcas geométricas e depois estenderemos a análise para $\mathbb{Z}_p$. Em $\mathbb{R}$ não é desafiador traçar o gráfico de uma curva elíptica. Uma propriedade que salta aos olhos na equação da curva elíptica é a existência de um termo quadrático na variável y , o que nos remete a uma simetria em relação ao eixo das abscissas, de fato vamos investigar da seguinte maneira: Seja $P = (x_p, y_p) \in E(\mathbb{R}) : y^2 = x^3 + Ax + B$ desejamos saber se $P' = (x_p, -y_p)$ também pertence à curva.

- $P = (x_p, y_p) \in E \Rightarrow y_p^2 - x_p^3 - Ax_p - B = 0$
- Substituindo as coordenadas de P' em E: $(-y_p)^2 - (x_p)^3 - A(x_p) + B = y_p^2 - x_p^3 - Ax_p - B$ entretanto, pelo item anterior: $y_p^2 - x_p^3 - Ax_p - B = 0$ e, consequentemente, $P' \in E$.
- Como P' tem a mesma abscissa de P, mas ordenada simétrica, então definimos que $P' = -P$

Para construir os gráficos das curvas elípticas em $\mathbb{R}$ nesse trabalho foram utilizados as ferramentas da calculadora gráfica on-line *desmos*, uma guia de uso desse aplicativo para construir curvas elípticas pode ser encontrado em [22]. Nos gráficos das figuras abaixo, podemos ver essa simetria em relação ao eixo-x nos diferentes exemplos de curvas singulares e não-singulares:

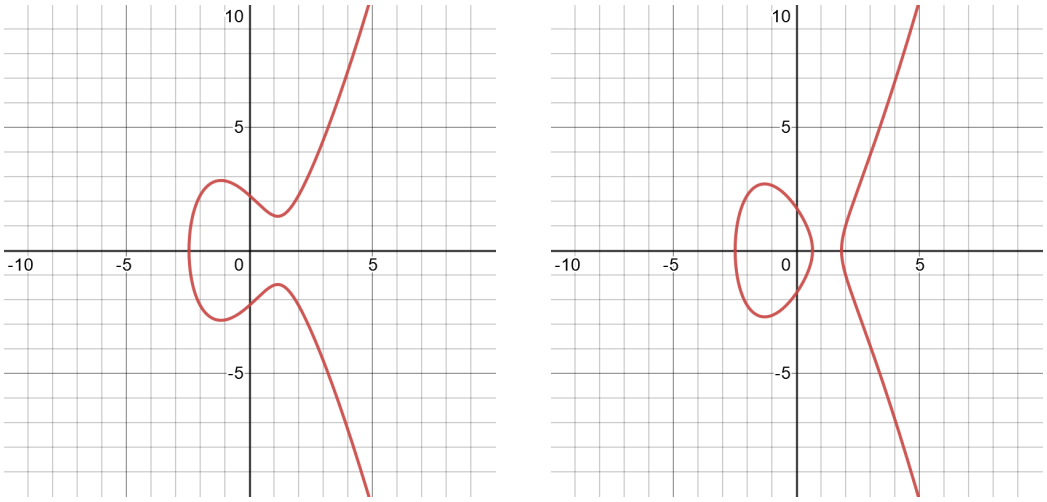


3.1(a): $y^2 = x^3$



3.1(b): $y^2 = x^3 - 3x + 2$

Figura 3.1: Exemplo de Curvas Elípticas Singulares



3.2(a): $y^2 = x^3 - 4x + 5$ 3.2(b): $y^2 = x^3 - 5x + 3$

Figura 3.2: Exemplo de Curvas Elípticas Não Singulares

3.4.2 "Somando" Pontos geometricamente em uma curva elíptica

Dada uma curva elíptica $E(\mathbb{R})$, é possível definir uma operação de "soma" entre dois de seus pontos quaisquer. O interessante dessa operação de soma é que obtemos um terceiro ponto que também pertence à curva e goza de várias propriedades. Mais adiante vamos mostrar que a estrutura algébrica definida por $E(\mathbb{R})$ e essa operação de "soma" formam um Grupo Abeliano.

Essa maneira de operar os pontos na curva elíptica é chamada de *Lei da Corda-Tangente* [23] e pode ser exemplificada da seguinte maneira: Sejam P e Q pontos distintos de uma curva elíptica $E(\mathbb{R})$, traça-se a reta que passa por P e Q (Figura 3.3), essa reta intersecta a curva novamente em outro ponto.

Vamos chamar esse outro ponto de R, como pode ser visto na Figura 3.4, se rebatermos R em relação ao eixo x, obtemos $R' = -R$, aqui definimos $R' = P \oplus Q$.

Agora suponha que os pontos P e Q sejam simétricos em relação ao eixo x, tais que $Q = -P$, nesse caso a reta que passa por P e Q é vertical e não encontraria outro ponto da curva, entretanto, analisando essa situação, teríamos que $P \oplus Q = P \oplus -P$ que nos induz a pensar em elemento neutro para a soma de um ponto com o seu simétrico. Pois bem, vamos definir esse elemento neutro como um ponto extra no infinito $\mathcal{O}$ [20] que, portanto, fará parte de toda curva elíptica e estará em toda linha vertical, no infinito (Figura 3.5), para entender completamente a origem desse raciocínio teríamos que recorrer ao espaço projetivo, que foge ao escopo desse trabalho, o leitor pode buscar mais detalhes em [23].

Com o advento do ponto no infinito, definido como o elemento neutro,

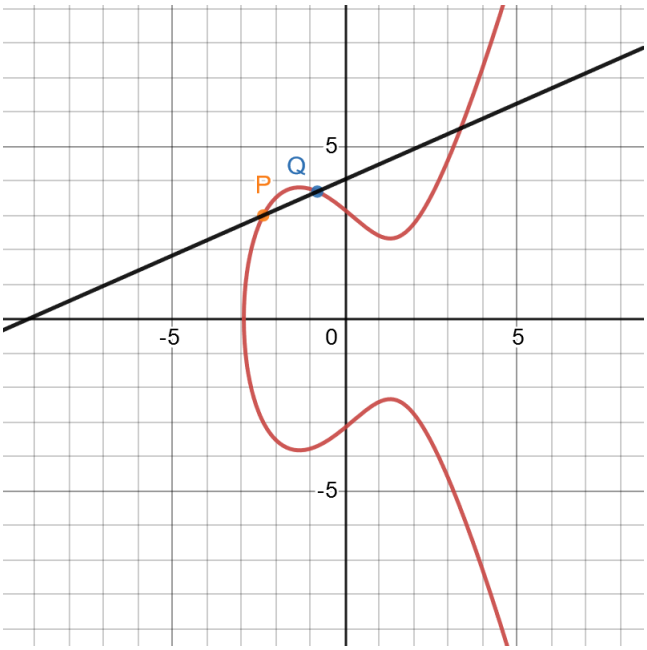


Figura 3.3: Reta que passa por P e Q em $E(\mathbb{R})$

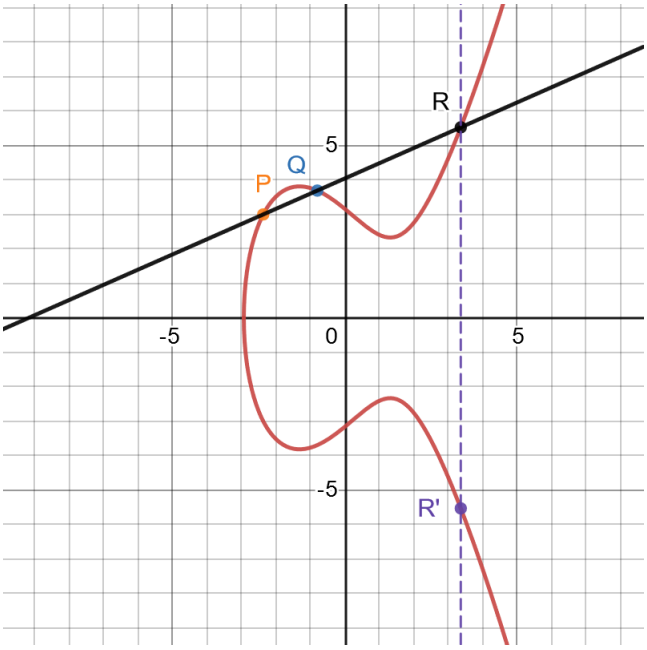


Figura 3.4: Rebatimento de R em relação ao eixo x, obtendo $R' = P \oplus Q$

ou zero, podemos enunciar a lei da corda-tangente da seguinte maneira: *três pontos têm soma zero se, e somente se, eles estão alinhados*. Da maneira como definimos essa soma, observando a Figura 3.4, podemos ver que $P \oplus Q \oplus R = P \oplus Q - (P \oplus Q) = 0$.

