**UNIVERSIDADE FEDERAL DO PAMPA**

**João Otávio Massari Chervinski**

# Deanonimização de dados de transações da criptomoeda Monero

Alegrete
2018

**João Otávio Massari Chervinski**

# Deanonimização de dados de transações da criptomoeda Monero

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Diego Luis Kreutz

Coorientador: Prof. Dr. Jiangshan Yu

Alegrete
2018

**João Otávio Massari Chervinski**

# Deanonimização de dados de transações da criptomoeda Monero

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Projeto de Trabalho de Conclusão de Curso defendido e aprovado em ..... de ..... de .....
Banca examinadora:

---

**Prof. Me. Diego Luis Kreutz**
Orientador
UNIPAMPA

---

**Prof. Dr. Marcelo Caggiani Luizelli**
UNIPAMPA

---

**Prof. Dr. Marcelo Resende Thielo**
UNIPAMPA

## RESUMO

A adoção de criptomoedas como forma de pagamento vem crescendo expressivamente desde o lançamento da primeira moeda digital descentralizada, a Bitcoin. Diferentemente do dinheiro tradicional, criptomoedas utilizam apenas pseudônimos para a identificação dos usuários, permitindo que transações sejam realizadas sem que informações pessoais sejam reveladas. Mas, apesar da privacidade fornecida pelo uso de pseudônimos, alguns trabalhos mostram que através da análise dos dados gerados pelas transações é possível identificar os usuários envolvidos. Com a intenção de proteger a identidade dos usuários, foram criadas criptomoedas cujo foco é a privacidade das transações, mas até mesmo essas podem ser vulneráveis à ataques de deanonimização. O objetivo deste trabalho é realizar um estudo sobre os ataques de deanonimização de dados de transações da criptomoeda Monero, cujo foco é a privacidade dos usuários, e investigar o sistema buscando novas vulnerabilidades. São apresentados resultados obtidos através da implementação de ataques existentes na literatura. São destacadas as vulnerabilidades do sistema Monero e é mostrada a importância da detecção e correção de falhas de segurança.



**Palavras-chave**: Criptomoedas. Monero. Rastreabilidade. Privacidade.

**ABSTRACT**

The adoption of cryptocurrencies as a form of payment has been growing significantly since the launch of the first decentralized digital currency, Bitcoin. Unlike fiat money, cryptocurrencies require only the use of pseudonyms to identify users, allowing them to participate in transactions without the need to disclose any personal information. Despite providing privacy, the use of pseudonyms is not enough, as many studies show that it is possible to identify the users through the analysis of transaction data. Some cryptocurrencies were created with the intent of providing better privacy for users, but even those may be vulnerable to deanonymization attacks. This work conducts a study on deanonymization attacks on Monero, a privacy-centered cryptocurrency, aiming to discover new vulnerabilities. We implemented existing attacks and our findings indicate the existence of vulnerabilities in Monero's privacy mechanisms. The results shown emphasize the importance of detecting and fixing security issues.

**Key-words**: Cryptocurrencies. Monero. Traceability. Privacy.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE SIGLAS

**BTC** Unidade da moeda do sistema Bitcoin

**CSV** Comma-separated values

**ECDSA** Elliptic Curve Digital Signature Algorithm

**I2P** Invisible Internet Project

**IP** Internet Protocol

**JSON** Javascript Object Notation

**PoW** Proof-of-Work

**RingCT** Ring Confidential Transaction

**RIPEMD** RACE Integrity Primitives Evaluation Message Digest

**SHA** Secure Hashing Algorithm

**STXO** Spent Transaction Output

**TCC** Trabalho de Conclusão de Curso

**UTXO** Unspent Transaction Output

**XMR** Unidade da moeda do sistema Monero

SUMÁRIO

# 1 INTRODUÇÃO

Desde o lançamento da criptomoeda Bitcoin (NAKAMOTO, 2008), vêm ocorrendo uma revolução nos sistemas de pagamento em todo o mundo. Um dos fatores que impulsionam a adoção deste tipo de moeda é natureza descentralizada do sistema sobre o qual as criptomoedas operam, uma tecnologia chamada de *blockchain*. A rede Bitcoin funciona através de conexões ponto-a-ponto, onde os participantes trabalham de maneira colaborativa para validar transações, utilizando assinaturas criptográficas para garantir a segurança e a verificabilidade das mesmas, por isso o nome criptomoeda. Diferentemente do que ocorre com dinheiro tradicional, não existe nenhuma autoridade central que regula o uso da Bitcoin, ao invés disto, a moeda é mantida por colaboradores e tanto seu código-fonte quanto o histórico de todas as transações é aberto, ou seja, pode ser acessado por qualquer um. Devido a popularidade alcançada pela Bitcoin, houve o surgimento de outras criptomoedas, chamadas de *altcoins*, como: Ethereum (WOOD, 2014), Zcash (SASSON et al., 2014) e Dash(DUFFIELD; DIAZ, 2014).

A proposta de criação de moedas digitais, no entanto, não é recente. Como é destacado no primeiro trabalho que trata sobre esse assunto, a privacidade é uma característica desejável neste tipo de moeda (CHAUM, 1983). Porém, no que diz respeito à Bitcoin, há trabalhos que apresentam métodos capazes de identificar usuários da rede, seja através de seus endereços Internet Protocol (IP), nomes de usuário em fóruns e serviços na *Web* ou outros meios (BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014; MEIKLEJOHN et al., 2013; FLEDER; KESTER; PILLAI, 2015).

Com o objetivo de solucionar os problemas de privacidade enfrentados pelos usuários de criptomoedas, em Abril de 2014 foi lançada uma moeda chamada Monero. Monero é uma criptomoeda de código aberto, cujo foco é garantir a privacidade e a irrastreabilidade das transações. Contudo, mesmo buscando oferecer um nível de segurança superior ao Bitcoin, as versões originais da criptomoeda Monero apresentavam falhas no que diz respeito à manutenção da privacidade dos usuários, como mostram alguns trabalhos (KUMAR et al., 2017; MILLER et al., 2017). Isto demonstra que a tarefa de desenvolver uma criptomoeda que seja imune contra ataques à privacidade dos usuários é uma tarefa complexa. Os trabalhos que apresentaram ataques contra o sistema Monero também evidenciam a importância da exploração de vulnerabilidades em criptomoedas. Através da exploração é possível identificar falhas e propor novas estratégias de segurança.

Com o objetivo de investigar falhas de privacidade e contribuir para o avanço do estado da arte do anonimato no uso de criptomoedas, este trabalho apresenta um estudo sobre técnicas de deanonimização de usuários de criptomoedas, com foco no sistema Monero. O primeiro passo para a realização do trabalho é estudar os ataques existentes nos sistemas das criptomoedas Bitcoin e Monero, com o intuito de comparar os níveis de segurança oferecidos por cada uma das moedas. Observando a literatura existente, os métodos de investigação para a descoberta de falhas de privacidade em criptomoedas

como a Bitcoin e Monero são:

- Análise dos grafos de transações: O objetivo é gerar grafos que representam as transações entre os usuários do sistema, onde as arestas representam as transações e os nodos representam os usuários. Os caminhos entre os nodos são usados para rastrear a origem das criptomoedas e identificar relações entre usuários.
- Análise do tráfego de rede: Os dados disseminados pelos usuários na rede durante a criação das transações pode ser analisado, permitindo correlacionar endereços de rede dos usuários com as transações realizadas por eles no sistema.
- Análise das chaves privadas de visualização: No sistema Monero, todos os usuários possuem uma chave privada de visualização. Essa chave é utilizada para analisar as transações e revelar quais são destinadas para o usuário detentor da chave. Algumas plataformas e serviços revelam suas chaves para tornar transparentes as operações nas quais estão envolvidos, transmitindo confiança aos usuários. As chaves de visualização que são tornadas públicas podem ser utilizadas para a descoberta de relações entre vários usuários, potencializando o rastreamento de transações.
- Análise dos dados do *blockchain*: As informações geradas por transações e recompensas pela criação de blocos são armazenadas no *blockchain*. Heurísticas de análise de dados podem ser aplicadas à essas informações, revelando as entradas reais de transações.

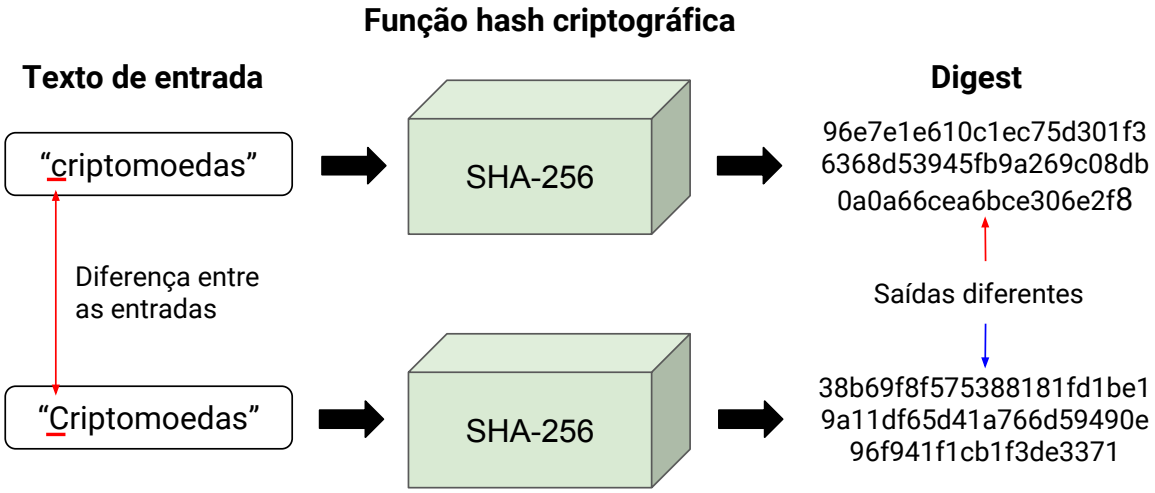
O restante deste documento está organizado da seguinte forma: No Capítulo 2 são apresentadas as tecnologias subjacentes aos sistemas de criptomoedas. O Capítulo 3 discute alguns trabalhos com propostas relacionadas ao tema deste trabalho. O Capítulo 4 apresenta resultados obtidos através da implementação de ataques existentes na literatura. Por fim, no Capítulo 5 é apresentado o cronograma de atividades a ser seguido durante o desenvolvimento do trabalho.

# 2 ESTADO DA ARTE

## 2.1 Tecnologia Blockchain

O *blockchain* é um registro distribuído formado por uma cadeia de blocos de dados, conectados uns aos outros por um sistema que utiliza funções *hash* criptográficas. A Figura 1 apresenta o resultado da aplicação da função *hash* criptográfica SHA-256 em duas entradas distintas. Como pode ser observado, a saída da função é completamente diferente para as duas entradas aparentemente idênticas.

Figura 1 – Aplicação de uma função *hash* criptográfica.




Funções *hash* tradicionais transformam dados de entrada de tamanho arbitrário em uma saída de tamanho fixo, chamada de *digest* ou *hash*. Funções *hash* criptográficas são um tipo especial de funções *hash* que devem possuir as seguintes propriedades:

- A aplicação da função sobre o mesmo dado deverá sempre retornar o mesmo resultado.
- Computar um *digest* para uma determinada entrada deve ser fácil.
- Descobrir quais dados foram utilizados como entrada da função analisando somente o *digest* deve ser muito difícil.
- Encontrar duas entradas que gerem o mesmo *digest* deve ser muito difícil.
- Uma pequena mudança na entrada deve alterar o *digest* resultante de tal forma que seja muito difícil encontrar alguma relação entre as saídas.

Este tipo de função é utilizado para ajudar a garantir a integridade de informações, já que uma pequena mudança nos dados de entrada altera o *digest* resultante, como pode ser observado na Figura 1. Ao aplicar uma função *hash* em um arquivo e enviar o *digest*

para a pessoa que irá recebê-lo, o recipiente poderá aplicar a função novamente e verificar se o resultado é igual ao *digest* recebido. Desta forma ela garante que o arquivo não foi modificado, desde que o *digest* tenha sido recebido de forma segura. Por exemplo, suponha que João trabalha no setor financeiro de uma empresa e foi encarregado de realizar uma transferência de dinheiro da conta da empresa para a conta de algumas empresas parceiras. João recebeu de Maria o arquivo contendo os dados das contas bancárias por email.

