Revista Brasileira de Computação Aplicada, Novembro, 2019

DOI: 10.5335/rbca.v11i3.9394
Vol. 11, Nº 3, pp. 12–27
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

TUTORIAL

# Introdução às tecnologias dos blockchains e das criptomoedas

# Introduction to blockchain and cryptocurrencies

João Otávio Massari Chervinski[1] and Diego Kreutz[1]

[1]Universidade Federal do Pampa, Alegrete-RS, 97546-550, Brazil

*joaootaviors@gmail.com; kreutz@acm.org

**Recebido:** 05/05/2019. **Revisado:** 14/08/2019. **Aceito:** 25/09/2019.

## Resumo

As tecnologias utilizadas pelas criptomoedas, como o *blockchain*, tornaram possível a realização de transações entre os usuários sem a necessidade de uma entidade intermediária (exemplo: um banco). As criptomoedas (como Bitcoin e Monero) podem ser definidas como sistemas descentralizados que operam sem a necessidade de intermediários, tornando possível a realização de pagamentos descentralizados e sem fronteiras, onde usuários podem participar de transações sem a necessidade de estabelecer confiança. Devido a essas características e possibilidades, o uso de *blockchain* difundiu-se rapidamente pelos sistemas financeiros tradicionais. Grandes bancos e instituições financeiras passaram a investir significativamente na tecnologia como forma de melhorar a eficiência e reduzir custos em transações financeiras entre bancos e/ou inter-países. Uma das primeiras criptomoedas voltadas para instituições financeiras foi a Ripple. Rapidamente, a aplicação teórica e prática das tecnologias utilizadas pelas criptomoedas difundiu-se também para os mais diversos domínios, com o objetivo de solucionar diferentes tipos de problemas. Este tutorial apresenta uma introdução às tecnologias utilizadas pelas criptomoedas. O principal objetivo é difundir conhecimento sobre o assunto e estimular o desenvolvimento de pesquisas relacionadas à essas tecnologias. No decorrer do tutorial, são discutidos os fundamentos de um *blockchain* e das criptomoedas Bitcoin e Monero, com especial atenção na segurança e privacidade dos dados.

**Palavras-Chave**: Blockchain; Bitcoin; Criptomoedas; Monero; Sistemas Distribuídos;

## Abstract

The technology used by cryptocurrencies, such as blockchains, made it possible for users to transact with each other without needing a trusted third-party entity (a bank for example). Cryptocurrencies (like Bitcoin and Monero) can be defined as trustless decentralized systems which operate without a financial intermediary, making it possible to build decentralized payment systems where users are able to send payments securely without establishing trust. Due to those characteristics, the use of blockchains has spread rapidly among traditional financial systems. Large banks and corporations are investing heavily in the technology, aiming to increase its systems efficiency and reduce the costs of international transactions. Ripple was one of the first cryptocurrencies directed to financial institutions. The application of technologies brought by cryptocurrencies spread quickly to different areas with the intent of solving a myriad of problems. This tutorial presents an introduction to the technologies utilized by cryptocurrencies. The main objective is to share knowledge about the topic and promote the development of research involving those technologies. We discuss the fundamentals of blockchains and the cryptocurrencies Bitcoin and Monero, with emphasis on data security and privacy.

**Keywords**:
Blockchain; Bitcoin; Cryptocurrencies; Distributed Systems; Monero;

## 1 Introdução

O lançamento de criptomoedas como a Bitcoin Nakamoto (2008), Monero Monero's Team (2019) e Ripple Ripple (2019) deu origem a uma revolução nos sistemas de pagamento no mundo todo. A Bitcoin é o primeiro e mais conhecido caso disruptivo de sucesso. Criptomoedas como a Bitcoin funcionam de maneira descentralizada, isto é, sem a necessidade de uma autoridade reguladora, permitindo que qualquer um participe da rede e efetue transações. Uma das grandes inovações trazidas pela Bitcoin foi a criação de um sistema descentralizado no qual participantes são capazes de enviar pagamentos uns aos outros sem a necessidade de estabelecer confiança prévia. Quando pagamentos são efetuados através de uma empresa de cartões de crédito convencional, o usuário estabelece uma relação de confiança com a empresa e delega a ela a responsabilidade de validar os seus dados e efetuar a transação corretamente. Criptomoedas como a Bitcoin conseguiram livrar-se da necessidade de uma autoridade intermediária através da tecnologia de *blockchain*.

Um *blockchain* é uma estrutura de dados distribuída, formada por uma série de blocos de informação encadeados. Cada participante da rede pode obter uma cópia completa dos dados e compartilhá-la com outros participantes. Numa rede de *blockchain* os usuários trabalham de maneira colaborativa para validar transações, utilizando criptografia para garantir a sua segurança e verificabilidade. Mecanismos criptográficos também são utilizados para garantir a ordem dos blocos e evitar a sua adulteração, visto que só deve existir uma sequência válida de blocos. Entretanto, apesar dos *blockchains* serem utilizados principalmente em criptomoedas, já existem propostas para a aplicação dessa tecnologia em áreas e temas como sistemas financeiros, registro e proteção de propriedade intelectual, sistemas baseados em reputação, cadeia de suprimentos, setor de energia, setor de saúde, e-gov, marketing e sistemas de votação eletrônica Zheng et al. (2018), Korpela et al. (2017), Hoy (2017), Kelly and Williams (2016), Chen et al. (2018), Kar (2016), Lacey (2016), Christidis and Devetsikiotis (2016), Mizrahi (2015), Walport (2016), Olnes et al. (2017), Marinoff (2018).

Este tutorial tem como objetivo apresentar os principais conceitos inerentes às tecnologias de *blockchain* e criptomoedas, contribuindo para a difusão do conhecimento desta área que está no seu auge de pesquisa e desenvolvimento. No decorrer do tutorial, são abordadas também questões relacionadas à privacidade e segurança em criptomoedas como a Bitcoin e a Monero. Por fim, vale ressaltar que atualmente há muitas oportunidades, tanto na academia (exemplo: novas conferências específicas, criadas em 2018 e 2019) quanto no mercado (exemplo: muitas empresas investindo e apostando nessas novas tecnologias), relacionadas diretamente aos temas aqui discutidos.

O restante do tutorial está dividido como segue. Na Seção 2 são apresentados e discutidos os detalhes da tecnologia *blockchain*. A Seção 3 discute as criptomoedas e métodos de mineração e criação de novas moedas. O funcionamento da criptomoeda Bitcoin é detalhado na Seção 4. A questão da privacidade no contexto das criptomoedas é discutida na Seção 5. Nas Seções 6 e 7 são apresentadas a criptomoeda Monero e as considerações finais do tutorial.

## 2 Blockchain

O *blockchain* é um registro de informações distribuído formado por uma cadeia de blocos de dados, conectados uns aos outros por um sistema que utiliza funções *hash* criptográficas. A Fig. 1 apresenta o resultado da aplicação da função *hash* criptográfica SHA-256 (Secure Hash Algorithm-2) NIST (2018) em duas entradas distintas. Como pode ser observado, a saída da função é completamente diferente para as duas entradas aparentemente idênticas. A utilização deste tipo de função é essencial para a garantir a segurança e a integridade dos dados presentes no *blockchain*.

