Capítulo 4

Uso de Blockchain para Privacidade e Segurança em Internet das Coisas

Vanessa R. L. Chicarino, Emanuel Ferreira Jesus, Célio V. N. de Albuquerque, Antônio A. de A. Rocha

Instituto de Computação (IC), Universidade Federal Fluminense (UFF), Rio de Janeiro.

Abstract

The Internet of Things (IoT) are increasingly a reality today, nevertheless some key challenges still need to be given special attention so that IoT solutions further support the growing demand for connected devices and the services offered. Due to the potential relevance and sensitivity of services, IoT solutions should address the security and privacy concerns surrounding these devices and the data they collect, generate, and process. Recently, the Blockchain technology has gained a lot of attention in IoT solutions. Its main usage scenarios are in the financial domain, where Blockchain creates a world of promising applications and can be leveraged to solve security and privacy issues. However, this emerging technology has a great potential in the most diverse technological areas, and can significantly help achieve the Internet of Things view in different aspects, increasing the capacity of decentralization, facilitating interactions, enabling new transaction models and allowing autonomous coordination of the devices. The goal of this short, is to provide the concepts about the structure and operation of Blockchain and, mainly, to analyze how the use of this technology can be used to provide security and privacy in IoT.

Resumo

Internet das Coisas (IoT) é cada vez mais uma realidade atual, embora alguns desafios-chave carecem ainda de uma atenção especial para que soluções de IoT dêem um suporte ainda maior para a crescente demanda por dispositivos conectados e os serviços oferecidos. Devido a possível relevância e sensibilidade dos serviços, as soluções de IoT devem abordar as preocupações de segurança e privacidade em torno desses dispositivos e dos dados que eles coletam, geram e processam. Recentemente, a tecnologia Blockchain ganhou muita atenção em soluções de Internet das Coisas. Seus principais cenários de uso estão no domínio financeiro, onde a Blockchain cria um mundo de promissoras aplicações e pode ser aproveitado para resolver os problemas de segurança e privacidade.

Porém, essa emergente tecnologia tem um grande potencial nas mais diversas áreas tecnológicas, e pode ajudar significativamente a alcançar a visão da Internet das Coisas em diferentes aspectos, aumentando a capacidade de descentralização, facilitando interações, possibilitando novos modelos de transações e permitindo a coordenação autônoma dos dispositivos. Este minicurso, tem como objetivo fornecer conceitos sobre a estrutura e funcionamento da Blockchain e, principalmente, de analisar como o uso desta tecnologia pode ser usada para prover segurança e privacidade em IoT.

4.1. Introdução e motivação

Internet das Coisas (IoT) e Blockchain são consideradas tecnologias emergentes na atualidade. Ao mesmo tempo que transformam conceitos e criam novas possibilidades, cada uma em seus respectivos cenários, existe a oportunidade de criar aplicações que podem compartilhar as características intrínsecas de ambos, explorando como a IoT pode se beneficiar da natureza descentralizada da Blockchain.

Blockchain (também conhecido como "o protocolo da confiança") é um conceito que visa a descentralização como medida de segurança. São bases de registros e dados distribuídos e compartilhados que possuem a função de criar um índice global para todas as transações que ocorrem em uma determinada rede. Funciona como um livro-razão, só que de forma pública, compartilhada e universal, que cria consenso e confiança na comunicação direta entre duas partes, ou seja, sem o intermédio de terceiros. Está constantemente crescendo à medida que novos blocos completos são adicionados a ela por um novo conjunto de registros. A cadeia de blocos também pode ser usada para comunicações em cadeia de fornecimento, contratos inteligentes, gerenciamento de identidade digital e em uma série de outras aplicações [Pilkington, 2016].

A Internet das Coisas é um termo abrangente referente aos esforços em curso para conectar uma grande variedade de coisas físicas às redes de comunicação. Atualmente não apenas computadores convencionais estão conectados à Internet, como também uma grande heterogeneidade de equipamentos, tais como TVs, laptops, geladeira, fogão, eletrodomésticos, automóveis, smartphones, entre outros. Nesse novo cenário, a pluralidade é crescente e previsões indicam que mais de 50 bilhões de dispositivos estarão conectados até 2020 [Evans, 2011]. Dentro do domínio IoT existem vários tipos de aplicações, como por exemplo: cidades inteligentes (smart cities); saúde (smart healthcare); casas inteligentes (smart home) entre outras.

Ao mesmo tempo que a IoT poderá nos proporcionar benefícios valiosos, ela também aumentará os nossos riscos de exposição a diversas ameaças de segurança e privacidade, algumas dessas ameaças são novas e bem particulares desta tecnologia. Antes do advento da Internet das Coisas, a maioria das ameaças de segurança estavam relacionadas ao vazamento de informações e a negação de serviço. Com a IoT, as ameaças à segurança vão muito além do roubo de informações ou da impossibilidade de uso de determinados serviços. Essas ameaças podem agora estar potencialmente relacionadas com as vidas reais, inclusive de segurança física.

Soluções de segurança e privacidade devem ser implementadas conforme as características de dispositivos IoT heterogêneos. Há uma demanda por soluções de segurança que sejam capazes de fornecer níveis de segurança equivalentes para vários tipos de dis-

positivos. IoT trouxe consigo um aumento da quantidade de informações pessoais que serão entregues e compartilhadas entre os dispositivos conectados. Assim, embora não seja uma demanda nova ou exclusiva deste novo cenário, a privacidade é um elemento importante que, em virtude de suas especificidades, demanda mecanismos capazes de auditar e controlar acesso nestes ambientes.

Neste contexto que Blockchain também se insere, pois essa tecnologia pode ser usada para Autenticar, Autorizar e Auditar os dados gerados pelos dispositivos. Além disso, em virtude de sua natureza descentralizada, elimina a necessidade de confiança em terceiros e não possui um ponto único de falha.

Este minicurso tem por objetivo familiarizar novos interessados, bem como atualizar os leitores que possuem algum conhecimento anterior, a esta nova e promissora tecnologia chamada Blockchain. Isso inclui as recentes aplicações em segurança e privacidade, e como o seu uso pode alavancar a IoT. A abordagem oferecida neste minicurso será orientada a estudos de caso em que o uso de Blockchain já é uma realidade ou que tem sido movido esforços nesta direção. Estudos de caso, como por exemplo o uso de Blockchain para prover controle de acesso e anonimidade em sistemas de armazenamento de dados distribuídos ou o seu uso para controle de operações financeiras, nortearão este minicurso.

O minicurso está estruturado em cinco seções, sendo esta a primeira. A Seção 4.2 irá apresentar os fundamentos teóricos básicos para a compreensão da solução proposta, como os princípios fundamentais de segurança e os conceitos de IoT; criptografia e funções hash e redes peer-to-peer (P2P). A Seção 4.3 apresentará todos os mecanismos de funcionamento da tecnologia Blockchain e descreverá os Contratos inteligentes, uma das aplicações da Blockchain. A Seção 4.4 descreve alguns casos de uso do Blockchain para prover segurança e privacidade em IoT e apresenta alguns casos reais de ataques a rede Blockchain, apresentando os resultados e análises de simulações realizadas. Finalmente, a Seção 4.5 apresenta as considerações finais e questões em aberto, apresentando referências para que o leitor possa complementar seus estudos.

4.2. Fundamentação teórica básica

Esta seção visa ambientar os leitores com as informações mais básicas para entender o que é a Blockchain. Para isso, serão apresentados: os princípios de segurança (Confidencialidade, Integridade, Disponibilidade, Privacidade, Auditoria, Autenticação e Não Repúdio) relacionados à Blockchain, recentes pesquisas a respeito de Internet das Coisas, abordando as classificações e taxonomias propostas em IoT.

O Bitcoin [Bitcoin, 2009] é um conjunto de conceitos e tecnologias que formam a base de um ecossistema de dinheiro digital. Seus usuários comunicam-se através da Internet utilizando uma rede Par-a-Par (ou P2P, do termo em inglês *Peer-to-Peer*) própria, mas outras formas de rede também podem ser usadas. Sua implementação está disponível como software de código aberto, e pode ser executada em diversos tipos de dispositivos, o que torna a tecnologia de fácil acesso. Será explicado o funcionamento desta rede P2P e os conceitos de funções hash e criptografia de chaves públicas utilizados.

4.2.1. Internet das Coisas

A Internet das Coisas (ou IoT, do termo em inglês *Internet of Things*) abrange o processamento de dados e a comunicação entre dispositivos de plataformas e capacidades diferentes de forma autonômica, sem intervenção humana. Nas últimas décadas esse termo despontou como uma evolução da internet e um novo paradigma tecnológico, social, cultural e digital.

A IoT é considerada uma extensão da internet atual, pois proporciona aos objetos do dia-a-dia (eletrodomésticos, meios de transporte e até acessórios, como por exemplo, óculos e relógios) com capacidade computacional e de comunicação a se conectarem a Internet. A conexão com a rede mundial de computadores viabilizará o controle remoto dos objetos e permitirá que os próprios objetos sejam acessados como provedores de serviços, tornando-os objetos inteligentes ou *smart objects*. Os objetos inteligentes possuem capacidade de comunicação e processamento aliados a sensores.

O primeiro dispositivo IoT foi apresentado em 1990 na *INTEROP '89 Conference* por John Romkey que criou uma torradeira que poderia ser ligada e desligada pela Internet, conectando a torradeira a um computador com rede TCP / IP. Em setembro de 1999, Kevin Ashton [Ashton, 2011], cofundador e diretor executivo do Auto - ID Center, proferiu uma palestra para a Procter & Gamble, apresentando a ideia de utilizar etiquetas eletrônicas nos produtos da empresa, para facilitar a logística da cadeia de produção, através de identificadores de rádio frequência (RFID, em inglês), para chamar a atenção dos executivos, ele colocou no título da apresentação a expressão *"Internet of Things"*, sendo considerado o criador desse termo, ao descrever que os objetos do mundo físico poderiam se conectar à internet, criando um mundo mais inteligente.

A partir de 2005, a discussão sobre a Internet das Coisas se generalizou, começou a ganhar a atenção dos governos e aparecer relacionada a questões de privacidade e segurança de dados. Foi neste ano que a Internet das Coisas se tornou a pauta do *International Telecommunication Union* (ITU), agência das Nações Unidas para as tecnologias da informação e da comunicação, que publica anualmente um relatório sobre tecnologias emergentes. Assim, depois da banda larga e da internet móvel, a Internet das Coisas ganhou a atenção do órgão e passou a figurar como o "próximo passo da tecnologia em comunicações *'always on'*, que prometem um mundo de dispositivos interconectados em rede" [Peña-López et al., 2005].

Entre os anos de 2008 e 2010, devido ao amadurecimento das Redes de Sensores Sem Fio (RSSF) (do inglês *Wireless Sensor Networks* - WSN), que trazem avanços na automação residencial e industrial, e técnicas para explorar as diferentes limitações dos dispositivos, como por exemplo, memória, energia, escalabilidade e robustez da rede, o termo Internet das Coisas ganhou popularidade rapidamente. Nesse período foi publicado o livro *The Internet of Things* por Rob Van Kranemburg, que aborda esse termo sob um novo paradigma em que os objetos produzem informação. Esse livro é uma das grandes referências teóricas sobre Internet das Coisas [Cui, 2016]. Em 2011 o termo Internet das Coisas foi adicionado, como tecnologia emergente, ao *Gartner Hype Cycle* [Fenn and LeHong, 2011], que fornece uma representação gráfica da maturidade e adoção de tecnologias e aplicativos e fornece uma visão de como uma tecnologia ou aplicação irá evoluir ao longo do tempo. A Figura 4.1 apresenta o gráfico deste ano, onde as

Plataformas IoT permanecem como tecnologia promissora.

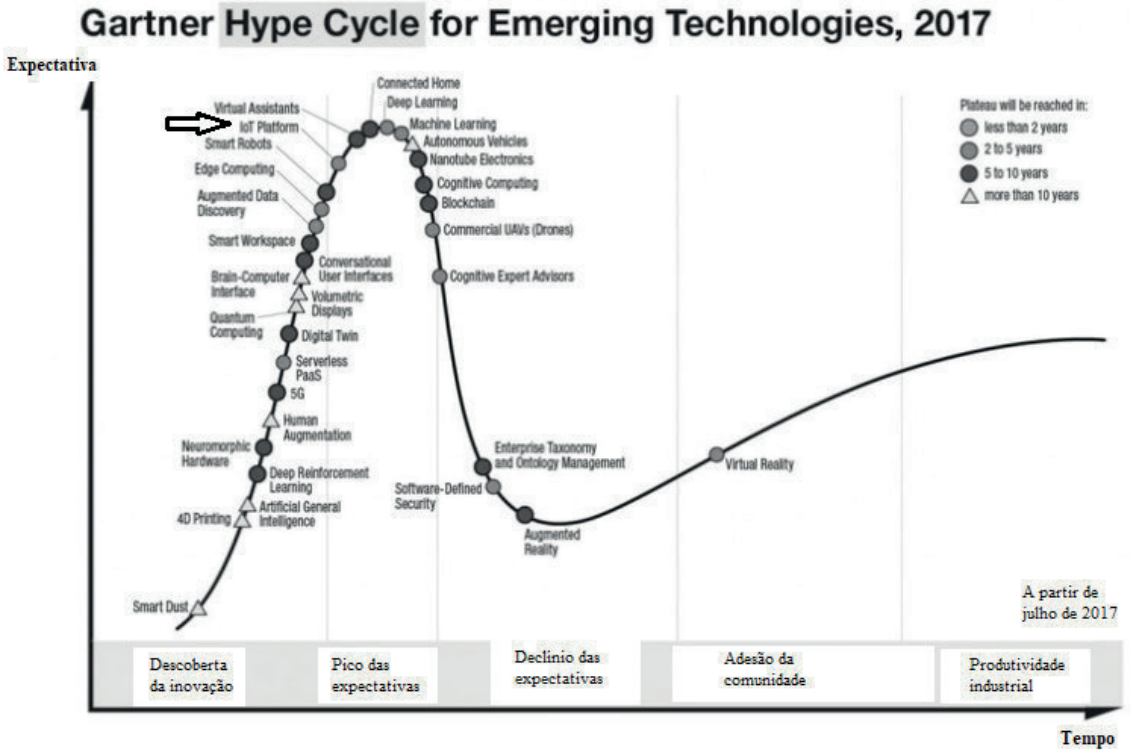

Figura 4.1. Gartner Hype Cycle para Tecnologias Emergentes Fonte: https://http://www.gartner.com acessado em: 10/09/2017

Atualmente não há um único conceito de Internet das Coisas definido na literatura, porém vários autores e instituições contribuíram para a construção da visão de IoT. Uma das referências mais citadas em [Atzori et al., 2010], o autor descreve IoT como sendo uma variedade de coisas ou objetos - como tags de identificação de radiofrequência (RFID), sensores, atuadores, telefones celulares, entre outros em torno de nós que interagem uns com os outros e cooperam com seus vizinhos, através de esquemas de endereçamento únicos, com a finalidade de alcançar metas comuns, sendo o resultado da convergência de três paradigmas: orientado para a internet (middleware), orientado a coisas (sensores e atuadores) e orientado à semântica (a representação e o armazenamento das informações trocadas), porém a utilidade do IoT pode ser desencadeada somente em um domínio de aplicação onde os três paradigmas se cruzam.

Algumas instituições relevantes enfatizaram o conceito de que o IoT deve se concentrar principalmente nas "Coisas"e que o caminho para sua implantação total deve começar a partir do aumento na inteligência das coisas. Algumas definições na literatura, derivam dessa visão, uma delas, proposta pelo Grupo de pesquisa europeu em IoT (European Research Projects on the Internet of Things - IERC) apresenta IoT como "Uma infra-estrutura de rede global e dinâmica com capacidades de autoconfiguração baseadas em protocolos de comunicação padronizados e interoperáveis onde as "coisas"físicas e virtuais têm identidades, atributos físicos e personalidades virtuais e usam interfaces inteligentes, sendo integradas perfeitamente na rede de informações" [Guillemin et al., 2009].

O IERC está ativamente envolvido no Grupo de Estudos, que lidera o trabalho da União Internacional de Telecomunicações (International Telecommunications Union - ITU) sobre padrões para redes de próxima geração (NGN) que apresentou a seguinte definição [Strategy and Unit, 2005]: Internet of things (IoT) é uma infra-estrutura global de informação, em constante evolução para a sociedade, permitindo serviços avançados, interligando fisicamente e virtualmente as "coisas"providas de um certo grau de tecnologia de comunicação e informação interoperáveis existentes. A IoT fará pleno uso das "coisas"para oferecer serviços a todos os tipos de aplicações, através da exploração da capacidades de identificação, captura de dados, processamento e comunicação, garantindo simultaneamente que os requisitos de segurança e privacidade sejam cumpridos. Essa perspectiva faz com que IoT seja percebida como uma visão com implicações tecnológicas e sociais.

Em [Recommendation, 2012] foi proposto o modelo de referência IoT pelo Setor de Normalização das Telecomunicações (Telecommunication Standardization Sector (ITU-T)), como visto na Figura 4.2, composto por quatro camadas:

- **camada de aplicação:** responsável por prover os serviços para os clientes;
- **camada de suporte a aplicação e ao serviço:** consiste em suporte genérico e suporte específico. Suporte genérico é aquele em que os recursos de suporte são comuns que podem ser usados por diferentes aplicações de IoT, como processamento ou armazenamento de dados. Suporte específico é aquele que possui recursos particulares que atendem aos requisitos de aplicações específicas, eles podem consistir em vários grupos com recursos detalhados, a fim de fornecer diferentes funções de suporte.
- **camada de rede:** responsável por funções relevantes de controle de conectividade de rede, tais como funções de controle de recursos de acesso e transporte, gerenciamento de mobilidade ou autenticação, autorização e contabilidade, fornecimento de conectividade para o transporte do serviço IoT e informações de dados específicos da aplicação, bem como o transporte de informações de controle e gerenciamento relacionadas ao IoT.
- **camada de dispositivos:** Os recursos dessa camada podem ser logicamente categorizados em recursos do dispositivo e recursos do *gateway*:
  - Recursos do dispositivo incluem: Interação direta com a rede de comunicação sendo os dispositivos capazes de coletar e fazer upload de informações diretamente, sem usar recursos do *gateway*, para a rede de comunicação e podendo receber diretamente informações (por exemplo, comandos) da rede de comunicação; Interação indireta com a rede de comunicação, onde os dispositivos são capazes de coletar e fazer o upload de informações para a rede de comunicação, através de recursos do gateway; Rede ad hoc em que os dispositivos podem ser capazes de construir redes de forma ad hoc em alguns cenários que precisam de escalabilidade aumentada e implantação rápida; *Sleeping and waking-up* onde os recursos do dispositivo podem utilizar mecanismos de "dormir"e "acordar"para economizar energia.

– Os recursos do *Gateway* incluem: Suporte a várias interfaces, por exemplo, na camada de dispositivos, os recursos do *gateway* suportam dispositivos conectados através de diferentes tipos de tecnologias com ou sem fio, ZigBee, Bluetooth ou Wi-Fi e na camada de rede, os recursos do *gateway* podem se comunicar através de várias tecnologias, como rede telefônica comutada (PSTN), redes de segunda geração ou de terceira geração (2G ou 3G), Ethernet ou assinante digital linhas (DSL); Conversão de protocolo, quando as comunicações na camada do dispositivo usam diferentes protocolos, por exemplo, protocolos de tecnologia ZigBee e protocolos de tecnologia Bluetooth e quando comunicações envolvendo camada de dispositivo e camada de rede usam protocolos diferentes, por exemplo, um protocolo de tecnologia ZigBee na camada de dispositivo e um protocolo de tecnologia 3G na camada de rede.