Um usuário malicioso, realizando um ataque de interceptação de dados, pode alterar o documento contido no email antes que ele seja recebido por João, adicionando novas contas bancárias na lista, por exemplo. Para certificar-se de que João receberá o arquivo com as mesmas informações enviadas originalmente, Maria computa o *digest* do documento anexado no email. Considere que a empresa utiliza um canal seguro para comunicação auxiliar, para o envio de dados como os *digests*. Maria envia o *digest* do arquivo para João utilizando a rede privada segura. Ao fazer o *download* do arquivo com os dados das contas, João precisa certificar-se de que nada foi modificado. Para isto, ele computa o *digest* do arquivo recebido e compara-o com o *digest* enviado por Maria. Se eles forem iguais, o documento não foi modificado.

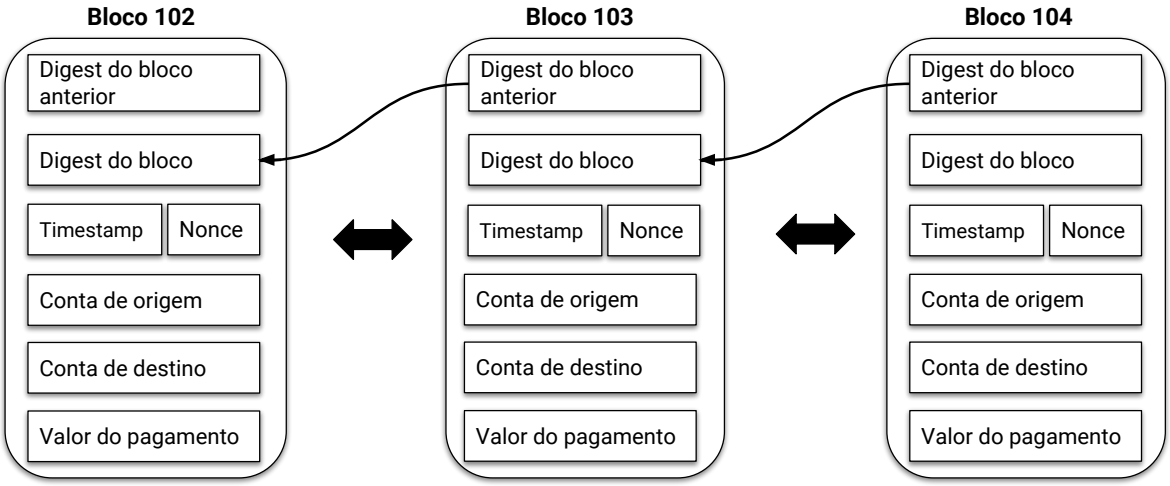
Os *blockchains* possuem uma propriedade interessante para o registro de transações: quando um bloco é adicionado ao final da cadeia, torna-se uma tarefa muito difícil alterá-lo. Suponha a existência de um *blockchain* para uma aplicação bancária fictícia. Cada bloco da aplicação armazena a informação de uma transação entre duas contas. Os dados armazenados em cada bloco são:

- *Digest* do bloco anterior: Resultado da aplicação de uma função *hash* criptográfica sobre os dados do bloco anterior.
- *Digest* do novo bloco: Resultado da aplicação de uma função *hash* criptográfica sobre os dados do novo bloco.
- *Timestamp*: Representação da data e hora de criação do bloco.
- *Nonce*: Número auxiliar utilizado para realização do cálculo da função *hash* criptográfica.
- Conta de origem: A conta de onde o dinheiro será retirado.
- Conta de destino: A conta que receberá o dinheiro.
- Valor da transação: Quantia de dinheiro a ser transferida.

Considerando a estrutura de bloco apresentada, existem duas informações que garantem que a ordem dos blocos irá manter-se correta, o *digest* do bloco anterior na cadeia e o *digest* do próprio bloco. A Figura 2 ilustra a conexão entre uma cadeia de blocos. A utilização de uma função *hash* criptográfica garante que cada bloco irá gerar

um *digest* único, permitindo estabelecer um vínculo forte e único entre os blocos da cadeia. Isso garante que só existirá uma única sequência válida de blocos.

Figura 2 – Representação dos blocos em um *blockchain*.




Imagine que um atacante deseja modificar o conteúdo de um bloco para fins maliciosos, como direcionar o valor de uma transação a si mesmo. Para isso, será necessário que os novos dados do bloco gerem um *digest* igual ao anterior, caso contrário, a ligação entre a cadeia de blocos será desfeita. Se a cadeia de blocos for desfeita, o *blockchain* ficará em um estado inconsistente. Como as funções *hash* criptográficas garantem que é muito díficil gerar o mesmo *digest* a partir de dados de entrada diferentes, seria mais fácil alterar o campo que contém o *digest* que aponta para o bloco anterior na cadeia em cada um dos blocos seguintes. Bastaria que o atacante mudasse os campos nos blocos posteriores, calculasse seus *digests* e repetisse o processo até o final do *blockchain*. Um sistema utilizado para controlar a criação de novos blocos em alguns *blockchains*, chamamado de Proof-of-Work (PoW), ajuda a evitar este ataque. Através deste sistema, não basta apenas calcular o *digest* do bloco para que ele seja adicionado à cadeia. O resultado da função *hash* deve obedecer à uma restrição que requer um grande esforço computacional para ser atendida. Um exemplo de restrição é a geração de um *digest* cujos primeiros dez *bits* sejam iguais a zero. Para variar a saída da função *hash* é necessário alterar os dados de entrada, para esta finalidade há em cada bloco um campo chamado de *nonce*.

Um usuário que deseja criar um *digest* válido altera os dados do campo *nonce* até que a execução da função *hash* gere o resultado desejado. Como não é possível prever o resultado da aplicação de uma função *hash* criptográfica, para cumprir o desafio, quem deseja criar um bloco válido deve tentar valores diferentes no campo *nonce* até que o *digest* resultante do bloco atenda às exigências do sistema. Após descoberto o *nonce* que torna o bloco válido, calcular novamente o *digest* do bloco torna-se trivial. Dois blocos diferentes podem possuir o mesmo dado no campo *nonce*, porém, os dados de transações

não podem ser iguais. Isso faz com que *digests* iguais não possam ser gerados a partir de blocos diferentes. A descentralização do *blockchain* também dificulta a execução do ataque de modificação de blocos, pois alterações na cadeia de blocos implica em mudar todas as outras cópias do *blockchain*, armazenadas por outros usuários. Para que um atacante possa controlar a criação de novos blocos, ele deve possuir mais de 50% do poder computacional de toda a rede. Somente assim seria possível criar blocos válidos com uma velocidade maior do que o resto de toda a rede. Além do PoW, foram propostos outros esquemas de criação de blocos, como o *Proof-of-Stake*, o *Proof-of-Activity* e o *Proof-of-Publication* (TSCHORSCH; SCHEUERMANN, 2016).

A tecnologia dos *blockchains* possui potencial para substituir plataformas digitais, não somente na área de finanças, mas também na cadeia de suprimentos, no setor de energia e no setor de agricultura(KSHETRI, 2018).

## 2.2 Criptomoedas

A adoção de criptomoedas como forma de pagamento vem crescendo rapidamente devido aos benefícios que elas oferecem em relação às formas de pagamento tradicionais (MEDFAR87, 2018). 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 implica em menores taxas de transações, maior controle, e mais privacidade. Mas, apesar dos benefícios oferecidos por esse tipo de moeda, existem desvantagens como a impossibilidade de reverter transações e de obter suporte caso ocorram erros no sistema. A volatilidade dos preços da moedas é outro fator negativo. No caso da Bitcoin, o preço pode variar mais de 10% em poucas horas (ADKISSON, 2018).

### 2.2.1 Mineração

Para obter criptomoedas, um usuário pode realizar uma compra através de serviços especializados em vendas de criptomoedas, chamados de *cryptocurrency exchanges*. Algumas moedas, como a Bitcoin e a Monero, oferecem aos usuários a possibilidade de obtê-las diretamente através do sistema, participando de um processo comumente chamado de mineração. É importante notar que criptomoedas como a Ripple (SCHWARTZ et al., 2014) e a IOTA (POPOV, 2014) não possuem um sistema através do qual os usuários possam obter moedas ao criar blocos de transações, ou seja, não podem ser mineradas. Esta seção discute o processo de mineração que ocorre em criptomoedas similares à Bitcoin e Monero.

O processo de mineração consiste em participar da criação de blocos válidos através do esquema de PoW ou de outro similar. Esse processo é essencial para garantir o funcionamento do *blockchain* e da moeda, pois é através dele que as transações são confirmadas e adicionadas ao final da cadeia de blocos.

Devido ao esforço computacional necessário para a criação de blocos válidos pelo esquema de PoW, é necessário um incentivo para que os participantes da rede, ou mineradores, emprestem o seu poder de processamento para ajudar no funcionamento do sistema. Esse incentivo é dado através de uma recompensa em criptomoedas para o participante da rede que conseguir validar primeiro um determinado bloco de transações.

A primeira transação de cada bloco, chamada de *coinbase transaction*, é de um tipo especial que não possui entradas e cuja finalidade é enviar o valor da recompensa para o usuário que efetuou a validação do bloco. O sistema de recompensas é geralmente desenvolvido de maneira que, com o passar do tempo, a recompensa por bloco diminua. Isto é necessário para controlar a quantidade de novas moedas criadas devido ao aumento no preço da moeda e outros fatores econômicos. Na Bitcoin, essa diminuição ocorre a cada 210,000 blocos minerados, quando a recompensa é reduzida pela metade. Em um dado momento, a validação de novos blocos não irá gerar mais recompensas e o pagamento pela criação de blocos válidos será feito somente através das taxas de transações, pagas pelos usuários. Atualmente, em 2018, a quantia recompensada por bloco na Bitcoin é de 12,5 unidades da moeda, chamada de BTC.

A Tabela 1 mostra a mudança na recompensa por bloco válido criado ao longo do tempo. O período estimado para que ocorra a criação de 210,000 novos blocos e ocorra uma diminuição no valor da recompensa por bloco é de 4 anos. Na prática este período pode ser maior ou menor, apesar do aumento no número de mineradores ao longo do tempo. Esta oscilação ocorre devido ao fato de o sistema elevar automaticamente a dificuldade de criação de novos blocos quando a velocidade da rede aumenta. Esta estratégia é empregada pelo sistema para manter o tempo entre a criação de novos blocos por volta de dez minutos. Isto também evita a emissão de muitas moedas novas em um curto período de tempo. Quando a dificuldade de criar blocos aumenta, o retorno pela criação de blocos diminui dado o esforço necessário, causando uma redução no número de mineradores. A redução no número de mineradores diminui a capacidade de criação de blocos da rede, que ajusta a dificuldade de acordo. Essas variações causam mudanças no intervalo de tempo entre a diminuição da recompensa por bloco.

Tabela 1 – Recompensa pela validação de blocos na Bitcoin.

| **Número de Blocos** | **Recompensa por bloco** | **Ano** |
|---|---|---|
| 0 | 50 BTC | 2009 |
| 210,000 | 25 BTC | 2012 |
| 420,000 | 12,5 BTC | 2016 |
| 630,000 | 6,25 BTC | 2020 (estimado) |

A recompensa por bloco no sistema Monero é regulada continuamente de acordo com a Equação 2.1, que considera a quantidade de moedas já emitidas até o momento

da validação. Os autores do protocolo chamam esse processo de emissão de moedas de *smooth emission* (SABERHAGEN, 2013):

$$Recompensa = (M - A) * 2^{-20} * 10^{-12} \tag{2.1}$$

onde:
$M$ = Suprimento máximo de Monero, $2^{64} - 1$ unidades atômicas da moeda.
$A$ = Moedas emitidas até o momento presente.

É comum que mineradores organizem-se em grupos que trabalham em conjunto para criarem blocos válidos, chamados de *mining pools*. Quando um usuário cria um bloco válido, a recompensa é dividida entre todos os participantes do grupo. Este método diminui o valor da recompensa individual de cada minerador, porém garante um fluxo mais estável de renda para todos. A participação em *mining pools* é vantajosa pois é difícil computar sozinho um *hash digest* válido para a criação de um bloco.