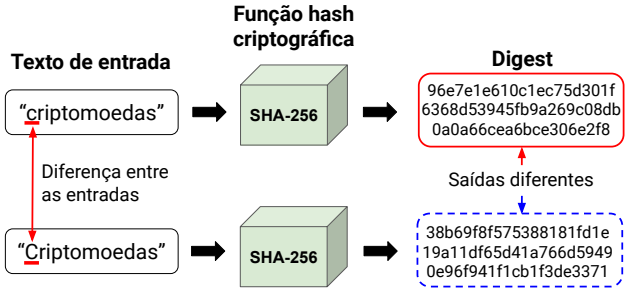


**Figura 1:** Aplicação de uma função *hash* criptográfica.

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:

($p_1$) A aplicação da função sobre o mesmo dado deverá sempre retornar o mesmo resultado.
($p_2$) Computar um *digest* para uma determinada entrada deve ser fácil, i.e, a operação pode ser realizada em menos de um segundo.
($p_3$) Descobrir quais dados foram utilizados como entrada da função analisando somente o *digest* deve ser muito difícil. Para isso é necessário aplicar a função em entradas aleatórias distintas até que a saída gerada seja igual ao *digest* sendo analisado, uma operação que pode necessitar de vários anos considerando as capacidades atuais de processamento.
($p_4$) Encontrar duas entradas que gerem o mesmo *digest* deve ser muito difícil, necessitando de vários anos de processamento.
($p_5$) Uma pequena mudança na entrada deve alterar o *digest* resultante de tal forma que não seja possível encontrar alguma relação entre as saídas geradas pelo dado original e pelo dado alterado.

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 Fig. 1. Ao aplicar uma função *hash* criptográfica em um arquivo e enviar o *digest* para a pessoa que irá recebê-lo, o receptor poderá aplicar a função novamente sobre o arquivo e verificar se o resultado é igual ao *digest* recebido. Desta forma, ela garante que os dados não foram modificados, desde que o *digest* tenha sido recebido de forma segura. Por exemplo, suponha que João trabalhe no setor financeiro de uma empresa e que tenha sido 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 pequenas quantidades de dados como os *digests*. Maria envia o *digest* do arquivo para João utilizando o canal de comunicação seguro. 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.

No *blockchain*, a conexão lógica entre os blocos de dados é estabelecida e garantida através de *digests* de uma função *hash* criptográfica. Cada bloco na cadeia aponta para um bloco anterior identificado por um *digest* único (*digest* do bloco). Isto fornece uma propriedade interessante para o registro de transações: quando um bloco é adicionado ao final da cadeia de informações, 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. Um exemplo de um conjunto de dados armazenados em cada bloco de um *blockchain* fictício voltado para o armazenamento de transações financeiras é:

($d_1$) Versão: Versão do sistema utilizada.
($d_2$) *Timestamp*: Representação da data e hora de criação do bloco.
($d_3$) *Nonce*: Número auxiliar utilizado para realização do cálculo da função *hash* criptográfica.
($d_4$) *Digest* do bloco anterior: Resultado da aplicação de uma função *hash* criptográfica sobre os dados do bloco anterior.
($d_5$) Conta de origem: A conta de onde o dinheiro será retirado.
($d_6$) Conta de destino: A conta que receberá o dinheiro.
($d_7$) Valor da transação: Quantia de dinheiro a ser transferida.

A Fig. 2 ilustra a conexão entre uma cadeia de blocos. Considerando a estrutura de bloco apresentada, a informação que garante a ordem correta dos blocos é o ponteiro para o *digest* do bloco anterior na cadeia. A utilização de uma função *hash* criptográfica nos dados de cada bloco gera um *digest* único, permitindo estabelecer um vínculo forte e único entre os dados. Isso garante que exista uma única sequência válida de blocos.

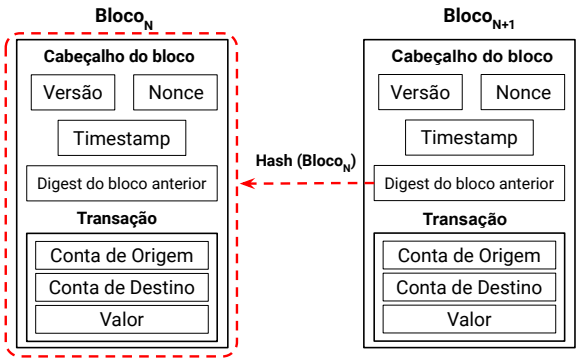


**Figura 2:** Representação dos blocos em um *blockchain*.

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 a probabilidade de gerar o mesmo *digest* a partir de dados de entrada diferentes é extremamente baixa, seria mais fácil alterar também o campo que aponta para o *digest* do bloco anterior nos blocos seguintes. Bastaria que o atacante mudasse os campos nos blocos posteriores, calculasse seus *digests* e repetisse o processo alterando os ponteiros até o final do *blockchain*. No entanto, um dos sistemas utilizado para controlar a criação de novos blocos, chamado de Prova de Trabalho, do inglês *Proof-of-Work* (PoW), evita que isto aconteça. 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. A restrição adotada na prática pela criptomoeda Bitcoin é encontrar um bloco cujo *digest* resultante possua os primeiros $n$ bits iguais a zero, onde $n$ depende da dificuldade de mineração determinada pelo sistema. Para variar a saída da função *hash* e encontrar um bloco que atenda a restrição é necessário alterar os dados de entrada, para esta finalidade há em cada bloco um campo chamado de *nonce*, que é alterado até que o resultado desejado seja obtido.

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 implicam 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 dos usuários. Além do PoW, foram propostos outros esquemas de criação de blocos, como o *Proof-of-Stake* (PoS), o *Proof-of-Activity* (PoA) e o *Proof-of-Publication* (PoP) Tschorsch and Scheuermann (2016).

Eventualmente, pode ocorrer uma bifurcação no *blockchain* quando novos blocos são adicionados, causada pelo tempo necessário para a propagação dos blocos na rede. A Fig. 3 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 dados 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 são os últimos blocos. Não é possível saber qual das cadeias é a correta.

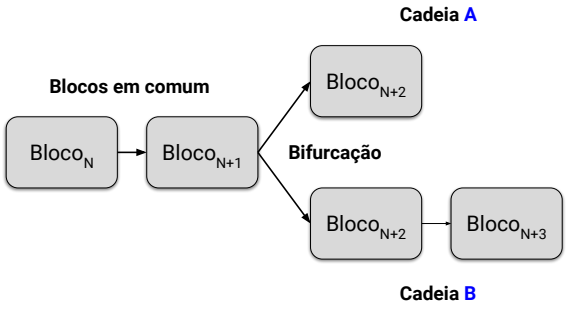


**Figura 3:** Bifurcação na cadeia de blocos.

Para resolver o problema da bifurcação na cadeia de blocos, alguns sistemas empregam 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, em várias criptomoedas é recomendado que os usuários aguardem até que mais blocos sejam adicionados após o bloco onde sua própria transação foi validada. Isso ajuda a garantir que a transação não será desfeita.

A tecnologia de *blockchain* está se popularizando e sua aplicação como uma estrutura de armazenamento de dados vem sendo investigada em uma grande gama de sistemas em diversas áreas IEEE (2017). *Blockchains* podem (potencialmente) substituir plataformas digitais em diferentes setores, além da área de finanças. Porém, o investimento em pesquisa e desenvolvimento ainda é embrionário e necessário para acelerar a criação e comercialização de soluções baseadas nesta tecnologia.

## 3 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 leva a menores taxas de transações, maior controle, e mais privacidade. 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*. Entretanto, 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, segundo estatísticas de mercado, o preço pode variar mais de 10% em poucas horas Adkisson (2018).

### 3.1 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 existem criptomoedas como a Ripple Schwartz et al. (2014) e a IOTA Popov (2014) que não possuem um sistema através do qual os usuários possam obter moedas ao criar blocos de transações, ou seja, não podem ser mineradas. Neste tutorial, iremos focar a discussão em processos de mineração similares aos que ocorrem em criptomoedas como a Bitcoin e a Monero.