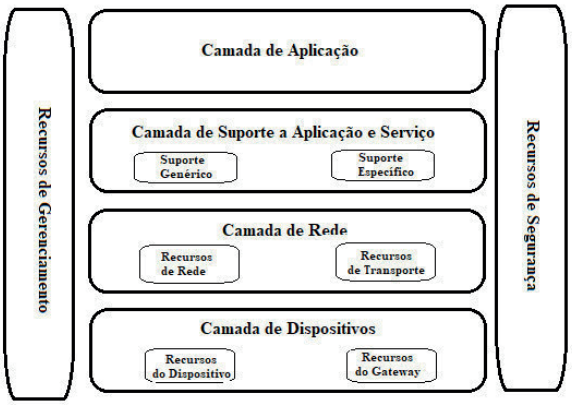

Figura 4.2. Modelo de Referência ITU-T para IoT. Adaptado de: Recommendation ITU-T Y.2060

Esse modelo inclui Recursos de Gerenciamento e Recursos de Segurança associados às quatro camadas. Os Recursos de Gerenciamento cobrem as classes tradicionais de falhas, configurações, contabilidade, desempenho e segurança (FCAPS), como gerenciamento de falhas, gerenciamento de configurações, gerenciamento de contabilidade, gerenciamento de desempenho e gerenciamento de segurança, também são divididos em recursos genéricos e específicos:

- Recursos genéricos de gerenciamento na IoT incluem: gerenciamento de dispositivos, como ativação e desativação remota de dispositivos, diagnóstico, atualização de firmware e/ou software, gerenciamento de status de funcionamento do dispositivo; gestão de topologia de rede local; gerenciamento de tráfego e congestionamento, como a detecção de condições de transbordamento de rede e a implementação de reserva de recursos para fluxos de dados críticos de tempo e/ou vida.
- Recursos específicos de gerenciamento na IoT estão intimamente associadas aos

requisitos específicos da aplicação, por exemplo, requisitos de monitoramento da linha de transmissão de energia da rede inteligente.

De forma similar, também existem dois tipos de recursos de segurança: recursos genéricos de segurança e recursos específicos de segurança:

- Recursos genéricos de segurança são independentes dos aplicativos. Eles incluem na camada de aplicação: autorização, autenticação, confidencialidade e proteção de integridade dos dados da aplicação, proteção de privacidade, auditoria de segurança e antivírus. Na camada de rede incluem: autorização, autenticação, confidencialidade dos dados de uso e da sinalização e proteção da integridade de sinalização. Na camada do dispositivo: autenticação, autorização, validação da integridade do dispositivo, controle de acesso, confidencialidade de dados e proteção de integridade.
- Recursos específicos de segurança estão intimamente associados aos requisitos específicos da aplicação, por exemplo, aplicação de pagamento com mobilidade.

As aplicações de Internet das Coisas são inúmeras e diversas, e permeiam praticamente a vida diária das pessoas, das empresas e sociedade como um todo, transformando o mundo em *smart world* que permite que a computação se torne "invisível" aos olhos do usuário, por meio da relação entre homem e máquina, tornando um mundo mais eficiente e eficaz [Gubbi et al., 2013]. A Figura 4.3 a seguir mostra um panorama da atuação da internet das coisas:

- **Produtos Inteligentes** - Bens adquiridos pelos consumidores, tais como *smartphones, smart house, smart car, smart TV e wearables.*
- **Saúde Inteligente *(eHealth)*** - Fitness, bioeletrônica e cuidados com a saúde. Por exemplo: monitoramento e controle da frequência cardíaca durante os exercícios; Monitoramento das condições dos pacientes em hospitais e em casas de idosos.
- **Transporte Inteligente** -Notificação das condições de tráfego, controle inteligente de rotas, monitoramento remoto do veículo, coordenação das rodovias e integração inteligente de plataformas de transporte.
- **Distribuição Inteligente de Energia *(smart grid)*** - Acompanhamento de instalações de energia, subestações inteligentes, distribuição de energia automática e medições remotas de relógios residenciais.
- **Logística** - *Smart e-commerce*, rastreabilidade, gerenciamento na distribuição e inventário.
- **Indústria Inteligente** - Economia de energia, controle da poluição, segurança na manufatura, monitoramento do ciclo de vida dos produtos, rastreamento de produtos manufaturados na cadeia de abastecimento, monitoramento de condições ambientais e controle de processos de produção.

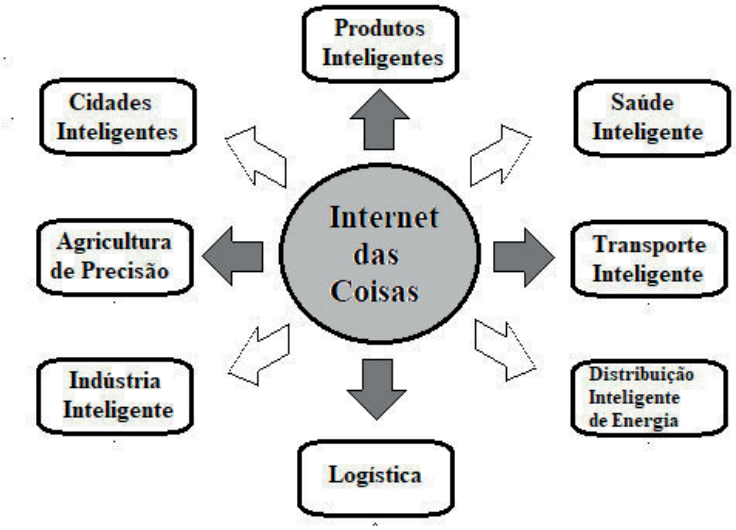

Figura 4.3. Aplicações de IoT

- **Agricultura de Precisão** - Segurança e rastreabilidade de produtos agrícolas, gerenciamento de qualidade, monitoramento ambiental para produção e cultivo, gerenciamento no processo de produção, utilização de recursos para a agricultura.
- **Cidades Inteligentes** - Monitoramento estrutural: monitoramento de vibrações e condições dos materiais em edifícios, pontes e monumentos históricos. Energia elétrica: iluminação inteligente e adaptável conforme a rua. Segurança: monitoramento por meio de vídeo digital, gerenciamento de controle de incêndio e sistemas de anúncio público. Transporte: estradas inteligentes com avisos, mensagens e desvios de acordo com as condições climáticas e eventos inesperados como acidentes ou engarrafamentos. Estacionamento: monitoramento em tempo real da disponibilidade de espaços de estacionamento, sendo possível identificar e reservar vagas disponíveis. Gestão de resíduos: detecção de níveis de lixo em recipientes para otimizar a rota de coleta de lixo.

4.2.2. Princípios Fundamentais de Segurança

Segurança e privacidade são princípios basilares de qualquer sistema de informação. Nos referimos a segurança como a combinação de Integridade, Disponibilidade e Confidencialidade. Normalmente é possível obter segurança usando uma combinação de autenticação, autorização e identificação. Esses conceitos são definidos a seguir [Stallings, 1995]:

- **Integridade:** Certeza que uma informação não foi alterada, exceto por quem tem

o direito de realizar estas alterações. A integridade dos dados é a garantia de que os dados não foram manipulados, estão corretos. No contexto da Blockchain é a garantia de que os dados que constam nas transações não podem ser modificados intencionalmente ou por eventos fortuitos, como surtos de energia ou erros na propagação dos dados. Mecanismos criptográficos de verificação de integridade são comumente utilizados para sua confirmação.

- **Disponibilidade:** Garante que os usuários de um determinado sistema conseguirão utilizá-lo sempre que for necessário. Em outras palavras, os serviço estará sempre ativo quando solicitado por um usuário legítimo. Isso requer que tanto a infraestrutura de comunicação quanto as bases de dados possam ser utilizadas. A Blockchain alcança este objetivos ao permitir que os usuários estabeleçam conexão com vários usuários e ao manter os blocos de maneira descentralizada com várias cópias dos blocos na rede.

- **Confidencialidade:** É a garantia de que a informação não será obtida por pessoas não autorizadas. Isto é, apenas aqueles com os direitos e privilégios necessários serão capazes de acessar a informação, esteja ela armazenada, em processamento ou em trânsito. Na rede Bitcoin, para garantir este princípio são utilizados mecanismos de pseudo anonimização do usuário, como uso dos endereços Bitcoin. Os endereços Bitcoin são resumos criptográficos das chaves públicas.

- **Autenticação, Autorização e Auditoria:** Busca verificar a identidade de quem realiza uma determinada função em um sistema, verificar que direitos esse usuário possui e armazenar informações de uso desse usuário. A estrutura da Blockchain é totalmente desenvolvida para garantir estas três funções, pois somente os usuários que possuem as chaves privadas podem realizar transações, e todas as transações são públicas e auditáveis.

- **Não Repúdio:** Garantia que a pessoa não negue ter feito uma determinada ação em um sistema. O não repúdio fornece provas de que um usuário realizou uma determinada ação, como transferir dinheiro, autorizar uma compra, ou enviar uma mensagem. Como todas as transações são assinadas, um usuário não pode negar que a realizou.

A privacidade pode ser definida como o direito que um indivíduo tem em compartilhar suas informações. Os usuários do Bitcoin usam um pseudônimo (endereço) para realizar suas transações. Normalmente cada usuário possui centenas de endereços. Uma transação pode ser vista como uma cadeia de assinaturas que comprovam a posse e a transferência de valores, de maneira auditável. Assim uma das preocupações é que essas transações possam revelar informações do usuário que vão além de simplesmente uma identificação, como hábitos de compra e locais frequentados do usuário.

O conceito de privacidade em Blockchain consiste em manter o anonimato e a desvinculação de transações. O anonimato de transações exige que não seja possível vincular uma transação particular a um usuário, para isto, o usuário utiliza um endereço diferente a cada nova transação. A desvinculação das transações exige que duas transações do mesmo indivíduo não possam ser vinculadas como tal.

4.2.3. Funções Hash e Criptografia

Toda a posse de recursos e transações na rede são feitas utilizando-se o conceito de chaves e assinaturas digitais. As chaves usadas são geradas aplicando o conceito de criptografia de chaves públicas. São geradas um par de chaves, uma pública que pode ser compartilhada e uma secreta que somente o dono tem acesso. Toda transação requer uma assinatura para ser considerada válida e para provar a posse do recurso dispendido.

Resumos criptográficos

Resumos criptográficos, ou hash, são funções matemáticas que geram um resumo, uma espécie de impressão digital dos dados de entrada. Quando aplicadas a um determinado conjunto de dados ela irá gerar como saída um valor, que a princípio, é único[1]. Um dos usos mais frequentes para o hash é verificar a integridade de arquivos. Por exemplo, ao assinar digitalmente um documento, a pessoa que o recebe, além de verificar a chave usada para a assinatura, também compara o hash fornecido pelo emissor do documento com o calculado na hora do recebimento. Caso o documento sofra alguma alteração, os resumos serão diferentes. O tamanho da saída do hash depende do algoritmo usado, mas o importante é que ela seja sempre do mesmo tamanho, não importando o tamanho da entrada. Exemplos de algoritmos de hash são o SHA-256 e o RIPEMD160, ambos usados pelo Bitcoin. Os algoritmos de hash devem possuir algumas características:

- **Unidirecionalidade:** Deve ser computacionalmente muito difícil encontrar a entrada a partir do resumo.
- **Compressão:** É desejável que o tamanho do resumo represente uma fração pequena dos dados.
- **Facilidade de cálculo:** Não deve ser custoso calcular o valor do resumo.
- **Difusão:** Para dificultar a engenharia reversa do algoritmo ao mudar um bit dos dados de entrada o valor do resumo deve ser alterado de uma quantidade de bits próxima a 50%.
- **Colisão:** Deverá ser computacionalmente difícil encontrar dois valores de entrada que gerem o mesmo resumo.

Criptografia

Todos os sistemas de criptografia usam uma transformação de uma mensagem clara em uma ilegível. Para realizar esta transformação são usadas algumas transformações matemáticas sobre a mensagem clara juntamente com uma chave. Após essas transformações é obtido um texto cifrado que poderá ser lido apenas por quem possuir a chave para decifrar.

[1]Podem existir dois conjuntos de dados com o mesmo hash, mas a probabilidade dessa ocorrência é extremamente baixa.

Um dos sistemas de criptografia mais simples foi o utilizado por Júlio Ceasar, que consistia em substituir as letras do texto por outras posteriores espaçadas da mesma chave. Por exemplo, escrever SBC com chave 13 resulta em FOP. Este tipo de criptografia também pode ser classificado como Criptografia de Chave Privada ou Simétrica, onde uma única chave é usada para criptografar e de-criptografar o segredo.

Outro tipo de criptografia é a Assimétrica ou de Chave Pública. Onde usa-se um par de chaves, uma pública e outra privada, a primeira para criptografar e a segunda para de-criptografar e vice-versa. Isso é possível graças ao uso de algumas funções matemáticas que possuem a propriedade de serem irreversíveis. As mais usadas são a fatoração em números primos (IFP - *Integer Factorization Problem*), curvas elíptica s(ECDLP - *Elliptic Curve Discrete Logarithm Problem*) ou logaritmos discretos (DLP - *Discrete Logarithm Problem*). A eficiência de um sistema de criptografia pode ser medida considerando:

- **Carga Computacional:** Mede a eficiência com que os algoritmos podem implementar as transformações com as chaves públicas e privadas.
- **Tamanho da Chave:** O NIST indica o uso de pares de chave (pública, privada) com tamanhos, em bits, para cada tipo de implementação: RSA (1088,2048), DSA (1026,160), e ECC (161,160). O ECC apresenta grande vantagem nesse aspecto.
- **Tamanho de Banda:** Corresponde a quantidade de bits necessária para transmitir uma mensagem, após codificar ou assinar.

[Jansma and Arrendondo, 2004] comparou o ECC com os RSA e chegou a conclusão de que para um mesmo nível e segurança a ECC possui uma menor carga computacional, menor tamanho de chave e menor tamanho de banda. Por esses motivos O Bitcoin adotou o sistema de curvas elípticas definida em um padrão chamado *secp256k1*, estabelecido pelo Instituto Nacional de Padronização e Tecnologia (NIST). Para maiores informações sobre curvas elípticas é recomendado [Hankerson et al., 2006].

Assinatura Digital, Endereço e Carteira

Uma assinatura digital é a cifragem do hash de um documento usando uma chave privada, tal que a chave pública, da mesma pessoa que assinou, seja usada para provar que foi ela quem assinou aquele documento.

O Bitcoin adota o *Elliptic Curve Digital Signature Algorithm* (ECDSA) para realizar assinaturas. É uma versão baseada em curvas elípticas. Assume-se que a dificuldade do logaritmo não permita que terceiros assinem um documento sem que tenha conhecimento da chave privada de uma pessoa. Pensando de modo inverso, se é impossível forjar a assinatura, então uma assinatura válida não pode ser refutada pelo dono da chave. Normalmente, o processo de assinar um documento é realizado sobre seu resumo criptográfico. Uma vantagem de se usar estas funções é que elas sempre geram como saída uma pequena quantidade de bits de mesmo tamanho. A assinatura deve ser capaz de prover integridade, não repúdio e autenticidade.

No Bitcoin a chave privada é obtida gerando um número aleatório de 256bits, uma chave pública é obtida ao efetuar a multiplicação da chave privada por um ponto na curva conhecido como "ponto gerador". Ele é sempre o mesmo para todos os usuários do Bitcoin e é definido na especificação secp256k1. O resultado da multiplicação da chave privada pelo ponto gerador é um ponto na curva, este ponto é a chave pública. Os nós armazenam somente as suas chaves privadas, pois ele pode a qualquer momento gerar a pública correspondente.

A partir deste ponto, o nó já possui um par de chaves ele pode gerar o endereço. O endereço, não confundir com endereço IP, é um número obtido usando a chave pública do nó. Ele é usado para informar ao sistema quem é o dono daquela transação, pois somente quem possuir a chave privada que gerou aquele endereço poderá usar o que estiver na transação, seja um montante monetário ou um dado. Para gerar o endereço o nó deve realizar uma operação de duplo hash, primeiro SHA-256 depois RIPEMD160: $Endereco = RIPEMD160(SHA256(ChavePublica))$, após deve converter o resultado para Base58. Este é o endereço Bitcoin.

Os usuários do Bitcoin possuem chaves que permitem provar a posse de transações. Essas chaves precisam ser armazenadas, e geralmente são armazenadas em uma carteira digital. A carteira tem a função de gerar as chaves dos usuários. Existem dois tipos de carteira: as determinísticas e as aleatórias. As determinísticas usam uma chave inicial, chamada de semente, para criar as demais através de uma função hash. Armazena apenas a primeira chave, pois todas as outras podem ser recalculadas. As carteiras Aleatórias precisam usar algum algoritmo de geração de números aleatórios para gerar as chaves, e precisa armazenar todas as chaves criadas.

4.2.4. Rede peer-to-peer(P2P)

A rede do Bitcoin foi pensada para ser uma rede de consenso descentralizada, pois essa descentralização é um dos pontos chaves de sua mentalidade. Desta forma foi desenvolvida uma rede par-a-par (ou P2P, do termo em inglês *peer-to-peer*), onde todos os participantes da rede são iguais, não existindo um nó centralizador, e todos são onerados para manter a rede funcionando. Todos os nós se interconectam na forma de uma rede sobreposta.

Existem quatro funções que podem ser assumidas por um nó na rede: Roteamento; Base de dados Blockchain; Mineração; e Carteira. Um nó completo possui todas as quatro funções, mas todos os nós possuem pelo menos a função de roteamento. Estas funções foram separadas pois nem todos os participantes da rede precisam executar todas as funções. Um usuário comum, por exemplo, que busca somente um meio de pagamento possui apenas a carteira e o roteamento. Desta forma ele pode se conectar a rede e realizar transações somente com um celular, sem a necessidade de armazenar toda a cadeia de blocos.

Para entrar na rede, é necessário conhecer ao menos um nó participante. Cada nó pode iniciar até 8 (*Outbounds Conections*) conexões e aceitar até 117 (*Inbounds Conections*). No core do Bitcoin existe gravado uma lista com alguns nós conhecidos como *Seeders*, que tem o objetivo de entregar uma lista de outros nós ativos da rede, para que o novo nó estabeleça conexão. Mas o nó não é obrigado a se conectar aos *Seeders*. Todas

as conexões são TCP. Para estabelecer a conexão inicial o nó realiza um *HandShake* com uma mensagem *version*. Assim que a conexão é estabelecida, o nó envia uma mensagem GETADDR solicitando uma lista de endereços IP conhecidos. De posse desta lista, inicia o processo novamente para outros nós, desta lista, a fim de se tornar bem conectado. Após a primeira vez que o nó é conectado, ele guarda em disco uma lista com todos os nós que estabeleceu conexão recentemente. Assim das próximas vezes que se ligar a rede pode não necessitar do auxílio dos *Seeders*.