### 2.2.2 Bitcoin

Introduzida através de um *white paper* em uma lista de correio eletrônico sobre criptografia em 2008 e lançada em 2009, Bitcoin foi a primeira criptomoeda. Anos após seu lançamento, ainda permanece sendo a mais utilizada, possuindo um valor total de mercado de US$ 158 bilhões[1]. A Bitcoin baseia-se em material de pesquisas anteriores, como o esquema de PoW que controla a criação de blocos válidos no *blockchain*, esquemas de assinaturas digitais que garantem que os usuários que utilizam moedas realmente as possuem e técnicas de *timestamping* que marcam a data e a hora da realização das operações. A principal contribuição da Bitcoin foi eliminar a necessidade de uma autoridade central que regule a emissão de moedas e a confirmação de transações. Isto foi possível pela forma descentralizada pela qual o sistema funciona, utilizando uma rede ponto-a-ponto onde usuários participam do processo de validação e verificação da autenticidade das transações. A descentralização também é fruto da maneira como os dados estão armazenados. Todas as transações ocorridas estão armazenadas em um registro chamado de *blockchain* e cada partipante da rede possui uma cópia desse registro.

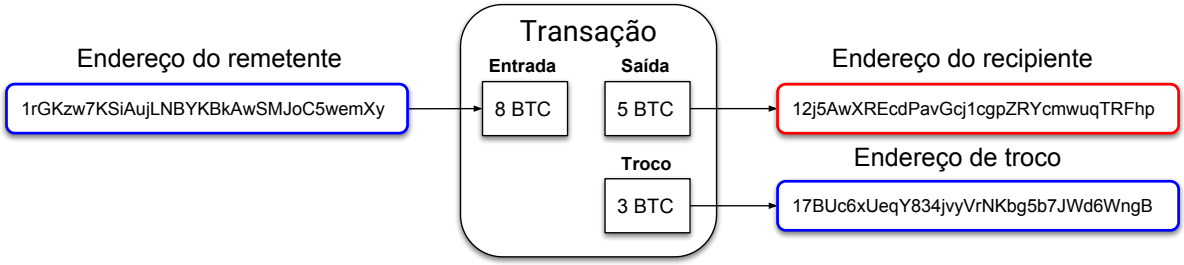
Para enviar e receber transações em Bitcoin, um usuário necessita de um par de chaves criptográficas composto por uma chave pública e uma chave privada. No caso da Bitcoin, a chave pública é utilizada como endereço de envio e recebimento de pagamentos. Um usuário pode divulgar a sua chave pública e outras pessoas poderão enviar Bitcoins para esse endereço. A chave privada é utilizada para comprovar que um usuário é dono da chave pública que a acompanha, podendo assim utilizar os fundos recebidos. A chave privada não deve ser compartilhada e deve ser armazenada em segurança para que ninguém

[1] <https://coinmarketcap.com/>

além do proprietário possa acessá-la. O endereço de um usuário é derivado de sua chave pública, ao aplicar o algoritmo de *hashing* SHA-256 e depois o algoritmo RIPEMD-160, adicionar números para controle de erro e controle de versões e por fim codificá-lo em BASE58 (TSCHORSCH; SCHEUERMANN, 2016). O processo de derivação da chave é realizado para fornecer segurança adicional, ajudando a ocultar a verdadeira chave pública. Os usuários também podem optar pela utilização de sua chave pública original como endereço.

Em uma transação, existem endereços de entrada e endereços de saída. A Figura 3 ilustra o procedimento de uma transação de Bitcoins contendo uma única entrada.

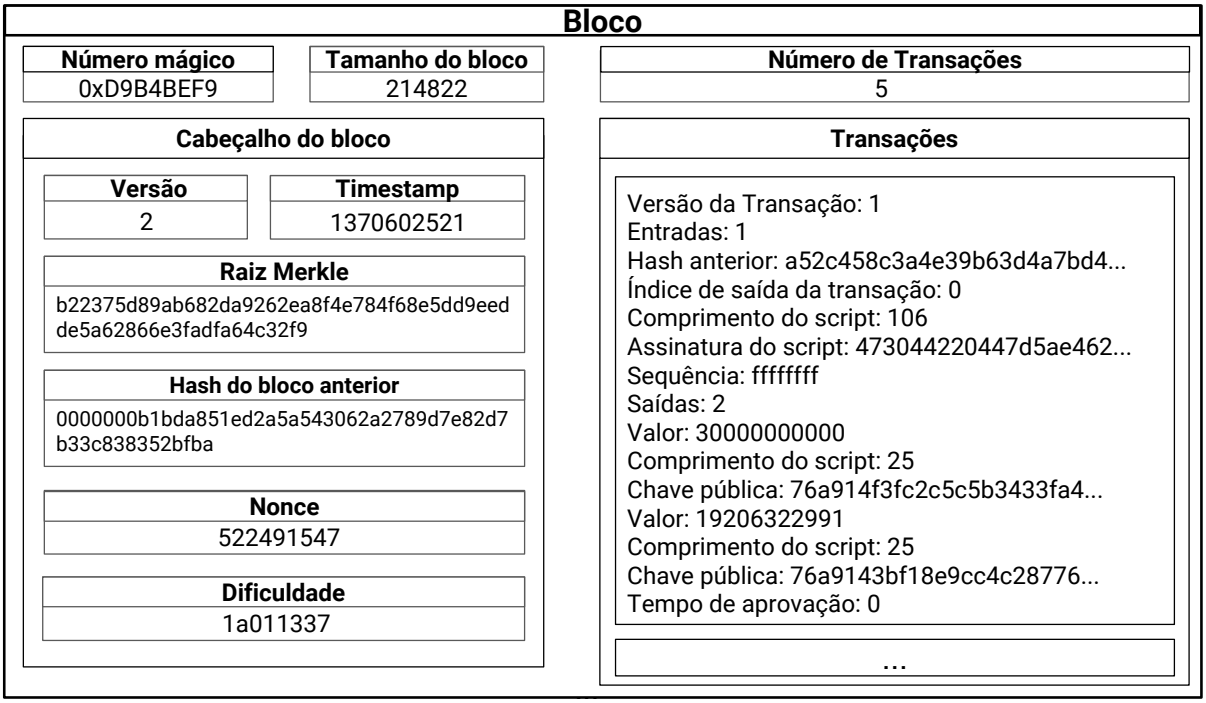
Figura 3 – Transação de Bitcoins contendo uma única entrada.




A Figura 4 mostra como são estruturados os dados de transações dentro de um bloco no *blockchain* do sistema Bitcoin. Endereços de entrada são chaves que foram geradas como saída em uma transação anterior. Endereços de saída são os endereços do usuários que receberão as criptomoedas. Quando um valor é recebido, o usuário obtém uma saída de transação. O saldo de um usuário da Bitcoin consiste na soma dos valores de todas as saídas de transações que ele já recebeu e ainda não utilizou. Antes de serem utilizadas, as chaves que contêm criptomoedas recebem a denominação de "saída de transação não-utilizada", do inglês Unspent Transaction Output (UTXO). Sempre que uma chave de entrada é utilizada em uma transação, todo o seu conteúdo em Bitcoins deve ser gasto, ou seja, não é possível usar somente parte da quantia armazenada em um endereço. Após uma UTXO ser utilizada, seu estado muda para "chave de saída utilizada", do inglês Spent Transaction Output (STXO), para indicar que a quantia armazenada nesta chave já foi gasta. Para permitir que os remetentes mantenham o dinheiro que sobra do pagamento, existe a idéia de troco, onde o pagamento é feito para o recipiente e o restante é enviado para um endereço de escolha do remetente. O endereço de troco pode ser o mesmo endereço utilizado como entrada, mas isso é desencorajado porque quanto mais um endereço é utilizado, mais fácil torna-se o processo de rastrear informações do usuário. É recomendado que os usuários utilizem uma carteira de Bitcoins, um tipo de programa que auxilia no gerenciamento das chaves e endereços ao criar automaticante novos endereços para o recebimento de troco.

Um bloco é capaz de armazenar milhares de transações. A Figura 4 mostra os dados contidos um bloco do sistema Bitcoin.

Figura 4 – Estrutura de um bloco do *blockchain* do sistema Bitcoin.




Os blocos criados para armazenar as transações no *blockchain* possuem a seguinte estrutura:

- Número mágico: Valor utilizado para identificar o tipo de estrutura contida nos dados. Neste caso, um bloco. Este valor é específico do protocolo Bitcoin.
- Tamanho do bloco: Especifica o tamanho em *bytes* dos dados contidos no bloco.
- Cabeçalho do bloco: Contém dados que identificam o bloco atual.
  - Versão: Especifica a versão do sistema no momento da criação do bloco.
  - *Timestamp*: Representa o momento no tempo em que o bloco foi criado.
  - Raiz Merkle: Um tipo de *hash* utilizado para verificar a validade das transações contidas nos blocos sem a necessidade de verificar todas as informações das transações. Para realizar a verificação é necessário o cabeçalho dos blocos e uma estrutura chamada de Árvore de Merkle.
  - *Nonce*: Campo cujo valor deve ser modificado até que o *digest* resultante do bloco atenda as exigências do sistema.
  - Dificuldade: Especifica o número de *bits* 0 necessários à esquerda do *digest* do bloco para que ele atenda às exigências do sistema.

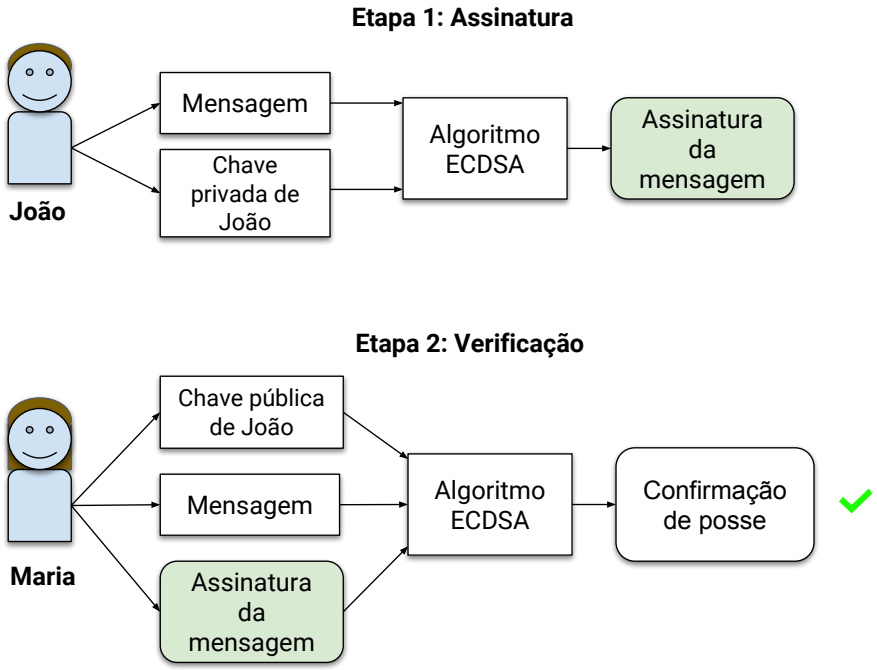
- Número de transações: Contém o número de transações presentes no bloco atual.
- Transações: Contém os dados de cada uma das transações contidas no bloco.

Como o *blockchain* da Bitcoin é um registro transparente, o histórico de transações de todos os usuários está disponível abertamente. Através da análise dos endereços e do fluxo de transações é possível efetuar ataques que correlacionam os pseudônimos com as identidades dos usuários (MEIKLEJOHN et al., 2013). Para reduzir o impacto de ataques que efetuam a análise das transações é sugerida a utilização de uma nova chave e endereço para cada transação (NAKAMOTO, 2008).

A quantidade de Bitcoins associada a cada usuário não é armazenada explicitamente nos registros. O saldo de um endereço pode ser verificado ao checar todo o seu histórico de transações, calculando quantas Bitcoins foram recebidas e quantas foram enviadas a partir do endereço. Por isso, sempre que um usuário instala pela primeira vez uma carteira de Bitcoins em seu sistema, é necessário que ele verifique todas as transações já ocorridas. A verificação é realizada para garantir que os usuários que estão efetuando pagamentos possuem de fato os valores sendo gastos.

Para garantir a segurança das transações na Bitcoin, um sistema de assinaturas baseado no algoritmo ECDSA é utilizado. A Figura 5 ilustra o processo de criação de uma assinatura digital.

Figura 5 – Processo de assinatura de uma transação.