O processo de mineração consiste em participar da criação de blocos válidos através do esquema de PoW, PoS ou outro similar, conforme determinado pelo respectivo sistema. Esse processo é essencial para garantir

o funcionamento do *blockchain* e da moeda digital, pois é através dele que as transações são confirmadas e adicionadas ao final da cadeia de blocos.

Mineradores são participantes da rede que utilizam sua capacidade computacional para tentar computar *digests* válidos para os blocos e adicioná-los ao final do *blockchain*. Quando um usuário efetua uma transação, seus dados são compartilhados na rede para que mineradores possam validá-la. Para cada transação efetuada, os usuários devem incluir uma taxa como pagamento para recompensar os mineradores. Transações que pagam taxas maiores são geralmente escolhidas primeiro e são incluídas em blocos mais rapidamente. Cada minerador escolhe as transações (disponíveis na rede) que irão fazer parte do bloco. Em seguida, inicia o processo de encontrar um *digest* que atenda às restrições do sistema.

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 bloco de transações. Quando um minerador valida um bloco, sua informação é disseminada na rede para que os outros mineradores atualizem suas cópias do *blockchain* e escolham novas transações para validar.

A primeira transação de cada bloco, chamada de *coinbase transaction*, é um tipo especial de transação 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.

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. A redução da recompensa também contribui para a valorização da moeda, que passa a valer mais quando a procura aumenta e a oferta diminui. Em um momento no futuro, 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 2019, 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 porque o sistema eleva automaticamente a dificuldade de criação de novos blocos quando a velocidade da rede aumenta. Esta estratégia é empregada 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.

| Nº 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) |

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 interessante pelo fato de aumentar a probabilidade de retorno financeiro uma vez que é difícil um único usuário, sozinho, competir com os demais e as *mining pools*.

## 3.2 Hardware especializado para mineração

A mineração de blocos pode ser uma atividade muito lucrativa se um usuário for capaz de validar vários blocos em um curto período de tempo. Considerando a cotação atual da criptomoeda Bitcoin (20,313.72 reais em 27 de abril de 2019), a recompensa de 12,5 BTC pela criação de um bloco é equivalente à 253,921.50 reais. Porém, o processo de PoW torna pequenas as chances de um usuário conseguir validar até mesmo um único bloco.

Com o objetivo de aumentar o lucro proveniente da mineração de criptomoedas, algumas empresas desenvolveram equipamentos de hardware específicos para computar Provas-de-Trabalho da Bitcoin, chamados de circuitos integrados de aplicação específica, do inglês *Application Specific Integrated Circuit*s (sASICs). Estes equipamentos são projetados especificamente para computar *digests* de blocos em uma velocidade muito superior àquela alcançavel em hardware comum. Alguns dos ASICs mais poderosos, destinados ao sistema Bitcoin, como o *ANTMINER S9 Hydro* da fabricante Bitmain, possuem uma capacidade de processamento de até 18,000,000,000 (18 trilhões) de *hashes* por segundo, enquanto um processador Intel Core i7-3930k consegue computar apenas cerca de 98,000 *hashes* por

segundo Cointopper (2018).

O uso de ASICs permite que usuários monopolizem a criação de novos blocos, especialmente quando vários utilizadores desses dispositivos cooperam em *mining pools*. Algumas *Graphics Processing Unit*s (sGPUs) também são utilizadas por mineradores para computar *hashes* por serem mais rápidas do que processadores comuns, porém, sua capacidade também é muito inferior aos *hardwares* especializados. A popularização de ASICs impede que usuários obtenham lucro através do processo de PoW do Bitcoin a menos que invistam na aquisição de equipamentos especializados Jefferys (2018).

O sistema Monero utiliza um algoritmo resistente à ASIC, chamado de CryptoNight, no seu processo de PoW. Este algoritmo é usado para tornar o processo de mineração mais igualitário do que em outras criptomoedas. O algoritmo CryptoNight pertence à uma classe de funções denominada *memory-bound*. Este tipo de função funciona de maneira que o tempo necessário para resolver um problema computacional depende diretamente da quantidade e da velocidade da memória disponível para armazenar os dados utilizados durante a sua resolução. A utilização do algoritmo CryptoNight combate o uso de ASICs, que tornaram-se praticamente essenciais para usuários que desejam participar do processo de PoW da Bitcoin.

### 3.3 Mineração através de navegadores

A idéia de inserir códigos de mineração de criptomoedas em páginas da *Web* surgiu logo após o lançamento da Bitcoin Eskandari et al. (2018). Pouco tempo após a proliferação da idéia, várias aplicações de mineração desenvolvidas com a linguagem de programação JavaScript tornaram-se populares como: JSMiner, MineCrunch, Tidbit e BitcoinPlus. O processo de mineração utilizando navegadores foi divulgado como uma alternativa à exibição de propagandas para a monetização de conteúdos em páginas da *Web*. Ao visitar uma página que contém uma aplicação de mineração, os recursos computacionais do usuário são utilizados para computar *digests* de blocos de transações, uma etapa essencial na criação de blocos e obtenção de recompensas em sistemas de criptomoedas. Esse tipo de serviço possui a vantagem de funcionar em qualquer plataforma, bastando que os usuários estejam utilizando um navegador que permita a execução de códigos JavaScript.

Algumas páginas *Web* defendem o uso benigno de mineradores baseados em código JavaScript para a geração de lucro e manutenção de sua infraestrutura Saad et al. (2018), solicitando que os usuários emprestem seu poder computacional em troca de acesso ao conteúdo da página. Entretanto, o uso dessas aplicações por usuários maliciosos difundiu-se rapidamente devido à facilidade de execução de ataques, necessitando apenas que as vítimas possuam uma conexão com a Internet e visitem uma página da *Web* infectada com o código do minerador desenvolvido em JavaScript.

Após a difusão inicial da idéia e o surgimento de diversas aplicações de mineração via navegadores, desenvolvedores e páginas da *Web* que compactuavam com o uso desse tipo de aplicações passaram a enfrentar problemas judiciais devido ao uso não autorizado dos recursos computacionais dos usuários Blattberg (2014). Além disso, com a popularização da Bitcoin e o crescimento da base de usuários do sistema a dificuldade de mineração de novos blocos passou a aumentar rapidamente. O aumento constante da dificuldade, somado ao fato de que as aplicações para mineração desenvolvidas em JavaScript possuiam um desempenho inferior às aplicações nativas para *desktops*, desencorajou a utilização de aplicações de mineração via navegadores. Páginas *Web* que faziam uso dessas aplicações obtiam pouco retorno monetário e estavam sujeitas a envolverem-se em problemas judiciais.

No ano de 2017, anos após a primeira onda de aplicações de mineração codificadas em JavaScript, as atividades de mineração em navegadores aumentaram expressivamente devido ao lançamento de um novo serviço de mineração de criptomoedas em plataformas *Web*, chamado de Coinhive Rauchberger et al. (2018). De volta com a premissa do uso do poder computacional dos usuários para gerar lucros e fornecer uma alternativa à exibição de propagandas, o código de mineração do serviço Coinhive chegou a estar presente em cerca de 32.000 *websites* distintos em 2017 Krebs (2018). Desta vez, o foco da mineração deixou de ser a Bitcoin e voltou-se para a criptomoeda Monero. Um dos principais fatores que contribuiram para a escolha da Monero foi o seu algoritmo de PoW, CryptoNight, projetado para permitir que computadores com processadores de propósito geral sejam adequados para participar do processo de criação de blocos. Ao permitir que os processadores presentes em computadores comuns sejam capazes de calcular *hashes* de blocos de maneira mais eficiente do que *hardwares* especializados, o algoritmo CryptoNight torna viável a mineração de Monero através de códigos embutidos em páginas *Web*.