Existem duas tabelas chamadas de *Tried e New*. A primeira armazena os endereços que o nó já estabeleceu alguma conexão, iniciada ou recebida. Ela é uma estrutura de dados com 64 entradas, chamadas de recipiente, onde cada recipiente armazena 64 endereços. A segunda tabela possui 256 recipientes, também com a capacidade de armazenar 64 endereços, e é usada para guardar os endereços recebidos das mensagens ADDR e os endereços removidos da *Tried*. Quando um nó precisar estabelecer uma nova conexão, ele irá escolher um endereço de uma das duas tabelas: *Tried ou New*. Para isso, ele usa a seguinte fórmula, que dá a probabilidade de escolha da *Tried*:

$$P_{tried} = \frac{\sqrt{\theta(9-\varepsilon)}}{(\varepsilon+1)+\sqrt{\theta(9-\varepsilon)}}$$

Onde θ é a razão entre a quantidade de endereços armazenados na *Tried* sobre a *New* e ε é quantidade conexões iniciadas.

Além das mensagens *version e ADDR* o protocolo especifica mensagens para troca de dados, que são as mensagens para difusão das transações e dos blocos. A Figura 4.4 sintetiza as principais mensagens utilizadas.

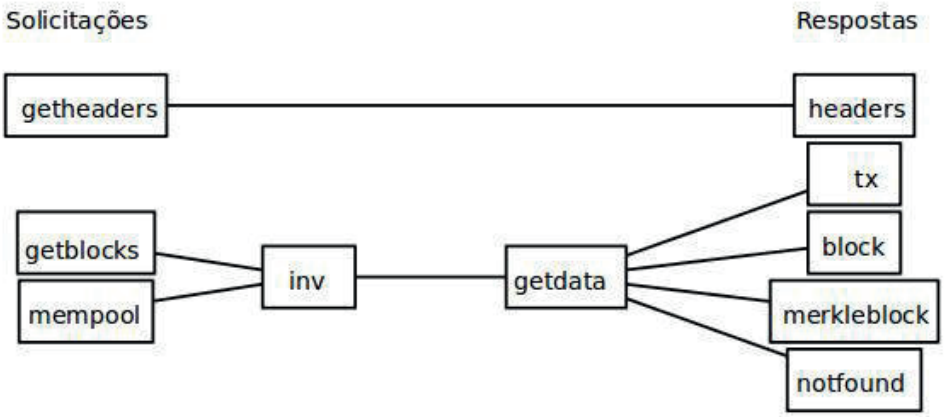

Figura 4.4. Mensagens do protocolo Bitcoin

Alguns nós da rede são nós simples, que possuem apenas funções de roteamento e carteira. Esses nós não possuem uma visão completa da rede e necessitam de ajuda de outros nós para fazer checagens de rotina, por exemplo, ao receber um pagamento um nó quer saber se o valor recebido é válido. O protocolo então específica que os nós completos podem realizar essas checagens e responder aos nós simples. Para isso eles

proveem um serviço de RPC(*Remote Procedure Call*) de modo que aqueles nós que são limitados possam realizar consultas sobre a rede e realizar operações típicas de carteira.

4.3. Blockchain

O conceito da Blockchain começa a deixar claro que vai muito além da inovação tecnológica. Está causando um grande impacto, primeiramente mudando a forma de fazer negócios de modo centralizado para uma forma descentralizada, conferindo confiabilidade na realização de transações entre agentes distribuídos e mutuamente não confiáveis, sem a necessidade de uma entidade intermediária confiável por ambos. Além disso tem a capacidade de mudar a maneira como são realizadas todos os tipos de transações e habilitar uma gama imensa de possibilidades em outras áreas, como computação segura entre múltiplos participantes (MPC - Multi-Party Computation) [Zyskind et al., 2015a], uso em Organizações Autônomas Descentralizadas(DAC - Descentralized Autonomous Corporation) [Swan, 2015b], e aplicações governamentais [Andrea, 2014].

É possível dividir sua evolução em três etapas [Swan, 2015a]: Blockchain 1.0, 2.0 e 3.0. Blockchain 1.0 é o uso comercial com transferência de moeda, remessa, e sistemas de pagamento digital, amplamente difundido pelo uso do Bitcoin e derivados. Blockchain 2.0 é o seu uso com contratos, toda a lista das questões econômicas, mercado e aplicações financeiras que o utilizam de maneira mais extensa do que transações simples de caixa, como: ações, títulos, empréstimos, hipotecas e contratos inteligentes. Blockchain 3.0 refere-se o seu uso em aplicações além da moeda, finanças e mercados, particularmente nas áreas de governo, saúde, ciência e etc.

4.3.1. Definição da Blockchain

Satoshi Nakamoto [Nakamoto, 2008] (pseudônimo dos desenvolvedores iniciais do Bitcoin) introduziu o Blockchain como mecanismo para garantir irretratabilidade, auditabilidade, e imutabilidade a fim de prover segurança a transações eletrônicas, servindo como um grande livro razão distribuído. Este mecanismo é visto como a principal inovação introduzida pelo Bitcoin. Ele representa uma forma de alcançar um consenso entre participantes não confiáveis. Normalmente instituições como bancos ou cartórios são responsáveis pela guarda e segurança do registro de transações e são chamados de terceiros de confiança. O sistema proposto por Nakamoto elimina a necessidade destas entidades, pois todos os registros são, além de públicos, mantidos de maneira descentralizada por diversos participantes da rede. A Figura 4.5 é uma visão simplificada da rede, onde pode-se observar as funções principais que cada nó pode utilizar. Observa-se que é uma rede sobreposta e que os vizinhos podem estar, inclusive, em outros continentes.

De maneira simplificada a Blockchain é uma estrutura de dados que armazena transações de forma ordenada e ligada ao bloco anterior, servindo como um sistema de registros distribuído. Essa estrutura é dividida em duas partes: cabeçalho e transações, e armazena informações detalhadas a respeito das transações que contém. Assim é possível associar uma transação ao seu endereço de origem e destino. Cada bloco possui uma identificação única gerada a partir de um resumo criptográfico de hash conforme explicado na seção anterior. O cabeçalho possui um campo que armazena o hash do bloco imediatamente anterior, desta forma conseguimos estabelecer uma ligação, um "elo", en-

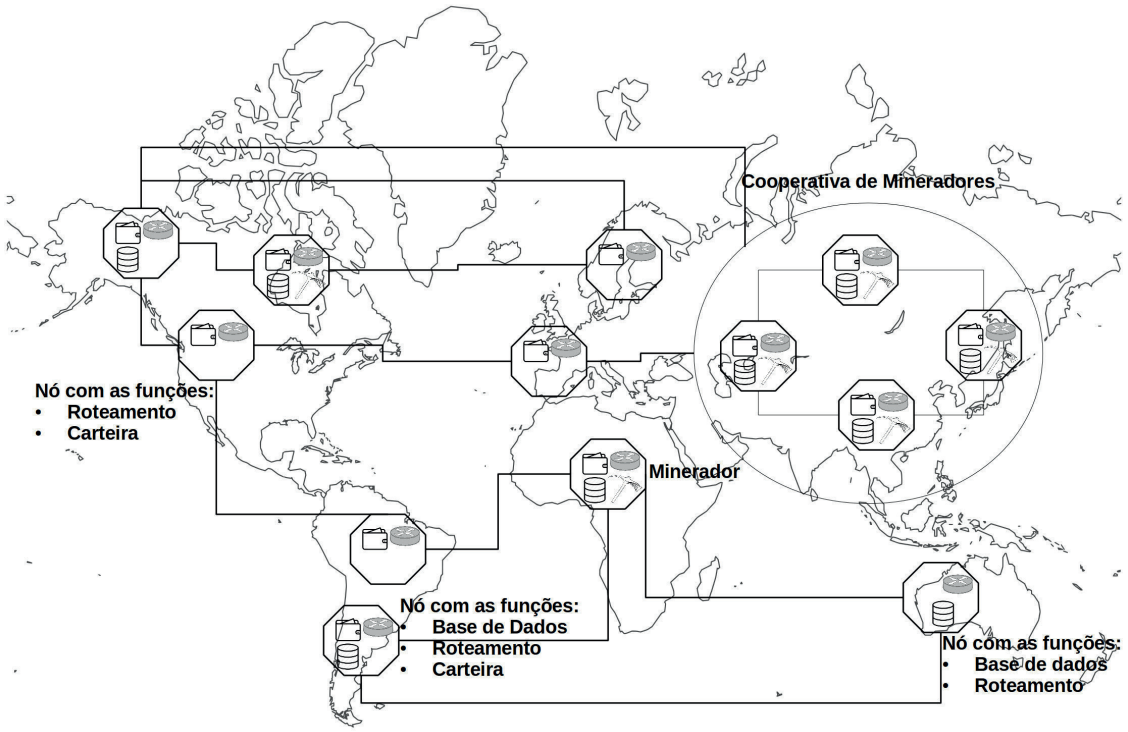

Figura 4.5. Visão Geral da Rede Bitcoin

tre os blocos. Por esse motivo, essa estrutura recebeu o nome de "corrente de blocos"ou Blockchain (ver Figura 4.6). Outra característica desta ligação é que esse hash é obtido a partir de uma colisão parcial, que será explicada mais detalhadamente a seguir, processo esse que requer um grande poder computacional para ser calculado. Como cada bloco faz referência ao seu antecessor, se um bit do bloco anterior for alterado, seu hash irá mudar e consequentemente será necessário recalcular o hash de todos os blocos descendentes. Por esse motivo assume-se que a existência de em uma cadeia longa de descendentes torna o bloco imutável, garantindo a segurança das transações armazenadas.

4.3.2. Estrutura de um Bloco

As partes principais de um bloco são o cabeçalho e as transações. As transações são o agrupamento dos dados que são armazenados no bloco. Por sua vez, o cabeçalho possui diversos campos, dois quais os mais importantes para seu funcionamento são: hash do bloco anterior, dificuldade, nonce, e raiz da árvore de merkle. Além destes, também é preciso entender dois metadados: altura do bloco e hash do cabeçalho, que são armazenados de forma a identificar o bloco e sua posição na cadeia. Estes campos serão detalhados abaixo, pois o correto entendimento da Blockchain depende deles.

4.3.2.1. Cabeçalho do Bloco

- **Altura:** Os blocos são incluídos na cadeia de forma linear em ordem cronológica, cada novo bloco recebe um número de ordem, a diferença entre o número do último

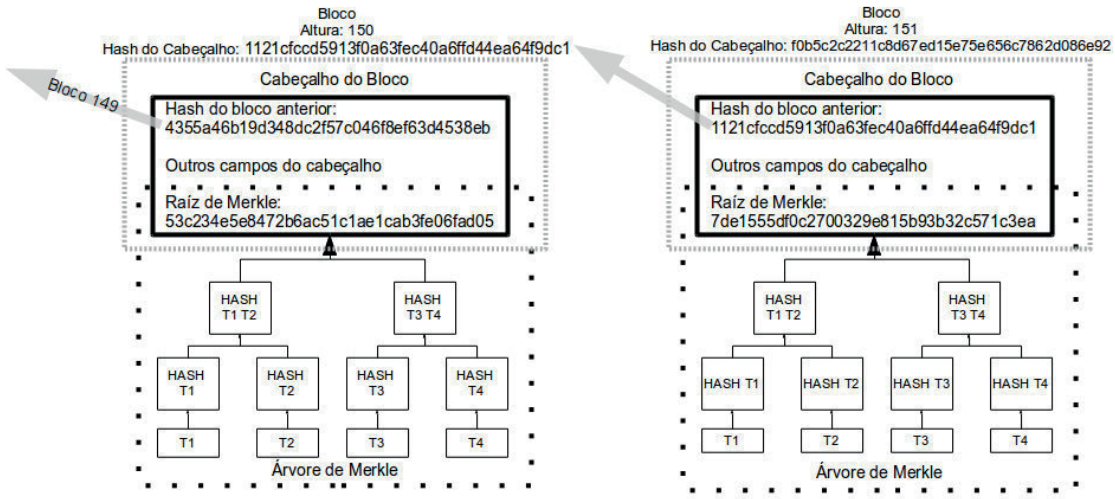

Figura 4.6. Estrutura dos Blocos Simplificada

bloco e o primeiro é chamada de altura. Este campo nem sempre é usado para identificar um bloco, pois pode haver, momentaneamente, dois ou mais blocos com a mesma altura. Neste caso ocorre um desvio, um fork, na cadeia.

- **Hash do cabeçalho:** É o principal identificador do bloco, é obtido ao realizar uma operação de resumo criptográfico no próprio cabeçalho. Ele não faz parte da estrutura de dados do bloco e também não é enviado pela rede junto com o bloco. Ele é computado isoladamente por cada nó completo no ato do recebimento de um novo bloco, e é armazenado em um banco de dados separado como parte dos metadados do bloco. Ao contrário da altura, o hash do cabeçalho pode ser usado para identificar um bloco de forma inequívoca.

- **Hash do bloco anterior:** Este campo é incluído no cabeçalho de modo a possibilitar que um novo bloco seja ligado ao anterior. Como vimos na Figura 4.6, o bloco 236 possui, em seu cabeçalho, o hash do bloco 235. Os nós completos armazenam os metadados dos blocos. Assim todos os nós possuem o hash do bloco 235, tão logo o bloco 236 seja recebido, por um nó completo, ele irá verificar este campo e definirá que o bloco 236 é filho do 235.

- **Nonce:** É um número usado como uma variável para modificar a saída da função hash do cabeçalho. Em conjunto com o campo dificuldade alvo ele é usado para provar que um minerador realizou um trabalho e encontrou um cabeçalho que atenda aos critérios estabelecidos para essa prova. Para ficar mais claro imagine que seja estabelecido que o hash do cabeçalho inicie com uma sequencia de três zeros, o minerador então irá por força bruta iterar o nonce até que o hash do cabeçalho atenda a esse requisito. No recebimento do novo bloco os nós completos irão calcular o hash do cabeçalho apenas uma vez.

- **Dificuldade:** A dificuldade nada mais é que uma colisão parcial de hash, ou seja, como descrito anteriormente, um algoritmo de hash gera sempre um mesmo resumo

para uma determinada entrada. Se for alterado um bit que seja desta entrada o hash resultante será completamente diferente. Assim depende do poder computacional do nó minerador achar um hash que satisfaça a essa colisão parcial. O mecanismo usado para gerar a colisão é o nonce. Como ele faz parte do cabeçalho, sempre que é alterado o resumo também muda. Quando a dificuldade é configurada para com 1bit (zero) basta achar um hash que inicie com um zero e qualquer valor para os outros 255bits, ou seja $2^2$55 possibilidades, será considerado válido. Caso seja setado com 2bits as possibilidades serão reduzidas para 254bits ou $2^2$54, com 10bits serão $2^2$46 possibilidades e assim por diante. É possível observar que a diminuição do espaço de possíveis valores que satisfaçam a colisão implica em maior dificuldade em achar um resumo que satisfaça a dificuldade, logo, mais computação é requerida, ou mais tempo de mineração, e maior gasto com energia.

O processo de incluir novos blocos à cadeia é chamado de mineração, e o nós que realizam o trabalho de gerar um novo bloco é chamado de minerador. A taxa pela qual novos blocos são incluídos na cadeia é definida pelos desenvolvedores de cada projeto de Blockchain. Na rede Bitcoin foi estabelecido um alvo de 10min, ou seja, a dificuldade é ajustada por todos os nós completos e mineradores para que, em média, a cada 10min um novo bloco seja incluído na cadeia. É esperado que novos mineradores se juntem a rede e novos equipamentos mais poderosos sejam lançados, com isso, em média, o tempo de inclusão de novos blocos tende a diminuir. Para evitar que novos blocos sejam incluídos a intervalos menores que 10min a dificuldade é ajustada, aumentando a quantidade de bits para a colisão. Assim, como será mais difícil achar o novo hash o tempo de inclusão de novos blocos irá se ajustar até ficar próximo ao alvo de 10 minutos. Cada nó minerador recalcula, independentemente, a nova dificuldade a cada 2016 novos blocos realizando a seguinte operação matemática:

$$Nova\ Dificuldade = Dificuldade\ Antiga * (\frac{Tempo\ n\ Blocos}{(Tempo\ Alvo * n\ Blocos)})$$

- **Transações:** No Bitcoin uma transação é uma transferência de valores. De maneira simplificada é um conjunto de entradas (endereços de onde os valores serão retirados) e saídas (endereços para onde os valores serão enviados). Um nó após criar uma transação a envia a todos os seus vizinhos. Os nós que receberam a transação a retransmitem aos seus vizinhos, para que a transação alcance todos os nós da rede. Quando um minerador recebe a transação ele irá guardá-la para que ela seja incluída em um próximo bloco que será minerado. Quando este bloco for incluído na cadeia, a transação se torna pública e imutável. As transações são assinadas com um sistema de chaves públicas. Para enviar um valor a alguém é necessário possuir a chave privada para assinar a transação, provando a posse do valor. Também é necessário conhecer a chave pública do usuário que irá receber o valor, para cifrar a transação de modo que somente o detentor da chave privada, que faz par com a pública de destino, conseguirá decifrá-la. Desta forma é possível que o sistema seja público e ainda assim somente quem realmente é dono da transação poderá usá-la.

  Existem outros dois tipos de transações na rede Bitcoin, os contratos inteligentes, que serão explicados melhor ao longo do capítulo, e o armazenamento de dados, chamado de *OP_RETURN*. Foi destinado um campo com 40bytes para ser usado

para armazenamento de dados diversos. É endereçado da mesma forma que uma transação financeira, assim o recebedor dos dados precisa possuir a chave necessária para usá-los.

- **Árvores de Merkle:** Uma árvore de Merkle [Merkle, 1987], ou árvore de hash binário, é definida como uma árvore binária completa com um valor de k bits associado a cada nó da árvore, de modo que cada valor de nó interior seja uma função unidirecional dos valores de seus filhos. São projetadas para que um valor de folha possa ser verificado em relação a um valor de raiz conhecido publicamente, bastando serem fornecidos os valores dos pares correspondentes no caminho da folha até a raiz.

  Na Blockchain ela é usada para resumir, eficientemente, as transações contidas em um bloco, para tal é necessário produzir $2 * \log_2 N$ hashes. Logo, ela fornece um processo muito eficiente para verificar se uma transação consta em um bloco. Para construir esta árvore deve-se iniciar pelas folhas, que contêm o hash das transações. Para criar a árvore é necessário um número par de transações, caso haja um número ímpar a última folha da árvore será duplicada. As folhas são então agrupadas duas a duas e seu hash produz um nó pai, os nós pai são então agrupados em pares e sofrem o mesmo processo de modo que esse processo continue até que não haja mais pares, gerando assim um nó raiz chamado de raiz de merkle, conforme Figura 4.7.