João precisa provar que possui um endereço para enviar dinheiro através dele. Para isso, ele cria uma mensagem contendo os dados transação que deseja realizar. O segundo

passo é criar uma assinatura digital, que servirá para provar que o João é o dono do endereço do qual as moedas estão sendo enviadas. Utilizando a sua chave privada, João gera a assinatura digital da mensagem e a envia para a rede juntamente com a mensagem e sua chave pública. Para verificar a validade da transação, Maria realiza uma operação matemática utilizando a chave pública e assinatura enviadas junto com a mensagem. O resultado da operação irá confirmar se a chave pública recebida corresponde a chave privada utilizada na geração da assinatura digital, sem revelar qualquer informação sobre a chave privada. Maria saberá que a chave pública enviada junto com a mensagem, que é o mesmo endereço de onde estão sendo enviadas moedas, pertence a pessoa que tem chave privada correspondente. João é então identificado como o dono do endereço.

Após a criação de uma transação, o remetente dissemina na rede uma mensagem avisando que possui uma nova transação. Os participantes interessados nos dados enviam um pedido explicito ao remetente. Os mineradores acumulam transações e organizam-as em blocos antes de iniciarem o processo de tentativa de criação de um bloco válido. O sistema ajusta a dificuldade de criação de um bloco válido a cada 2.016 blocos criados. O ajuste é realizado com a intenção de manter o tempo necessário para adicionar um bloco em cerca de dez minutos, para que o tempo de confirmação das transações seja razoável. A mudança na dificuldade ocorre de acordo com a seguinte equação:

$$D = D_{anterior} \times \frac{T_{atual}}{2016 \times 10} \tag{2.2}$$

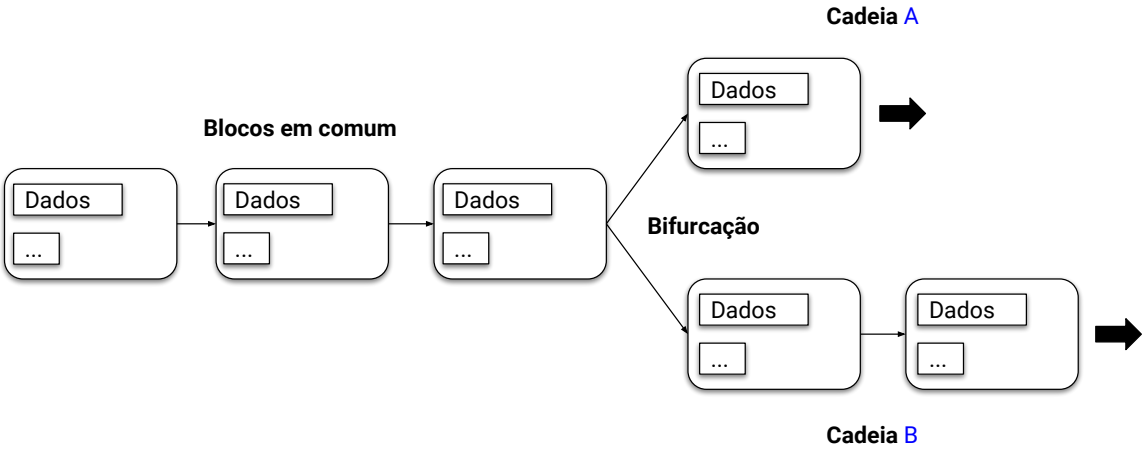
onde:
$D$ = Dificuldade da resolução do problema de criação de um bloco válido.
$T$ = Tempo ocorrido desde a última mudança de dificuldade.

Ao efetuar a criação de um bloco válido, o minerador dissemina a informação do bloco para que os outros participantes saibam que as transações contidas naquele bloco já foram confirmadas. Uma aplicação de carteira Bitcoin realiza uma varredura na cadeia de blocos para verificar as transações destinadas ao seu endereço público e saber quantas moedas o usuário possui. Devido à transparência das informações contidas no *blockchain*, qualquer um com acesso aos dados é capaz de descobrir o saldo de um endereço qualquer, diminuindo a privacidade dos usuários.

Existe um problema que gera uma bifurcação no *blockchain*, causado pelo tempo necessário à propagação dos blocos na rede. A Figura 6 ilustra o estado da cadeia de blocos quando ocorre uma bifurcação. O problema ocorre quando um usuário $A$ cria um um bloco válido ao mesmo tempo em que um usuário $B$ cria outro bloco válido contendo transações diferentes. Como é necessária uma quantia de tempo para que a informação seja propagada aos outros participantes da rede, os que recebem primeiro o bloco de $A$, o colocam no fim de suas cadeias, já os que recebem primeiro o bloco de $B$, colocam um bloco diferente no final de suas cadeias. Isso causa uma divergência momentânea no

*blockchain*. A partir desse momento existem duas cadeias cuja única diferença é o ultimo bloco. Não é possível saber qual das cadeias é a correta.

Figura 6 – Bifurcação na cadeia de blocos.




Para resolver o problema da bifurcação na cadeia de blocos, a Bitcoin emprega uma estratégia chamada de "a regra da cadeia mais longa". Com o passar do tempo, naturalmente outros mineradores irão adicionar novos blocos ao final de suas próprias cópias do *blockchain* e irão propagá-los na rede. O sistema irá selecionar a cadeia com o maior número de blocos e a tornará definitiva. As cadeias restantes serão descartadas e as transações contidas em seus blocos voltarão para o conjunto de transações que estão aguardando para serem confirmadas. Devido à esse tipo de ocorrência, é recomendado que os usuários aguardem até que pelo menos seis blocos sejam adicionados após o bloco onde sua própria transação foi validada. Isso ajuda a garantir que a transação não será desfeita.

Apesar de apresentar alguns problemas de privacidade e de possuir uma capacidade limitada de processar transações devido ao esquema de PoW, a Bitcoin continua sendo amplamente utilizada. Seu sucesso contribui para a criação de novas criptomoedas que almejam solucionar problemas existentes nos sistemas atuais, como a demora das confirmações e a rastreabilidade das transações.

### 2.2.3 Privacidade

O surgimento de trabalhos que relatam os problemas de privacidade na Bitcoin motivaram o desenvolvimento de novas criptomoedas com foco na segurança e privacidade dos usuários. A necessidade de privacidade nos sistemas de criptomoedas, porém, é motivo de polêmica, porque além de compras comuns tais quais as que podem ser feitas através de dinheiro tradicional, criptomoedas são utilizadas para a compra e venda de produtos ilegais, como armas de uso restrito e drogas (TORPEY, 2018).

Apesar de algumas atividades ilegais serem motivadas pela existência de criptomoedas, a privacidade continua sendo um direito dos usuários. Criptomoedas que buscam oferecer garantias de privacidade têm recebido mais atenção nos últimos anos (WILLIAMS, 2017).

Mesmo através do uso de pseudônimos como as chaves criptográficas, ainda existe a possibilidade de vincular diferentes endereços à um mesmo usuário (NAKAMOTO, 2008). Quando são efetuadas transações com múltiplas entradas, por exemplo, várias chaves públicas são utilizadas e têm seus valores somados para realizar o pagamento. Como todas as chaves devem ser adicionadas pelo usuário que cria uma transação, é possível assumir que quem efetuou o pagamento possui todas as chaves utilizadas. Dessa forma é possível vincular várias chaves ao usuário que efetua a transação. A divulgação das chaves públicas dos usuários contribui para vinculação de pseudônimos com as suas identidades reais. Esta associação pode ser feita ao associar nomes de contas e informações dos usuários que divulgam as chaves em plataformas online com as próprias chaves divulgadas.

Usuários legítimos podem se beneficiar da privacidade. Por exemplo, se a privacidade das transações não for protegida, empresas podem analisar os dados com o objetivo de predizer hábitos e gostos dos usuários. Através dos dados privados de cada usuário as empresas podem exibir propagandas e ofertas dirigidas de produtos, bem como comercializar os dados de perfil do usuário para outras empresas. Como os dados sobre a utilização de serviços vêm se tornando cada vez mais valiosos, as informações geradas por cada usuário deveriam ser de sua posse somente, cabendo a cada um autorizar ou não a divulgação ou comercialização dos seus dados. Em um cenário ideal, os próprios usuários poderiam realizar a comercialização dos seus dados.

Para garantir a privacidade dos usuários é necessário adotar técnicas que impeçam a identificação dos usuários a partir dos dados públicos do *blockchain*.

### 2.2.4 Monero

A Monero foi lançada em Abril de 2014 e é uma criptomoeda descentralizada. Seu código é aberto e seu foco é a privacidade dos usuários. Monero ganhou popularidade devido às suas características que fornecem um nível de privacidade mais elevado do que as chaves pseudoanônimas da Bitcoin e de outros sistemas (KUMAR et al., 2017). A atenção atraída pela criptomoeda fez com que ela subisse para a 12º posição em termos de valor de mercado se comparada a todas as criptomoedas[2].

Monero utiliza um protocolo chamado de CryptoNote (SABERHAGEN, 2013). Esse protocolo é utilizado em outras criptomoedas que focam na privacidade dos usuários, tais como Bytecoin e DashCoin. O protocolo oferece funcionalidades que são essenciais para a garantia da privacidade no uso de Monero, pois as transações do CryptoNote não podem ser rastreadas através da análise do *blockchain*. Porém, quando o protocolo é

[2] <https://coinmarketcap.com/>

utilizado sem precauções, são criadas brechas que permitem ataques à privacidade das transações, como é mostrado em detalhes na seção 4.2. As duas principais características de privacidade oferecidas pela Monero são:

- **Irrastreabilidade das transações**: Garante que dada uma transação com várias entradas, não é possível descobrir qual entrada foi utilizada, impedindo que seu histórico seja traçado.
- **Não-vinculação de endereços**: Garante que dadas transações diferentes, não é possível provar que elas foram originadas de um mesmo usuário.

Essas propriedades são baseadas nas funcionalidades de chaves de uso único e *ring signatures*, oferecidas pelo protocolo CryptoNote. O sistema Monero oferece outras garantias de segurança e privacidade: as Ring Confidential Transactions (RingCTs), e o protocolo de roteamento Kovri, descritos a seguir.

No lado do recipiente das transações, a identidade do usuário é protegida através da utilização dos endereços de uso único, chamadas na Monero de *stealth addresses*. Cada usuário possui dois pares de chaves pública e privada, um par de chaves de visualização e um par de chaves de utilização de fundos. Sempre que uma quantia em XMR, a unidade da moeda do sistema Monero, é enviada para um recipiente, é criado um endereço de uso único que permite que somente o recipiente saiba que recebeu um pagamento. Suponha que o usuário João deseja enviar uma quantia em XMR para a usuária Maria. João usa as duas chaves públicas de Maria e um número aleatório para gerar um endereço de uso único para a qual o pagamento será enviado. Maria realiza uma varredura no *blockchain* usando sua chave privada de visualização para identificar o endereço de uso único destinado a si. Isso permite ao recipiente identificar transações e evita que qualquer outra pessoa com acesso aos dados do *blockchain* consiga identificar quem recebe a saída de uma transação. Com a sua sua chave privada de utilização de fundos, Maria consegue provar que é a dona daquela chave de saída.

As *ring signatures* são um tipo de assinatura digital onde um grupo de possíveis remetentes são utilizados em conjunto para criar uma assinatura digital que autoriza a transação. Ao utilizar criptografia para esconder os dados da transação o remetente original mantem-se anônimo. A assinatura digital é composta pelo verdadeiro remetente juntamente com outros remetentes válidos. O verdadeiro remetente usa uma chave de uso único para efetuar o pagamento e as chaves restantes são retiradas de transações anteriores contidas no *blockchain* e são chamadas de *mixins* ou misturas. Todas essas chaves compõem as entradas de uma transação, tornando difícil para um observador deduzir qual entrada é a verdadeira.