Uma situação da qual ainda não discorremos é quando se deseja adicionar um ponto a ele mesmo, observe que nesse caso, dado um ponto P fixo na curva, à medida que um outro ponto Q da curva se aproxima de P até serem coincidentes, podemos concluir que a reta que passa por P e Q vai

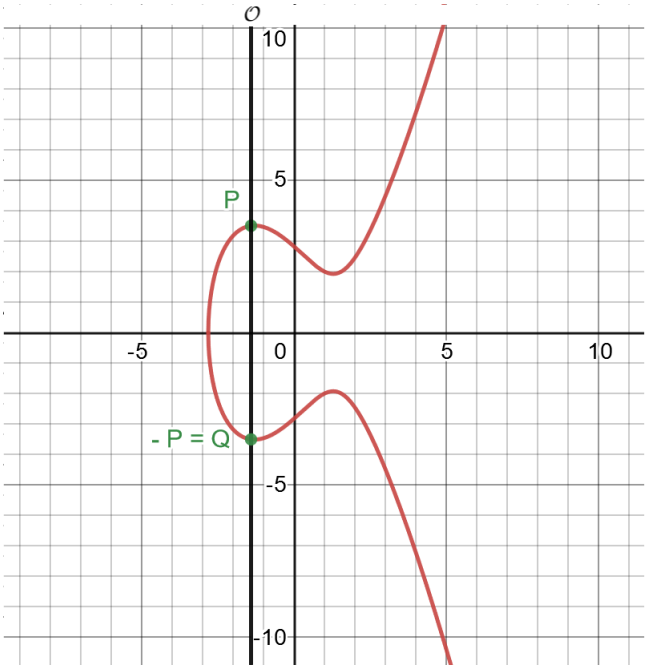


Figura 3.5: Reta vertical e o ponto no infinito $\mathcal{O} = P \oplus -P$

se aproximando da tangente ao gráfico no ponto $P = Q$ a como mostra a Figura 3.6:

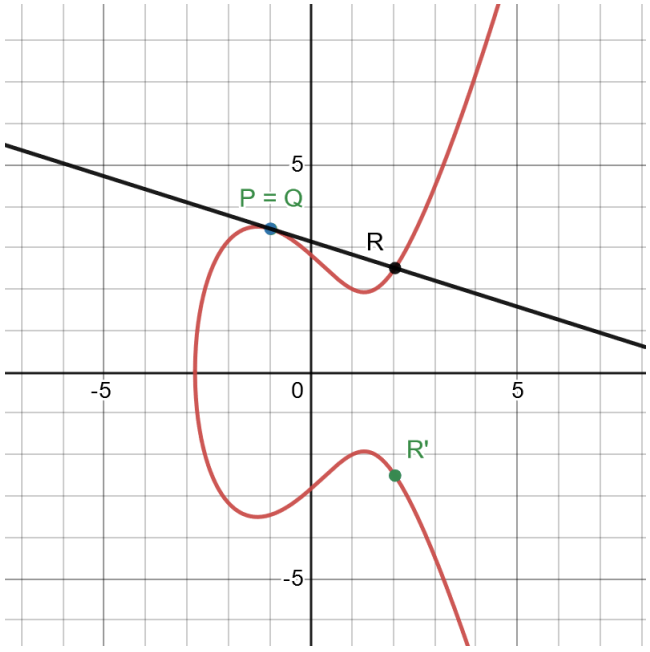


Figura 3.6: Adicionando um ponto a ele mesmo, $R' = P \oplus P = 2 \otimes P$

Assim, Podemos resumir as situações anteriores da seguinte maneira:

– Dados dois pontos distantes da curva, a reta que os une sempre intersecta a curva em um terceiro ponto;

– Se os dois pontos são coincidentes, então existe uma tangente à curva nesse ponto que sempre intersecta a curva em um segundo ponto;

– Se os dois pontos são simétricos em relação ao eixo x, então é definido o ponto no infinito, fora do plano xy $\mathcal{O} \in E(\mathbb{R})$ tal que para todo $P \in E(\mathbb{R})$ temos que $P \oplus \mathcal{O} = P$

3.4.3 Álgebra da operação de soma de pontos em uma curva elíptica

Vamos trabalhar com um pouco de geometria analítica e álgebra para concluir fórmulas gerais para operação de adição de pontos em uma curva elíptica [20]. Podemos seguir um raciocínio semelhante ao de um algoritmo para determinar essas coordenadas da seguinte maneira : sejam P_1 e P_2 pontos da seguinte curva elíptica: $E = \{(x, y) \in \mathbb{R} | : y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$ com $4A^3 + 27B^2 = 0$, então:

1. Se $P_1 = \mathcal{O}$, então $P_1 \oplus P_2 = P_2$
2. Se $P_2 = \mathcal{O}$, então $P_2 \oplus P_1 = P_1$
3. Se nenhum desses casos ocorrem, então podemos definir $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$
4. Se $x_1 = x_2$ e $y_1 = -y_2$ então $P_1 \oplus P_2 = \mathcal{O}$
5. Caso contrário, seja $y = \lambda x + \mu$ a reta que passa por P_1 e P_2 ($P_1 = P_2$ ou $P_1 \neq P_2$) e intersecta a curva em outro ponto $P_3' = (x_3', y_3')$ como P_3' está na reta e na curva elíptica, temos a igualdade:

$$(\lambda x + \mu)^2 = x^3 + Ax + B \Rightarrow x^3 - \lambda^2 x^2 + (A - 2\lambda\mu)x + B - \mu^2 = 0$$

Como x_1, x_2 e x_3' são raízes dessa equação, pelas relações de Girard, temos:

$$x_1 + x_2 + x_3' = \lambda^2 \Rightarrow x_3' = \lambda^2 - x_2 - x_1$$

Usando o fato de que P_1 pertence à reta, temos que $\mu = y_1 - \lambda x_1$ e para obter-se y_3' usamos o fato de P_3' também pertencer à reta e a expressão obtida para μ anteriormente:

$$y_3' = \lambda x_3' + \mu = \lambda x_3' + (y_1 - \lambda x_1) = \lambda(x_3 - x_1) + y_1$$

Pela definição da operação de soma, sabemos que $P_1 \oplus P_2 = -P_3' = P_3$, logo a expressão geral para as coordenadas da soma desses dois pontos dados sobre a curva é dada por $P_3 = (x_3, y_3)$:

$$x_3 = x_3' = \lambda^2 - x_2 - x_1$$
$$y_3 = -y_3' = -(\lambda(x_3 - x_1) + y_1) = \lambda(x_1 - x_3) - y_1$$

Se $P_1 \neq P_2$ então λ é o coeficiente angular da reta que passa por P_1 e P_2 e pode ser calculado como $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$.

No caso de termos $P_1 = P_2$, λ representa o coeficiente angular da reta tangente à curva no ponto $P_1 = P_2$, assim será obtido derivando a equação da curva elíptica de maneira implícita $y^2 = x^3 + Ax + B \rightarrow 2yy' = 3x^2 + A$, logo $\lambda = \frac{3x_1^2 + A}{2y_1}$.

Vamos ver que, o conjunto de pontos da curva elíptica e a operação de adição de pontos em uma curva elíptica forma um grupo abeliano $(E(\mathbb{R}), \oplus)$. Pode-se mostrar, sem muita dificuldade, a menos da propriedade da associatividade por requerer o uso das fórmulas explícitas, que as seguintes propriedades são satisfeitas:

1. *Associatividade*: sejam P, Q e $R \in E(\mathbb{R}) \Rightarrow P \oplus (Q \oplus R) = (P \oplus Q) \oplus R$;
2. *Existência de elemento neutro*: $\exists \mathcal{O} \in E(\mathbb{R})$ tal que $\forall P \in E(\mathbb{R})$, tem-se $P \oplus \mathcal{O} = \mathcal{O} \oplus P$;
3. *Existência de elemento inverso*: $\forall P \in E(\mathbb{R}), \exists$ um elemento $P' \in E(\mathbb{R})$ tal que $P \oplus P' = P' \oplus P = \mathcal{O}$. Dizemos que P' é o inverso de P em $(E(\mathbb{R}), \oplus)$
4. *comutatividade*: P, $Q \in E(\mathbb{R}) \Rightarrow P \oplus Q = Q \oplus P$

3.4.4 Curvas Elípticas sobre o Corpo dos Inteiros Módulo p ($\mathbb{Z}_p$)

Em termos de processamento computacional, trabalhar com pontos da curva sobre o corpo dos Reais pode não ser tão rápido pela necessidade de truncamanto da parte decimal dos números operados. Uma solução para se trabalhar apenas com números inteiros é utilizar a aritmética dos inteiros módulo p, com p primo e $p \neq \{2, 3\}$, pois como vimos nas seções anteriores, vamos trabalhar com característica diferente de 2 e 3.

Assim, as curvas elípticas sobre o corpo dos inteiros módulo p, podem ser definidas de maneira análoga à que fizemos para os Reais, com algumas particularidades:

1. A equação da curva se torna $E(\mathbb{Z}_p) : y^2 \equiv x^3 + Ax + B \bmod p$;
2. Os parâmetros A e B são escolhidos em $\mathbb{Z}_p$, entretanto vamos suprimir a notação de classes por simplificação , mas sem perda de generalidade;
3. $4A^3 + 27B^2 \not\equiv 0 \bmod p$
4. O eixo de simetria deixa de ser $y = 0$, como era em R e passa a ser $y = p/2$, de fato se (x, y) é um ponto da curva em $E(\mathbb{Z}_p)$, então $(x, -y+p)$ também pertence a curva, pois $(-y + p)^2 \equiv y^2 - 2py + p^2 \equiv y^2 \bmod p$

Exemplo 2: Determinar os pontos da curva $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$.