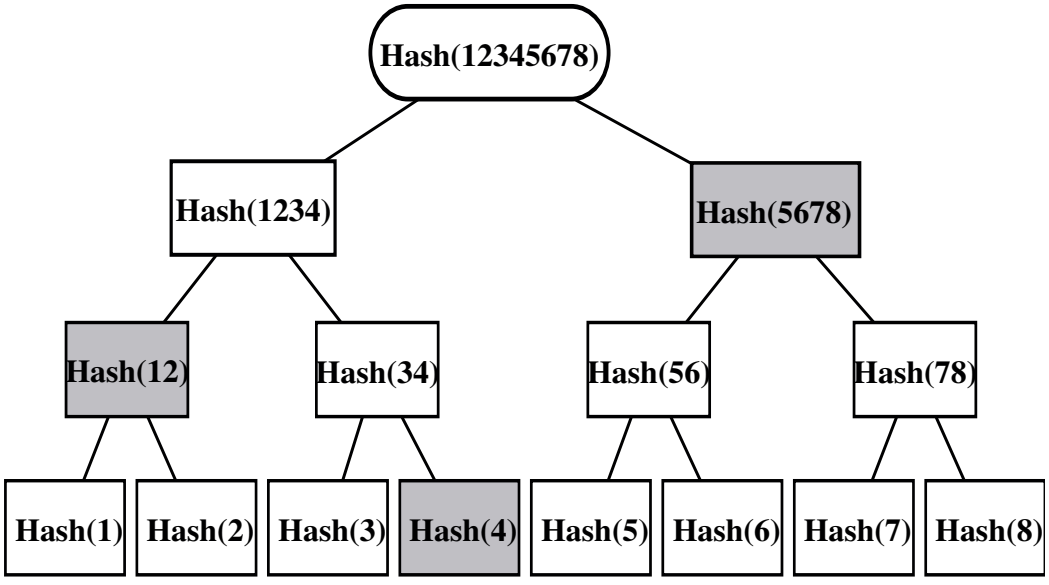

Figura 4.7. Árvore de merkle

Para provar que uma transação está inclusa em um bloco basta fornecermos o caminho que a transação ira percorrer na árvore, este caminho consiste no hash do complemento dos pares. Assim é possível realizar esta verificação rapidamente em meio a milhares de transações. Isso é particularmente útil pois para verificar se uma transação consta em um determinado bloco não é necessário solicitar o bloco inteiro à rede, basta o cabeçalho do bloco e o caminho até a transação. Como vimos anteriormente um nó simplificado não possui a cadeia de blocos armazenada.

Caso ele necessite confirmar uma transação precisará da ajuda de um nó completo. Por exemplo, na Figura 4.7 cada folha corresponde ao hash de uma transação e os valores em cinza correspondem ao caminho para provar que a transação consta no bloco. Para provar que a transação 3 consta no bloco o nó completo enviará o cabeçalho deste bloco e os hash(4), hash(12) e hash(5678) ao nó simplificado. De posse desses dados é possível calcular a raiz da árvore e comparar com o valor da raiz de merkle que consta no cabeçalho do bloco. O Nó simplificado fará o cálculo do hash(3) que juntamente com o hash(4) calculará o hash(34), pega o valor do hash(12) e chega a hash(1234) e por último usa o hash(5678) para calcular a raiz, cujo valor é hash(12345678).

4.3.3. Mineração

A mineração é o processo responsável por atualizar a Blockchain, pelo qual alguns nós especiais, chamados de mineradores, incluem as transações em um bloco e geram um cabeçalho válido para essas transações. O mineradores gastam muita energia para realizar a Prova de Trabalho, por esse motivo precisam ser recompensados. A primeira transação do bloco é sempre uma transação especial chamada de *Coinbase*. Ela tem dois propósitos, incluir novas moedas ao sistema e recompensar o minerador. Na rede Bitcoin, a mineração tem dois propósitos. Primeiramente, incluir novas moedas ao sistema e em segundo lugar proteger as transações realizadas. Para gerar esse cabeçalho os mineradores devem calcular a árvore de merkle das transações, verificar a dificuldade estabelecida, incluir a estampa de tempo e realizar uma série de cálculos a fim de encontrar um nonce que satisfaça a dificuldade em vigor. Assim será descrito a importância da dificuldade e como ela se ajusta automaticamente, além de mostrar um passo a passo do processo de mineração.

A mineração consiste em gerar um novo bloco. Para isso o minerador primeiro cria um "rascunho"de um bloco. É sobre esse rascunho que ele vai trabalhar até que obtenha um bloco viável para ser enviado aos todos os nós da rede. O rascunho é a estrutura de dados que vai comportar os dados do cabeçalho e as transações. Após criar essa estrutura em branco, o minerador preenche alguns campos do cabeçalho: hash do bloco anterior, estampa de tempo, versão e dificuldade. Restando preencher a raiz da arvore de merkle, o nonce e agrupar as transações.

As transações, ao serem geradas, são enviadas via *broadcast* a todos os nós vizinhos e estes reencaminham aos seus vizinhos. Os mineradores, ao receberem uma mensagem com uma transação, às armazenam em uma base de dados de transações ainda não mineradas. As transações permanecem temporariamente em uma espécie de fila com prioridade até que sejam retiradas para ser incluídas em um novo bloco. Cada minerador possui uma fila diferente de transações, e pode selecionar quais transações ele vai incluir nesse novo bloco. Após selecionar quais transações serão incluídas ele irá gerar uma árvore de merkle e incluir o valor da sua raiz no cabeçalho.

Agora falta achar o valor do nonce que fará parte do novo bloco. Esta é a etapa demorada do processo, requer um grande poder computacional dos mineradores e consequentemente um enorme gasto de energia, conforme foi explicado na seção anterior. Para se ter ideia da tempo para achar um hash válido atualmente são comercializados dispositivos especializados em calcular hash, esses dispositivos atingem a marca de 9TH/s,

ou seja conseguem calcular nove trilhões de hash por segundo, que com a dificuldade atual da rede Bitcoin seriam necessários 13 anos para achar um hash válido. A seguir um exemplo: Para achar o nonce que produza um hash válido para "Simpósio Brasileiro de Segurança"com a dificuldade alvo de "000"(12bits), ou seja o alvo é que o hash inicie com três zeros em sequência. para isso é pode-se concatenar o nonce com a informação que será resumida - sha256("Simpósio Brasileiro de Segurança+nonce") - incrementando o nonce a cada insucesso até que seja encontrado um hash válido. Em um terminal do Linux é possível fazer:

for nonce in $(echo {0..10000}); do echo "$(echo "Simpósio Brasileiro de Segurança +$nonce sha256sum) $nonce"; done | grep ^000

Como resultado será obtido: 000a45ed0ebded66019dff14fc916c429aac6021494 e4983960e27c917696db5 - 4060

O que significa dizer que o nonce 4060 ao ser acrescido a "Simpósio Brasileiro de Segurança"gera um hash que atende a dificuldade alvo. É importante notar que existem outros valores de nonce que geram resultados válidos, como o 5470. A partir deste momento qualquer um que possua uma implementação do sha256 pode calcular o resumo de "Simpósio Brasileiro de Segurança+4060"e comparar com o hash fornecido, demonstrando assim que o resultado é válido.

Assim que o nonce é encontrado o nosso rascunho fica completo e portanto o bloco está pronto para ser enviado a todos os nós da rede. Os nós da rede ao receberem um novo bloco iniciam uma série de verificações a fim de validar o bloco e chegar a um consenso em caso de bifurcações (*"forks"*).

4.3.4. Consenso e Prova de Trabalho

A cadeia de blocos não é criada por uma autoridade central. Os blocos são criados independentemente pelos mineradores da rede. Os nós, usando as informações que são transmitidas através de conexões inseguras, conseguem chegar à mesma conclusão e fabricar o mesmo registro público que todos os outros nós. Atingindo assim um consenso global. Os nós completos armazenam toda cadeia com os blocos que foram validados por ele. Quando diversos nós possuem os mesmos blocos em sua cadeia principal é considerado que eles chegaram ao consenso. Esta subseção descreve as regras de validação de cada bloco e como o consenso é alcançado e mantido e também explica alguns dos vários mecanismos de consenso que são utilizados atualmente.

O mecanismo de consenso é composto por duas etapas: validação do bloco e seleção da maior cadeia. Estas duas etapas são realizadas de maneira independente por cada nó. Os blocos são enviados em *broadcast* pela rede, e cada nó ao receber um novo bloco o retransmite aos seus vizinhos, mas antes desta retransmissão o nó faz a validação do bloco a fim de garantir que somente blocos válidos sejam propagados. Existe uma extensa lista de verificação a ser seguida, dentre elas:

- Estrutura do bloco;
- Verificar se o hash do cabeçalho atende a dificuldade estabelecida;
- Tamanho do bloco dentro dos limites projetados;

- Verificação de todas as transações; e
- Verificação da estampa de tempo.

Por definição da Blockchain cada bloco tem somente um pai, mas pode ocorrer uma situação em que um ou mais mineradores gerem novos blocos quase ao mesmo tempo, fazendo com que haja um ou mais filhos com um mesmo pai. Neste caso, entende-se que ocorreu um fork, uma bifurcação, na cadeia. A última etapa do mecanismo de consenso serve exatamente para selecionar qual destes blocos fará parte da cadeia principal e qual será descartado. Isso é possível em virtude da prova de trabalho, que será abordada nessa seção, fundamental para o mecanismo de consenso adotado, pois como vimos anteriormente para gerar o bloco, mineradores gastam muita energia em busca de um bloco válido.

Como é possível a ocorrência de bifurcações, os nós armazenam os blocos sem pai(órfãos) [2] e mantém duas cadeias, uma principal e uma secundária. Os blocos órfãos acontecem quando dois blocos são gerados em espaços curtos de tempo e chegam em ordem inversa, ou seja, um bloco foi recebido e não faz referência a um bloco na cadeia. Ele é armazenado por um período de tempo, caso o nó receba um bloco que seja pai do órfão ele será incluídos na cadeia em sua ordem correta. Note que neste caso não houve a ocorrência de uma bifurcação, os blocos apenas foram recebidos fora de ordem.

Como existem diversos mineradores gerando blocos de forma descentralizada, os novos blocos enviados por eles podem chegar a diferentes nós em momentos diferentes, o que pode resultar em visões diferentes. Para ficar mais claro quando dois mineradores geram blocos fazendo referência a um mesmo pai ocorre a bifurcação, e os outros mineradores deverão escolher qual bloco eles irão adotar como referência. Se uma parte dos mineradores adotar um bloco e outra parte adotar o outro, essas duas cadeias irão coexistir até que uma fique maior que a outra. Para resolver esta situação, os nós que se comportam de maneira honesta, de acordo com o mecanismo de consenso, sempre irão adotar a maior cadeia e o fork estará resolvido. A corrente principal é a maior cadeia, aquela onde há a maior quantidade de trabalho acumulada. Na Figura 4.8 os blocos cinza bifurcaram da cadeia principal, como alcançaram uma altura maior passaram a ser a cadeia principal. Os blocos brancos 127,128 e 129 são descartados e suas transações são consideradas como não confirmadas, devendo ser incluídas futuramente em outros blocos.

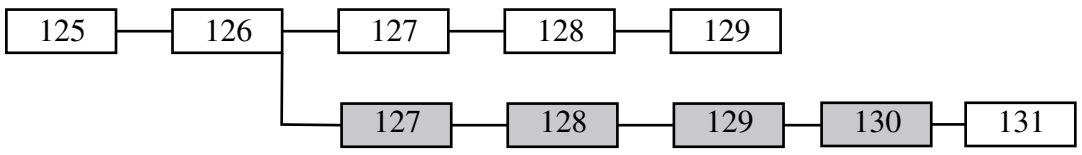

Figura 4.8. Bifurcação

Uma das preocupações mais comuns para os sistemas de moedas digitais é a possibilidade do gasto duplo, quando um usuário malicioso gasta um mesmo valor em duas

[2] Situação rara e temporária

transações diferentes da cadeia. Note que para ocorrer a tentativa de um gasto duplo é necessário uma bifurcação, pois se o gasto ocorrer na mesma cadeia, quando o novo bloco for criado, ele não passará nas verificações iniciais de consistência e será descartado. Com o fork, o usuário malicioso faz um gasto e envia para rede, gasta a mesma quantia novamente em outro lugar e começa a minerar sobre esse gasto. Desta forma, há a possibilidade de ele conseguir minerar um bloco e realizar o fork. A partir deste momento, a rede estará dividida e como mencionado anteriormente haverá uma corrida que será vencida pela maior cadeia. Uma das transações será descartada e o gasto duplo será rejeitado. Como uma das cadeias irá ser aceita pela rede e, a outra descartada, eventualmente o gasto duplo será detectado. É usualmente aceito na rede Bitcoin que uma transação é considerada confirmada quando existem seis novos blocos com altura maior que a sua, pois será necessário muito esforço para alterá-la.

Um cenário de ataque contra o mecanismo de consenso é chamado de "ataque de 51%". Nesse cenário, um grupo de mineradores, controlando uma maioria (51%) do poder de hash total da rede, conspira para atacar o Bitcoin. Com a habilidade de minerar a maioria dos blocos, os mineradores atacantes podem gerar bifurcações deliberadas na Blockchain, gerar transações de gasto duplo ou executar ataques de negação de serviço (DoS) contra endereços ou transações específicas. Um ataque de bifurcação/gasto duplo é um ataque onde o atacante faz com que blocos já confirmados sejam invalidados ao fazer uma bifurcação em um nível abaixo deles, com uma posterior re-convergência em uma cadeia alternativa. Com poder suficiente, um atacante pode invalidar seis ou mais blocos em uma sequência, invalidando transações que antes eram consideradas imutáveis (com seis confirmações). Note que o gasto duplo só pode ser feito nas transações do próprio atacante, para as quais o atacante pode produzir uma assinatura válida. Fazer um gasto duplo da própria transação é rentável quando, ao invalidar uma transação, o atacante puder receber um pagamento irreversível ou um produto sem ter que pagar por isso.

Alcançar o consenso em um sistema distribuído é um desafio. Os algoritmos de consenso devem ser resilientes a falhas de nós, particionamento da rede, atrasos de mensagens, mensagens que chegam fora de ordem e corrompidas. Eles também têm que lidar com nós egoístas e deliberadamente maliciosos. Vários algoritmos tem sido propostos para resolver isso, cada um realizando o conjunto de suposições necessárias em termos de sincronia, transmissões de mensagens, falhas, nós maliciosos, desempenho e segurança das mensagens trocadas. Para uma rede Blockchain, alcançar consenso garante que todos os nós na rede concordem com um estado global consistente da cadeia de blocos.

Segundo [Kim, 2014, Natoli and Gramoli, 2016], um protocolo de consenso tem três propriedades fundamentais com base nas quais sua aplicabilidade e eficácia podem ser determinadas:

- **Segurança:** Um protocolo de consenso é determinado para ser seguro se todos os nós produzirem o mesmo resultado (*agreement*) e os resultados produzidos pelos nós são válidos de acordo com as regras do protocolo (*validity*). Isso também é referido como consistência do estado compartilhado.

- **Vivacidade:** Um protocolo de consenso garante a vivacidade se todos os nós que seguem o protocolo eventualmente produzem um valor (*termination*), ou seja, se

um nó gerar uma transação e enviá-la a todos os nós da rede em algum momento um minerador irá incluí-la em um bloco.

- **Tolerância a falhas:** Capacidade de continuar a operar, chegar ao consenso, adequadamente, mesmo após a falha de alguns nós da rede.

O resultado de impossibilidade de Fischer Lynch Paterson (FLP) afirma que um sistema de consenso assíncrono determinista pode ter no máximo duas destas três propriedades. Este é um resultado comprovado, ou seja, qualquer sistema de consenso distribuído na Internet deve sacrificar uma dessas propriedades [Fischer et al., 1985].

A maioria das plataformas Blockchain existentes, mais de 90% da capitalização do mercado total de moedas digitais, utilizam o mecanismo de consenso, na sua forma original e computacionalmente cara, que é a Prova de Trabalho. Porém, existem vários outros mecanismos que oferecem certas vantagens desejadas em relação ao modelo original, como a prova de Prova de Posse (PoS, do inglês *Proof of Stake*) [King and Nadal, 2012], o Algoritmo de Tolerância a Falhas Bizantinas (PBFT, do inglês *Practical Byzantine Fault Tolerance*) [Castro and Liskov, 2002] e a Prova do Tempo Decorrido (PoeT, do inglês *Proof of Elapsed Time*) [Intel,] aparecem como outras alternativas e serão brevemente explicadas a seguir:

- **Prova de Trabalho:** A ideia principal da Prova de Trabalho (PoW, do inglês *Proof of Work*) é tentar evitar ataques cibernéticos. Para atingir este objetivo, utiliza-se de um sistema onde o usuário deve provar que gastou um certo tempo para encontrar alguma resposta que satisfaça algum requisito que o verificador pedir. A tarefa de encontrar tal resposta, é baseada em dois princípios. Em primeiro lugar, a PoW tem que ser difícil e trabalhosa, mas não impossível; e em segundo lugar a verificação dessa prova deve ser muito mais rápida e fácil de ser realizada. Este conceito foi inicialmente proposto por Adam Back [Back et al., 2002] e é utilizado por diversos sistemas de prova e também pelo Bitcoin.

  No Bitcoin, a Prova de Trabalho é gerada da seguinte forma: o remetente adiciona um número arbitrário à mensagem (chamado de nonce) e aplica uma função matemática de hash na mensagem. O SHA-256 [Gilbert and Handschuh, 2003] é usado pelo Bitcoin. O objetivo é encontrar uma resposta com um certo número de zeros na frente. Ele repete o procedimento variando o nonce até achar essa resposta. Como é relativamente difícil encontrar tal resposta, ao receber a mensagem, todo usuário será capaz de verificar que houve um grande esforço do remetente em gerá-la. Ao decifrar o problema, o minerador gera um novo bloco. A dificuldade da prova de trabalho é ajustada a cada 2016 blocos, a fim de que seja gerado em média um bloco a cada dez minutos. A segurança da PoW baseia-se no princípio de que nenhuma entidade deve reunir mais de 50% do poder de processamento da rede porque essa entidade poderá efetivamente controlar o sistema, manipulando a cadeia mais longa.

- **Prova-de-Posse (PoS)** - também é utilizada em plataformas Blockchain de criptomoedas. Enquanto a PoW recompensa os participantes que resolvem enigmas criptográficos complicados, baseados em hash, para validar transações e criar novos

blocos (mineração), a PoS requer uma certa quantidade de moeda para sua participação. O criador do próximo bloco é escolhido de uma maneira Probabilística, e a chance de uma conta ser escolhida depende de sua "riqueza"(ou seja, a posse). Os algoritmos de Prova-de-Posse são projetados para superar as desvantagens dos algoritmos PoW em termos do alto consumo de energia envolvido nas operações de mineração. A PoS substitui completamente a operação de mineração por uma abordagem alternativa envolvendo participação de usuários ou propriedade de moeda virtual no sistema de Blockchain. Dito de outra forma, em vez de um usuário gastar R$ 2000 comprando equipamentos de mineração para iniciar os trabalhos no algoritmo PoW e ganhar uma recompensa pela mineração, o mesmo usuário, utilizando a PoS pode comprar R$ 2000 em criptomoedas e usá-las como participação para comprar cotas proporcionadas de criação de blocos no sistema Blockchain, tornando-se um validador. Em criptografia de PoS, os blocos costumam ser validados, em vez de minerados. Essas implementações não oferecem incentivos para que os nós votem no bloco correto. Portanto, os nós podem votar em vários blocos que suportam vários forks para maximizar suas chances de ganhar uma recompensa. Neste caso não gastam nada ao fazê-lo em oposição a PoW, onde o nó dividiria seus recursos para votar em múltiplos forks. Este é o problema *"Nothing-at-Stake"*, que precisa ser abordado para uma implementação correta e eficiente de PoS. A seleção pelo saldo da conta resultaria em centralização (indesejável), pois o único membro mais rico teria uma vantagem permanente. Vários métodos de seleção diferentes foram planejados, por exemplo, Nxt [Kim, 2016] e BlackCoin [Vasin, 2014] usam aleatorização para prever o próximo gerador de blocos, usando uma fórmula que procura o menor valor de hash em combinação com o tamanho da participação. Uma vez que as apostas são públicas, cada nó pode prever, com precisão razoável, qual conta ganhará o direito de validar um bloco.