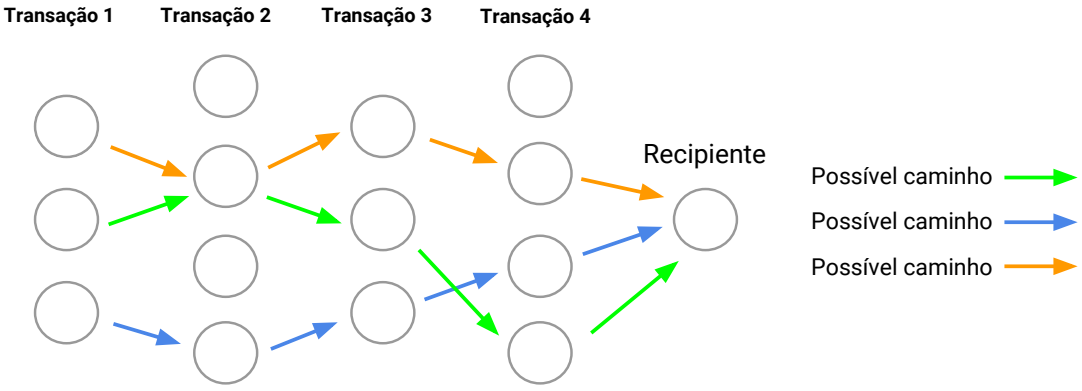
Os usuários podem escolher o número de *mixins* utilizados em uma transação. Em versões antigas do sistema, o usuário podia optar por não incluir nenhuma mixin, tornando a chave de entrada visível. Apesar da liberdade de escolha, não é recomendada

a utilização de um número muito elevado de *mixins*, pois isso faz com que a transação se destaque entre as outras presentes no *blockchain*. Utilizar um número muito elevado de *mixins* também gera custos adicionais, pois os usuários devem pagar uma taxa para realizar as transações de acordo com o seu tamanho em *bytes*.

Para evitar que a mesma chave seja usada duas vezes para realizar pagamentos, uma vez que os outros participantes da rede não são capazes de deduzir se ela já foi utilizada, existem as imagens de chaves. Imagens de chaves são chaves criptográficas únicas que são derivadas da chave real sendo utilizada na transação. Com a imagem da chave, outros participantes da rede são capazes de verificar que chave sendo utilizada não foi gasta em uma transação anterior, mas não são capazes de identificar qual das chaves inclusas na transação é a verdadeira.

A utilização das *ring signatures* faz com que as transações não sejam transparentes, dificultando a identificação de suas origens e o rastreamento dos seus históricos. Devido a quantidade de possíveis caminhos, a análise do grafo de transações se torna inviável. A Figura 7 ilustra uma tentativa de análise das transações do sistema Monero. Cada coluna de círculos representa uma transação do sistema e as chaves de entrada são representadas pelos círculos em cada coluna. A chave utilizada na transação pode ser qualquer uma das entradas, portanto, todas as combinações de caminhos da coluna inicial até o recipiente são válidas. As setas coloridas mostram três caminhos válidos no grafo de transações. Como o *blockchain* armazena milhões de transações, um atacante precisará analisar os milhares de caminhos possíveis para tentar deduzir a entrada correta de uma transação.

Figura 7 – Análise das transações do sistema Monero.

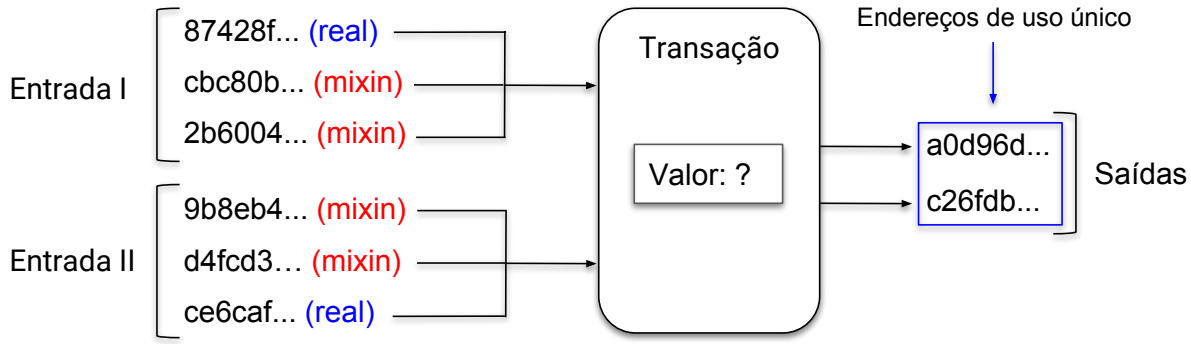



Adaptado: de CryptoNote (2015)

Para aumentar ainda mais o nível de privacidade nas transações de Monero, foi criado o protocolo RingCT. A Figura 8 apresenta uma transação utilizando o protocolo RingCT, onde o valor transacionado não é visível. Antes da criação deste protocolo, os valores das transações eram visíveis no *blockchain* e precisavam ser divididos em várias

partes, chamadas de denominações. A Tabela 2 mostra os valores de denominações disponíveis. A divisão dos valores era necessária porque chaves usadas como *mixins* em uma transação precisavam possuir o mesmo valor que a chave real sendo utilizada. Devido ao grande número de valores diferentes, algumas transações não encontravam *mixins* suficientes para que alcançassem um bom nível de segurança, o que levou a várias transações sem nenhum *mixin*. Isto gerou um problema de segurança que permite a identificação da chave real sendo utilizada nas transações. O valor da transação era constituído de diferentes tamanhos de denominações, até atingir o valor correto. Por exemplo, se um usuário precisasse enviar 16,5 XMR, a transação iria conter uma entrada de 10 XMR, seis entradas de 1 XMR e 5 entradas de 0,1 XMR, resultando no valor desejado. As RingCTs escondem o valor das transações, fazendo com que os valores de todas entradas de uma transação apareçam como 0 XMR. Uma transação pode escolher qualquer outra saída de uma transação que utilize RingCT para utilizar como *mixin*, independentemente do valor transacionado. Saídas de transações que não usam o novo protocolo não podem ser misturadas com saídas do RingCT para a realização de pagamentos.

Figura 8 – Representação de transação RingCT com duas entradas.




Apesar das garantias de privacidade oferecidas a nível de transações no sistema Monero, ainda existem maneiras pelas quais um atacante pode obter dados sobre os usuários do sistema. Uma delas é a coleta dos dados enviados pela rede durante a realização de uma transação, que pode contribuir para a identificação dos usuários (BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014).

Para a resolução desse problema, foi desenvolvida a tecnologia Kovri baseada nas especificações do Invisible Internet Project (I2P). Ao utilizar técnicas de roteamento e de criptografia, o protocolo Kovri estabelece uma rede sobreposta privada, permitindo que os usuários escondam suas informações geográficas e seu endereço IP. Kovri faz um tunelamento do tráfego através da rede I2P utilizando um processo chamado de *garlic routing*. As mensagens trafegam através de uma rede privada em mensagens que são

Tabela 2 – Denominações usadas nas antigas transações do sistema Monero.

| **Nome** | **Base 10** | **Quantidade(XMR)** |
|---|---|---|
| piconero | $10^{-12}$ | 0,000000000001 |
| nanonero | $10^{-9}$ | 0,000000001 |
| micronero | $10^{-6}$ | 0,000001 |
| millinero | $10^{-3}$ | 0,001 |
| centinero | $10^{-2}$ | 0,01 |
| decinero | $10^{-1}$ | 0,1 |
| monero | $10^{0}$ | 1 |
| decanero | $10^{1}$ | 10 |
| hectonero | $10^{2}$ | 100 |
| kilonero | $10^{3}$ | 1,000 |
| meganero | $10^{6}$ | 1,000,000 |

criptografadas em camadas, e as únicas informações visíveis são as instruções de envio das mensagens para outros participantes.

O sistema Monero oferece várias funcionalidades que fornecem segurança e privacidade para os usuários. É necessário que as técnicas empregadas sejam analisadas em profundidade, a fim de identificar possíveis falhas e contribuir com o desenvolvimento de criptomoedas privadas e descentralizadas.

# 3 TRABALHOS RELACIONADOS

Trabalhos presentes na literatura propõem técnicas de ataques à criptomoedas com o intuito de prejudicar a rede ou revelar informações sobre as entidades envolvidas nas transações. Alguns trabalhos baseiam-se na interceptação de tráfego de rede (APOSTOLAKI; ZOHAR; VANBEVER, 2017; BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014), outros realizam análises baseando-se somente nos dados contidos no histórico de transações do *blockchain* (KUMAR et al., 2017; MILLER et al., 2017; RON; SHAMIR, 2013). A seguir são apresentadas algumas estratégias utilizadas para a deanonimização de dados, apresentadas em trabalhos com temas relacionados ao tema deste trabalho.

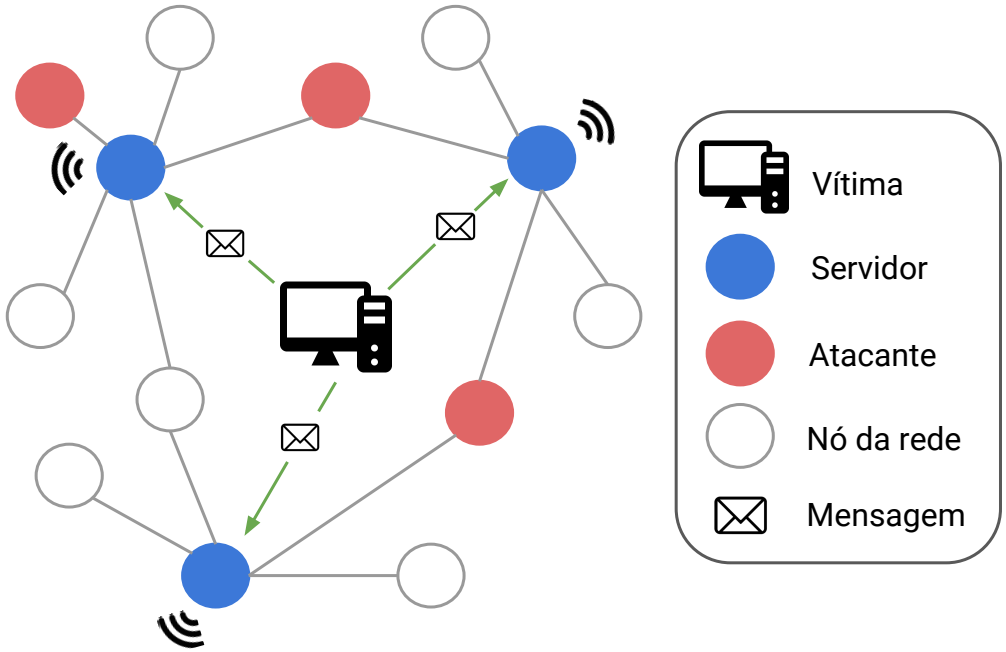
Em (KUMAR et al., 2017) é apresentada uma análise forense do *blockchain* da criptomoeda Monero com o objetivo de testar as garantias de irrastreabilidade das transações, um princípio de privacidade que é oferecido pela criptomoeda. Os autores analisam dados de transações que datam desde o primeiro bloco da criptomoeda, chamado de *genesis*, até a última transação realizada no dia 6 de Fevereiro de 2017. São propostas três heurísticas que definem estratégias de ataque aos dados das transações. A primeira explora uma falha no controle do número de *mixins* utilizados nas transações, permitindo que as transações que não utilizam nenhum *mixin* sejam rastreadas de maneira trivial, criando uma reação em cadeia que permite rastrear transações posteriores. A segunda heurística busca rastrear um novo tipo de transação introduzido recentemente no Monero, as RingCTs, assumindo que os *mixins* usados em uma transação geralmente são originados de um mesmo bloco e, assim, eliminando os *mixins* originários do mesmo bloco e obtendo as chaves reais usadas nas transações. A terceira e última estratégia de ataque realiza uma análise temporal nas chaves usadas como entrada em uma transação. Assumindo que transações mais antigas possuem maiores chances de serem usadas como *mixins*, o ataque seleciona a entrada mais recente como a chave real da transação.

É apresentada em (MILLER et al., 2017) uma análise empírica do conjunto de dados do *blockchain* do sistema Monero e são exploradas duas vulnerabilidades na seleção de *mixins* da criptomoeda. A primeira vulnerabilidade analisada consiste na mesma falha explorada pela primeira heurística explorada em (KUMAR et al., 2017), onde a maioria das transações não incluem nenhuma chave adicional para ofuscar os dados das transações. A segunda vulnerabilidade, de maneira similar ao trabalho anterior, é explorada através da análise da altura do bloco de origem da chave no *blockchain*. Quanto mais recente é um bloco, maior a sua altura no *blockchain*. As chaves cujo bloco de origem é o mais recente dentre as entradas da transação são selecionadas como a entrada real. Adicionalmente, os autores conduzem uma investigação sobre as práticas de mineração da criptomoeda, utilizando dados de comunidades que trabalham em conjunto para validar transações e obter XMR. Informações sobre as transações que remuneram participantes das comunidades de mineração são divulgadas com o objetivo de fornecer transparência ao serviço. Os autores usam estas informações para estimar a quantidade de blocos minerada

por algumas das maiores comunidades de usuários.