Além de servir como uma alternativa para a monetização de conteúdo, o serviço Coinhive despertou novamente o interesse de atacantes. A mineração de criptomoedas através de navegadores sem o consentimento dos usuários (*in-browser cryptojacking*) tornou-se novamente uma forma de ataque proeminente. Ataques de *cryptojacking* consistem na utilização dos recursos computacionais de uma ou mais vítimas, sem o seu consentimento, para a realização do processamento necessário à criação de blocos válidos em criptomoedas. No ano de 2017, o aumento na detecção de ataques de *in-browser cryptojacking* foi de 8.500% Mathur (2018). Em junho do ano de 2018, a plataforma Coinhive foi responsável por 1,18% dos blocos minerados no sistema Monero Rüth et al. (2018). Um dos fatores que contribuiram para o crescimento do número de ataques foi o conjunto de mecanismos de privacidade oferecidos pelo sistema Monero. Ao infectar páginas da *Web* com scripts de mineração maliciosos, um atacante é capaz de obter lucro ao receber pagamentos em criptomoedas pela validação de transações. Ao usar o sistema Monero os atacantes permanecem anônimos graças ao sistema de chaves de uso único.

Estima-se que 10 milhões de usuários sejam afetados por ataques de *in-browser cryptojacking* a cada

mês Hong et al. (2018). Páginas da *Web* nas quais os usuários permanecem por longos períodos de tempo como plataformas de *streaming* de vídeo, são alvos ideais para a injeção de scripts de *cryptojacking*, visto que o lucro obtido através de mineração é proporcional ao tempo que os usuários permanecem em uma página infectada.

Após reconhecer a importância de solicitar o consentimento dos usuários para a execução de scripts de mineração em *browsers*, a plataforma Coinhive lançou um serviço denominado AuthedMine. Esse serviço pergunta aos usuários se estes aceitam emprestar o seu poder computacional para a realização de cálculos antes de começar a computar *hashes*. No entanto, esse novo serviço gerou novas discussões à respeito da ética envolvida no uso do poder computacional dos usuários. A prática é controversa mesmo quando há consentimento, pois não fica claro se o usuários entendem o que estão consentindo ou o motivo pelo qual isso é necessário Eskandari et al. (2018). A Fig. 4 mostra a mensagem que solicita aos usuários o uso dos seus recursos computacionais. Embora, de um ponto de vista ético, o serviço AuthedMine seja uma alternativa mais apropriada do que os scripts de mineração padrão (sem consentimento), serviços que não requerem permissão continuam populares por causa do interesse de administradores maliciosos de páginas *Web* e atacantes.

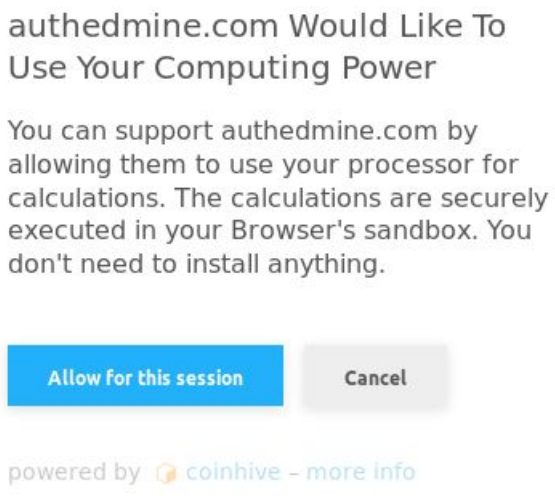


**Figura 4:** Janela de confirmação para execução do serviço AuthedMine.

A execução de um ataque de *in-browser cryptojacking* geralmente depende de dois componentes distintos, um código controlador e um código de mineração. A Fig. 5 ilustra o processo de comunicação necessário entre a máquina do usuário e os servidores do atacante para iniciar ao processo de mineração. A primeira etapa da comunicação ocorre quando um usuário visita uma página que executa um código de mineração. Ao comunicar-se pela primeira vez com a página, o navegador do usuário recebe e executa um código Javascript que inspeciona os recursos computacionais disponíveis em sua máquina. O script também verifica se o navegador da vítima suporta a execução de código WebAssembly (Wasm).

O formato de instruções Wasm é otimizado para permitir a execução de código em navegadores com um desempenho próximo ao de uma aplicação sendo executada nativamente na máquina *WebAssembly* (2018). Na próxima etapa o script comunica-se com um servidor externo definido pelo atacante, isto é, um servidor diferente do qual o usuário acessou. Se o navegador suporta a execução de código Wasm, o script do controlador faz o download de um código Wasm que é compilado na máquina da vítima, caso contrário, é realizado o download de instruções `asm.js`, um tipo de código JavaScript focado em desempenho.

Na última etapa, o código de mineração cria *threads* (processos leves) na máquina da vítima de acordo com a quantidade de recursos disponíveis, cria uma conexão com um serviço de *mining pool* e requisita tarefas de processamento. Ao terminar as tarefas recebidas, a máquina envia para o servidor da *mining pool* os *digests* computados e repete a última etapa até o encerramento da conexão.

Ataques de *in-browser cryptojacking*, quando configurados para utilizar apenas uma fração do poder de processamento das vítimas, são pouco intrusivos e dificilmente perceptíveis. Na intenção de prevení-los, foram criadas extensões de navegadores como NoCoin *NoCoin* (2018) e MinerBlock *MinerBlock* (2018) que mantêm listas de sites infectados e bloqueiam o acesso à esses sites. Porém, listas de bloqueio são ineficientes pois requerem constante manutenção para a adição de novas páginas, podem gerar falsos positivos e podem ser contornadas através de técnicas de evasão de detecção Segura (2018). Após a popularização das listas de bloqueio, os atacantes passaram a utilizar diversas técnicas de evasão:

($te_1$) **Servidores de retransmissão:** Ao configurar seu próprio servidor externo, um atacante é capaz de usá-lo como intermediário para a transmissão de dados até *mining pools*. O código malicioso comunica-se com o servidor do atacante, que é desconhecido e dificilmente será detectado através da análise de assinaturas maliciosas, e este comunica-se com *mining pools*, requisitando trabalho e enviando os *digests* computados.

($te_2$) **Ofuscação de código:** Alguns códigos de *cryptojacking* tentam ofuscar os comandos executados pelo script utilizando técnicas como a utilização de funções de codificação e decodificação de *strings*.

($te_3$) **Artifícios anti-depuração:** Códigos que utilizam artifícios anti-depuração verificam se estão sendo analisados através das ferramentas de depuração para desenvolvedores do navegador. Se estiver sendo analisado o código interrompe a sua execução.

($te_4$) **Ofuscação do protocolo de comunicação:** Para dificultar a sua detecção, alguns códigos codificam

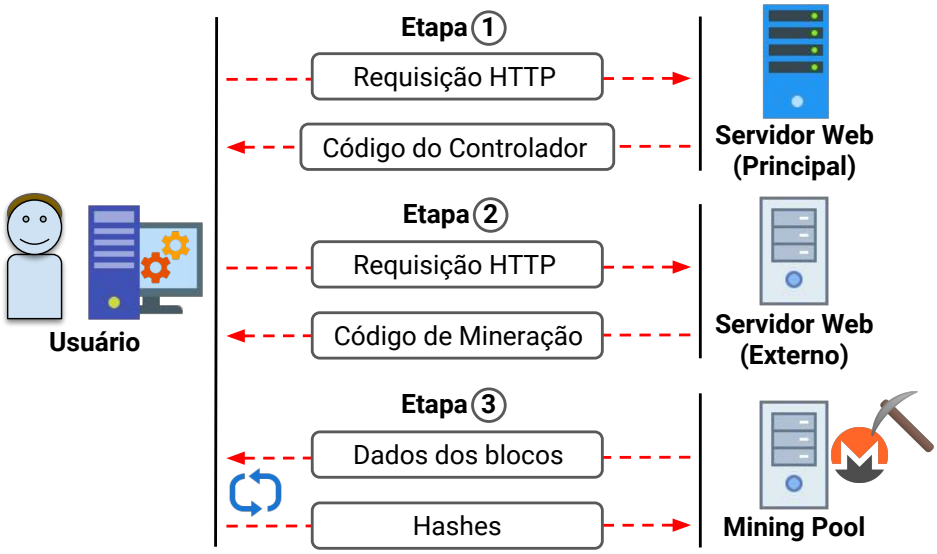


**Figura 5:** Diagrama ilustrando as etapas um ataque de *in-browser cryptojacking*.

ou cifram as mensagens de comunicação com *mining pools*, enviadas através de um protocolo chamado Stratum Slush Pool (2012).