- **Algoritmo de Tolerância a Falhas Bizantinas (PBFT)** - A função de um protocolo de consenso é manter a ordem das transações em uma rede de cadeias de blocos, apesar das ameaças a essa ordem. Uma dessas ameaças é a falha arbitrária simultânea, um dos tipos de falha bizantina, de múltiplos nós de rede. Usando PBFT, uma rede de nós Blockchain pode tolerar nós defeituosos até f, onde f é uma fração arbitrária conhecida do número total de nós - com uma máquina de estados replicada em nós diferentes (uma réplica sendo definida como primária). O algoritmo PBFT funciona da seguinte forma:

  – Um cliente envia uma solicitação de serviço para a máquina primária.
  – A primária replica o pedido para os backups.
  – As réplicas executam o pedido e enviam respostas.
  – O cliente aguarda $f + 1$ respostas idênticas de réplicas diferentes para considerar um resultado correto.

  Como o número total de nós precisa ser conhecido, o PBFT não é adequado para sistemas públicos, sendo utilizado apenas em sistemas privados. Uma rede PBFT garante a consistência e a integridade dos dados quando ocorrem falhas bizantinas em até $1/3$ dos nós da rede. Por exemplo, usando PBFT, uma rede de cadeia de

blocos de nós N pode suportar f número de nós Bizantinos, onde $f = (N-1)/3$. Em outras palavras, o PBFT garante que um mínimo de $2*f+1$ nós alcancem consenso sobre a ordem das transações antes de anexá-las ao livro razão compartilhado. A regra $2*f+1$ tem as seguintes implicações:

Como são necessários um mínimo de $2*f+1$ nós para chegar a um consenso antes de prosseguir para o próximo bloco de transações, o livro-razão em qualquer nó adicional (além de $2*f+1$) ficará temporariamente atrasado. Este atraso na sincronização do livro-razão geral compartilhado em todos os nós é uma limitação inevitável em qualquer rede PFBT.

- **Prova do Tempo Decorrido (PoET)** um algoritmo de consenso, projetado pela Intel. PoET usa um modelo de eleição aleatória de um líder, que irá validar os blocos. Funciona essencialmente da seguinte forma, existe um *hardware* especializado para gerar um valor de tempo aleatório. Cada validador solicita um tempo de espera a este *hardware*. O validador com o tempo de espera mais curto para um determinado bloco é eleito o líder, e espera este tempo para validar o bloco. Após este bloco ser incluído na cadeia o processo se repete. Este modelo é proposto para uso em Blockchains privados, pois, em teoria os validadores são honestos. A aleatoriedade na geração de tempos de espera garante que a função líder seja distribuída de forma uniforme e entre todos os validadores. Uma desvantagem desse algoritmo é a dependência de hardware especializado.

4.3.5. Categorias de Blockchain com base no acesso aos dados

Blockchain pode ser classificado com base no acesso aos dados e na participação do mecanismo de consenso sobre quaisquer mudanças propostas no seu livro razão. Podendo ser:

- **Blockchain sem permissão ou (Pública):** O mecanismo de consenso está aberto a todos. O objetivo de uma cadeia sem permissão é permitir que qualquer pessoa contribua com dados. Isso cria a chamada resistência da censura, o que significa que nenhum ator pode evitar que uma transação seja adicionada à cadeia. Os participantes mantêm a integridade da cadeia ao chegar a um consenso quanto ao seu estado. Qualquer um pode se juntar à rede e participar do processo de verificação de blocos para criar consenso e também criar contratos inteligentes. Ter um sistema sem permissão implica assumir que pode não haver confiança entre os nós, portanto, um mecanismo de consenso fortemente distribuído deve ser imposto. Em tal sistema, existe a possibilidade de um ataque Sybil [Douceur, 2002], onde um nó de rede tenta aparecer como vários nós distintos criando um grande número de pseudoidentidades. Uma influência desproporcionalmente grande por um único nó é uma ameaça, então a introdução do PoW na validação da transação é logicamente justificada e necessária.

- **Blockchain permissiva ou (Privada):** Participantes no processo de consenso estão pré-selecionados. Quando um novo registro é adicionado, a integridade do livro razão é verificada por um processo de consenso realizado por um número limitado

de atores confiáveis. Isso torna a manutenção de um registro compartilhado muito mais simples do que o processo de consenso sem permissão. As cadeias de blocos permitidas fornecem conjuntos de dados altamente verificáveis porque o processo de consenso cria uma assinatura digital, que pode ser vista por todas as partes. As características que derivam de sistemas confiáveis podem abrir a possibilidade de evitar um protocolo de consenso computacionalmente exigente, como a PoW.

Muitos projetos foram iniciados para tornar a Blockchain mais popular e viável para diferentes modelos de negócios e aplicações, aproveitando as categorias existentes. A Tabela 1.1 resume as principais características de algumas aplicações com base em Blockchain. Bitcoin, Ethereum [Wood, 2014] são exemplos de Blockchain sem permissão e Hyperledger [Cachin, 2016] e Ripple [Pilkington, 2015] são exemplos de Blockchain com permissão. É possível verificar uma diferença crítica entre essas duas categorias que é o modelo de mineração subjacente - Blockchains sem permissão usam a Prova de Trabalho (PoW) onde o poder de *hashing* é oferecido para criar confiança, já as Blockchains permissivas não precisam usar a mineração baseada em energia computacional para chegar a um consenso, já que todos os atores são conhecidos, eles acabam usando algoritmos de consenso como PBFT que podem ser usados para alcançar o consenso sem mineração por PoW, levando a um tempo de processamento de bloco bem inferior comparado ao tempo da Blockchain sem permissão, sendo praticamente considerado realizado em tempo real. .

Tabela 4.1. Comparação entre sistemas Blockchain

Blockchain	**Bitcoin**	**Ethereum**	**Hyperledger**	**Ripple**
Natureza	Sem Permissão	Sem Permissão	Permissiva	Permissiva
Validação	PoW SHA-256	PoW - ethash	PBFT	BFT customizado (RPCA)
Propósito	criptomoeda	contrato inteligente	Chaincode	criptomoeda
Linguagem	Scripts baseados em pilha	Código interno Turing completo	Go, Java	C++
Tempo de proc. do bloco	$\sim$ 600 s	$\sim$ 15 s	$\sim$ tempo real	$\sim$ tempo real

4.3.6. Contratos inteligentes

O uso da tecnologia Blockchain proporcionou múltiplas classes de funcionalidades de aplicações em todos os segmentos de negócios com criptomoedas, mercados e transações financeiras. Desde o início sua utilidade foi pensada para ser usada além da moeda e dos pagamentos prevista para Bitcoin; As possibilidades de dinheiro programável e contratos foram preparadas no Bitcoin, em sua invenção. Uma comunicação de 2010 de Satoshi Nakamoto indica que "o design suporta uma tremenda variedade de possíveis tipos de transações que eu planejei anos atrás: Transações de custódia, contratos vinculados, arbitragem de terceiros, assinatura multipartidária, etc. Se o Bitcoin se encaixar de forma

importante, estas são coisas que queremos explorar no futuro, mas todas elas deveriam ser projetadas no início para, com certeza, fazer que elas sejam possíveis mais tarde". Os conceitos e estrutura desenvolvidos para Bitcoin são extremamente portáteis e extensíveis.

Em 2014, surgiu o termo "Blockchain 2.0", sendo usado para descrever um novo projeto no campo de banco de dados distribuído da Blockchain. Uma parte da terminologia que se refere amplamente ao espaço Blockchain 2.0 e inclui Bitcoin 2.0 e seus protocolos, contratos inteligentes, propriedades inteligentes, Dapps (aplicativos descentralizados), DAOs (organizações descentralizadas autônomas) e DACs (corporações autônomas descentralizadas) [Buterin, 2014].

Considerando que o Blockchain 1.0 é para a descentralização de dinheiro e pagamentos, o Blockchain 2.0 é para a descentralização de mercados de forma mais geral e contempla a transferência de muitos outros tipos de ativos além da moeda usando a cadeia de blocos. A ideia-chave é que as funcionalidades do livro-razão de transações descentralizadas da cadeia de blocos podem ser usada para registrar, confirmar e transferir todo tipo de contratos e propriedades.

Os Contratos Inteligentes são *scripts* armazenados na Blockchain e executam suas instruções de maneira distribuída em todos os participantes do contrato. O termo Contratos Inteligente (do termo em inglês, *Smart Contracts*) guarda similaridade com o contrato legal. Neste sentido ele regula a interação entre diferentes entidades. Nick Szabo introduziu este conceito em 1994 e definiu um *Smart Contract* como "um protocolo de transação computadorizado que executa os termos de um contrato" [Szabo, 1994]. Szabo sugeriu a transposição de cláusulas contratuais em código e incorporá-las em propriedades (hardware ou software) que possam auto-executá-las [Szabo, 1997], de modo a minimizar a necessidade de intermediários confiáveis entre as partes das transações, e a ocorrência de exceções maliciosas ou acidentais.

Os Contratos Inteligentes possuem um endereço, e são acionados endereçando uma transação para ele. Em seguida, é executado de modo independente e automático da forma prescrita em cada nó da rede, de acordo com os dados que foram incluídos na transação desencadeadora. Isto implica que cada nó em uma Blockchain, habilitado por contrato inteligente, está executando uma máquina virtual e que a rede Blockchain atua como uma máquina virtual distribuída [Christidis and Devetsikiotis, 2016].

Três características dos contratos inteligentes os diferenciam são: autonomia, auto-suficiência e descentralização [De Filippi and Mauro, 2014]. Autonomia significa que, depois de lançado e em execução, um contrato e seu agente iniciador não precisam estar em contato. A segunda característica é que os contratos inteligentes podem ser auto-suficientes em sua capacidade de gerar recursos, isto é, arrecadar fundos ao fornecer serviços ou emitir equidade e gastá-los em recursos necessários, como o poder de processamento ou o armazenamento. E a terceira característica é que os contratos inteligentes são descentralizados, na medida em que não subsistem em um único servidor centralizado, sendo distribuídos e auto-executados em nós de rede.

O exemplo clássico usado para demonstrar contratos inteligentes sob a forma de código que é executado automaticamente é uma máquina de venda automática. Ao contrário de uma pessoa, uma máquina de venda automática se comporta de forma algorítmica,

ou seja, o mesmo conjunto de instruções será seguido toda vez em todos os casos. Quando você deposita o dinheiro e faz uma seleção, o item é lançado. Não há possibilidade da máquina não cumprir o contrato um dia, ou apenas cumpri-lo parcialmente (desde que não esteja quebrada). Um contrato inteligente de forma semelhante, executa o código pré-especificado.

A confiança mínima geralmente torna as coisas mais convenientes, tirando o julgamento humano da equação, permitindo uma automação completa. Um exemplo de um contrato inteligente básico na Blockchain são aqueles utilizados em apólices de seguro. As companhias de seguros podem automatizar políticas de seguro, escrevendo-as para um contrato inteligente, quando as condições de entrada do contrato inteligente mudam para um evento segurado, por exemplo, no caso de uma catástrofe natural, o processo de reivindicações é desencadeado imediatamente. Os parâmetros mensuráveis do evento, como a velocidade do vento, a localização de um furacão ou a magnitude de um terremoto podem ser registrados na Blockchain. À medida que os parâmetros atravessam certos limiares pré-acordados, o processo de reclamações é desencadeado imediatamente e a quantidade exata de pagamento financeiro pode ser entregue sem necessidade de intervenção humana. A transparência e a confiança no processo são visíveis para todas as partes interessadas e todos os órgãos reguladores.

Um outro exemplo de contratos inteligentes, agora para a plataforma de Internet das Coisas, é o monitoramento de entrega de encomendas. Atualmente, por exemplo, os pacotes podem se perder na postagem, porém com o advento da Internet das Coisas, com sensores em todos os lugares, da prateleira no armazém, ao endereço do destinatário. Cada sensor forma seu próprio nó em uma Blockchain e contratos inteligentes podem gravar a "posse"do dispositivo em cada sensor individual (e local subsequente). Um dispositivo de rastreamento na embalagem será lido em cada sensor no caminho para o destinatário. Cada vez que é lido por um novo sensor, sua localização é transmitida e acordada por todos os participantes da IoT na Blockchain. Um contrato inteligente, em seguida, mantém as guias de "posse"ao longo de todo o caminho, solidificando a confiança de saber exatamente onde encontrar o pacote.

Ethereum, é o *framework* mais conhecido e utilizado para contratos inteligentes. Ethereum é uma máquina virtual descentralizada, que executa programas chamados contratos a pedido dos usuários. Contratos são escritos em uma linguagem *Turing-completa bytecode*, chamado *EVM bytecode* [Wood, 2014]. Um contrato é um conjunto de funções, cada uma definida por uma sequencia de instruções *bytecode*. Uma característica notável dos contratos é que eles podem transferir éter (uma criptomoeda similar ao Bitcoin) para/de usuários e para outros contratos. As transações são usadas para:

- criar novos contratos;
- invocar funções de um contrato;
- transferência de éter para contratos ou para outros usuários.

Todas as transações são registradas em uma Blockchain pública. A sequência de transações na Blockchain determina o estado de cada contrato, e o saldo de cada usuário.

Na Figura 4.9 é apresentado um exemplo de código para um contrato inteligente básico escrito para uso na Blockchain Ethereum. É apresentado um código básico, na linguagem solidity, para a criação de um *token* digital que no ecossistema Ethereum pode representar qualquer bem negociável: moedas, certificados de ouro, etc. Como todos os *tokens* implementam algumas características básicas de uma maneira padrão, isso também significa que esse *token* será instantaneamente compatível com a carteira Ethereum e qualquer outro cliente ou contrato que use os mesmos padrões.

O comando *mapping*, cria uma matriz associativa, onde é associado endereços com saldos. Os endereços estão no formato hexadecimal básico da ethereum, enquanto os saldos são inteiros, variando de 0 a 115 *quattuorvigintillion* (que corresponde a uma incontável quantia de *vigintillions*). A palavra-chave *public* significa que esta variável será acessível por qualquer pessoa no bloco, o que significa que todos os saldos são públicos (como devem ser, para que os clientes possam exibi-los). A função *MyToken* tem o mesmo nome que o contrato *MyToken*, se for renomear um, deve-se renomear o outro também: esta é uma função de inicialização especial que é executada apenas uma vez e uma vez somente quando o contrato é carregado pela primeira vez para a rede. Esta função irá definir o saldo do *msg.sender*. A função *transfer* é muito direta, ela tem um destinatário e um valor como parâmetro e sempre que alguém a chama, subtrairá o valor de seu saldo e o adicionará ao saldo do destinatário. De imediato, haverá um problema óbvio que acontece se a pessoa quiser enviar mais do que possui. Para solucionar esse problema é necessário a implementação de uma verificação rápida e, se o remetente não tiver fundos suficientes, a execução do contrato simplesmente irá parar.

```
contract MyToken {
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(
        uint256 initialSupply
        ) {
        balanceOf[msg.sender] = initialSupply;              // Give the creator all initial tokens
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) {
        require(balanceOf[msg.sender] >= _value);           // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        balanceOf[msg.sender] -= _value;                    // Subtract from the sender
        balanceOf[_to] += _value;                           // Add the same to the recipient
    }
}
```

Figura 4.9. Exemplo de Contrato inteligente fonte: https://www.ethereum.org/token acessado em: 22/08/2017

Uma vez que os contratos têm um valor econômico, é crucial garantir que a sua execução seja executada corretamente. Os potenciais conflitos na execução de contratos (devido, por exemplo, a falhas ou ataques) são resolvidos através de um protocolo de consenso baseado em PoW. Idealmente, a execução de contratos é garantida, mesmo com a presença de um usuário malicioso, desde que ele não possua a maioria do poder

computacional da rede. A rede Ethereum atualmente usa um algoritmo de consenso PoW, chamado Ethash, criado especificamente para Ethereum. Foi construído para dificultar seu processamento por meio de hardwares específicos, como o chips ASICs. Está prevista a alteração do mecanismo de consenso da rede Ethereum até o final do ano de 2017, será utilizado o PoS.

A segurança do processo de consenso baseia-se na suposição de que é mais conveniente para um minerador seguir o protocolo do que tentar atacá-lo. Para manter esse pressuposto, os mineradores recebem alguns incentivos econômicos para executar os cálculos exigidos pelo protocolo. Parte desses incentivos é dada pelas taxas de execução pagas pelos usuários em cada transação, ou etapa de execução de um contrato. Um atacante até poderia então criar um contrato com uma execução longa, mas ele ficaria caro demais, pois ele necessitaria pagar taxas a cada etapa do processo. Desta forma, as taxas limitam a quantidade de etapas de execução de um contrato, impedindo assim ataques de negação de serviço que usam computações demoradas.

4.4. Casos de uso do Blockchain para prover segurança e privacidade em IoT

Os dispositivos na IoT coletam, geram e processam dados, enviam estas informações através da internet, produzindo uma gigantesca massa de informação a ser usada pelos mais diversos serviços.

Apesar dos benefícios, problemas críticos relacionados a privacidade podem emergir. A Blockchain pode ter um papel fundamental no desenvolvimento de aplicações descentralizadas que irão executar em bilhões de dispositivos. Entender como e quando esta tecnologia pode ser usada para prover segurança e privacidade é um desafio, diversos autores apontam esses desafios, dos quais cita-se [Conoscenti et al., 2016, Dorri et al., 2016, Dorri et al., 2017a].

Nesta seção será explorado como a Blockchain pode ser usada para beneficiar as aplicações de segurança para Internet das Coisas, como aplicações descentralizadas que permitam aos objetos inteligentes interagir com segurança, estabelecer mecanismos de pagamentos [Wörner and von Bomhard, 2014], criar serviços de Infra-Estrutura Pública de Chaves (PKI) [Nguyen et al., 2015, Ali et al., 2016], realizar Computação Segura entre Múltiplos Participantes (MPC) [Zyskind et al., 2015a], suportar Ambientes Inteligentes [Dorri et al., 2017b] e etc. Segundo [Zyskind et al., 2015b] a privacidade pode ser alcançada, por exemplo, ao se combinar o uso do Blockchain com um sistema de armazenamento de dados descentralizado via P2P.

Além do problema da privacidade, segurança é também uma questão fundamental para os sistemas que farão uso de Blockchain. Isto porque a Blockchain pode ser usada para prover não repúdio, autenticidade, confidencialidade e autorização de forma descentralizada. Serão apresentados os ataques mais comuns abordados na literatura, tais como: minerador egoísta [Eyal, 2015, Nayak et al., 2016], gasto duplo [Gervais et al., 2016] e o eclipse [Heilman et al., 2015]. Será descrito como a Blockchain pode ser usado para prover controle de acesso em armazenamento descentralizado [Ouaddah et al., 2017], e para realizar Computação Segura entre Múltiplos Participantes [Zyskind et al., 2015a]. Será realizada a simulação de um ataque demonstrativo, que visa mostrar como a latência de propagação dos blocos e a taxa de inclusão de novos blocos [Gervais et al., 2016] podem

influenciar a segurança do mecanismo de consenso.