Em (BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014) é descrito um método para revelar a identidade de uma fração dos usuários do sistema Bitcoin, correlacionando seus pseudônimos com seus endereços IP públicos. O ataque baseia-se na identificação dos clientes através dos dados dos nós de entrada aos quais os mesmos conectam-se. Primeiramente, os autores realizam um procedimento para bloquear conexões com a rede Bitcoin utilizando nós da rede Tor comprometidos. O procedimento é realizado através da criação de um grande número de conexões com a rede Tor, selecionando para cada conexão um nó de saída distinto. Isto é feito para comprometer o máximo de nós possível. A próxima etapa é a disseminação de mensagens malformadas através das conexões criadas, fazendo com que a rede a penalize os nós de saída até que eles sejam banidos da rede. Este passo prejudica a comunicação de nós anônimos com a rede, fazendo com que os usuários conectem-se utilizando seu endereço IP real. Os atacantes obtêm o endereço IP público dos usuários a partir da inspeção das mensagens que são enviadas por eles na rede. A segunda parte do ataque explora a disseminação de mensagens do protocolo Bitcoin, que informa aos usuários conectados sobre a chegada de novos participantes na rede. A Figura 9 ilustra a organização da rede durante o ataque.

Figura 9 – Organização da rede durante o ataque de interceptação de tráfego.




Os atacantes executam instâncias do cliente do sistema Bitcoin e estabelecem conexões com o maior número de servidores possíveis. Quando um novo participante entra na rede, os servidores disseminam as informações dele para que os outros participantes possam conectar-se com o novo nó. Os atacantes armazenam as informações do novo participante juntamente com as informações dos servidores que disseminaram a mensagem.

O passo final, para revelar a identidade dos usuários, é detectar quando os servidores disseminam informações de novas transações na rede. Quando uma nova transação é compartilhada, o atacante cria uma lista dos servidores que disseminaram a informação. Usando os dados obtidos anteriormente sobre os participantes da rede, os atacantes procuram por um usuário que esteja conectado exatamente aos servidores que disseminaram a mensagem. Ao encontrar o participante que está conectado aos mesmos servidores, há uma grande probabilidade de que seja ele quem gerou a transação. O endereço IP do usuário é identificado como a origem da transação recém criada.

Em (RON; SHAMIR, 2013) é apresentada uma análise do histórico da rede Bitcoin através do grafo de transações, incluindo estatísticas sobre a quantidade de transações, a distribuição de entidades e o uso da criptomoeda. Os autores obtiveram os dados do *blockchain* e utilizaram uma variação do algoritmo *Union-Find* para encontrar endereços pertencentes à uma mesma entidade. Adicionalmente, são estudadas as maiores transações já ocorridas na história do Bitcoin e é mostrado o comportamento do fluxo de moedas na rede. Através da análise dessas transações é possível identificar os padrões de comportamento de usuários que enviam grandes quantias de criptomoedas. As moedas são inicialmente divididas em pequenas quantias e enviadas para um grande número de endereços diferentes. Posteriormente, elas são enviadas desses endereços para um endereço único novamente. Este procedimento é realizado para dificultar o rastreamento do histórico das transações.

O trabalho de (FLEDER; KESTER; PILLAI, 2015) apresenta um *framework* capaz de agrupar dados das atividades desempenhadas pelos usuários do sistema. O *framework* constrói um grafo dirigido, que ilustra o fluxo de transações do sistema Bitcoin, onde os usuários são representados por nodos e as transações por arestas. No grafo resultante, as entidades que possuem o maior número de conexões são consideradas como mais importantes, pois participaram de uma quantidade maior de transações. Para ajudar a classificar os nós de acordo com a sua importância, os autores utilizam o algoritmo *PageRank* (PAGE et al., 1999). Este algoritmo é comumente utilizado em mecanismos de busca para classificar páginas de acordo com a sua relevância. Após a criação do grafo de transações, é utilizado um sistema que coleta dados através de *web scraping*, uma técnica que utiliza algoritmos para vasculhar páginas da *web* automaticamente em busca de informações. São realizadas buscas em páginas como fóruns, redes sociais e plataformas de doação, procurando por informações de usuários que divulgaram seus pseudônimos da rede Bitcoin. O sistema utiliza os dados coletados para realizar anotações no grafo de transações da Bitcoin, associando as chaves presentes no grafo aos usuários através das informações encontradas na *web*.

Em (MEIKLEJOHN et al., 2013) é apresentado um estudo que coleta endereços públicos do sistema Bitcoin e realiza o agrupamento de endereços pertencentes à um mesmo usuário ou entidade. Os endereços são marcados com informações de identificação

dos usuários, coletados na *web* usando *web scraping*. Para obter as informações de identificação, os autores realizaram buscas em fóruns e outras páginas *web*, onde são divulgados endereços Bitcoin junto com nomes de usuários para fins de trocas, negociações e recebimento de doações. Além disso, os autores participaram de transações com plataformas fornecedoras de serviços como armazenamento, trocas de moedas, vendas e apostas a fim de identificar os endereços Bitcoin utilizados pelos serviços. Após a obtenção dos endereços, os autores geram um grafo de transações no qual as entidades são representadas pelo agrupamento de endereços pertencentes a um mesmo serviço ou usuário. O agrupamento permite a visualização da quantidade de transações nas quais as entidades estão envolvidas, pois todas as arestas de transações são direcionadas a um único nó que representa todos os endereços de um mesmo usuário. Desta forma é possível quantificar as moedas enviadas e recebidas por cada usuário.

A Tabela 3 apresenta uma comparação dos trabalhos relacionados. As colunas da tabela representam as estratégias utilizadas nos trabalhos para fins de deanonimização de dados e identificação dos usuários.

Tabela 3 – Comparação dos trabalhos relacionados.

| | **Análise dos dados de transações** | **Análise do tráfego de rede** | **Análise do grafo de transações** | ***Web Scraping*** | **Agrupamento de endereços** | **Quantificação de atividades dos usuários** |
|---|---|---|---|---|---|---|
| (KUMAR et al., 2017) | X | | | | | X |
| (MILLER et al., 2017) | X | | | | | X |
| (BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014) | X | X | | | | |
| (RON; SHAMIR, 2013) | X | | X | | X | X |
| (FLEDER; KESTER; PILLAI, 2015) | X | | X | X | X | |
| (MEIKLEJOHN et al., 2013) | X | | X | X | X | X |

Cada uma das estratégias utilizadas nos trabalhos relacionados serve propósitos diferentes. A combinação de técnicas pode contribuir para a geração resultados melhores, causando a identificação de um número maior de usuários.

- Análise dos dados de transações: Consiste na extração das informações sobre as transações armazenadas no *blockchain*. Após a extração dos dados é possível aplicar

neles as heurísticas de ataque. A análise dos dados de transações é fundamental para a criação de grafos de transações.

- Análise do trafego de rede: É realizada através da interceptação do tráfego de rede gerado pelos usuários conectados aos servidores do sistema. Quando um nó entra na rede, ele solicita aos seus nós de entrada, com os quais estabeleceu conexão, que disseminem seu endereço IP para que outros nós possam adicioná-lo à suas listas de conexões. Ao escolher um usuário para atacar, monitora-se quais nós disseminam o endereço desse usuário. Depois de conhecer os nós de entrada do usuário, é possível atribuir as novas transações disseminadas por estes nós aos usuários que estejam conectados a eles. Com os endereços e as transações, é possível associar um endereço IP à um pseudônimo do sistema. Esta estratégia é útil para descobrir os endereços públicos dos usuários do sistema. Entranto, vale ressaltar que o volume de usuários identificados através desta técnica é pequeno, pois é necessário aguardar até que as vítimas enviem dados de transações pela rede para que o ataque possa ser realizado.

- Análise do grafo de transações: O grafo de transações é criado utilizando os endereços públicos dos usuários do sistema. Quando um usuário efetua uma transação, é criada uma conexão entre o endereço do remetente e o endereço do recipiente. O grafo representa as transações realizadas e a interação entre os usuários do sistema.

- *Web Scraping:* Esta estratégia é utilizada para a obtenção de informações de usuários disponíveis na *web*. São realizadas buscas por dados de usuários que compartilham seus endereços públicos em fóruns e sites de venda de moedas. Esta técnica pode ser utilizada para a identificação de nós após ter sido construído o grafo de transações do sistema.

- Agrupamento de endereços: É utilizado para facilitar a visualização e a organização dos dados do grafo de transações. Ao construir o grafo de transações, endereços diferentes pertencentes à um mesmo usuário são representados por nós distintos. O agrupamento permite representar todos os endereços do mesmo usuário através de um único nó do grafo. Isto contribui para a identificação das relações entre usuários distintos e permite quantificar as atividades dos usuários.

- Quantificação das atividades dos usuários: Estudo das informações obtidas através da aplicação de outras técnicas. Permite identificar mudanças no comportamento dos usuários ao longo do tempo, como por exemplo, a variação do número médio de *mixins* utilizados no sistema Monero e o volume de moedas transacionadas por dia no sistema Bitcoin. Esta técnica contribui para a identificação dos serviços de criptomoedas mais utilizados pelos usuários, inclusive os que envolvem atividades criminosas, caso seus endereços públicos sejam conhecidos.

# 4 RESULTADOS PRELIMINARES

Neste capítulo são apresentados resultados obtidos com a implementação e verificação de ataques ao sistema Monero, utilizando estratégias existentes na literatura. O objetivo da realização destes ataques foi a extração dos dados do *blockchain*, exploração de falhas, e familiarização com os mecanismos de privacidade utilizados pelo sistema Monero. A realização dos ataques contribui para a obtenção de informações que ajudem a identificar novas falhas no sistema. Foi utilizado o gerenciador de repositórios GitLab (GITLAB TEAM, 2018) e o sistema de controle de versões Git (SOFTWARE FREEDOM CONSERVANCY, 2018) para realizar o versionamento do código e auxiliar na organização dos arquivos.

## 4.1 Extração de Dados

A primeira etapa na extração dos dados do *blockchain* foi a obtenção do cliente de linha de comando do sistema Monero, versão 0.11.1.0, e a sincronização dos dados locais com os dados das transações disponíveis na rede.

Para acessar os dados do *blockchain*, a ferramenta *Onion Monero Blockchain Explorer*, disponível no repositório de exemplos do Monero no GitHub, foi utilizada. Esta ferramenta serve como uma interface de acesso aos dados das transações. Ela retorna os dados solicitados atráves de consultas utilizando a API Javascript Object Notation (JSON). Um script foi desenvolvido para automatizar a extração dos dados dos blocos, utilizando a linguagem de programação Python, versão 3.6.

Os dados extraídos contêm desde o primeiro bloco, datado de 18 de Abril de 2014, até o bloco de número 1,514,000, criado no dia 20 de Fevereiro de 2018. No total, os 1,514,000 blocos extraídos contêm 3,976,181 transações, e destas, 1,514,000 são transações *coinbase*, que não possuem nenhuma entrada e remuneram mineradores pela criação do bloco. A primeira transação de um bloco é sempre uma transação *coinbase*. As informações extraídas foram armazenadas em um arquivo no formato CSV. Cada um dos blocos é composto pelos seguintes campos:

- Número do bloco
- *Timestamp* da criação
- Número de transações contidas no bloco
- Para cada transação:
  - *Hash* da transação
  - Número de entradas da transação
  - Número de *mixins* em cada entrada

- Endereços de entrada da transação
- Endereços de saída da transação
- Valores de saída da transação em XMR

Estas informações constituem o conjunto de dados utilizados na realização da análise descrita na seção 4.2.

## 4.2 Análise dos Dados

Uma análise das transações do sistema Monero foi realizada utilizando heurísticas apresentadas em trabalhos existentes na literatura. Os ataques exploram o princípio de irrastreabilidade, apresentado na subseção 2.2.4, o princípio de não-vinculação de endereços permanece inafetado. Os resultados obtidos demonstram a existência de problemas de privacidade nas transações efetuadas antes do lançamento do protocolo RingCT. Após o ínicio do uso das RingCTs, a porcentagem de transações afetadas pelas heurísticas começou a ser reduzida, pois o novo protocolo torna as transações resistentes aos ataques apresentados.