Para determinar esses pontos, devemos resolver a seguinte equação: $y^2 \equiv x^3 + 7 \bmod 11$, podemos ir atribuindo valores às variáveis x e y e tabelar os resultados do lado esquerdo e do lado direito da equação para comparar as soluções por inspeção:

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

Tabela 3.7: Lado Direito da equação: atribuindo valores à variável x

y	0	1	2	3	4	5	6	7	8	9	10
$y^2 \bmod 11$	0	1	4	9	5	3	3	5	9	4	1

Tabela 3.8: Lado Esquerdo da equação: atribuindo valores à variável y

Analisando os resultados vemos que quando x é nulo, o lado direito da equação deixa resíduo 7 mod 11, entretanto, para nenhum valor de y, o lado esquerdo deixa esse mesmo resíduo. Agora, veja que, por exemplo, quando $x = 2$ e $y = 2$ os resíduos são iguais.

Assim $(2, 2)$ pertence ao conjunto de pontos de nossa curva sobre Z_{11} e lembrando que o ponto $\mathcal{O}$ está em todas as curvas elípticas que estudamos, podemos concluir que o conjunto dos pontos da curva $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$ são os 12 pontos dados abaixo:
$E(\mathbb{Z}_{11}) : \{\mathcal{O}, (2, 2), (2, 9), (3, 1), (3, 10), (4, 4), (4, 7), (5, 0), (6, 5), (6, 6), (7, 3), (7, 8)\}$

Conhecendo os pontos da curva, podemos explorar as mesmas operações definidas para as curvas elípticas em R, como a soma de pontos. Utilizaremos o algoritmo da soma de pontos na curva elíptica já apresentado anteriormente para gerar uma tábua de soma desses pontos.

- Se $P_1 = (2, 2)$ e $P_2 = \mathcal{O}$, então $P_1 \oplus P_2 = (2, 2)$.
- Se $P_1 = (2, 2)$ e $P_2 = (2, 9)$, então $P_1 \oplus P_2 = \mathcal{O}$. Nesse caso P_1 e P_2 são simétricos em relação a um eixo horizontal pois a abscissa é a mesma e as ordenadas são tais que $9 \equiv -2 \bmod 11$.

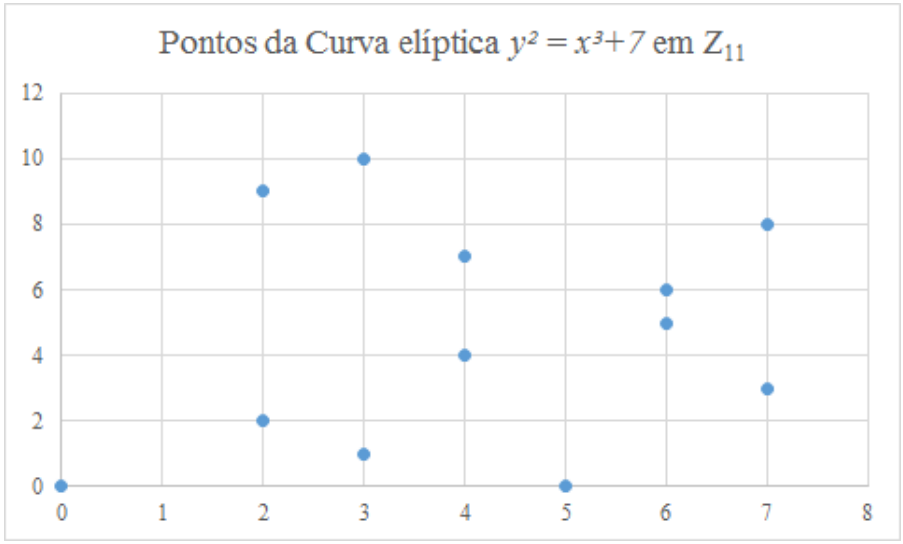


Figura 3.7: Pontos da curva elíptica $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$

– Se $P_1 = (2, 2)$ e $P_2 = (4, 7)$, então $P_1 \neq P_2$ e para obter-se $P_1 \oplus P_2 = P_3$ se faz:

$$\lambda \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p \Leftrightarrow \lambda \equiv (y_2 - y_1)\,(x_2 - x_1)^{-1} \bmod p$$

Substituindo as coordenadas de P_1 e P_2 temos:

$$\lambda \equiv (7 - 2)\,(4 - 2)^{-1} \bmod 11$$

Mas pela Tábua do produto em Z_{11}, temos que $2^{-1} \equiv 6 \bmod 11$, assim $\lambda \equiv 30 \equiv 8 \bmod 11$

De posse de λ, P_1 e P_2, obtemos as coordenadas de P_3 da seguinte maneira:

- $x_3 = \lambda^2 - x_2 - x_1 \equiv 8^2 - 4 - 2 \equiv 3 \bmod 11$
- $y_3 = \lambda(x_1 - x_3) - y_1 \equiv 8(2 - 3) - 2 \equiv -10 \equiv 1 \bmod 11$

Assim, temos que :

$$(2, 2) \oplus (4, 7) = (3, 1)$$

– Se $P_1 = (2, 2)$ e $P_2 = (2, 2)$, então $P_1 = P_2$ e para obter-se $P_1 \oplus P_1 = 2P_1$ se faz:

$$\lambda \equiv (3x_1^2 + a)(2y_1)^{-1} \bmod p$$

Substituindo as coordenadas de P_1 e P_2 temos:

$$\lambda \equiv \left(3.2^2 + 0\right)(2.2)^{-1} \bmod 11$$

Mas pela Tábua do produto em $\mathbb{Z}_{11}$, temos que $4^{-1} \equiv 3 \bmod 11$, assim $\lambda \equiv 36 \equiv 3 \bmod 11$

Assim como fizemos no caso anterior, obtemos as coordenadas de P_3 da seguinte maneira:

- $x_3 = \lambda^2 - x_2 - x_1 \equiv 3^2 - 2 - 2 \equiv 5 \bmod 11$
- $y_3 = \lambda(x_1 - x_3) - y_1 \equiv 3(2-5) - 2 \equiv -11 \equiv 0 \bmod 11$

Assim, temos que :

$$(2,2) \oplus (2,2) = (5,0)$$

Repetindo essas operações para a soma de todos os pontos , podemos obter a tábua da soma dos pontos de $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$

⊕	O	(2,2)	(2,9)	(3,1)	(3,10)	(4,4)	(4,7)	(5,0)	(6,5)	(6,6)	(7,3)	(7,8)
O	O	(2,2)	(2,9)	(3,1)	(3,10)	(4,4)	(4,7)	(5,0)	(6,5)	(6,6)	(7,3)	(7,8)
(2,2)	(2,2)	(5,0)	O	(7,3)	(4,4)	(6,5)	(3,1)	(2,9)	(7,8)	(4,7)	(6,6)	(3,10)
(2,9)	(2,9)	O	(5,0)	(4,7)	(7,8)	(3,10)	(6,6)	(2,2)	(4,4)	(7,3)	(3,1)	(6,5)
(3,1)	(3,1)	(7,3)	(4,7)	(3,10)	O	(2,2)	(7,8)	(6,6)	(5,0)	(6,5)	(4,4)	(2,9)
(3,10)	(3,10)	(4,4)	(7,8)	O	(3,1)	(7,3)	(2,9)	(6,5)	(6,6)	(5,0)	(2,2)	(4,7)
(4,4)	(4,4)	(6,5)	(3,10)	(2,2)	(7,3)	(6,6)	O	(7,8)	(4,7)	(2,9)	(5,0)	(3,1)
(4,7)	(4,7)	(3,1)	(6,6)	(7,8)	(2,9)	O	(6,5)	(7,3)	(2,2)	(4,4)	(3,10)	(5,0)
(5,0)	(5,0)	(2,9)	(2,2)	(6,6)	(6,5)	(7,8)	(7,3)	O	(3,10)	(3,1)	(4,7)	(4,4)
(6,5)	(6,5)	(7,8)	(4,4)	(5,0)	(6,6)	(4,7)	(2,2)	(3,10)	(3,1)	O	(2,9)	(7,3)
(6,6)	(6,6)	(4,7)	(7,3)	(6,5)	(5,0)	(2,9)	(4,4)	(3,1)	O	(3,10)	(7,8)	(2,2)
(7,3)	(7,3)	(6,6)	(3,1)	(4,4)	(2,2)	(5,0)	(3,10)	(4,7)	(2,9)	(7,8)	(6,5)	O
(7,8)	(7,8)	(3,10)	(6,5)	(2,9)	(4,7)	(3,1)	(5,0)	(4,4)	(7,3)	(2,2)	O	(6,6)

Figura 3.8: Tábua das somas dos pontos de $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$

Observe que a diagonal da tábua foi destacada para representar as operações que dobram os pontos, por exemplo $(4,7) \oplus (4,7) = (6,5)$, isso facilita quando calculamos multiplos de pontos, como o exemplo abaixo:

Exemplo 3: Seja $P = (7,3)$ pertencente à curva $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$. Calcule $9 \otimes P$.