4.4.1. Blockchain para prover anonimidade e controle de acesso em IoT

Primeiramente é bom ressaltar que a anonimidade provida pelo uso da Blockchain não é absoluta, por isso ela é comumente chamada de pseudo-anonimidade. É possível, em certas circunstâncias, de-anonimizar o dono da transação, ou seu endereço IP. Para de-anonimizar as transações existem algumas técnicas específicas, [Conoscenti et al., 2016] as dividiu em quatro:

- **Múltiplas Entradas:** Em alguns casos para realizar determinado gasto é necessário reunir saldo de diversas contas. Caso seja preciso guardar o saldo total da carteira em uma única conta, é possível realizar a transferência dos saldo menores para uma única conta, esse procedimento é chamado de transação com múltiplas entradas. Como, para realizar esta transação, é necessário possuir a chave privada de cada entrada é sensato supor que todas as contas pertencem a um mesmo usuário. A partir deste momento é possível associar os endereços a um usuário. Esta abordagem foi utilizada em [Spagnuolo et al., 2014, Moser et al., 2013, Herrera-Joancomartí, 2015].

- **Endereços de Troco:** Como já visto, todas as transações no Bitcoin são transferências de recursos. Por definição do protocolo, É obrigatório gastar todo o saldo associado a uma determinada chave. Caso o valor da transação seja menor que o valor da entrada, essa transação irá gerar troco. O valor do troco deve retornar ao dono e por isso deve ser a endereçado uma saída com destino ao próprio usuário. Caso o usuário use sempre o mesmo endereço para receber o troco de suas transações, pode-se associar este endereço aos endereços de entrada anteriores e descrever exatamente todos os gastos de um usuário, além da possível correlação com fontes secundárias de informação como sites de redes sociais. Esta abordagem foi utilizada em [Spagnuolo et al., 2014, Moser et al., 2013, Herrera-Joancomartí, 2015].

- **Associação ao IP:** A rede Bitcoin é uma rede sobreposta a rede IP. Grande parte das mensagens da rede são transmitidas em *broadcast* para os vizinhos diretos de cada nó. Uma grande quantidade de vizinhos permite a um nó é extrair algum conhecimento da rede, como sua topologia, quem são os nós mineradores, localização dos nós e seu endereço IP. Em [Koshy et al., 2014], o autor conseguiu associar o endereço IP ao endereço do usuário ao escutar o tráfego da rede e utilizar um algoritmo de clusterização.

- **Uso de Serviços Centralizados:** Os usuários podem, por diversos motivos, não guardar e gerenciar suas próprias chaves privadas e delegam essa função a serviços terceirizados. Alguns autores [Moser et al., 2013, Valenta and Rowan, 2015] acham um isso risco a privacidade, pois esta entidade pode vazar seus dados, com isso suas identidades e seus recursos, e até mesmo utilizar os recursos de terceiros, pois a prova de propriedade se dá pela posse da chave privada que está nas mãos de terceiros.

Segundo [Conoscenti et al., 2016] são necessários cuidados extras a fim de mitigar estes problemas. Os dispositivos IoT devem se configurados para: sempre usar um endereço diferente para receber troco; sempre gerar um endereço novo para cada recebimento recursos; não usar serviços terceirizados. Essas medidas não são suficientes para prover anonimidade total, mas proverão um certo grau de segurança em manter as identidades preservadas, evitando principalmente correlacionar um determinado dispositivo ao seu dono.

É possível também usar a Blockchain para armazenar dados e prover controle de acesso a eles. Suponha que um sensor de presença queira armazenar seus histórico diário na Blockchain. Ele irá gerar uma transação com os dados a serem armazenados, e assinará essa transação com sua chave secreta, assim todos saberão qual sensor é dono desses dados. O sensor indicará como saída da transação as chaves públicas com direito de ler seus dados. Ele enviará esta transação aos mineradores de sua rede, que autenticarão a transação e a incluirão no próximo bloco. Como a Blockchain é pública todos os usuários tem acesso a seus dados, e saberão que um determinado usuário tem o direito de ler o histórico produzido pelo do sensor de presença. Mas, somente aqueles detentores das chaves privadas, que fazem par com a públicas indicadas pelo sensor, conseguirão ler o histórico diário que foi disponibilizado pelo sensor.

Ouaddah [Ouaddah et al., 2017] propôs o FairAccess, um *framework*, que usa a Blockchain para habilitar aos usuários controlar seus próprios dados. Ele reutiliza o código do Bitcoin e introduz alguns novos tipos de transação usados para controlar o acesso aos dados, como: *"grant"e "revoke"access*. O modelo prevê a existência de alguns atores: recurso a ser compartilhado; o dono do recurso; e os usuários. As transações são usadas para controlar o acesso dos usuários e a Blockchain é usada servir como local para armazenamento e leitura das permissões. Os autores fizeram uma prova de conceito com um *Raspberry PI* com uma câmera("Recurso"). Foi criado um usuário "dono do recurso"e outro usuário "utilizador". O Dono controla o acesso ao recurso através de transações enviadas a Blockchain. Assim para conceder acesso ao usuário ele envia uma transação do tipo *grant access* e a envia ao usuário, como se estivesse vendendo um produto com Bitcoin. Esta transação será minerada pela rede, o que significa dizer que será verificado que o dono possui a chave privada referente ao recurso e a transação será incluída na Blockchain. A partir deste ponto o utilizador solicitará acesso diretamente ao recurso que verifica na Blockchain se existe uma transação que lhe garanta acesso, neste caso o usuário conseguirá acessar a câmera.

Uma das principais críticas ao armazenamento de dados na Blockchain é o uso de estrutura de dados que não foram projetadas para armazenar grandes quantidades de informação. Assim, caso o bloco comece a ser usado com esse fim, haverá diversas cópias de um mesmo arquivo sendo mantidas na rede, uma vez que a cadeia inteira é mantida de foma descentralizada. Além do desperdício de espaço, há a forma ineficiente de gerenciar estes dados. Com o intuito de usar a segurança provida pela Blockchain, Zyskind [Zyskind et al., 2015b] combina o uso do armazenamento de dados fora da cadeia com o controle de acesso na cadeia de blocos. O armazenamento é realizado com um sistema de DHT (distibuted hash table) sendo mantido por um conjunto de nós da rede previamente selecionados. Os dados são replicados de maneira eficiente pelos nós de forma a garantir a alta disponibilidade e repartida de forma que nenhum nó tenha o

arquivo inteiro. A Blockchain é então usada de forma a gerenciar onde esses dados estão distribuídos e quem tem acesso a eles. Para isso, são gerados dois novos tipos de transação, uma para prover o controle de acesso e outro para controlar a distribuição dos dados no DHT.

Como a Blockchain não possui ponto central de falha e não é governada por uma única entidade, ela permite uma nova classe de aplicativos e serviços descentralizados, como por exemplo, um servidor raiz DNS ou uma autoridade de certificação raiz. Esses benefícios motivaram Ali et al. [Ali et al., 2016] a usar a Blockchain para construir um novo sistema de PKI descentralizado e um sistema de identidade, chamado *Blockstack ID*. O formato para publicar chaves públicas é semelhante ao PGP [Zimmermann, 1995]. *O Blockstack* desacopla o registro do nome e propriedade da disponibilidade de dados associados, separando os planos de controle e dados. O plano de controle define o protocolo para o registro de nomes legíveis para humanos, criando elos (nome, hash). O plano de controle consiste em um bloco e uma camada logicamente separados do plano de controle, sendo responsável pelo armazenamento e disponibilidade de dados. Ele consiste em um local para pesquisa de dados por hash ou URL, e um sistema de armazenamento externo de dados. Todos os dados armazenados são assinados pela chave do respectivo proprietário do nome. Ao armazenar os dados fora da cadeia de blocos, o *Blockstack* permite valores de tamanho arbitrário e permite uma variedade de *backends* de armazenamento. Os usuários não precisam confiar na camada de armazenamento porque podem verificar a integridade dos valores de dados no plano de controle.

4.4.2. Uso de Blockchain em cenários econômicos para garantir transações eletrônicas em IoT

O futuro da IoT é se tornar uma rede de dispositivos autônomos que podem interagir uns com os outros e com seu ambiente, e tomar decisões inteligentes sem a intervenção humana. Este é o lugar onde a Blockchain pode ajudar a alavancar a IoT e formar uma base que suportará a economia compartilhada baseada em comunicações da máquina-a-máquina (M2M). Em [Wörner and von Bomhard, 2014] os autores descreveram uma implementação prototípica simples do processo de troca de dados por dinheiro eletrônico entre um sensor e um requisitante, utilizando a rede Bitcoin. O sistema é composto de três partes:

- **Dispositivo IoT Cliente:** Precisa cumprir as seguintes tarefas: anotar uma solicitação de dados ao receber um pagamento e ser capaz de criar e publicar uma transação contendo os dados solicitados.

- **Cliente requisitante:** Precisa poder enviar pagamento ao endereço Bitcoin do sensor e deve monitorar alterações na Blockchain até detectar a transação com os dados enviados pelo dispositivo IoT.

- **Repositório de dispositivo IoT:** Onde os sensores podem ser registrados ou podem ser encontrados pelos solicitantes. Uma entrada no repositório de sensores deve conter pelo menos o endereço do Bitcoin, quais os dados que ele oferece, o preço e metadados adicionais como localização, tags, etc.

Um endereço Bitcoin é anexado ao dispositivo IoT que precisa anotar uma solicitação de dados ao receber Bitcoins e precisa ser capaz de criar e publicar uma transação contendo os dados solicitados. A maneira usada pelos desenvolvedores para utilizar a própria rede Bitcoin para vender dados é a utilização da transação do tipo *OP RETURN*.

Esse trabalho demonstrou um processo simples usando a rede Bitcoin com certas limitações, como por exemplo: os dados adquiridos estão disponíveis publicamente na Blockchain. Isso pode ser resolvido criptografando os dados com a chave pública do solicitante, em que o cliente requisitante descriptografa os dados usando sua chave privada.

Em [Zhang and Wen, 2015], os autores propõem uma arquitetura de comércio eletrônico projetada especificamente para as mercadorias IoT, baseada no protocolo do Bitcoin. Foram utilizadas Corporações Autônomas Distribuídas (DACs) como a entidade de transação para lidar com os dados de dispositivos e propriedade inteligente negociados. Nesse modelo as pessoas podem negociar com DACs para obter mercadorias IoT, utilizando criptomoedas baseadas no protocolo Bitcoin. Para trocar os dados do dispositivo são usados chaves eletrônicas e contratos inteligentes.

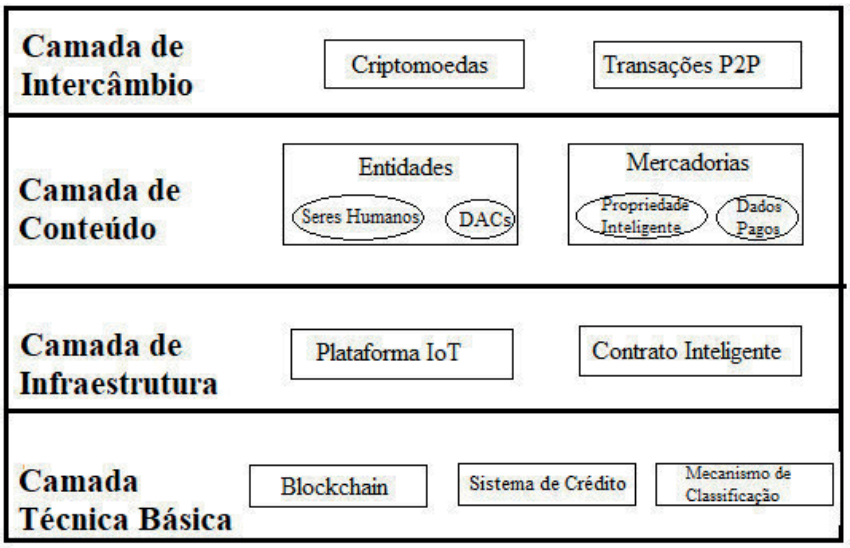

Figura 4.10. Modelo de Negócio para IoT usando Blockchain. Adaptado de: [Zhang and Wen, 2015]

Conforme mostrado na figura 4.10, são propostas 4 camadas para o modelo IoT de comércio eletrônico, que são: camada técnica básica; camada de infraestrutura; camada de conteúdo e camada de intercâmbio. A camada técnica básica inclui o módulo do mecanismo de classificação das mercadorias, módulo de algoritmo de crédito para efetuar o gerenciamento das carteiras e o módulo Blockchain Bitcoin, que foi a criptomoeda adotada pelo projeto. A camada de infraestrutura contém a plataforma do serviço de informações IoT e a plataforma de contratos inteligentes. A camada de conteúdo inclui duas partes: entidades participante e as mercadorias IoT. As Entidades consistem em DAC e seres humanos. DACs são executados automaticamente sem a interferência do ser humano, cada DAC pode comprar produtos de outros DAC como clientes, enquanto isso, todos podem emitir suas próprias mercadorias IoT. As mercadorias são propriedades inteligentes e dados coletados de sensores. As propriedades inteligentes podem ser obras

de arte, bens duráveis como carros, casas e energia como eletricidade, água, gás e óleo que podem ser controlados e quantificadas por dispositivos digitais por chaves eletrônicas ou sistema de controle de acesso. A camada de intercâmbio inclui o sistema de transações P2P que é o núcleo do modelo de negócios da IoT juntamente com com a criptomoeda escolhida que é a Bitcoin.

O emprego da tecnologia Blockchain com a finalidade de introduzir a funcionalidade de transações econômicas para IoT foi abordado por uma série de propostas e aplicativos, incluindo:

- **ADEPT** [Panikkar et al., 2014] - Telemetria Autônoma Peer-to-Peer descentralizada (do inglês, Automated Decentralized P2P Telemetry) é um sistema IoT descentralizado criado por uma parceria entre a IBM e a Samsung que utiliza elementos do projeto do Bitcoin para construir uma rede de dispositivos distribuídos (uma IoT descentralizada) permitindo que bilhões de dispositivos transmitam transações entre pares e realizem auto-manutenção, fornecendo identificação e autenticação seguras aos usuários. O conceito ADEPT utiliza a Blockchain para fornecer a espinha dorsal do sistema, utilizando uma mistura de prova de trabalho e prova de participação para transações seguras. Esta plataforma foi testada em vários cenários, incluindo um que envolve uma máquina de lavar inteligente que pode automaticamente comprar e pagar por detergente com Bitcoin ou ether e será capaz de negociar os melhores preços dos produtos de limpeza baseado nas preferências do seu proprietário. De acordo com o documento, uma máquina de lavar Samsung W9000 reconfigurada para funcionar dentro do sistema ADEPT usando contratos inteligentes para emitir comandos para um revendedor de detergente para receber novos suprimentos. Esses contratos dão ao dispositivo a capacidade de pagar pela própria encomenda e depois receber uma mensagem do revendedor de que o detergente foi pago e enviado. Essa informação é transmitida para o *smartphone* do proprietário da lavadora, um dispositivo que também é conectado à rede da *smart home*.

- **Filament** [Crosby et al., 2016] - É um sistema desenvolvido para permitir que os dispositivos tenham identidades únicas em um livro público e possam descobrir, comunicar e interagir uns com os outros de forma autônoma e distribuída. Além disso, os dispositivos envolvidos podem trocar valor diretamente ou indiretamente com uma ampla gama de entidades. Por exemplo, eles poderiam vender dados sobre condições ambientais para uma agência meteorológica. O objetivo é criar um diretório de dispositivos inteligentes que permita que os dispositivos IoT Filament se comuniquem de forma segura, executem contratos inteligentes e enviem microtransações. A pilha de tecnologia de Filament usa cinco tecnologias: blockname; telehash; contratos inteligentes; pennybank e BitTorrent. Usando o blockname, os dispositivos são capazes de criar um identificador exclusivo que é armazenado em uma parte do chip incorporado no dispositivo e gravado no bloco. O Telehash, por sua vez, fornece comunicações criptografadas de ponta a ponta e o BitTorrent permite o compartilhamento de arquivos. Os pagamentos pelo uso dos dispositivos são tratados por contratos inteligentes, o que permite que os termos dos pagamentos e o acesso ao dispositivo sejam controlados por esses programas. O Filament usa

um protocolo baseado em Bitcoin que foi desenvolvido para microtransações em sua plataforma, chamado Pennybank, devido a restrições específicas de dispositivos IoT. Os dispositivos IoT não são de alta potência e nem sempre estão online. Assim, o Pennybank cria um serviço de garantia entre dois dispositivos IoT, permitindo que eles liquidem as transações quando estiverem conectados on-line.

- **Tilepay** [Kennedy and Duranleau,] - É um sistema que oferece um mercado online seguro e descentralizado onde os usuários podem registrar seus dispositivos na cadeia de blocos e vender seus dados em tempo real em troca de moeda digital, utilizando a cadeia de bloco Bitcoin. Ao usar a Blockchain do Bitcoin, o registro, a autenticidade, a segurança e os dados disponíveis serão transmitidos pelo livro razão descentralizado. O proprietário dos dispositivos publicam seu perfil de dados através do *Marketplace* e as empresas podem comprar dados.

- **Watson IoT Platform** [Kshetri, 2017] - Essa plataforma da IBM permite que os dispositivos IoT enviem dados para Blockchain privadas. Todos os parceiros de negócios, que possuem uma Blockchain IBM podem acessar e fornecer dados de seus dispositivos IoT sem a necessidade de um controle ou gerenciamento central. Podem verificar cada transação, evitando disputas e garantindo que cada parceiro seja responsabilizado por suas funções individuais na transação global. A Blockchain IBM fornece a infra-estrutura de uma rede Blockchain privada que replica os dados do dispositivo e valida a transação através de *smart contract* seguros. A plataforma Watson IoT traduz os dados existentes do dispositivo, de um ou mais tipos de dispositivo, para o formato necessário para as APIs do contrato Blockchain. O contrato Blockchain não precisa conhecer os detalhes específicos dos seus dispositivos IoT. A plataforma Watson IoT filtra os eventos do dispositivo e envia apenas os dados necessários para o contrato.

- **IOTA** [Popov, 2016] - é uma criptomoeda desenvolvida especificamente para a comercialização de dados de dispositivos IoT. Ao invés de utilizar uma Blockchain global a IOTA usa um DAG (Grafo Acíclico Dirigido), as arestas correspondem as transações e os pesos a quantidade de vezes que ela foi confirmada. A ideia principal é que para realizar uma transação um nó precisa primeiro realizar uma série de verificações em transações anteriores com o objetivo de aprová-las. Não há diferenciação entre os nós, e todos são responsáveis por aprovar as transações, o que, segundo o autor, garante à rede uma maior escalabilidade: quanto maior a quantidade de transações, mais eficiente ela se torna. A rede IOTA foi projetada para a IoT, com uma estrutura modular e leve, que pode ser facilmente integrada a equipamentos eletrônicos, permitindo automação e a formação de um mercado M2M, em que os nós não precisam estar ligados diretamente à Internet, bastando uma conexão *bluetooth* entre eles.

Alguns outros casos de uso interessantes envolvendo monetização de dados com Blockchain e dispositivos IoT estão sendo estudados, como por exemplo, organizações como Nasdaq e Chain of Things conduzem pesquisas sobre aplicações que podem ajudar a tornar as fontes de energia renováveis disponíveis para o público em geral, onde a energia produzida por painéis solares IoT gera valor em criptomoedas que serão registrados na

Blockchain. Assim, qualquer pessoa que ingressar na rede pode fazer investimentos em tecnologia de energia renovável.