Duas heurísticas de deanonimização foram selecionadas para serem aplicadas aos dados extraídos. A primeira heurística, apresentada na subseção 4.2.1, baseia-se na exploração de uma falha de implementação do sistema para descobrir as entradas reais sendo utilizadas nas transações. A segunda heurística, apresentada na subseção 4.2.2, explora uma deficiência no algoritmo de escolha de *mixins* para descobrir as entradas reais das transações.

Algumas ocorrências interessantes foram observadas nos dados do *blockchain* durante a análise dos dados:

1. No bloco de número 8,330, a primeira transação não *coinbase* possui 10 entradas, e nove destas possuem um *mixin* além da chave real. Curiosamente, a entrada remanescente, que não utiliza *mixins*, possui um valor dezenas de vezes maior do que as outras. Isto é provavelmente um erro do sistema, pois todas as entradas de uma transação devem possuir o mesmo número de *mixins*.

2. O bloco de número 202,612 possui 513 transações. Destas, 511 têm como valor de entrada 1 piconero($10^{-12}$), a menor unidade da moeda. Nenhuma das 511 transações gera alguma saída. Em todas elas o valor de entrada é usado para pagar as taxas de transação.

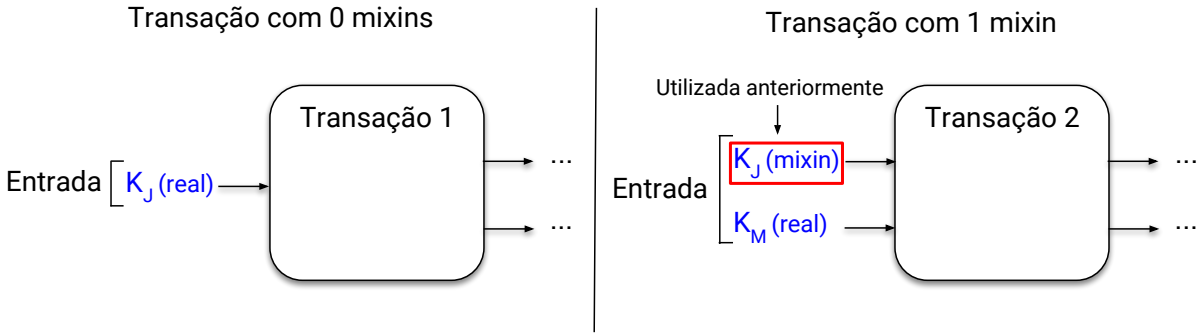
3. No bloco de número 981,002, a mesma chave aparece como saída nas duas transações do bloco. Isto não deve ocorrer, uma vez que todas as chaves de saída são geradas de forma que sejam únicas.

### 4.2.1 Análise de *Mixins*

A primeira estratégia de ataque aplicada foi a análise de transações com zero *mixins* (MILLER et al., 2017; KUMAR et al., 2017). Este ataque explora uma falha que permite a efetuação de transações sem a utilização de *mixins*. Isto ocorre por escolha do usuário ou pela falta de transações anteriores com valores iguais aos valores sendo transacionados. Mais tarde, o uso de denominações foi introduzido como meio de mitigar a falta de entradas com o mesmo valor, conforme discutido na subseção 2.2.4.

A falta de *mixins* em uma transação torna suas entradas reais visíveis a qualquer atacante que possua acesso aos dados do *blockchain*. A não utilização de *mixins* gera uma reação em cadeia que permite descobrir a entrada real de transações futuras. Isto afeta outras transações além da própria transação sem *mixins*. A Figura 10 ilustra o efeito que transações sem *mixins* desencadeam em outras transações, onde a primeira transação é realizada sem a utilização de qualquer *mixin* e posteriormente, a mesma chave é usada como *mixin* em outra transação. Suponha que João efetue uma transação com uma única entrada $K_J$ e nenhum *mixin*. Esta entrada é sabidamente a chave sendo utilizada na transação. Agora suponha que Maria efetue uma transação com apenas um *mixin* além da chave verdadeira, $K_M$, e que o sistema escolha a chave $K_J$ para ser utilizada como *mixin*. Neste caso, como a chave $K_J$ já foi utilizada, é possível deduzir que a entrada real da transação é a chave $K_M$.

Figura 10 – Efeito em cadeia de transações com 0 *mixins*.




Além das transações que são afetadas pela quebra do princípio de irrastreabilidade, outras transações são afetadas, tendo o tamanho efetivo do seu conjunto de *mixins* diminuído. Considerando o conjunto de *mixins* para uma entrada em uma transação, para cada *mixin* sabidamente já utilizado presente na entrada, o tamanho do conjunto de possíveis chaves utilizadas na transação é diminuído, tornando mais fácil o processo de adivinhar a chave real.

O ataque realizado foi baseado no pseudocódigo da heurística de análise de zero *mixins* (KUMAR et al., 2017). Foi desenvolvido um algoritmo na linguagem de progra-

**Algoritmo 1** Procedimento de análise de mixins

```
1:  procedure ANALISA-MIXINS(blocos)
2:      for each bloco ∈ blocos do                          ▷ Para cada bloco extraído
3:          transacoes ← extraiTransacoes(bloco)
4:          for each transacao ∈ transacoes do
5:              entradas ← extraiEntradas(transacao)
6:              for each entrada ∈ entradas do
7:                  chaves ← extraiChaves(entrada)
8:                  chavesParaAnalisar.add(chaves)
9:              end for
10:         end for
11:     end for
12:
13:     chavesUtilizadas ← {}
14:     while chavesUtilizadas for alterado do          ▷ Enquanto deduzir novas chaves
15:         for each entrada ∈ chavesParaAnalisar do
16:             chavesNaoRastreadas ← {}
17:             for each chave ∈ entrada do
18:                 if chave ∉ chavesUtilizadas then
19:                     chavesNaoRastreadas ← chavesNaoRastreadas ∪ chave
20:                 end if
21:             end for
22:             if |chavesNaoRastreadas| == 1 then
23:                 chavesUtilizadas ← chavesUtilizadas ∪ naoRastreadas
24:             end if
25:         end for
26:     end while
27:     return chavesUtilizadas
28: end procedure
```

mação C++ para a realização da análise das chaves das transações.

O Algoritmo 1, adaptado de (KUMAR et al., 2017), descreve o processo de descoberta de chaves utilizadas em transações anteriores. A primeira tarefa é extrair as chaves de entrada de cada uma das entradas presentes em cada transação. As transações contidas em cada bloco são extraídas (Linhas 2-3) e cada uma tem as chaves de suas entradas armazenadas em uma lista de chaves que será utilizada na análise (Linhas 4-8). A variável *chavesParaAnalisar* é uma lista composta por listas menores que representam as chaves de cada uma das entradas. As chaves extraídas das entradas são compostas por todas as chaves incluídas nas transações, ou seja, as chaves reais das transações e aquelas que foram utilizadas como *mixins*. O próximo passo é realizar a análise das chaves e repetir o processo enquanto o algoritmo for capaz de marcar novas chaves como utilizadas (Linha 14). Se após uma iteração nenhuma nova chave for identificada, o algoritmo retorna as chaves deduzidas. Para cada lista de entradas presente em *chavesParaAnalisar* é criado um conjunto vazio (Linha 16). Para cada chave da entrada é realizada uma verificação.

Se ela ainda não foi marcada como utilizada, é adicionada ao conjunto de chaves não rastreadas (Linhas 17-21). Se ao final da análise das chaves da entrada, apenas uma não foi rastreada, isto significa que a chave é a única presente na entrada ou que todas as outras chaves já foram marcadas como utilizadas. Em ambos os casos, a chave restante é chave real da transação e é adicionada ao conjunto de chaves utilizadas (Linhas 22-23). Na primeira iteração do algoritmo, as chaves presentes em transações com zero *mixins* são identificadas como utilizadas. Nas iterações seguintes, o número de chaves marcadas como utilizadas cresce. Algumas transações com múltiplas entradas têm algumas de suas chaves identificadas a cada iteração até que em algum momento a chave real seja deduzida. Por fim, o conjunto de chaves identificadas como utilizadas é retornado (Linha 27).

A Tabela 4 mostra os 13 primeiros tamanhos de *mixins* utilizados em entradas de transações, em quantas entradas são utilizados e a quantas destas entradas foram deduzidas utilizando a análise de *mixins*.

Tabela 4 – Frequência de mixins e quantia deduzida.

| **Quantidade de mixins** | **Nº de entradas** | **Entradas deduzidas** | **Taxa de dedução** |
|---|---|---|---|
| 0 | 12,207,748 | 12,207,748 | 100% |
| 1 | 707,788 | 609,383 | 86,09% |
| 2 | 4,496,449 | 1,774,903 | 39,47% |
| 3 | 1,486,633 | 951,153 | 63,98% |
| 4 | 2,966,970 | 446,946 | 15,06% |
| 5 | 301,068 | 73,845 | 24,52% |
| 6 | 425,726 | 201,659 | 47,36% |
| 7 | 20,416 | 4,264 | 20,88% |
| 8 | 27,115 | 3,483 | 12,84% |
| 9 | 15,956 | 2,144 | 13,43% |
| 10 | 111,064 | 22,863 | 20,58% |
| 11 | 2,576 | 372 | 14,44% |
| 12 | 3,793 | 716 | 18,87% |

É importante observar que a taxa de dedução apresentada na Tabela 4 é afetada pela quantia de *mixins*. A probabilidade de deanonimização aumenta com a redução do número de *mixins*. Outro fator que impacta na probabilidade de dedução das transações são as *mixins* escolhidas pelo sistema. Se as *mixins* escolhidas foram utilizadas em transações deduzíveis ou com nenhuma outra *mixin*, o efeito em cadeia afeta a privacidade da transação atual.

A incorporação das RingCTs ao sistema, a partir de 19 de Setembro de 2016, teve

um impacto na taxa de dedução de transações com duas e quatro *mixins*. Isto ocorreu porque após o lançamento do protocolo, o número mínimo de *mixins* em uma transação foi elevado para duas e meses depois, para quatro. Como os usuários tendem a utilizar a quantidade mínima de *mixins* permitida, pois este é o valor padrão em muitas aplicações que gerenciam endereços do sistema Monero, foi criado um grande número de transações não-rastreáveis utilizando duas e quatro *mixins*. As transações com zero, uma e três *mixins* possuem uma taxa de dedução alta porque a maioria das transações desses grupos foram efetuadas antes do lançamento do protocolo RingCT, quando as transações eram mais vulneráveis à dedução.

Para mostrar o impacto que este ataque pode causar até mesmo em transações com um número elevado de *mixins*, a Tabela 5 mostra algumas entradas com uma quantidade elevada de *mixins* que foram deanonimizadas pela heurística.

Tabela 5 – Entradas com número elevado de *mixins* afetadas pelo ataque.

| **Quantidade de mixins** | **Entradas deduzidas** |
|---|---|
| 50 | 149 |
| 70 | 22 |
| 90 | 10 |
| 100 | 39 |
| 153 | 1 |

Com o passar do tempo, a porcentagem de transações deduzíveis irá ser reduzida porque o uso do protocolo RingCT tornou-se obrigatório em todas as transações do sistema, a partir de setembro de 2017[1]. As transações RingCT só podem utilizar como *mixins* entradas de transações do mesmo tipo. Como o protocolo foi tornado obrigatório juntamente com uma política que requer o uso de pelo menos quatro *mixins* por entrada, não haverão mais transações transparentes que possam causar reações em cadeia.

A Tabela 6 sumariza os resultados da aplicação do ataque de análise de *mixins*.

Tabela 6 – Resumo dos resultados da análise de mixins.

| **Observável** | **Quantidade** |
|---|---|
| Número total de entradas | 22,845,510 (100%) |
| Entradas que contêm 0 mixins | 12,196,416 (53,39%) |
| Entradas vulneráveis à dedução | 4,097,846 (17,94%) |
| Total de entradas deanonimizadas | 16,294,262 (71,32%) |

[1] <https://getmonero.org/resources/moneropedia/ringCT.html>

### 4.2.2 Análise Temporal

A segunda estratégia de ataque utilizada foi baseada na heurística de análise temporal (MILLER et al., 2017; KUMAR et al., 2017). A análise temporal considera que uma chave gerada por uma transação não permanece sem ser utilizada por muito tempo. Chaves geradas em transações antigas tem uma probabilidade muito maior de terem sido utilizadas do que chaves geradas recentemente, portanto, é possível assumir que a chave real de uma transação é aquela cujo bloco de origem no *blockchain* é o mais recente dentre todas as entradas.