Vemos que $9 \otimes P$ pode ser escrito como $2 \otimes (2 \otimes (2 \otimes (P))) \oplus P$, essa maneira de decomposição é conveniente, pois agiliza o processamento e diminui o custo computacional. Em vez de realizarmos 8 operações de soma com P, fazemos 3 operações de dobra de P e uma operação de soma com P.

Assim, pela tábua acima, temos que:

$2 \otimes P = (7,3) \oplus (7,3) = (6,5)$

$4 \otimes P = 2 \otimes (2 \otimes P) = (6,5) \oplus (6,5) = (3,1)$

$8 \otimes P = 2 \otimes (2 \otimes (2 \otimes P) = (3,1) \oplus (3,1) = (3,10)$

$9 \otimes P = 8 \otimes P \oplus P = (3,10) \oplus (7,3) = (2,2)$

3.5 Parâmetros das Curvas Elípticas

3.5.1 Ordem de uma Curva elíptica

A ordem de uma curva elíptica sobre um corpo finito de p elementos é a quantidade de pontos que essa curva possui (N). O Teorema de Hasse [20] estabelece o seguinte:

$$p + 1 - 2p \leqslant N \leqslant p + 1 - 2p$$

Tomando o exemplo de $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$ temos que $p = 11$, portanto $6 \leqslant N \leqslant 18$. Como já sabemos todos os pontos de E, determinamos $N = 12$. Existe um algoritmo (algoritmos de schoof) de ordem polinomial [6] para determinar a ordem de uma curva elíptica, o leitor pode obter mais detalhes em [24].

3.5.2 Subgrupos cíclicos

Seja $P = (2, 9)$ pertencente à curva $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$, podemos utilizar a tábua construida anteriormente (Figura 3.8) para observar os múltiplos de P:

$0 \otimes P = \mathcal{O}$
$1 \otimes P = (2, 9)$
$2 \otimes P = (2, 9) \oplus (2, 9) = (5, 0)$
$3 \otimes P = (2, 9) \oplus (5, 0) = (2, 2)$
$4 \otimes P = (2, 9) \oplus (2, 2) = O$
$5 \otimes P = (2, 9) \oplus \mathcal{O} = (2, 9)$
$6 \otimes P = (2, 9) \oplus (2, 9) = (5, 0)$
...

Podemos verificar que os múltiplos de P são apenas 4 $(\mathcal{O}, (2, 9), (5, 0), (2, 2))$ e eles se repetem ciclicamente, assim podemos dizer que esse conjunto dos múltiplos de P é um subgrupo cíclico de $E : y^2 = x^3 + 7$ sobre $\mathbb{Z}_{11}$ e o Ponto P é chamado Ponto Gerador desse subgrupo. Os subgrupos cíclicos são a base da aplicação das curvas elípticas na criptografia [11].

3.5.3
Ordem e Cofator de um Subgrupo

Definimos a ordem como a quantidade de pontos de um grupo, mas para um subgrupo cíclico a ordem se torna o menor inteiro positivo (n) tal que $n \otimes P = \mathcal{O}$. No exemplo anterior, de fato, o subgrupo contém 4 pontos ($n = 4$) e $4 \otimes P = \mathcal{O}$.

O cofator de um subgrupo (h), pode ser definido como a razão entre a ordem da curva elíptica sobre o corpo finito e a ordem do subgrupo: $h = N/n$. No nosso exemplo, $N = 12$ e $n = 4$, logo $h = 3$;

3.6
Problema do logaritmo discreto elíptico

Baseado no problema do logaritmo discreto é possível construir analogias aplicáveis à operação de soma de dois pontos de uma curva elíptica sobre $\mathbb{Z}_p$, pois essa operação forma um grupo abeliano e os resultados estudados anteriormente são válidos. Nesse caso, o grupo multiplicativo é substituído por $E(\mathbb{Z}_p)$ e analogamente ao que vimos na proposição 2, estamos considerando um grupo finito e cíclico e precisamos de um elemento que seja gerador do grupo. Assim, seja P um ponto de $E(\mathbb{Z}_p)$ tal que P seja um gerador de $E(\mathbb{Z}_p)$. Deste modo, para cada $Q \in E(\mathbb{Z}_p)$ existe um inteiro $n \in \mathbb{Z}_p$ tal que:

$$Q = n \otimes P$$

Podemos dizer que n é o Logaritmo Discreto Elíptico de Q em relação a P. O Problema do Logaritmo Discreto Elíptico consiste em determinar n para cada ponto Q. Entretanto, vimos anteriormente que a operação de soma entre pontos de uma curva elíptica é mais complexa que a operação de multiplicação, o que torna esse problema mais difícil de ser resolvido do que o problema do logaritmo discreto em $\mathbb{Z}_p$.

3.7
Troca de Chaves - sistema DHM aplicado às curvas elípticas sobre Z_p

Vários métodos criptográficos se baseiam na aritmética das curvas elípticas para serem implementados. Todos eles são regidos por um protocolo de troca de chaves, no qual cada agente da comunicação possui uma chave secreta e são conhecidas a curva elíptica sobre a qual serão feitas as operações e um ponto público dessa curva. Voltemos à situação hipotética de que João e Maria desejarem trocar entre si uma chave secreta em um meio de comunicação inseguro:

1. João e Maria escolhem um primo p, uma curva $E(\mathbb{Z}_p) : y^2 = x^3 + Ax + B$, com $4A^3 + 27B^2 \neq 0$, e um ponto $P \in E(\mathbb{Z}_p)$ gerador do grupo. Esses dados serão públicos

2. Maria escolhe uma chave secreta $n_m \in \mathbb{Z}_p$, calcula uma chave pública $Q_m = n_m \otimes P$ e envia Q_m para João.

3. João escolhe uma chave secreta $n_j \in \mathbb{Z}_p$, calcula uma chave pública $Q_j = n_j \otimes P$ e envia Q_j para Maria.

4. Maria calcula $R_m = n_m \otimes Q_j$:

$$R_m = n_m \otimes Q_j = n_m \otimes (n_j \otimes P) = (n_m \otimes n_j) \otimes P$$

5. João calcula $R_j = n_j \otimes Q_m$:

$$R_j = n_j \otimes Q_m = n_j \otimes (n_m \otimes P) = (n_m \otimes n_j) \otimes P$$

6. Dessa maneira João e Maria compartilham a chave $R_m = R_j = R$

Caso um agente externo intercepte a comunicação terá descoberto as chaves públicas Q_m, Q_j mas teria que resolver o problema do logaritmo discreto elíptico para descobrir as chaves secretas de João e Maria. Com essa chave secreta trocada João e Maria podem se comunicar utlizando um criptosistema qualquer, como o ElGamal, por exemplo.

Vale ressaltar que estamos falando aqui de primos muito grandes, a título de comparação, se p é da ordem $\approx 2^{160}$, em termos computacionais o problema do logaritmo discreto elíptico, apresenta uma dificuldade equivalente ao problema do logaritmo discreto em um grupo multiplicativo de ordem $\approx 2^{1000}$ [6]. Uma implicação disso é que a criptografia baseada em curvas elípticas requer chaves menores, isso explica o motivo dos esquemas de assinatura digital empregados no Bitcoin e em outras criptomoedas utilizarem essa criptografia.

4
Criptografia aplicada no protocolo Bitcoin

Nesse capítulo vamos apresentar a curva elíptica utilizada para geração de chaves no protocolo Bitcoin, explorar o conceito de assinatura digital com o algoritmo ECDSA e as etapas que compõem uma transação.

4.1
Parâmetros das Curvas Elípticas sobre um corpo finito $(\mathbb{Z}_p)$

Para padronizar a aplicação das curvas elípticas em criptografia se convencionou, por meio de normas específicas desenvolvidas por associações de empresas da área [25], parametriza-las pela seguinte sêxtupla:

$$T = (p, a, b, G, n, h)$$

onde :
p = Dimensão do corpo finito;
a e $b \in \mathbb{Z}_p$ são os coeficientes de $E(\mathbb{Z}_p) : y^2 \equiv x^3 + ax + b \bmod p$;
G = Ponto gerador $\in \mathbb{Z}_p$ de coordenadas (x_G, y_G);
n = ordem do subgrupo gerado por G (nessa convenção n deve ser primo)
h = cofator do subgrupo gerado por G

4.1.1
A Curva elíptica do Protocolo Bitcoin

O SCGE (Standards for Efficient Cryptography Group) é um consórcio internacional de empresas que concentra essa padronização das curvas elípticas para fins de criptografia e emite documentos com diretrizes e parâmetros recomendados para cada curva a fim de aumentar a eficiência e interoperabilidade das aplicações [26]. No documento [27] é definida a curva $secp256k1$, que é a curva sobre a qual ocorre a criptografia do protocolo Bitcoin, nessa nomenclatura há informações importantes, como sinaliza SEGUIAS, 2020 [6]:

• "sec": abreviação de "Standards for Efficient Cryptography" indica que a curva pertence ao rol de curvas estudadas e padronizadas pelo SECG;