($te_5$) **Ocultação de *payload*:** Alguns atacantes não inserem seus códigos maliciosos diretamente nas páginas sendo atacadas, optando por incluí-los dentro de suas próprias versões de arquivos contendo código de bibliotecas já conhecidas. No momento do carregamento das bibliotecas nas páginas *Web*, o código malicioso também é inicializado.

($te_6$) **Controle do uso do processador:** Se um código malicioso utiliza todos os recursos computacionais da máquina da vítima, o dispositivo pode sofrer travamentos e lentidão, permitindo a identificação da aplicação de mineração. Alguns scripts limitam a utilização dos recursos da máquina da vítima, muitas vezes passando despercebidos em meio à outras aplicações que estão sendo executadas no momento, e.g. navegador e gerenciador de janelas.

Estima-se que a utilização de plataformas de mineração como Coinhive não gere um retorno monetário tão expressivo quanto a exibição de propagandas em páginas da *Web* Saad et al. (2018). A menos que as páginas utilizadoras de mineração em navegadores atraiam muitos usuários e ofereçam conteúdos que os mantenham no *site* por longos períodos de tempo, a utilização desse tipo de serviço não é recomendado.

Outro fator que influencia no retorno das páginas que contêm mineradores é a quantidade de recursos utilizados durante o processo. Utilizar todo o poder computacional dos usuários pode prejudicar as suas experiências ao navegar na própria página, impactando negativamente na sua popularidade. No entanto, atacantes podem beneficiar-se economicamente de ataques de *in-browser cryptojacking* ao infectar páginas que oferecem entretenimento e conteúdo envolvendo pirataria, pois usuários passam mais tempo procurando por recursos nessa categoria de páginas Konoth et al. (2018), Hong et al. (2018).

## 4 A criptomoeda 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 de sucesso e utilizada em large escala Nakamoto (2008). Anos após seu lançamento, ainda permanece sendo a mais utilizada, possuindo um valor total de mercado de US$ 101 bilhões[1] em maio de 2019. A Bitcoin baseia-se em material de pesquisas anteriores, como o esquema de PoW para controlar a criação de blocos válidos no *blockchain*, esquemas de assinaturas digitais para garantir 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 *blockchain* público, isto é, cada participante da rede pode optar por obter uma cópia desse registro. Usuários que não desejam realizar o download dos dados do *blockchain* da criptomoeda podem acessá-los através de clientes leves, do inglês *Light-Clients*. Clientes leves são programas que permitem aos usuários acessar dados através de uma conexão com um nó remoto confiável

[1] https://coinmarketcap.com/

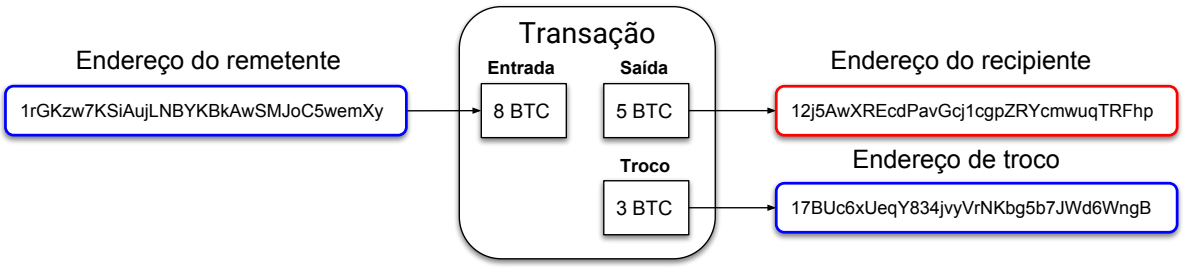


**Figura 6:** Transação de Bitcoins contendo uma única entrada.

que mantém uma cópia do *blockchain*.

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 podem 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 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 and 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 Fig. 6 ilustra o procedimento de uma transação de Bitcoins contendo uma única entrada. O endereço do remetente é a chave pública que corresponde ao endereço da carteira de Bitcoins do emissor do pagamento. O endereço do recipiente é o endereço da carteira do receptor do pagamento. O endereço de troco deve ser definido pelo emissor do pagamento para que ele receba o troco da operação, caso a chave de entrada utilizada exceda o valor do pagamento.

Só podem ser utilizadas como entradas de transações as chaves que foram geradas como saída em uma transação anterior. Endereços de saída de transações são chaves recebidas pelos usuários que recebem as criptomoedas. 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. Uma carteira é capaz de gerenciar diferentes endereços de um mesmo usuário.

Um único bloco é capaz de armazenar milhares de transações, desde que os dados de todas elas somados não ultrapassem o tamanho de 1 MB. A Fig. 7 resume os dados contidos um bloco do sistema Bitcoin. Os blocos criados para armazenar as transações no *blockchain* possuem os seguintes campos:

($c_1$) Número mágico: Valor utilizado para identificar o tipo de estrutura contida nos dados. Neste caso, um bloco. Este valor é específico do protocolo Bitcoin.
($c_2$) Tamanho do bloco: Especifica o tamanho em *bytes* dos dados contidos no bloco.
($c_3$) Cabeçalho do bloco: Contém dados que identificam o bloco atual.
($c_4$) Versão: Especifica a versão do sistema no momento da criação do bloco.
($c_5$) *Timestamp*: Representa o momento no tempo em que o bloco foi criado.
($c_6$) 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.
($c_7$) *Nonce*: Campo cujo valor deve ser modificado até que o *digest* resultante do bloco atenda as exigências do sistema.
($c_8$) Dificuldade: Especifica o número de *bits* 0 necessários à esquerda do *digest* do bloco para que ele atenda às exigências do sistema.