4.4.3. Uso de Blockchain em Computação Segura entre Múltiplos Participantes

Considere o seguinte problema: dois milionários interessados em saber qual deles possui a maior fortuna sem revelar a sua própria para o outro ou para terceiros. Este é o famoso problema dos milionários proposto por Yao [Yao, 1982], que foi resolvido com um protocolo para computação segura entre dois participantes. O MPC é a generalização desta solução para múltiplos participantes, é definido como o problema de *N* participantes para calcular uma função com suas entradas de forma segura, onde a segurança significa garantir o resultado correto e a privacidade das entradas, mesmo com a presença de alguns participantes maliciosos. Ao final, cada participante obterá apenas o resultado da função e não conhecerá as entradas dos demais participantes. Seu uso abre caminho para diversas aplicações como, votação na internet, mineração e compartilhamento de dados dentre outras.

Partindo do princípio que para realizar qualquer função são necessários apenas circuitos aditivos e multiplicativos, basta construir blocos MPC para adições e multiplicações e depois usar estes blocos para qualquer outras funções aritméticas. Assim, os protocolos propostos para MPC buscam realizar estas duas funções principais, normalmente usando circuitos de Yao [Yao, 1982] ou baseados em compartilhamento de segredo [Shamir, 1979] ou suas variantes. Para realizar estas funções, é necessários que os participantes troquem mensagens, para os circuitos aditivos as trocas de mensagens crescem linearmente com o número de participantes, mas para os circuitos multiplicativos são necessários $O(n^2)$ comunicações. Este fato torna a implementação da MPC restrita a poucos participantes e cenários específicos. Ao longo dos anos surgiram propostas para otimizar as soluções e aumentar a quantidade de participantes [Cramer et al., 1999, Gennaro et al., 1998, Zyskind et al., 2015a].

Na formulação de problemas de computação segura entre múltiplos-participantes, é comum adotar dois modelos de protocolos: O Modelo Semi-honesto e o Modelo Malicioso.

- **Modelo Semi-Honesto:** No modelo semi-honesto as partes seguem o protocolo apropriadamente e guardam todos os passos intermediários das computações para uma posterior análise, a fim de conseguir inferir informações secretas da outra parte.
- **Modelo Malicioso:** No modelo malicioso uma parte,maliciosa, não precisa seguir o protocolo, podendo agir de forma arbitrária ao executar o protocolo, abortar a execução a qualquer instante, usar informações falsas e guardar os passos intermediários para análise posterior.

Enigma [Zyskind et al., 2015a] é uma plataforma para MPC com garantia de privacidade. Usa a Blockchain como controladora da rede, gerenciando o controle de acesso, e servindo como log de eventos para o compartilhamento dos segredos. Tem a capacidade de computar as funções em ambos os modelos de adversários, e é escalável. Cada nó recebe suas entradas e grava as suas saídas na Blockchain. A computação é dividida entre parcelas dos participantes, assim cada parcela realiza uma tarefa e as partes são unidas ao final

para chegar ao resultado. Essa compartimentação permite que haja um maior controle na replicação dos dados, melhorando a escalabilidade do sistema e permitindo um maior número de participantes. Logo um dos seus possíveis usos é exatamente em dispositivos IoT que lidem com dados sensíveis.

Outro trabalho [Yue et al., 2016] usa a Blockchain Blockchain para realizar controle de acesso e armazenamento dos dados dos pacientes. O autor considera que o uso dos dados dos pacientes sem seu consentimento é um ataque a privacidade, mas também descreve a importância da utilização destes dados par a pesquisa médica. Ele classifica os dados em dois tipos: privados e públicos. Qualquer pesquisador ou entidade governamental poderá utilizar os dados públicos. Os dados privados somente poderão ser utilizados via MPC. Assim, o uso de MPC torna possível saber, por exemplo, a quantidade de pacientes que possuem AIDS e pertencem a grupos de risco. Torna possível extrair conhecimento sobre esses dados sem revelar a privacidade dos pacientes.

4.4.4. Uso de Blockchain para garantir segurança em Ambientes Inteligentes

A segurança e a privacidade da Internet da Coisas (IoT) continuam a ser um grande desafio, principalmente devido à enorme escala e à natureza distribuída das redes IoT. As abordagens baseadas em blocos oferecem segurança descentralizada e privacidade, mas envolvem consumo excessivo de energia e atrasos que não são adequados para a maioria dos dispositivos IoT com recursos limitados. Dorri [Dorri et al., 2016, Dorri et al., 2017b] apresenta uma solução leve de um Blockchain especialmente orientada para uso com IoT, eliminando a Prova de Trabalho e o conceito de moedas. Sua abordagem foi exemplificada em uma implementação em Ambientes Inteligentes (*Smart Home*) e consiste em três estruturas principais: armazenamento em nuvem, uma camada de sobreposição e casa inteligente. Cada casa inteligente está equipada com um dispositivo com maior poder computacional que está sempre *on-line*. Este dispositivo é chamado de "minerador", pelo autor, e é responsável por lidar com todas as comunicações dentro e fora da casa. O minerador mantém uma Blockchain privada, usada para controlar e auditar as comunicações e prover controle de acesso entre os dispositivos. Neste cenário a PoW torna-se desnecessária, pois somete um dispositivo terá o trabalho de manter a Blockchain. Os demais dispositivos da casa recebem um par de chaves para que possam realizar transações. Como exemplo, caso um sensor de presença queira ligar uma lampada ele enviará uma transação para a lampada, que irá verificar na Blockchain se aquele sensor tem permissão para acendê-la.

4.4.5. Ataques à Blockchain

No mundo de Bitcoin, as transações são consideradas válidas quando estão em um bloco e confirmadas sempre que exista uma certa quantidade blocos com altura maior na cadeia. Conforme descrito anteriormente, podem surgir bifurcações de curta duração, que tendem a ser resolvidas de acordo com a regra da cadeia mais longa. A maioria dos das bifurcações ocorre naturalmente, sem má intensão, fazendo com que as poucas transações do lado descartado sejam atrasadas. Esta abordagem funciona bem, sob o pressuposto crucial de que nenhum atacante deve ser capaz de reunir tanto poder computacional que seja capaz de falsificar e publicar uma "cadeia alternativa"que tenha maior dificuldade total. Nesse caso, as regras de consenso fariam com que cadeia alternativa fosse adotada ao invés da

cadeia principal, a partir do ponto de bifurcação. Isso é teoricamente possível e é chamado de ataque dos 51% [Kroll et al., 2013]. Conforme amplamente debatido até agora a segurança da Bitcoin depende do consenso distribuído alcançado pela prova de trabalho. Assume-se, que não há conluio de mineradores, ou seja, nenhum grupo de coordenado de mineradores (ou um único) pode possuir mais de 50% da capacidade computacional da rede.

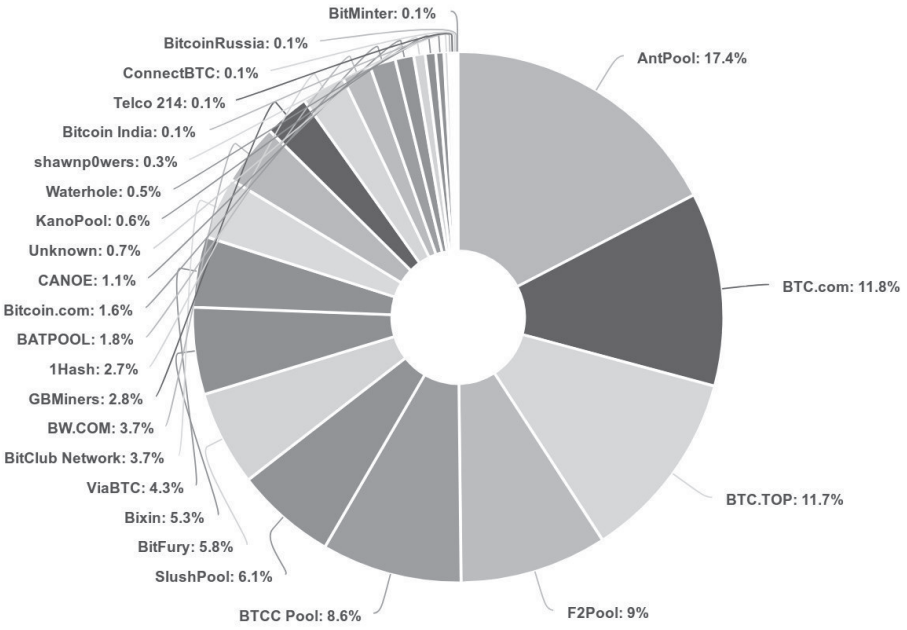

Figura 4.11. Cooperativas de mineradores. Fonte: www.blockchain.info acessado em: 21/08/2017.

No entanto, esta suposição é questionável. Primeiramente, é observado que há algum tempo que a mineração passou a ser organizada em cooperativas de mineradores (*mining pool*) que juntam esforços e compartilham as recompensas. Na Figura 4.11 é possível observar as maiores cooperativas em atividade. Em segundo lugar, não existe nenhuma entidade reguladora e nenhum minerador é obrigado a seguir o protocolo. Portanto uma cooperativa, que possua maioria de poder de computação, pode alterar as regras que são aplicadas pelo mecanismo de consenso e os mineradores que não participam da cooperativa provavelmente serão obrigados a se juntar a ela. Por exemplo, uma cooperativa, com mais de 50% do poder computacional da rede, pode escolher aceitar blocos vindos de outros mineradores na razão de 2:1, ou seja, a cada dois blocos enviados pelos nós honestos apenas um será aceito, e isso será possível pois a cooperativa possuirá o poder de manipular o consenso. Os mineradores honestos terão seus blocos ignorados e, portanto, perderão os pagamentos. O comportamento da cooperativa pode ser inclusive de realizar a negação de serviço a algum minerador ou a alguma transação, pois eles terão o poder de não incluir estas transações em nenhum de seus blocos, e caso outros mineradores o façam, eles podem gerar forks, rejeitando assim uma transação.

Os mineradores maliciosos podem desviar seu comportamento do padrão ao não divulgarem imediatamente os seus blocos recém minerados. Estes ataques são chama-

dos na literatura de *Selfish Mining* [Eyal and Sirer, 2014, Nayak et al., 2016], o qual será abordado com mais detalhes adiante. Primeiramente é preciso entender como a latência na propagação dos blocos e o tempo alvo para inclusão de novos blocos afetam o mecanismo de consenso.

- **Análise da latência de propagação dos blocos:** A propagação de mensagens na rede segue o protocolo P2P próprio. Um nó, ao receber um novo bloco, o retransmitirá aos seus vizinhos. Antes de iniciar a transmissão, ele faz uma série de verificações a fim de garantir que somente blocos válidos sejam propagados. Cada nó que recebe um novo bloco faz essas verificações independentemente. Após isso, o nó envia uma mensagem de inventário (*INV*), informando aos seus vizinhos que possui um novo bloco e sua altura. Caso estes vizinhos ainda não tenham recebido este bloco por outros nós eles enviam uma mensagem solicitando o novo bloco(*GET_DATA*). Somente então será iniciada a transmissão efetiva do novo bloco, Fig. 4.12. A soma de todos esses tempos, verificação e propagação da mensagens, durante a propagação de um bloco, é chamada de Latência de propagação dos blocos. Decker [Decker and Wattenhofer, 2013] fez a análise do tempo de propagação de 10.000 blocos, com diferentes tamanhos, e verificou que a mediana desde a criação do bloco até o recebimento por um determinado nó foi de 6,5 segundos e a média de 12,6 segundos. Outra observação interessante foi que após 40 segundos, 5% dos nós ainda não haviam recebido o novo bloco.

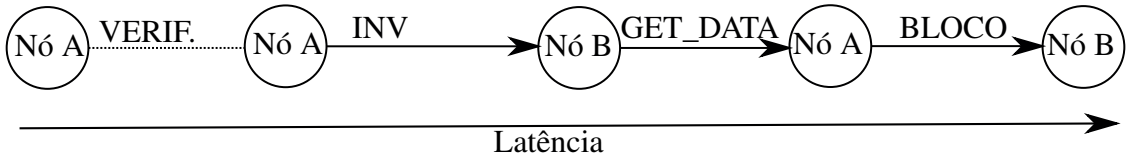

Figura 4.12. Latência de propagação dos blocos

- **Análise do tempo para inclusão de novos blocos:** O intervalo para a inclusão de novos blocos é crucial para a quantidade de *forks* observados na rede. Quanto menor for esse intervalo, maior será a quantidade de blocos gerados e consequentemente maior será a probabilidade de ocorrência de forks e de blocos órfãos.

  Decker [Decker and Wattenhofer, 2013] verificou a existência de 169 forks durante sua observação de 10.000 blocos, ou seja 1,69% dos blocos foram descartados pela ocorrência de forks. Gervais [Gervais et al., 2016] analisou o impacto da diminuição deste tempo, variando de 0,5 segundos a 25 minutos e usando 10.000 blocos em simulações no NS-3, conforme na Tabela 4.2

Caso um atacante resolva desviar do comportamento padrão e manter secretamente uma cadeia onde somente ele produza blocos, e adote uma heurística própria para divulgação destes blocos. Neste momento, todos, inclusive o atacante estariam minerando sobre o bloco *n*. Ao realizar esta ação, caso ele consiga produzir o próximo bloco, o atacante conseguirá uma vantagem em relação aos demais mineradores, mesmo sem possuir maior poder computacional, pois ele poderá iniciar o processo de mineração do

Tabela 4.2. Impacto do intervalo entre blocos na taxa de forks

Intervalo entre blocos	Taxa de Forks(%)	Mediana do tempo de propagação
0,5s	38,15	0,82
1s	26,74	0,82
2s	16,65	0,84
5s	8,64	0,89
10s	4,77	1
20s	3,2	1,21
30s	2,54	1,43
1m	2,15	2,08
2,5m	1,82	4,18
10m	1,51	14,7
25m	1,72	35,73

bloco $n+1$ antes de todos os outros mineradores. Se o atacante receber um bloco da rede ele pode decidir adotar este bloco e jogar fora seu trabalho, ou ignorar o bloco recebido e continuar minerando na cadeia privada. Como ele começou a minerar antes, há uma probabilidade de liberar bloco $n+1$ antes dos outros, gerando um fork com altura maior que a cadeia principal. Como os demais nós se comportam honestamente, eles também irão adotar esta cadeia e o atacante atingirá seu objetivo. Um esquema simples é mostrado na Figura 4.13 onde após a liberação dos blocos n e $n+1$ pelo atacante os nós honestos adotaram esta cadeia e produziram o bloco $n+2$. Este ataque é conhecido como *Selfish Mining* [Eyal and Sirer, 2014] e o atacante é chamado de "minerador egoísta".

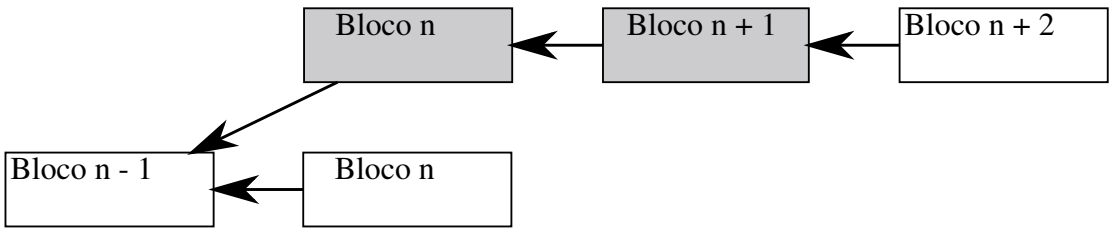

Figura 4.13. fork

Como vimos acima, uma parcela da latência de propagação dos blocos é gerada pela obrigatoriedade de verificação dos novos blocos por todos os nós. Caso um atacante controle alguns nós da rede ele pode tirar vantagem deste fato para amplificar o ataque do minerador egoísta. Estes nós controlados pelo atacante podem ser configurados para não realizar a verificação dos blocos minerados pelo atacante e retransmiti-los tão logo eles cheguem. Assim os blocos do atacante alcançam os outros nós em menor tempo que os nós honestos. Isso pode ser uma vantagem. Outra observação a respeito deste ataque é que era de se esperar que o total de blocos adicionados à cadeia seja a soma dos blocos produzidos pelo atacante e pelos nós honestos. Mas a ocorrência de forks gera blocos que serão descartados, *stale blocks*. Assim, a quantidade total de blocos incluídos é menor

que o total de blocos produzidos. Isso é importante pois se for desconsiderada a entrada de novos mineradores na rede quando os nós forem recalcular a nova dificuldade, esta será menor que a dificuldade anterior.

Em [Eyal and Sirer, 2014], Eyal faz uma análise matemática e propõe um modelo de transição de estados para capturar o melhor momento do atacante liberar seus blocos privados. Ele analisa as probabilidades de ocorrência de cada estado e chega a conclusão que para o atacante conseguir êxito, ou seja, publicar mais blocos que a cadeia honesta, seu poder de mineração deve satisfazer a seguinte inequação:

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2}$$

, onde α é o poder de mineração do atacante e γ é a razão dos nós honestos que escolhem minerar a cadeia desonesta. Isso é uma vantagem ao atacante, pois ele necessitará de menor poder para conseguir suplantar os nós honestos. Assim, a partir desta inequação é obtido o menor valor de α para um determinado γ. O pior caso para o atacante é quando nenhum nó honesto adota a sua cadeia, neste caso é necessário que o atacante possua um terço do poder computacional da rede.

Nayak et al. [Nayak et al., 2016] amplia a pesquisa de Eyal e verifica que o ataque proposto anteriormente não é ótimo. Ele propõe novas estratégias para aumentar o ganho do atacante, levando em conta não apenas o tamanho das cadeias, mas também seu poder computacional. Por exemplo, mesmo que o atacante esteja perdendo a corrida caso ele possua uma parcela significa de poder computacional, é melhor que ele continue a minerar em sua cadeia privada, pois ele terá uma grande chance de alcançar e ultrapassar a cadeia honesta. Outra contribuição de seu trabalho é mostrar que se o *Selfish Mining* for combinado com o *Eclipse* [Heilman et al., 2015], quando o atacante controla todas as conexões com um determinado nó, o atacante aumentará seus ganhos e surpreendentemente, com determinados parâmetros, o nó que foi alvo do *Eclipse* também conseguirá publicar mais blocos, em relação aos nós honestos.

O autor modela a mineração como um processo de decisão de Markov, que usa como informação as transições de estado quando os mineradores acham um novo bloco. Quando um nó honesto acha o novo bloco ele o publica imediatamente, enquanto o atacante mantém seus blocos ocultos. Depois, ele captura quantos blocos cada cadeia estão a frente uma da outra, e se os nós honestos estão minerando sobre a cadeia honesta ou desonesta. Suas estratégias chamadas de *Stubborn Mining*, são:

- ***Lead Stubborn:*** Vimos anteriormente que γ é um fator importante, pois quanto mais nós honestos estão minerando na cadeia do atacante menor será o α necessário. Vimos também que a latência de propagação dos blocos influencia fortemente quais blocos cada nós recebe primeiro. Assim, uma das estratégias adotadas pelo autor define que, caso o atacante esteja liderando por um bloco, tão logo os nós honestos liberem um bloco, o atacante também libera o seu. O objetivo é que seu bloco alcance uma parcela dos nós honestos que irão adotá-lo como referência para mineração, aumentando o γ e a probabilidade do atacante vencer a corrida. Caso esteja vencendo por dois ou mais ele mantém sua cadeia privada, somente revelando os blocos quando a diferença alcançar 1.