• "p": referência ao fato de a curva ter sido definida sobre um corpo finito de ordem p, com p primo;

• "256": indica que os pontos da curva têm abscissas e ordenadas de 256 bits;
• "k": Estabelece que essa curva é da família das curvas de Koblitz, que admitem implementações mais eficientes no cálculo da soma dos pontos, maiores detalhes podem ser encontrados em GALLANT et al, 2001 [28].
• "1": Indica que é a primeira curva do seu tipo que satisfaz os atributos acima;

Para essa curva, a sêxtupla de parâmetros associados em base hexadecimal são os seguintes [27]:

• Parâmetro p:
p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F$_{(16)}$
Cada dígito representa um número de 4 bits, como são 64 dígitos o parâmetro p é representado por um número de 256 bits, que convertido para a base decimal
$= 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 =$
$=$ 115792089237316195423570985008687907853269984665640564039457584007908834671663;

• Parâmetro a:
a = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000$_{(16)} = 0$;

• Parâmetro b:
b = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007$_{(16)} = 7$;

Como $p \notin \{2, 3\}$ e $4a^3 + 27b^2 \neq 0 \bmod p$; essa curva é não-singular e pode ser escrita na forma de Weierstrass:

$$E(\mathbb{Z}_p) = \left\{(x, y) \in \mathbb{Z}_p^2 \mid y^2 \equiv x^3 + 7 \bmod p\right\} \cup \{\mathcal{O}\}$$

Em particular, quando $p = 11$ encontra-se exatamente a curva que foi trabalhada em exemplos no capítulo 3, para a qual foram calculados todos os pontos como mostramos na (Figura 3.7). Entrentanto, quando se trabalha com implementações comerciais, como as propostas pelo SECG, os valores de p tornam impossível se conhecer a tábua de soma dos pontos, da mesma maneira que foi feito na (Figura 3.8).

• Parâmetro G:
G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8.
Como vimos, G é um ponto da curva, o ponto gerador, a partir do qual serão obtidas as chaves públicas e suas coordenadas são obtidas da seguinte maneira

convencionada em [27]: $G = 04_{(16)} \parallel (x_G)_{(16)} \parallel (y_G)_{(16)}$.
Assim, $G = (x_G, y_G)$, com
x_G = 50662630222773436695787188951685343262506034537775941755001873603891167292 40;
y_G = 32670510020758816978083085130507043184471273380659243275938904335757337482424;

• Parâmetro n:
n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141$_{(16)}$
= 11579208923731619542357098500868790785283756427907490438260516314151816149433 7;

• Parâmetro h:
Sendo n a ordem do subgrupo gerado por G, sabe-se que n deve ser divisor da ordem da curva elíptica (N) por conta do Teorema de Lagrange que vimos no Capítulo 3, o que de fato acontece quando se define o cofator $h = N/n$ como a unidade: $h = 01_{(16)} = 1$;

Isso significa que a ordem do subgrupo gerado por G é igual a ordem da curva, isto é, $n = N$, como n é primo, N também é primo, assim os pontos da curva $secp256k1$ formam um grupo cíclico e qualquer um dos seus elementos é um potencial ponto gerador.

4.2 Mecanismo de geração de chaves públicas na Curva Elíptica

A curva $secp256k1$ tem um papel muito importante no protocolo Bitcoin: é a partir dela que obtemos uma chave pública a partir de uma chave privada, a chave privada, por sua vez, é dada por um inteiro qualquer escolhido no intervalo $[1, (n-1)]$.

Como curiosidade, vale notar que n, na base decimal, é um número da ordem de grandeza de 10^{77}, o que comparando com um universo "imaginável" de quantidades, teria a mesma ordem de grandeza que o produto da quantidade aproximada de átomos em um corpo humano [29] pela quantidade de átomos aproximada do planeta terra [30], como d é um inteiro no intervalo $[1, n-1]$ pode-se concluir que tentar determinar a chave privada por tentativa e erro seria inviável computacionalmente. Utilizando um supercomputador para tentar checar todas as possibilidades desse valor, se estima que seriam necessários centenas de anos para processar todas as possibilidades [31].

De posse de uma chave privada d do ponto gerador G, definido anteriormente como parâmetro, podemos obter um chave pública $H = (x_H, y_H)$ fazendo a seguinte

operação sobre a curva $secp256k1$:

$$H = d \otimes G = \underbrace{G \oplus G \oplus G ... \oplus G}_{d\ parcelas}$$

Como vimos no final do Capitulo 3, fazer a operação no sentido de gerar a chave pública é relativamente simples utlizando as operações de adição e "dobra" dos pontos já definidas, entretanto, no sentido inverso, há que se resolver o problema do logaritmo discreto elíptico que, até o momento se mostra um problema intratável computacionalmente, pois requer um tempo de processamento muito extenso para ser utilizado na prática.

Assim, toda a segurança do protocolo Bitcoin reside na dificuldade de se obter a chave secreta dada a chave pública e a curva sobre a qual a operação é feita.

4.3 Assinando uma mensagem com ECDSA

Para que uma transação seja realizada na blockchain do protocolo Bitcoin é necessário garantir que o usuário que detém a chave secreta, de fato deu origem à transação. Imagine que Maria escolheu sua chave privada d e obteve sua chave pública $H = d \otimes G$, agora Maria gostaria de realizar uma transação de compra ou venda que pode ser interpretada como uma mensagem m. Essa mensagem precisa ser assinada por Maria para ser validada. A rede Bitcoin utiliza o algoritmo ECDSA (Elliptic Curve Digital Signature Algorithm) como meio dos usuários assinarem suas mensagens.

Primeiramente Maria, para aumentar a segurança, decide criptografar sua mensagem e aplica uma função Hash criptográfica à mensagem que escreveu (no caso do protocolo bitcoin utiliza-se a função SHA-256), obtendo um número na base hexadecimal $Hash(m)_{(16)}$ e convertendo para a base decimal obtemos o inteiro z, a partir de então serão seguidos os passos a seguir [10] para se obter a assinatura digital da mensagem, que denotaremos como $\sigma(m)$:

1. Escolhe-se aleatoriamente um parâmetro $k \in \mathbb{Z}_n^*$, com n primo. É importante que esse parâmetro seja diferente para cada assinatura realizada, pois repeti-lo gera vulnerabilidades na criptografia [7].

2. Calcula-se o ponto $P \in secp256k1$:

$$P = (x_P, y_P) = k \otimes G = G \oplus G \oplus \ldots \oplus G \ (k \text{ vezes})$$

3. Define-se $r \equiv x_P \bmod n$. Se $r = 0$, então deve-se retornar ao passo 1 e escolher outro valor de k.

4. Define-se $s \equiv k^{-1}(z + rd) \bmod n$. onde k^{-1} é o inverso multiplicativo de $k \bmod n$. Se $s = 0$ então deve-se retornar ao passo 1 e escolher outro valor de k. Note que para garantir que qualquer k escolhido admita um inverso

multiplicativo, temos que ter n primo, caso contrário o algoritmo ECDSA não poderá ser implementado.

Assim, através da aplicação do algoritmo ECDSA obtem-se o par ordenado $\sigma(m) = (r, s)$ que representa a assinatura digital, esse par ordenado é intrínseco à mensagem que foi transmitida, quer dizer que se algum agente tentar mudar o conteúdo da mensagem, desviando a transação por exemplo, a assinatura não é mais validada, pois alteraria o seu hash e o valor de s seria alterado.

4.4 Verificando uma Assinatura Digital

Como garantir que uma determinada pessoa, de fato executou uma dada transação? Faz-se necessária a existência de uma função de verificação, mas como verificar a veracidade sem conhecer a chave secreta dessa pessoa? A resposta está na chave pública, se a conhecermos, e tivermos acesso à mensagem assinada seremos capazes de verificar se, de fato, a pessoa executou a operação ou se trata-se de uma fraude. Suponha que João quer verificar se a assinatura digital pertence à Maria, para tanto, ele vai precisar da mensagem m, da assintura digital $\sigma(m) = (r, s)$ e da chave pública de Maria H.

A primeira ação de João é realizar algumas checagens básicas, como conferir se H é um ponto de $secp256k1$, se $H \neq \mathcal{O}$ e se $r, s \in \mathbb{Z}_n^*$ se qualquer uma dessas condições não for satisfeita a assinatura não é válida. Caso contrário:

1. Definem-se $u \equiv z.s^{-1} \bmod n$ e $v \equiv r.s^{-1} \bmod n$, nos quais s^{-1} denota o inverso multiplicativo de $s \bmod n$
2. Define-se $W = (x_w, y_w) = (u \otimes G) \oplus (v \otimes H)$
3. Se $r \equiv x_w \bmod n$ então a assinatura é verdadeira, caso contrário, a assinatura é falsa.