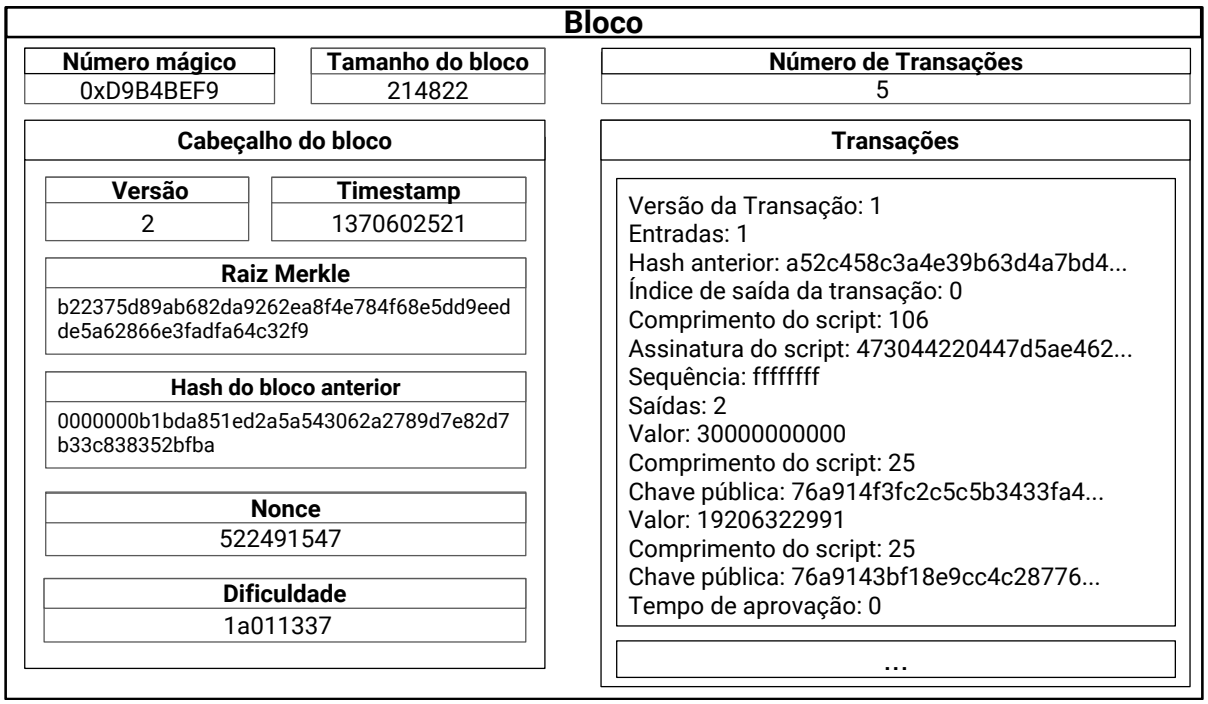


**Figura 7:** Estrutura de um bloco do *blockchain* do sistema Bitcoin.

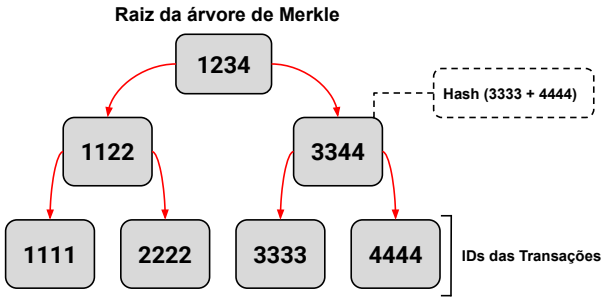


**Figura 8:** Representação de uma Árvore de Merkle.

($c_9$) Número de transações: Contém o número de transações presentes no bloco atual.
($c_{10}$) Transações: Contém os dados de cada uma das transações contidas no bloco.

É interessante ressaltar a importância do campo que armazena a raiz da Árvore de Merkle. A Fig. 8 ilustra a estrutura de uma Árvore de Merkle. Os valores contidos nos nós da árvore foram simplificados para fins didáticos. Em uma estrutura real, as informações armazenadas dentro dos nós de uma Árvore de Merkle são os *digests* gerados pela função SHA-256. A raiz de Merkle funciona como um identificador para as transações que estão contidas em um bloco.

Imagine que para cada bloco criado seja computado um resumo (ou *digest*) dos identificadores de todas as transações nele contidas. Para verificar se todas as transações contidas no bloco foram incluídas no *digest* (para verificar a integridade dos dados), seria necessário reunir os identificadores de todas as transações e computar o *digest* novamente. Em uma Árvore de Merkle, cada nó da estrutura armazena o *digest* dos dados contidos em seus nós filhos. Os nós folha da árvore armazenam os dados originais, no caso da Bitcoin, os identificadores das transações contidas no bloco. Com esta estrutura é possível verificar a integridade das transações presentes em um bloco através dos *digests* que resumem os dados, acelerando o processo.

Considerando a estrutura apresentada na Fig. 8, suponha que um usuário foi informado de que recebeu Bitcoins através da transação com o identificador 4444, em um bloco cuja raiz de Merkle é 1234. O receptor deseja verificar se esta transação está de fato presente no bloco. O usuário já conhece o valor da raiz da árvore e do identificador de sua transação. Para realizar a verificação, serão necessários também os valores 3333 e 1122. Primeiro, é computado o *digest* dos valores 3333 e 4444 e é gerado o valor 3344, que resume os dados das folhas do lado direito da árvore. O valor que resume os dados das folhas do lado esquerdo, 1122, já é conhecido. Na sequência é gerado um *digest* a partir dos valores que resumem os dois lados da árvore, 1122 e 3344, gerando a raiz da Árvore de Merkle, que é igual a 1234 e resume ambos os lados da árvore. Se a raiz de Merkle resultante é igual ao valor contido dentro do bloco no *blockchain*, isto significa que a transação 4444 está contida nas transações do bloco examinado.

Em blocos com poucas transações, a diferença entre o número de dados necessários para realizar a verificação e o número de dados no bloco é pequena. Porém, conforme o número de transações aumenta para cen-

tenas ou milhares, o número de operações com uma Árvore de Merkle torna-se muito menor do que o número de transações necessárias para verificar o *digest*, tornando evidente o benefício da utilização de Árvores de Merkle.

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 checar se os pagadores 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 Fig. 9 ilustra o processo de criação de uma assinatura digital. João precisa provar que possui um endereço para enviar dinheiro através dele. Para isso, ele cria uma mensagem contendo os dados da transação que deseja realizar. O segundo passo é criar uma assinatura digital, que servirá para provar que 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.

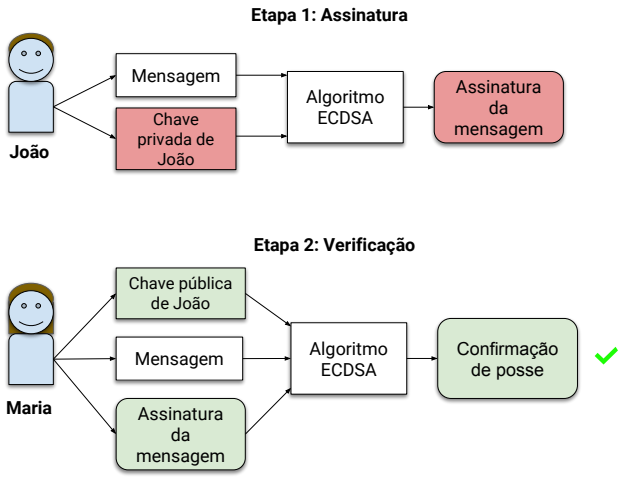


**Figura 9:** Processo de assinatura de uma transação.

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 à 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 à 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 explícito ao remetente. Os mineradores acumulam transações e as organizam em blocos antes de iniciarem o processo de tentativa de criação de um bloco válido. A dificuldade de criação dos blocos é regulada pelo sistema e é alterada com base no tempo que foi necessário para a criação dos últimos 2.016 blocos. O ajuste é realizado com a intenção de manter o tempo necessário para adicionar um bloco a cada dez minutos, para que o tempo de confirmação das transações seja razoável. Levando em consideração o tempo ideal de criação de um bloco, 10 minutos, 2.016 blocos devem ser criados em exatamente duas semanas. Se o tempo necessário para a criação dos últimos 2.016 blocos exceder duas semanas, a dificuldade da criação dos blocos é reduzida, se o tempo for inferior a duas semanas, a dificuldade é elevada. O sistema ajusta a dificuldade da criação de blocos válidos de acordo com a Eq. (1).

$$D = D_{anterior} \times \frac{T}{2016 \times 10min} \tag{1}$$

onde:

$D$ = Dificuldade da resolução do problema de criação de um bloco válido.