Apesar de ser uma heurística fácil de ser aplicada aos dados, a análise temporal é capaz de adivinhar corretamente a chave de mais de 90% das entradas das transações de versões antigas do sistema. Isto deve-se à uma característica da antiga distribuição matemática utilizada para a seleção de *mixins*, que favorecia a escolha de chaves antigas. A distribuição matemática utilizada na escolha de *mixins* foi atualizada em 13 de Dezembro de 2016, na versão 0.10.1 do sistema. Mudanças foram realizadas para garantir que pelo menos 25% das entradas de uma transação sejam escolhidas dentre as chaves geradas nos últimos cinco dias (MILLER et al., 2017). Isto ajuda a evitar que a chave verdadeira de uma entrada seja sempre a chave mais recente.

O Algoritmo 2, baseado em (KUMAR et al., 2017; MILLER et al., 2017), mostra o pseudocódigo do procedimento de análise temporal. O mesmo processo é realizado para cada uma das entradas. Primeiro, o conjunto de chaves pertencentes a entrada é extraído e armazenado em uma lista de chaves (Linhas 3-6). Então, é identificada qual das chaves dentre as presentes na entrada é originária do bloco mais recente (Linhas 7-14). As chaves restantes recebem uma marcação para identificá-las como *mixins*. Por fim, as entradas são retornadas com suas *mixins* marcadas (Linha 14).

Para tornar mais fácil o processo de identificação dos blocos de origem, as chaves foram armazenadas utilizando a sua posição relativa no *blockchain*. A posição relativa de uma chave é dada no formato (1)-(2)-(3), onde: (1) é o número do bloco de origem da chave, (2) é o índice da transação onde a chave aparece no bloco e (3) é o índice que representa a posição da chave nas saídas da transação. Os índices são considerados a partir de zero. Por exemplo, uma chave originária da segunda transação do bloco 645020, sendo a 3ª saída da transação é representada como 645020-1-2.

Esta heurística pode gerar falsos positivos porque baseia-se na probabilidade das chaves mais recentes serem as chaves reais. Para medir a taxa de acertos da heurística, a análise temporal foi aplicada às mesmas transações que foram deanonimizadas com sucesso pela heurística de análise de *mixins*, para as quais é possível garantir que os resultados estão corretos. Desta forma é possível estimar a eficácia da análise temporal sobre as transações para as quais não se sabe a chave real.

Do conjunto de 4,097,846 entradas deduzidas utilizando a estratégia da análise de *mixins*, 3,727,410, ou seja, aproximadamente 90,96% foram corretamente deduzidas com

**Algoritmo 2** Procedimento de análise temporal

```
1: procedure ANALISE-TEMPORAL(entradas)
2:     for each entrada ∈ entradas do                ▷ Para cada uma das entradas
3:         chaves = []
4:         for each chave ∈ entrada do
5:             chaves.add(chave)
6:         end for
7:         chaveReal = chaveMaisRecente(chaves)
8:         for each chave ∈ entrada do
9:             if chave != chaveReal then
10:                chave += '*'          ▷ Marca as chaves identificadas como mixins
11:            end if
12:        end for
13:    end for
14:    return entradas                           ▷ Retorna a lista de chaves marcadas
15: end procedure
```

a utilização da análise temporal das chaves. Isto significa que apenas em 9,04% dos casos a chave real não é a mais recente dentre as chaves as chaves de entrada.

Não é possível, porém, assumir que 90,96% das transações que resistiram à análise de *mixins* podem ser deduzidas pela análise temporal, pois novas políticas de escolha de *mixins* foram implementadas em 16 de dezembro de 2016, garantindo a inclusão de *mixins* recentes nas entradas, conforme discutido na subseção 2.2.4.

Com a finalidade de estimar o máximo de transações que já foram passíveis de serem deanonimizadas, os ataques foram aplicados aos dados do *blockchain* que precedem a atualização na política de seleção de *mixins*, contidos nos blocos 0 ao 1200078.

Tabela 7 – Resultados dos ataques e estimativa de entradas deanonimizáveis.

| **Observável** | **Quantidade** |
|---|---|
| Número total de entradas | 17,374,129 (100%) |
| Entradas que contêm 0 mixins | 12,130,656 (53,39%) |
| Entradas vulneráveis à dedução | 3,481,943 (20,04%) |
| Entradas deanonimizadas | 15,612,599 (89,86%) |
| Entradas não-deduzíveis pelo Algoritmo 1 | 1,761,530 (10,13%) |
| Taxa de acertos da análise temporal | 92,48% |
| Total de entradas deanonimizáveis (estimado) | 17,195,509 (98,97%) |

A Tabela 7 apresenta os resultados da execução dos ataques nos dados do *blockchain* anteriores a atualização da política de seleção de *mixins*. Como pode ser observado, a análise temporal obteve uma taxa de acertos de 92,48% sobre as entradas para as quais as chaves reais são conhecidas. Isto permite assumir que das 1,761,530 entradas restantes,

92,48%, ou seja, 1,629,053 entradas, podem ser corretamente deduzidas pela análise temporal. Isto leva a uma taxa de deanonimização estimada em 98,97% de todas as entradas presentes no *blockchain*, um resultado preocupante dado o foco da criptomoeda Monero em proteger as informações dos usuários.

Em resumo, os resultados apresentados evidenciam a necessidade de buscar e contribuir para a correção de falhas nas estratégias de privacidade empregadas no sistema Monero. A resolução dos problemas segurança contribui para a criação de criptomoedas mais seguras, que atendam as necessidades de privacidade dos seus usuários.

# 5 CRONOGRAMA DE ATIVIDADES

A Tabela 8 apresenta as atividades que serão realizadas durante o período de desenvolvimento do TCC II.

A primeira atividade é a realização de correções e melhorias no material elaborado durante o desenvolvimento do TCC I, levando em consideração as críticas e sugestões dos membros da banca de avaliação.

A segunda atividade consiste em explorar os mecanismos de privacidade do sistema Monero na busca por falhas de segurança. Para este propósito, serão utilizadas estratégias existentes na literatura como a interceptação de tráfego de rede, a criação de grafos de transações e *web scraping*. Também serão utilizados novos ataques a serem elaborados durante o desenvolvimento do trabalho, como por exemplo a exploração das chaves privadas de visualização. Serão codificados ataques de teste nos dados do *blockchain*, com o objetivo de verificar a relevância das heurísticas.

O próximo passo é a codificação completa das rotinas de ataque selecionadas como relevantes, para que os ataques possam ser realizados em qualquer conjunto de dados extraídos do *blockchain*.

A quarta etapa é a validação dos resultados. Os dados obtidos através da utilização de novos ataques serão comparados com os dados obtidos no TCC I. Um exemplo é a checagem da corretude dos resultados ao compará-los com os dados do ataque de análise de *mixins*, para o qual é possível provar que os resultados estão corretos.

A atividade final, realizada em paralelo com a validação dos dados, é a escrita do texto do TCC II. O documento irá apresentar resultados, observações e direções para pesquisas futuras sobre o tema.

Tabela 8 – Cronograma de atividades do TCC II.

| | **Julho** | **Agosto** | **Setembro** | **Outubro** | **Novembro** |
|---|---|---|---|---|---|
| Realização de correções e melhorias conforme avaliação do TCC I | X | X | | | |
| Exploração do sistema Monero e investigação de possíveis falhas | X | X | X | X | |
| Codificação de ataques à privacidade | | X | X | X | |
| Validação dos resultados | | | | X | X |
| Escrita do TCC II | | | | X | X |

# REFERÊNCIAS

ADKISSON, J. **Why Bitcoin is So Volatile**. 2018. Disponível em: <https://www.forbes.com/sites/jayadkisson/2018/02/09/why-bitcoin-is-so-volatile/#680d382239fb>. Citado na página 22.

APOSTOLAKI, M.; ZOHAR, A.; VANBEVER, L. Hijacking bitcoin: Routing attacks on cryptocurrencies. In: IEEE. **Security and Privacy (SP), 2017 IEEE Symposium on**. [S.l.], 2017. p. 375–392. Citado na página 35.

BIRYUKOV, A.; KHOVRATOVICH, D.; PUSTOGAROV, I. Deanonymisation of clients in bitcoin p2p network. In: ACM. **Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security**. [S.l.], 2014. p. 15–29. Citado 5 vezes nas páginas 17, 33, 35, 36 e 38.

CHAUM, D. Blind signatures for untraceable payments. In: SPRINGER. **Advances in cryptology**. [S.l.], 1983. p. 199–203. Citado na página 17.

CRYPTONOTE. **CryptoNote Technology**. 2015. Disponível em: <https://cryptonote.org/inside/>. Citado na página 32.

DUFFIELD, E.; DIAZ, D. **Dash: A PrivacyCentric CryptoCurrency**. [S.l.]: September, 2014. Citado na página 17.

FLEDER, M.; KESTER, M. S.; PILLAI, S. Bitcoin transaction graph analysis. **arXiv preprint arXiv:1502.01657**, 2015. Citado 3 vezes nas páginas 17, 37 e 38.

GITLAB TEAM. **GitLab**. 2018. Disponível em: <https://https://about.gitlab.com/>. Citado na página 41.

KSHETRI, N. 1 blockchain's roles in meeting key supply chain management objectives. **International Journal of Information Management**, Elsevier, v. 39, p. 80–89, 2018. Citado na página 22.

KUMAR, A. et al. A traceability analysis of monero's blockchain. In: SPRINGER. **European Symposium on Research in Computer Security**. [S.l.], 2017. p. 153–173. Citado 7 vezes nas páginas 17, 30, 35, 38, 43, 44 e 47.

MEDFAR87. **Cryptocurrency Growth & Adoption Statistics**. 2018. Disponível em: <https://steemit.com/cryptocurrency/@medfar87/cryptocurrency-growth-and-adoption-statistics>. Citado na página 22.

MEIKLEJOHN, S. et al. A fistful of bitcoins: characterizing payments among men with no names. In: ACM. **Proceedings of the 2013 conference on Internet measurement conference**. [S.l.], 2013. p. 127–140. Citado 4 vezes nas páginas 17, 27, 37 e 38.

MILLER, A. et al. An empirical analysis of traceability in the monero blockchain. **arXiv preprint arXiv:1704.04299**, 2017. Citado 5 vezes nas páginas 17, 35, 38, 43 e 47.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. 2008. Citado 3 vezes nas páginas 17, 27 e 30.

PAGE, L. et al. **The PageRank citation ranking: Bringing order to the web.** [S.l.], 1999. Citado na página 37.

POPOV, S. The tangle. 2014. Citado na página 22.

RON, D.; SHAMIR, A. Quantitative analysis of the full bitcoin transaction graph. In: SPRINGER. **International Conference on Financial Cryptography and Data Security**. [S.l.], 2013. p. 6–24. Citado 3 vezes nas páginas 35, 37 e 38.

SABERHAGEN, N. V. **Cryptonote v 2. 0**. 2013. Citado 2 vezes nas páginas 24 e 30.

SASSON, E. B. et al. Zerocash: Decentralized anonymous payments from bitcoin. In: IEEE. **Security and Privacy (SP), 2014 IEEE Symposium on**. [S.l.], 2014. p. 459–474. Citado na página 17.

SCHWARTZ, D. et al. The ripple protocol consensus algorithm. 2014. Citado na página 22.

SOFTWARE FREEDOM CONSERVANCY. **Git**. 2018. Disponível em: <https://git-scm.com/>. Citado na página 41.

TORPEY, K. **Study Suggests 25 Percent of Bitcoin Users Are Associated With Illegal Activity**. 2018. Disponível em: <https://bitcoinmagazine.com/articles/study-suggests-25-percent-bitcoin-users-are-associated-illegal-activity1/>. Citado na página 29.

TSCHORSCH, F.; SCHEUERMANN, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 3, p. 2084–2123, 2016. Citado 2 vezes nas páginas 22 e 25.

WILLIAMS, S. **Meet the Newest Cryptocurrency Trend: Privacy Coins**. 2017. Disponível em: <https://www.fool.com/investing/2017/12/27/meet-the-newest-cryptocurrency-trend-privacy-coins.aspx>. Citado na página 30.

WOOD, G. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum Project Yellow Paper**, v. 151, p. 1–32, 2014. Citado na página 17.