Agora vamos mostrar porque essa verificação funciona, ou seja, toda assinatura gerada como mostrado apresenta uma saída verdadeira quando verificada pelo procedimento acima. Note que definimos s:

$$s \equiv k^{-1}.(z + r.d) \bmod n$$
$$k.s \equiv (z + r.d) \bmod n$$
$$k \equiv s^{-1}.(z + r.d) \bmod n$$

Utilizando o algoritmo de verificação:

$W = (u \otimes G) \oplus (v \otimes H)$	Mas $H = d \otimes G$
$W = (u \otimes G) \oplus (v.d \otimes G)$	Substituindo u e v
$W = (z.s^{-1} \otimes G) \oplus (r.s^{-1}.d \otimes G)$	fatorando
$W = s^{-1}.(z + r.d) \otimes G$	

$W = k \otimes G = P$

Da igualdade acima pode-se concluir que $r \equiv x_p \equiv x_w \bmod n$

Pudemos realizar tais operações por conta das propriedades que discutimos no Capítulo 3. Esse algoritmo de assinatura digital é o "coração" da segurança das transações de Bitcoin. A partir dele se confere autenticidade às operações e se constrói a confiança dos usuários na rede Bitcoin.

4.5 Transações de Bitcoin

Uma transação de Bitcoin consiste basicamente na transferência do controle de gastos de um detentor de BTC para outro, esse controle refere-se ao desbloqueio de um determinado valor em BTC. A menor unidade transacionável do BTC é denominada *Satoshi* e representa 10^{-8} de uma unidade de BTC. Assim, uma transação de 100 Satoshis de Maria para João, transfere o controle de gastos de 10^{-6} BTC de Maria para João, que agora pode gastá-los como bem entender.

Para o propósito desse trabalho, é suficiente destacar que uma transação de Bitcoin é uma estrutura de dados que abrange quatro categorias principais de informações [7]:

1. *Fonte(s) dos fundos a serem transferidos*, também conhecida como saída(s) de transação não gasta ou UTXO (Unspent Transaction Outputs). Cada UTXO contém uma quantidade específica de Satoshis, cujo controle foi transferido de um detentor anterior para aquele que iniciou a transação Bitcoin atual. UTXOs tornam-se entradas para a transação Bitcoin atual.

2. *Destino(s) dos fundos*, ou seja, o(s) destinatário(s) pretendido(s), que terão controle sobre gastos dessas UTXOs.

3. *Quantia exata a ser enviada* para cada endereço de destino.

4. *Informações contendo a(s) assinatura(s) ECDSA* dos UTXO(s) correspondentes(s) para verificar a autenticidade do(s) remetente(s) de cada UTXO.

Cada transação de Bitcoin que ocorre na blockchain pode ser vista por todos os nós da rede, há vários sites que se dedicam a publicação desses dados como o *blockchain.info* que exibe, dentre várias informações específicas para cada transação dados sobre o emissor e receptor, quantidade de bitcoins, horário e parâmetros que permitem a qualquer nó da rede verificar se a transação é verdadeira.

De posse da chave privada e das informações contidas na transação é gerada uma assinatura digital que garante que o detentor dos Bitcoins unicidade e garantia de autencidade, pois só com sua chave privada e seus dados de transação se obtem essa assinatura, como vimos na seção anterior. Primeiro é aplicada a função Hash criptográfica SHA-256 à mensagem e depois são feitas as operações para se obter

o par ordenado que representa a assinatura digital $\sigma(r, s)$. Ilustramos no esquema abaixo da Fig 4.1 adaptada de [32] :

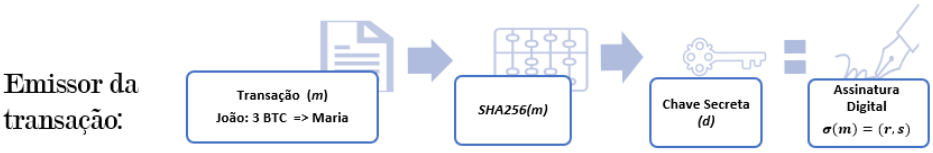


Figura 4.1: Esquema das etapas do processo de assinatura digital

Em caso de qualquer pessoa tentar alterar a mensagem, mudando, por exemplo o destinatário dos fundos, a assinatura se alteraria e a função de verificação não vai reconhecer essa assinatura como verdadeira. O que chamamos aqui de função de verificação nada mais é do que uma implementação do ECDSA, que utiliza a chave pública do emissor para, a partir da assinatura, obter o hash criprográfico da mensagem e comparar com o Hash criptográfico dos dados da mensagem, se os hash forem iguais então a assinatura é verdadeira, caso contrário é falsa. como ilustrado na Fig 4.2 adatpada de [32].

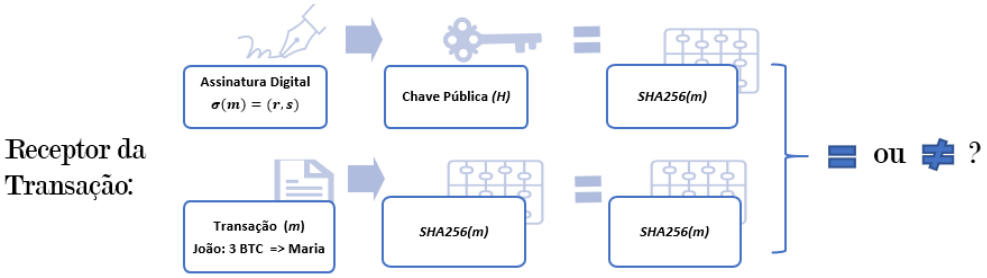


Figura 4.2: Etapas do processo de autenticação da assinatura digital

Utilizaremos o esquema da Fig 4.3 adaptada de [18] para discutirmos os elementos de segurança nas transações. Basicamente, como já vimos, tudo se inicia com a escolha de uma chave privada, que pode ser comparada à sua senha do banco. A partir desssa, obtemos a chave pública (através de operações na curva elíptica $secp256k1$), que pode ser comparado ao seu CPF, por exemplo, pois é uma informação que te identifica, mas não permite acesso aos seus fundos. Entretanto, na rede Bitcoin as transações não se dão entre chaves públicas, mas sim entre endereços (*address*), que podem ser comparados à um endereço de email, pois ainda te identifica, mas não permite uma consulta a nenhum dado sobre as suas dívidas, como o CPF possibilita. Vamos ver adiante como um endereço é formado a partir de uma chave pública.

Em resumo, um endereço Bitcoin é uma sequência de caracteres alfanuméricos usada para representar um destino de fundos. É importante destacar que um endereço não é uma carteira e não contém saldos de fundos, sempre que um determinado endereço é usado para gastar alguns de seus Bitcoins, todo o seu conteúdo é debitado: parte vai para o destinatário, parte é paga como uma taxa

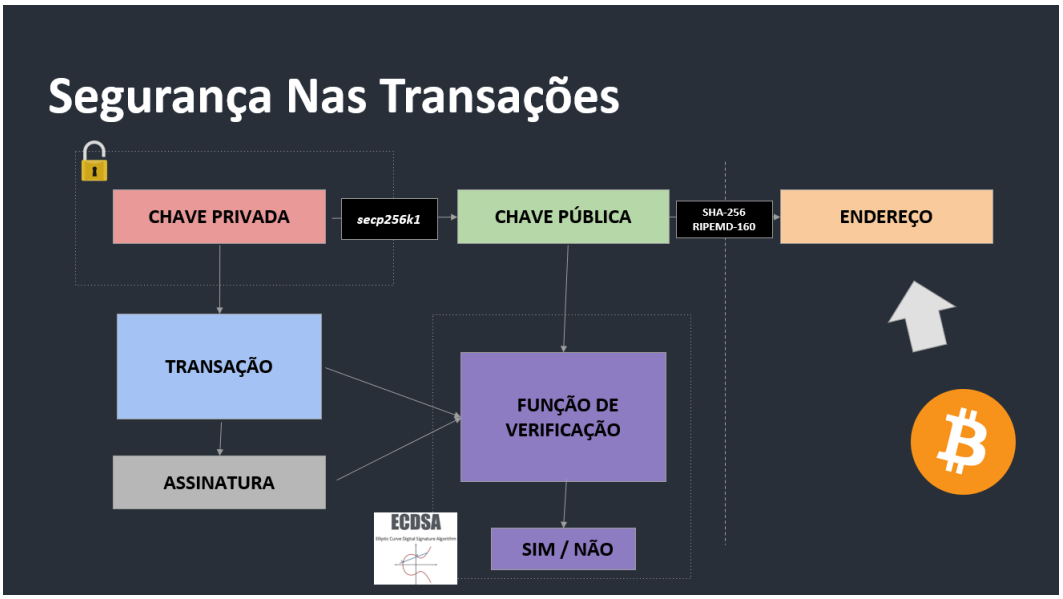


Figura 4.3: Esquema das operações que garantem segurança às Transações

ao minerador, e o saldo restante (se houver) é armazenado em um novo endereço conhecido como endereço de alteração. Qualquer pessoa ou entidade pode ter quantos endereços Bitcoin desejar. Na verdade, é recomendável criar um novo endereço a cada nova transação (uma prática que as carteiras modernas implementam automaticamente).

Os endereços Bitcoin são de dois tipos: os que iniciam com o dígito "1" e os que se iniciam com o dígito "3", correspondentes aos dois casos a seguir[9]:

1. O destino dos fundos é um único destinatário que tem controle total sobre os fundos e, por isso, pode gastá-los como quiser.

2. O destino dos fundos é uma estrutura mais complexa que especifica certas regras que precisam ser cumpridas para que os fundos sejam gastos ou desbloqueados.