$T$ = Tempo ocorrido desde a última mudança de dificuldade em minutos.

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 foram confirmadas por ele. 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.

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 buscam solucionar problemas existentes nos sistemas atuais, como a demora das confirmações e a rastreabilidade das transações.

## 5 Em busca da privacidade no ecosistema das criptomoeadas

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, como 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 a vinculação de pseudônimos com as suas identidades reais. Esta associação pode ser feita ao relacionar nomes e informações de contas dos usuários com as chaves públicas compartilhadas em fóruns e páginas da *Web*.

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. A partir 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.

## 6 A criptomoeda 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 12ª posição em termos de valor de mercado se comparada a todas as criptomoedas[2].

Monero utiliza um protocolo chamado de CryptoNote Van Saberhagen (2013). Esse protocolo é utilizado também em outras criptomoedas que focam na privacidade dos usuários, tais como Bytecoin Van Saberhagen (2013) e DashCoin Duffield and Diaz (2014). 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 utilizado sem precauções, são criadas brechas de segurança que permitem ataques à privacidade das transações. 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 demonstrar que elas foram originadas de um mesmo usuário.

Estas duas propriedades são baseadas nas funcionalidades de chaves de uso único e *ring signatures*, oferecidas pelo protocolo CryptoNote. Além destas, o sistema Monero oferece outras garantias de segurança e privacidade: as *Ring Confidential Transaction*s (sRingCTs), e o protocolo de roteamento Kovri. 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* verificando as saídas de transações com a sua chave privada de visualização. Dessa forma, Maria é capaz de identificar os endereços de uso único destinados 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 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

[2] https://coinmarketcap.com/

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. As imagens das chaves fornecem provas de que a chave sendo utilizada para o pagamento em uma entrada de transação não foi utilizada anteriormente. Isto é feito sem revelar qualquer informação sobre a chave real. 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. A Fig. 10 ilustra algumas transações do sistema Monero.

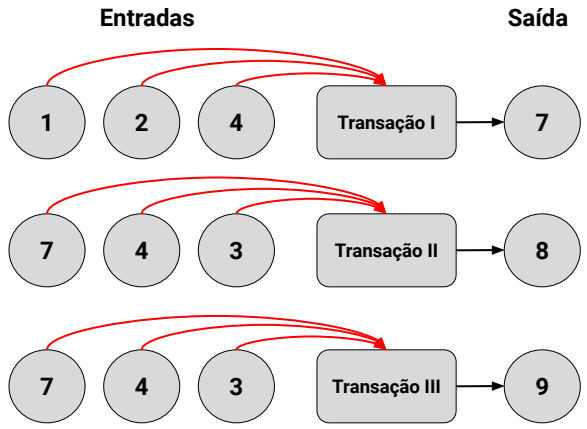


**Figura 10:** Representação de transações do sistema Monero.

Os círculo representam as chave de entrada e saída de uma transação. A chave utilizada na transação pode ser qualquer uma das entradas, portanto, existem várias combinações válidas. Por exemplo, as chaves utilizadas nas transações I, II e III respectivamente podem ser {1, 4, 3}, {1, 7, 4}, {2, 3, 4}, {4, 7, 3} dentre outras combinações. A chave número 4, apesar de estar presente nos conjuntos de entrada de todas as transações, pode ainda não ter sido utilizada. Desse modo, a única maneira de adivinhar a chave utilizada como entrada seria ao tentar adivinhar a chave correta. No exemplo da Fig. 10 a chance de um atacante adivinhar a chave correta em qualquer uma das transações é de $\frac{1}{3}$. Como o *blockchain* armazena milhões de transações, tentar adivinhar a entrada de cada uma das transações é inviável.

Para aumentar ainda mais o nível de privacidade nas transações de Monero, foi criado o protocolo RingCT. A Fig. 11 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. Os valores de denominações variam de $10^{-12}$ à $10^{6}$ e são identificadas por um prefixo composto pelo nome da unidade (pico à mega) e o sufixo "nero". 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.

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 et al. (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 *Internet Protocol* (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 criptografadas em camadas, e as únicas informações visíveis são as instruções de encaminhamento das mensagens ao longo do trajeto até o seu destino.

O sistema Monero utiliza o algoritmo CryptoNight no seu processo de validação de blocos. O algoritmo CryptoNight baseia-se em acessos aleatórios à memó-

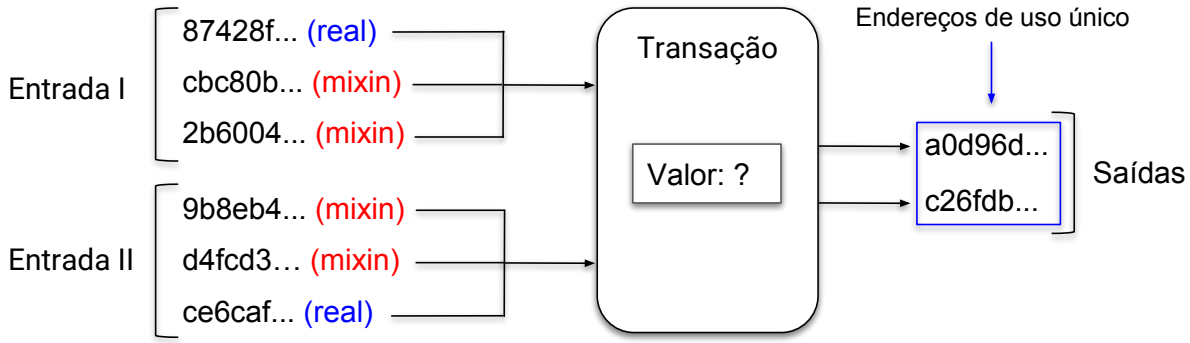


**Figura 11:** Representação de transação RingCT com duas entradas.

ria e possui ênfase na latência de acesso. Além disso, o algoritmo requer cerca de 2 Megabytes de armazenamento por instância, o que significa que os dados cabem em uma memória *cache* L3 em processadores modernos CryptoNote (2015). A velocidade das memórias presentes em GPUs e ASICs é muito inferior à velocidade das memórias *cache* de um processador e isso diminui a eficiência desses dispositivos ao executar algoritmos como o CryptoNight. Essas características fazem com que processadores comuns sejam os dispositivos ideais para computar *hashes* utilizando o algoritmo CryptoNight, permitindo que usuários participem de maneira competitiva do processo de mineração da Monero sem a necessidade de adquirir *hardware* especializado.

O conjunto de algoritmos e protocolos utilizados pela criptomoeda Monero contribui para o fornecimento de segurança e privacidade para os usuários. Entretanto, assim como em qualquer sistema que dependa da sua capacidade de proteger os usuários, é necessário que as técnicas empregadas no sistema Monero sejam recorrentemente analisadas em profundidade, a fim de identificar possíveis falhas e contribuir com o desenvolvimento de criptomoedas privadas e descentralizadas.

## 7 Conclusão

Este tutorial apresentou uma introdução às tecnologias de *blockchain* e criptomoedas, com especial atenção à segurança e privacidade dos dados. As discussões foram acompanhadas de ilustrações e exemplos com o objetivo de auxiliar a compreensão do leitor. Foram discutidos os detalhes de funcionamento de duas criptomoedas, a Bitcoin, que é a moeda digital mais utilizada atualmente, e a Monero, cujo foco é a privacidade dos usuários.