- ***Trail Stubborn:*** Quando a cadeia privada do atacante está atrás da pública. O atacante continua a mineração na cadeia privada ao invés de abandoná-la, na esperança de alcançar e ultrapassar a cadeia pública. Esta estratégia se mostra promissora caso o atacante possua um certo poder computacional.

- ***Equal Fork Stubborn:*** Esta estratégia é usada quando os nós honestos igualam a corrida e o atacante continua minerando em sua cadeia até que fique um bloco a frente, quando então ele libera sua cadeia.

O Ataque *Eclipse* [Heilman et al., 2015] é um ataque a nível de rede. Ocorre quando um atacante monopoliza todas as conexões de um determinado nó, isolando a vítima e filtrando todas as mensagens enviadas e recebidas. Como resultado, a vítima tem uma visão diferente da cadeia, pode ter seus blocos impedidos de serem incluídos na cadeia e pode ser forçada a trabalhar na cadeia do atacante. Na rede Bitcoin, os nós mantém no máximo 125 conexões com seus vizinhos, sendo 8 conexões de saída, e 117 de entrada. As conexões de saída são aquelas iniciadas pelo próprio nó, e as de entrada são as que outros nós solicitam, conforme vimos na seção que trata da rede P2P. Para armazenar o endereço destas conexões os nós usam duas tabelas, uma tabela de conexões bem sucedidas, onde são armazenadas as informações de todas as conexões estabelecidas, de entrada e de saída, e uma tabela de endereços fornecidos, por solicitação ou não. A primeira é chamada de *Tried Table* e a segunda de *New Table*.

- ***Tried Table:*** É formada por 64 recipientes que podem armazenar 64 endereços cada. Os recipientes são escolhidos da seguinte forma: Quando o nó é iniciado ele escolhe um valor aleatório SK, e calcula:

  $$Recipiente = Hash(SK, Grupo, Hash(SK, IP)\%4)\%64,$$

  onde o Grupo é o prefixo /16 do endereço IP. Quando um nó estabelece uma conexão ele então mapeia o IP do novo vizinho para um recipiente. Caso o recipiente esteja cheio, o nó então chama uma função para remover endereços do recipiente. Quatro endereços são escolhidos aleatoriamente e o mais antigo é então movido para a *New Table*.

- ***New Table:*** É formada por 256 recipientes e cada um armazena até 64 endereços. É preenchida pelos endereços removidos da *Tried Table* ou por endereços fornecidos pelos *DNS seedres* ou por mensagens *ADDR*, que são mensagens para informar novos endereços aos vizinhos. De forma similar ao item anterior, aqui também é executada uma função para mapear o recipiente e existe uma função de remoção de endereços antigos.

O ataque consiste em encher a *Tried Table* com endereços controlados pelo atacante e encher a *New Table* com endereços inválidos. Desta forma sempre que a vítima buscar uma nova conexão ela se conectará a um endereço controlado pelo atacante. A *New Table* é preenchida com endereços inválidos para que o atacante economize IPs. Assim, basta realizar duas rotinas repetidas vezes para popular as tabelas da vítima. Primeiro, para estabelecer conexões de entrada, basta que o atacante solicite uma conexão. Em

seguida, desconecta e solicita uma nova conexão com outro endereço. Isto basta para preencher a *Tried Table*. Para o atacante preencher a *New Table*, é necessaŕio enviar diversas mensagens *ADDR* com endereços inválidos para a vítima. Cada *ADDR* pode conter até 1000 endereços. São usados endereços classe C ou reservados para uso futuro, como a faixa destinada ao multicast, pois o atacante precisa de no mínimo 16384 endereços para preencher a *New Table*. Em seus experimentos, com uma *botnet* de apenas 400 bots foi capaz de popular totalmente a *New Table* e 60% da *Tried Table*, alcançando sucesso em controlar todas as conexões da vítima em 80% das vezes.

Com o intuito de exemplificar melhor o ataque *Selfish Mining* forma realizadas diversas simulações utilizando o módulo do NS-3 desenvolvido por [Gervais et al., 2016]. O módulo foi desenvolvido com o objetivo de analisar o impacto na taxa de forks, *throughput* da rede, o tempo de propagação dos blocos e ganho com gasto duplo. A conexão entre os blocos é feita com canais ponto-a-ponto, abstraindo dispositivos intermediários. Para configurar as características do canal (latência e banda) foram usados dados estatísticos de diversas fontes, como Verizon e testmy.net. A prova de trabalho é modelada atribuindo valores de poder de mineração aos nós, distribuindo estatisticamente a geração dos blocos. Os dados de entrada são: a taxa de inclusão de novos blocos, tamanho do bloco e valor do gasto duplo. Ao analisar a Figura 4.11, chega-se a conclusão que as 15 maiores cooperativas possuem 96,3% do poder de mineração. Por esse motivo, foram simulados 16 nós, representando as 15 cooperativas e os outros mineradores agrupados. Os nós honestos adotam o protocolo padrão, enquanto o atacante segue a heurística proposta pelo autor:

- **Adotar:** O Adversário adota a cadeia honesta. Isto corresponde a reiniciar o ataque, pois o atacante infere que os nós honestos possuem uma probabilidade maior de vencer a corrida.
- **Sobrepor:** Ocorre quando o atacante possui um bloco a mais que a cadeia honesta. É uma boa estratégia quando existe uma parcela dos nós honestos minerando sobre a cadeia do atacante.
- **Igualar:** O atacante publica tantos blocos quanto os publicados pelos mineradores honestos. Esta ação tem por objetivo fazer que alguns nós honestos minerem sobre a cadeia do atacante. Em seguida, o atacante pode usar a ação *Sobrepor*.
- **Esperar:** O atacante minera constantemente em sua cadeia, sem a revelar.
- **Publicar:** Corresponde a revelar sua cadeia.

Primeiramente foram realizadas 30 rodadas de simulação gerando 10.000 blocos em cada. As simulações foram feitas com 16 nós mineradores e todos seguiam o protocolo padrão, sem atacante. Como resultado uma taxa de 0,13% de forks foi obtida. Na segunda rodada de simulações um dos nós foi escolhido como atacante. Inicialmente ele possui 20% do poder de mineração total, que foi incrementado até 50%. É possível observar na Figura 4.14 que, a medida que a força do atacante é incrementada, há um considerável aumento na taxa de blocos descartados e consequentemente diminuição da quantidade de blocos na cadeia principal.

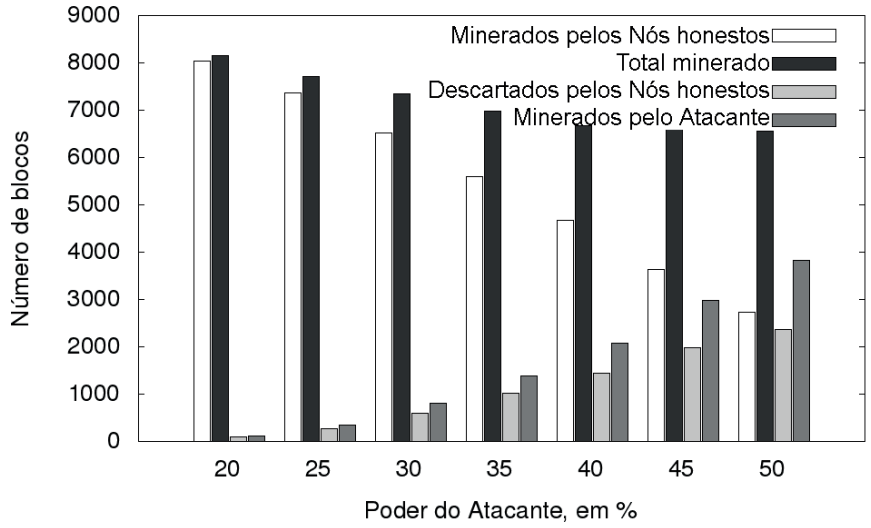

Figura 4.14. Evolução dos blocos descartados(*Stale Blocks)*

4.5. Considerações Finais, Perspectivas Futuras e Problemas em Aberto

A IoT processa e troca grandes quantidades de dados sem a intervenção humana e estes dados frequentemente possuem informações que podem ser críticas em relação a segurança e privacidade. Portanto, são alvos atraentes aos atacantes. Normalmente esses dispositivos são de baixa energia e de baixo poder computacional e devem dedicar seus parcos recursos a suas atividades principais, o que torna a tarefa de suporte a segurança e privacidade bastante desafiadora. Métodos de segurança tradicionais tendem a ser caros em termos computacionais e energéticos. Além disso, muitos dos *frameworks* de segurança são altamente centralizados e, portanto, não são necessariamente adequados para o cenário IoT devido à dificuldade de escalabilidade, e ao fato de se tornar um ponto único de falha. Consequentemente, a IoT exige uma proteção de segurança e privacidade leve, escalável e distribuído. A tecnologia Blockchain, que sustenta o Bitcoin, tem o potencial de superar esses desafios como resultado de sua natureza distribuída, segura e privada. Porém não é leve, necessitando de adaptações e otimizações.

A combinação de Blockchain e IoT pode ser bastante poderosa, pois o Blockchain pode dar resiliência a ataques e a capacidade de interagir com os pares de forma confiável e auditável. A contínua integração de Blockchain no domínio IoT causará transformações significativas em vários setores, trazendo novos modelos de negócios e nos fazendo reconsiderar como os sistemas e processos existentes são implementados.

A "cadeia de blocos"possibilita não meramente o movimento do dinheiro, mas pode também ser usada para transferir informações e a alocar recursos entre os dispositivos, habilitando o uso da Blockchain como um serviço [Swanson, 2015]. O mundo conectado pode incluir de forma útil a tecnologia Blockchain como uma camada para o que cada vez mais dispositivos (vestíveis, sensores, IoT, *smartphones, tablets, laptops*, casas, carros e cidades inteligentes) possam se beneficiar de suas características.

Blockchain, portanto, apresenta muitas promissoras oportunidades para o futuro da IoT. Os desafios, no entanto permanecem, como modelos de consenso e os custos

computacionais de verificação de transações. Mas ainda está nos estágios iniciais do desenvolvimento de cadeias de blocos, e esses obstáculos serão eventualmente superados, abrindo o caminho para muitas possibilidades.

Referências

[Ali et al., 2016] Ali, M., Nelson, J. C., Shea, R., and Freedman, M. J. (2016). Blockstack: A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference*, pages 181–194.

[Andrea, 2014] Andrea, A. (2014). *Mastering BitCoin*, volume 50.

[Ashton, 2011] Ashton, K. (2011). That 'internet of things' thing. *RFiD Journal*, 22(7).

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54.

[Back et al., 2002] Back, A. et al. (2002). Hashcash-a denial of service counter-measure.

[Bitcoin, 2009] Bitcoin (2009). Bitcoin developer reference. https://bitcoin.org/en/developer-reference. Accessed: 2017-07-30.

[Buterin, 2014] Buterin, V. (2014). Daos, dacs, das and more: An incomplete terminology guide. *Ethereum (accessed 12 July 2016) https://blog. ethereum. org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide*.

[Cachin, 2016] Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*.

[Castro and Liskov, 2002] Castro, M. and Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461.

[Christidis and Devetsikiotis, 2016] Christidis, K. and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303.

[Conoscenti et al., 2016] Conoscenti, M., Vetrò, A., and De Martin, J. C. (2016). Blockchain for the internet of things: a systematic literature review. *Porto.Polito.It*.

[Cramer et al., 1999] Cramer, R., Damgård, I., Dziembowski, S., Hirt, M., and Rabin, T. (1999). Efficient multiparty computations secure against an adaptive adversary. In *Eurocrypt*, volume 99, pages 311–326. Springer.

[Crosby et al., 2016] Crosby, M., Pattanayak, P., Verma, S., and Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10.

[Cui, 2016] Cui, X. (2016). The internet of things. In *Ethical Ripples of Creativity and Innovation*, pages 61–68. Springer.

[De Filippi and Mauro, 2014] De Filippi, P. and Mauro, R. (2014). Ethereum: the decentralised platform that might displace today's institutions. *Internet Policy Review*, 25.

[Decker and Wattenhofer, 2013] Decker, C. and Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE.

[Dorri et al., 2016] Dorri, A., Kanhere, S. S., and Jurdak, R. (2016). Blockchain in internet of things: Challenges and Solutions. *arXiv:1608.05187 [cs]*.

[Dorri et al., 2017a] Dorri, A., Kanhere, S. S., and Jurdak, R. (2017a). Towards an Optimized BlockChain for IoT. *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation - IoTDI '17*, 6.

[Dorri et al., 2017b] Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017b). Blockchain for iot security and privacy: The case study of a smart home.

[Douceur, 2002] Douceur, J. R. (2002). The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer.

[Evans, 2011] Evans, D. (2011). A internet das coisas: como a próxima evolução da internet está mudando tudo. *CISCO IBSG*.

[Eyal, 2015] Eyal, I. (2015). The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE.

[Eyal and Sirer, 2014] Eyal, I. and Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer.

[Fenn and LeHong, 2011] Fenn, J. and LeHong, H. (2011). Hype cycle for emerging technologies, 2011. *Gartner, July*.

[Fischer et al., 1985] Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382.

[Gennaro et al., 1998] Gennaro, R., Rabin, M. O., and Rabin, T. (1998). Simplified vss and fast-track multiparty computations with applications to threshold cryptography. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 101–111. ACM.

[Gervais et al., 2016] Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. (2016). On the Security and Performance of Proof of Work Blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*.

[Gilbert and Handschuh, 2003] Gilbert, H. and Handschuh, H. (2003). Security analysis of sha-256 and sisters. In *International workshop on selected areas in cryptography*, pages 175–193. Springer.

[Gubbi et al., 2013] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

[Guillemin et al., 2009] Guillemin, P., Friess, P., et al. (2009). Internet of things strategic research roadmap. *The Cluster of European Research Projects, Tech. Rep*.

[Hankerson et al., 2006] Hankerson, D., Menezes, A. J., and Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer Science & Business Media.

[Heilman et al., 2015] Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. (2015). Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security*, pages 129–144.

[Herrera-Joancomartí, 2015] Herrera-Joancomartí, J. (2015). Research and challenges on bitcoin anonymity. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*, pages 3–16. Springer.

[Intel,] Intel. Proof of elapsed time (poet). available from: http://intelledger.github.io/.

[Jansma and Arrendondo, 2004] Jansma, N. and Arrendondo, B. (2004). Performance comparison of elliptic curve and rsa digital signatures. *nicj. net/files*.

[Kennedy and Duranleau,] Kennedy, S. and Duranleau, C. Tilepay. https://http://www.tilepay.org. Accessed: 2017-09-08.

[Kim, 2014] Kim, J. (2014). Safety, liveness and fault tolerance—the consensus choices stellar.

[Kim, 2016] Kim, T. H. (2016). A study of digital currency cryptography for business marketing and finance security. *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, 6(1):365–376.

[King and Nadal, 2012] King, S. and Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19.

[Koshy et al., 2014] Koshy, P., Koshy, D., and McDaniel, P. (2014). An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*, pages 469–485. Springer.

[Kroll et al., 2013] Kroll, J. A., Davey, I. C., and Felten, E. W. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013.

[Kshetri, 2017] Kshetri, N. (2017). Can blockchain strengthen the internet of things? *IT Professional*, 19(4):68–72.

[Merkle, 1987] Merkle, R. C. (1987). A digital signature based on a conventional encryption function. In *Conference on the Theory and Application of Cryptographic Techniques*.

[Moser et al., 2013] Moser, M., Bohme, R., and Breuker, D. (2013). An inquiry into money laundering tools in the bitcoin ecosystem. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–14. IEEE.

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

[Natoli and Gramoli, 2016] Natoli, C. and Gramoli, V. (2016). The blockchain anomaly. In *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on*, pages 310–317. IEEE.

[Nayak et al., 2016] Nayak, K., Kumar, S., Miller, A., and Shi, E. (2016). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 305–320. IEEE.

[Nguyen et al., 2015] Nguyen, K. T., Laurent, M., and Oualha, N. (2015). Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks*, 32.

[Ouaddah et al., 2017] Ouaddah, A., Elkalam, A. A., and Ouahman, A. A. (2017). FairAccess : a new Blockchain-based access control framework for the Internet of Things. *Security and Communication Networks*, 9.

[Panikkar et al., 2014] Panikkar, S., Nair, S., Brody, P., and Pureswaran, V. (2014). Adept: An iot practitioner perspective. *IBM Institute for Business Value*.

[Peña-López et al., 2005] Peña-López, I. et al. (2005). Itu internet report 2005: the internet of things.

[Pilkington, 2015] Pilkington, M. (2015). Blockchain technology: principles and applications. *Browser Download This Paper*.

[Pilkington, 2016] Pilkington, M. (2016). Blockchain technology: principles and applications. research handbook on digital transformations, edited by f. xavier olleros and majlinda zhegu.

[Popov, 2016] Popov, S. (2016). The tangle. *Available electronically at http://iotatoken.com/IOTA Whitepaper. pdf*.

[Recommendation, 2012] Recommendation, Y. (2012). 2060 «overview of internet of things». *ITU-T, Geneva*.

[Shamir, 1979] Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

[Spagnuolo et al., 2014] Spagnuolo, M., Maggi, F., and Zanero, S. (2014). Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer.

[Stallings, 1995] Stallings, W. (1995). *Network and internetwork security: principles and practice*, volume 1. Prentice Hall Englewood Cliffs.

[Strategy and Unit, 2005] Strategy, I. and Unit, P. (2005). Itu internet reports 2005: The internet of things. *Geneva: International Telecommunication Union (ITU)*.

[Swan, 2015a] Swan, M. (2015a). *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc.".

[Swan, 2015b] Swan, M. (2015b). Blockchain thinking: The brain as a dac (decentralized autonomous organization). In *Texas Bitcoin Conference*, pages 27–29.

[Swanson, 2015] Swanson, T. (2015). Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. *Report, available online, Apr*.

[Szabo, 1994] Szabo, N. (1994). Smart contracts. *Unpublished manuscript*.

[Szabo, 1997] Szabo, N. (1997). The idea of smart contracts. Nick Szabo's Papers and Concise Tutorials.

[Valenta and Rowan, 2015] Valenta, L. and Rowan, B. (2015). Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 112–126. Springer.

[Vasin, 2014] Vasin, P. (2014). Blackcoin's proof-of-stake protocol v2.

[Wood, 2014] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151.

[Wörner and von Bomhard, 2014] Wörner, D. and von Bomhard, T. (2014). When your sensor earns money. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct*.

[Wörner and von Bomhard, 2014] Wörner, D. and von Bomhard, T. (2014). When your sensor earns money: exchanging data for cash with bitcoin. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 295–298. ACM.

[Yao, 1982] Yao, A. C. (1982). Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 160–164. IEEE.

[Yue et al., 2016] Yue, X., Wang, H., Jin, D., Li, M., and Jiang, W. (2016). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218.

[Zhang and Wen, 2015] Zhang, Y. and Wen, J. (2015). An iot electric business model based on the protocol of bitcoin. In *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pages 184–191. IEEE.

[Zimmermann, 1995] Zimmermann, P. R. (1995). *The official PGP user's guide*. MIT press.

[Zyskind et al., 2015a] Zyskind, G., Nathan, O., and Pentland, A. (2015a). Enigma: Decentralized Computation Platform with Guaranteed Privacy. *arXiv:1506.03471 [cs]*.

[Zyskind et al., 2015b] Zyskind, G., Nathan, O., and Pentland, A. S. (2015b). Decentralizing privacy: Using blockchain to protect personal data. *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015*.