Nesse texto vamos nos ater aos endereços do primeiro tipo, conhecidos como P2PKH *(Pay to Public Key Hash)* . A justificativa para o nome vem do fato de que tudo o que é necessário para criar esse endereço é um hash da chave pública para gerarmos o endereço Bitcoin associado. Utilizamos duas funções hash unilaterais (não admitem inversa), a saber, SHA-256 e RIPEMD-160 (Função hash criptográfica RIPE Message Digest). Enquanto a saída da função SHA-256 é um número de 256 bits (ou seja, 32 bytes), a RIPEMD-160 produz números de 160 bits (ou seja, 20 bytes). Em resumo, o procedimento é o seguinte:

1. Dada uma chave pública, aplica-se SHA-256;

2. Aplica-se RIPEMD-160 na saida da função SHA-256 do item anterior;

3. Após algumas operações aritméticas e aplicações sucessivas de SHA-256 (detalhadas no site abaixo) codifica-se o resultado para a base 58, obtendo-se o endereço Bitcoin. Essa base é composta pelos 58 elementos: 123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz, note que são

exluídos o algarismo zero, as letras "O"e "I" maiúsculas e letra "i" minúscula para evitar confusões e erros de digitações.

Em *http://gobittest.appspot.com/Address*, podem ser verificados os passos detalhados para geração de um endereço. Um exemplo de endereço segue abaixo:

16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM

Podemos entender o uso de endereços nas transações como uma camada adicional de segurança, que dificulta ainda mais o acesso à chave pública e, como a função hash criptográfica RIPEMD-160 retorna números de 20 bytes, os endereços são computados mais rapidamente na rede.

Agora suponha que João gostaria de pagar 100 satoshis para Maria, primeiro Maria deve fornecer o seu endereço Bitcoin. Então João precisa criar uma transação com destino a esse endereço e que deve utilizar como entrada os recursos de João obtidos em uma transação anterior na qual recebeu Bitcoins, ele pode ter comprado numa exchange, minerado ou recebido de um amigo, por exemplo.

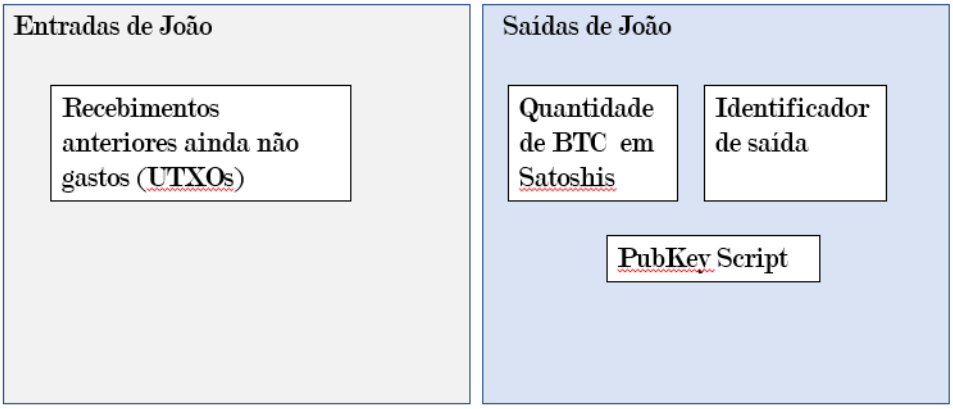


Figura 4.4: Esquema transação de João para Maria

Nesse esquema adaptado de [32] vemos que a saída da transação de João é composta por:

- Quantidade de BTC em satoshis a serem enviados;
- PubKey Script: é uma rotina que ao ser implementada vai bloquear a quantidade de satoshis (100 nesse caso) no endereço de Maria;
- Identificador de saída que ajuda a rede a rastrear rapidamente o registro dessa transação quando Maria precisar gastar esses fundos.

Com isso, Maria agora é a única pessoa que detem a chave secreta capaz de resgatar os satoshis recebidos nessa transação.

Assim que esta transação for validada e minerada pelos mineradores. A carteira de Maria irá computar que ela possui 100 Satoshis. Na realidade, a carteira tem um algoritmo que percorre as transações da blockchain e apenas rastreia as saídas

que correspondem ao endereço de Maria. Essas saídas são chamadas de Unspent Transaction Outputs (UTXOs) e ficam ociosas até que o detentor da chave privada forneça as informações necessárias para desbloquear a saída e enviar os Bitcoins para outro endereço criando outro UTXO.

A rede Bitcoin é apenas uma teia de UTXOs que esperam para ser desbloqueadas e enviadas para outros endereços como novos UTXOs. O protocolo Bitcoin não é um sistema baseado em contas, o que significa que os usuários não têm uma conta que acumula Bitcoin como em uma conta bancária. Em vez disso, os usuários possuem chaves privadas que desbloqueiam UTXOs na rede. Isso também significa que você não pode enviar uma quantidade parcial de satoshis de um UTXO. Cada vez que uma nova transação é criada todos os satoshis bloqueados nesse UTXO são enviados ao endereço de destino.

Se o usuário não quiser enviar todos os satoshis para outro usuário, ele também deve incluir um endereço para os satoshis restantes que deseja manter e esses satoshis são enviados de volta. Embora não seja a melhor opção, o mesmo endereço pode ser reutilizado para enviar de volta os satoshis restantes. Entretanto, para aumentar a privacidade, é mais seguro criar um novo endereço para enviar os satoshis restantes.

Agora que Maria tem acesso ao UTXO que recebeu de João com 100 satoshis bloqueados nele, sua amiga Ana também quer entrar na rede e usar bitcoin. Para enviar seus 100 satoshis, Maria deve criar uma nova transação que consiste em uma entrada e uma saída.

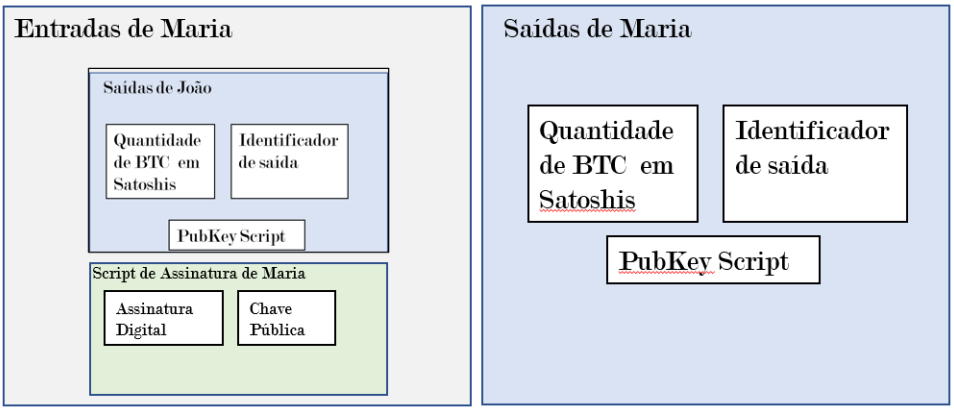


Figura 4.5: Esquema de Transação de Maria para Ana

Primeiramente, será utilizado o identificador da transação de saída de João para encontrar corretamente o UTXO e o PubKey Script correspondente que alguém deve satisfazer para poder gastar os bitcoins bloqueados.

Em seguida, Maria cria um Script de assinatura, que será usado para satisfazer o PubKey Script criado por João e desbloquear os 100 satoshis. Esse script de assinatura digital é responsável pela autenticidade de toda a transação que Maria está criando para Ana. Pois nele há a informação que desbloqueia os satohis do PubKey script criado por João, do novo PubKey Script para Ana e a quantidade de satoshis que será destinada ao endereço de Ana. Todos esses dados são criptografados

duas vezes usando a função hash criptográfica SHA-256. Em seguida, Maria assina esse conteúdo com sua chave privada para criar a assinatura digital, que combinado com sua chave pública, forma o script de assinatura para essa transação

Assim como a saída da transação anterior, a saída consiste em: um novo indicador de saída para localizar essa transação, a quantidade de satoshis e um novo PubKey Script usando o endereço que Ana forneceu para Maria, que bloqueia os Bitcoins no endereço de Ana. Agora, apenas Ana, que possui a chave privada que corresponde ao endereço, pode enviar esses Bitcoins para outra pessoa.

Uma vez que a transação de Maria para Ana for criada, será transmitida para a rede de mineradores, que por sua vez irão comparar o script de assinatura por meio do PubKey Script, como representado na Fig 4.2. Se o resultado do PubKey Script for verdadeiro, essa transação será adicionada a um bloco e validada na rede.

5
Considerações finais

O presente trabalho tratou de apresentar suscintamente o universo da criptomoeda mais conhecida, o Bitcoin. Discorreu-se sobre a matemática implícita ao processo de assinaturas digitais com o emprego de criptografia sobre as curvas elípticas em $\mathbb{Z}_p$, em particular sobre a curva $secp256k1$ e como se dão de fato as transações entre os endereços da rede.