A principal contribuição do tutorial foi a disseminação do conhecimento relativo a estes temas emergentes, atuais e importantes no mundo da tecnologia. Muitos especialistas consideram que a tecnologia de *blockchain* e criptomoedas estão iniciando e potencializando a maior revolução tecnológica depois da Internet. De fato, nos últimos anos, a gama de oportunidades, tanto na academia quanto no mercado, cresceu de maneira exponencial na pesquisa e desenvolvimento de novas soluções baseadas em *blockchain* para resolver problemas reais nos mais diversos domínios.

## Referências

Adkisson, J. (2018). Why bitcoin is so volatile. Disponível em http://tiny.cc/8fp35y.

Biryukov, A., Khovratovich, D. and Pustogarov, I. (2014). Deanonymisation of clients in bitcoin p2p network, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 15–29. https://doi.org/10.1145/2660267.2660379.

Blattberg, E. (2014). New jersey slaps mit bitcoin hackers with subpoena — and they're fighting back. Disponível em https://tinyurl.com/y3owapf4.

Chen, W., Xu, Z., Shi, S., Zhao, Y. and Zhao, J. (2018). A survey of blockchain applications in different domains, *Proceedings of the 2018 International Conference on Blockchain Technology and Application*, ICBTA 2018, ACM, New York, NY, USA, pp. 17–21. http://doi.acm.org/10.1145/3301403.3301407.

Christidis, K. and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things, *Ieee Access* **4**: 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339.

Cointopper (2018). Difference between asic, gpu and cpu mining. Disponível em http://tiny.cc/ljp35y.

CryptoNote (2015). Cryptonote technology. Disponível em https://cryptonote.org/inside/.

Duffield, E. and Diaz, D. (2014). Dash: A privacycentric cryptocurrency. Disponível em http://blockchainlab.com/pdf/Dash-WhitepaperV1.pdf.

Eskandari, S., Leoutsarakos, A., Mursch, T. and Clark, J. (2018). A first look at browser-based cryptojacking, *arXiv preprint arXiv:1803.02887* . Disponível em https://arxiv.org/abs/1803.02887.

Hong, G., Yang, Z., Yang, S., Zhang, L., Nan, Y., Zhang, Z., Yang, M., Zhang, Y., Qian, Z. and Duan,

H. (2018). How you get shot in the back: A systematical study about cryptojacking in the real world, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 1701–1713. https://doi.org/10.1145/3243734.3243840.

Hoy, M. B. (2017). An introduction to the blockchain and its implications for libraries and medicine, *Medical reference services quarterly* **36**(3): 273–279. http://doi.org/10.1080/02763869.2017.1332261.

IEEE (2017). Special Report on Blockchain World, *IEEE Spectrum* **10**. Disponível em https://spectrum.ieee.org/static/special-report-blockchain-world.

Jefferys, K. (2018). The problem with asics. Disponível em http://tiny.cc/kqu35y.

Kar, I. (2016). Estonian citizens will soon have the world's most hack-proof health-care records. Disponível em http://tiny.cc/jnp35y.

Kelly, J. and Williams, A. (2016). Forty big banks test blockchain-based bond trading system. Disponível em http://tiny.cc/2pp35y.

Konoth, R. K., Vineti, E., Moonsamy, V., Lindorfer, M., Kruegel, C., Bos, H. and Vigna, G. (2018). Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 1714–1730. https://doi.org/10.1145/3243734.3243858.

Korpela, K., Hallikas, J. and Dahlberg, T. (2017). Digital supply chain transformation toward blockchain integration, *Proceedings of the 50th Hawaii international conference on system sciences*. doi.org/10.24251/HICSS.2017.506.

Krebs, B. (2018). Who and what is coinhive? https://bit.ly/2DV2OGJ.

Kumar, A., Fischer, C., Tople, S. and Saxena, P. (2017). A traceability analysis of monero's blockchain, *European Symposium on Research in Computer Security*, Springer, pp. 153–173. Disponível em https://eprint.iacr.org/2017/338.pdf.

Lacey, S. (2016). The energy blockchain: How bitcoin could be a catalyst for the distributed grid, *GreenTech Media* **26**. Disponível em http://tiny.cc/6up35y.

Marinoff, N. (2018). South korea is trialing blockchain voting — here's what that means. Disponível em http://tiny.cc/swp35y.

Mathur, N. (2018). Cybersecurity: Cryptojacking attacks exploded by 8,500% in 2017, says report. Disponível em https://tinyurl.com/y84alobt.

medfar87 (2018). Cryptocurrency growth & adoption statistics. Disponível em http://tiny.cc/oxp35y.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M. and Savage, S. (2013). A fistful of bitcoins: characterizing payments among men with no names, *Proceedings of the 2013 conference on Internet measurement conference*, ACM, pp. 127–140. https://doi.org/10.1145/2504730.2504747.

*MinerBlock* (2018). Disponível em https://github.com/xd4rker/MinerBlock.

Mizrahi, A. (2015). A blockchain-based property ownership recording system. Disponível em http://tiny.cc/01p35y.

Monero's Team (2019). MONERO. Disponível em https://ww.getmonero.org.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em https://bitcoin.org/bitcoin.pdf.

NIST (2018). Hash functions. Disponível em https://csrc.nist.gov/projects/hash-functions.

*NoCoin* (2018). Disponível em https://github.com/keraf/NoCoin.

Olnes, S., Ubacht, J. and Janssen, M. (2017). Blockchain in government: Benefits and implications of distributed ledger technology for information sharing, *Government Information Quarterly* **34**(3): 355 – 364. https://doi.org/10.1016/j.giq.2017.09.007.

Popov, S. (2014). The tangle. Disponível em https://bit.ly/2YUFTqc.

Rauchberger, J., Schrittwieser, S., Dam, T., Luh, R., Buhov, D., Pötzelsberger, G. and Kim, H. (2018). The other side of the coin: A framework for detecting and analyzing web-based cryptocurrency mining campaigns, *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ACM, p. 18. https://doi.org/10.1145/3230833.3230869.

Ripple (2019). RippleNet. Disponível em https://ripple.com.

Rüth, J., Zimmermann, T., Wolsing, K. and Hohlfeld, O. (2018). Digging into browser-based crypto mining, *Proceedings of the Internet Measurement Conference 2018*, ACM, pp. 70–76. https://doi.org/10.1145/3278532.3278539.

Saad, M., Khormali, A. and Mohaisen, A. (2018). End-to-end analysis of in-browser cryptojacking, *arXiv preprint arXiv:1809.02152* . Disponível em https://arxiv.org/abs/1809.02152.

Schwartz, D., Youngs, N., Britto, A. et al. (2014). The ripple protocol consensus algorithm. Disponível em https://arxiv.org/abs/1802.07242.

Segura, J. (2018). Malicious cryptomining and the blacklist conundrum. Disponível em http://tiny.cc/2dq35y.

Slush Pool (2012). Stratum mining protocol. Disponível em https://bit.ly/2Lv5QdY.

Torpey, K. (2018). Study suggests 25 percent of bitcoin users are associated with illegal activity. Disponível em https://tinyurl.com/ycw3svef.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies, *IEEE Communications Surveys & Tutorials* **18**(3): 2084–2123. https://doi.org/10.1109/COMST.2016.2535718.

Van Saberhagen, N. (2013). Cryptonote v 2. 0. Disponível em http://bit.do/eRrpv.

Walport, M. (2016). Distributed ledger technology: beyond block chain, *Technical report*, U.K. Government Office Sci. Disponível em https://bit.ly/1KwRayJ.

*WebAssembly* (2018). Disponível em https://webassembly.org/.

Williams, S. (2017). Meet the newest cryptocurrency trend: Privacy coins. Disponível em http://bit.do/eRro5.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X. and Wang, H. (2018). Blockchain challenges and opportunities: a survey, *International Journal of Web and Grid Services* **14**(4): 352–375. http://doi.org/10.1504/IJWGS.2018.095647.