O contexto das criptomoedas é diverso e apresenta várias aplicações diferentes, como o Bitcoin foi a pioneira, hoje carrega o status de "ouro digital" de maneira que sua escassez é garantida pela política monetária de emissões de novas moedas como vimos no Capítulo 2. Também com grande destaque, tem-se a Criptomoeda Ethereum, muito utilizada em aplicações comerciais com o conceito de "smart contracts"(contratos inteligentes) que se executam automaticamente a partir da leitura de um determinado parâmetro. Mais recentemente houve também o desenvolvimento da Criptomoeda Cardano, cuja proposta é mais inovadora possibilitando que a sua blockchain se comunique com as blockchains de outras criptomoedas. Enfim, hoje o mundo conta com centenas de criptomoedas, cada uma com sua aplicação específica.

Como se vê, esse campo do conhecimento evoluiu muito desde a criação do Bitcoin em 2008[14], tornando a nossa realidade mais complexa e, esse trabalho é uma tentativa de apresentar de forma simples as aplicações matemáticas inerentes à segurança dessas transações.

Dos números primos ao Bitcoin, pudemos percorrer esse caminho fazendo pontes capazes motivar vários estudantes e educadores. Sabemos que nem todo programa de ensino de matemática abrange o ensino de congruências (infelizmente), entretanto há círculos nos quais existe esse espaço e o paralelo com a matemática que subsisida a segurança de uma criptomoeda famosa é de grande valia como atrativo para despertar o interesse por esse tema.

A fenômeno da "transformação digital" vem crescendo ao longo das últimas décadas e faz bastante sentido lançarmos o olhar da educação matemática sobre esse movimento em suas diversas manifestações, desde a aplicação de softwares em aula, bem como na abordagem que é escolhida para o estudo de determinados conteúdos. Assim, nesse texto procurou-se trazer o tema da virtualização da moeda para apresentar as aplicações muinto interessantes da matemática que permitem a um sistema dessa complexidade ser implementado.

Referências bibliográficas

[1] CHERVINSKI J., K. D.. **Introdução às tecnologias dos blockchains e das criptomoedas**. Revista Brasileira de Computação Aplicada, 11(3):12–27, set. 2019.

[2] ANTONOPOULOS, A. M.. **Mastering Bitcoin: Unlocking Digital Crypto-Currencies**. O'Reilly Media, Inc., 1st edition, 2014.

[3] DE ANDRADE, E. G.. **Criptografia com curvas elípticas**. Mestrado profissional em matemática- PROFMAT, Universidade Federal do Pará, Belém, 2016.

[4] LANA, M. C. A.. **Curvas elípticas e criptografia**. Mestrado profissional em matemática- PROFMAT, Universidade Federal de Juiz de Fora, Juiz de Fora, 2016.

[5] DA SILVA LEÃO, L. C.. **Uma introdução ao estudo de bitcoins e blockchains**. Mestrado profissional em matemática- PROFMAT, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

[6] SEGUIAS, B.. **Crypto theoretical minimum - elliptic curve groups**. Disponível em: `https://delfr.com/wp-content/uploads/2018/06/Elliptic_Curve_Group.pdf`, Acesso em: 04 Nov de 2020, 2018.

[7] SEGUIAS, B.. **Bitcoin elliptic curve digital signature algorithm (ecdsa)**. Disponível em: `https://delfr.com/wp-content/uploads/2018/12/Bitcoin_ECDSA.pdf`, Acesso em: 04 Nov de 2020, 2018.

[8] SEGUIAS, B.. **Crypto theoretical minimum - groups and finite fields**. Disponível em: `https://delfr.com/wp-content/uploads/2018/05/Groups_Fields.pdf`, Acesso em: 04 Nov de 2020, 2018.

[9] SEGUIAS, B.. **Bitcoin transactions (pre-segwit)**. Disponível em: `https://delfr.com/wp-content/uploads/2020/02/Bitcoin_Transactions_Legacy.pdf`, Acesso em: 04 Nov de 2020, 2020.

[10] CORBELLINI, A.. **Elliptic curve cryptography: Ecdh and ecdsa**. Disponível em: `https://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa/`, Acesso em: 18 Out de 2020.

[11] CORBELLINI, A.. **Elliptic curve cryptography: finite fields and discrete logarithms**. Disponível em: https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/, Acesso em: 12 Out de 2020, 2015.

[12] DEVELOPERS, W.. **Whatsapp encryption overview - technical white paper. version 3 updated october 22, 2020**. https://www.whatsapp.com/ Disponível em:https://scontent.whatsapp.net/v/t39.8562-34/122249142_469857720642275_2152527586907531259_n.pdf/WA_Security_WhitePaper.pdf?ccb=1-3&_nc_sid=2fbf2a&_nc_ohc=o2E6IrSusDAAX-W5v7u&_nc_ht=scontent.whatsapp.net&oh=cc790aed9da886ee25beb5cb00d1b5db&oe=60E0CD99, Acesso em: 10 Mai de 2021.

[13] LISA, A.. **Here's a bitcoin timeline for everything you need to know about the cryptocurrency**. Yahoo Finance Disponível em:https://finance.yahoo.com/news/bitcoin-timeline-everything-know-cryptocurrency-120003591.html, Acesso em: 02 Jun de 2021.

[14] NAKAMOTO, S.. **Bitcoin: A peer-to-peer electronic cash system**. www.bitcoin.org. Disponível em:www.bitcoin.org, Acesso em: 10 Ago de 2020.

[15] WIKIPEDIA CONTRIBUTORS. **Cryptocurrency — Wikipedia, the free encyclopedia**. https://en.wikipedia.org/w/index.php?title=Cryptocurrency&oldid=1024368321, 2021. [Acesso em 22-Ago-2020].

[16] NARAYANAN A., BONNEAU J., F. E. M. A. G. S.. **Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction**. Princeton University Press, USA, 2016.

[17] MISTRY, N.. **An introduction to bitcoin, elliptic curves and the mathematics of ecdsa**. www.bitcoin.org. Disponível em:https://raw.githubusercontent.com/bellaj/Bitcoin_Ethereum_docs/6bffb47afae6a2a70903a26d215484cf8ff03859/ecdsa_bitcoin.pdf, Acesso em: 30 Ago de 2020.

[18] EREMENKO K., P. H. L. T.. **Blockchain a-z™: Learn how to build your first blockchain**. Udemy. Disponível em:https://www.udemy.com/course/build-your-blockchain-az/.

[19] HEFEZ, A.. **Aritmética - Coleção PROFMAT**. Sociedade Brasileira de Matemática, Rio de janeiro, 2016.

[20] SILVERMAN, JOSEPH H. **The arithmetic of elliptic curves. Graduate Texts in Mathematics, 106**. Springer-Verlag, New York, 1992.

[21] STINSON, D. R.. **Criptography: theory and practice. 4rd edition**. Chapman Hall, Boca Raton-Florida, 1992.

[22] DEVELOPERS, D.. **Elliptic curve points**. Disponível em: `https://www.desmos.com/calculator/ialhd71we3?lang=pt-BR`, Acesso em: 07 Mar de 2021.

[23] MARTINEZ, F. B.; MOREIRA, C. G.; SALDANHA, N. ; TENGAN, E.. **Teoria dos números: um passeio com primos e outros números familiares pelo mundo inteiro**. IMPA, 2010.

[24] SCHOOF, R.. **Elliptic curves over finite fields and the computation of square roots mod p**. Mathematics of computation, vol 44 (170), p. 483-494, 1985.

[25] RESEARCH, C.. **Standards for efficient cryptography, SEC 1: Elliptic curve cryptography**, May 2009. Version 2.0.

[26] WIKIPEDIA CONTRIBUTORS. **Secg — Wikipedia, the free encyclopedia**. `https://en.wikipedia.org/w/index.php?title=SECG&oldid=889332866`, 2019. [Acesso em: 6 Mar 2021].

[27] RESEARCH, C.. **Standards for efficient cryptography, SEC 2: Recommended elliptic curve domain parameters**, January 2010. Version 2.0.

[28] R. P. GALLANT, R. J. L.; VANSTONE, S. A.. **Faster point multiplication on elliptic curves with efficient endomorphisms**. Advances in Cryptology — CRYPTO 2001.Lecture Notes in Computer Science 2139, pp. 190–200. International Association for Cryptologic Research, Springer, 2001.

[29] KROSS, B.. **How many atoms are in the human body?** Thomas Jefferson National Accelerator Facility - Office of Science Education. Disponível em: `https://education.jlab.org/qa/mathatom_04.html`, Acesso em: 07 Mar de 2021.

[30] WEISENBERGER, D.. **How many atoms are there in the world?** Thomas Jefferson National Accelerator Facility - Office of Science Education. Disponível em:`https://education.jlab.org/qa/mathatom_05.html`, Acesso em: 07 Mar de 2021.

[31] GEIGER, F.. **The bitcoin math 5: Transaction 1 – private key (2)**. Youtube. Disponível em:`https://www.youtube.com/watch?v=D9o1MjvAzA0&t=96s`, Acesso em: 10 Mar de 2021.

[32] MARSHAL, B.. **How does a bitcoin transaction actually work?** Disponível em: `https://medium.com/@blairlmarshall/`

how-does-a-bitcoin-transaction-actually-work-1c44818c3996, Acesso em: 20 Nov de 2020.