UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

NATALIA PAOLA DE VASCONCELOS COMETTI

UM ESTUDO SOBRE A TECNOLOGIA *BLOCKCHAIN* E SUA APLICAÇÃO EM SISTEMAS DE VOTAÇÃO

TRABALHO DE GRADUAÇÃO

RECIFE
2016

NATALIA PAOLA DE VASCONCELOS COMETTI

UM ESTUDO SOBRE A TECNOLOGIA *BLOCKCHAIN* E SUA APLICAÇÃO EM SISTEMAS DE VOTAÇÃO

TRABALHO DE GRADUAÇÃO

Trabalho de Graduação apresentado à graduação de Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Ruy José Guerra Barretto de Queiroz (ruy@cin.ufpe.br)

RECIFE
2016

NATALIA PAOLA DE VASCONCELOS COMETTI

UM ESTUDO SOBRE A TECNOLOGIA *BLOCKCHAIN* E SUA APLICAÇÃO EM SISTEMAS DE VOTAÇÃO

TRABALHO DE GRADUAÇÃO

Trabalho de Graduação apresentado à graduação de Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Recife, ___ de janeiro de 2016.

BANCA EXAMINADORA

_____

Prof.º Ruy José Guerra Barretto de Queiroz
(Orientador)

_____

Prof.º Vinicius Cardoso Garcia
(Avaliador)

AGRADECIMENTOS

Agradeço a minha família pelo apoio dado durante todo o curso, que foi essencial para eu chegar ao fim, e por compreender minha ausência em tantos momentos atarefados, como durante a realização desse trabalho.

Agradeço a meu namorado por toda a dedicação e paciência. Seu incentivo foi fundamental para que eu superasse os obstáculos encontrados nessa jornada.

Agradeço a meus amigos por sempre acreditarem em mim. Aos que moram distante, agradeço também por entenderem minha ausência durante a maior parte desse curso. Aos colegas de curso, agradeço por tornarem esses quatro anos e meio mais divertidos.

Agradeço a meus professores do curso por todos os ensinamentos, oportunidades e incentivos. Em especial, agradeço a meu orientador, professor Ruy, por todo o conhecimento passado durante o curso, e por guiar minha pesquisa nesse trabalho final.

Vocês são meus grandes incentivadores!

Muito, muito obrigada!

RESUMO

Os sistemas de votação utilizados atualmente são baseados em tecnologias sujeitas a vulnerabilidades e, por conta disso, eleições podem receber acusações de ilegitimidade. A tecnologia *blockchain* tornou-se conhecida pela segurança e transparência das transações financeiras do Bitcoin, que dispensa a necessidade de uma autoridade central. O objetivo desse trabalho é estudar o funcionamento da *blockchain*, bem como aspectos de sua segurança e privacidade, e, de posse desses conhecimentos, analisar sistemas de votação que propõem o aplicação da tecnologia para tornar votações auditáveis, seguras e menos custosas.

Palavras-chave: Blockchain. Votação. Segurança. Privacidade.

ABSTRACT

Voting systems currently used are based on technologies that may present vulnerabilities, and therefore, elections may receive accusations of illegitimacy. Blockchain technology became known by the security and transparency of Bitcoin's financial transactions that exclude the need of a central authority. The aim of this work is to study blockchain's behavior, as well as its security and privacy aspects, and based on this knowledge analyze voting systems that propose the usage of this technology to turn voting auditable, secure and cheaper.

Keywords: Blockchain. Voting. Security. Privacy.

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização e motivação

Em 2008, Satoshi Nakamoto propôs em um artigo [1] uma nova moeda criptográfica, o Bitcoin. Essa moeda atraiu muito interesse por conta da nova tecnologia, hoje conhecida como *blockchain*, empregada na sua concepção. Tal inovação permite que o Bitcoin funcione sem nenhuma dependência com instituições bancárias ou qualquer entidade. Ao mesmo tempo, existe o benefício da verificação de transações de modo a evitar irregularidades e fraudes.

Inicialmente, após o invento do Bitcoin, várias cripto moedas surgiram, seguindo o mesmo modelo descentralizado porém com suas próprias peculiaridades, e por isso ficaram conhecidas como Altcoins [2]. Algumas dessas moedas procuraram melhorar a ideia por trás do Bitcoin em algum aspecto, mas a tecnologia do Bitcoin não está intrinsicamente ligada a operações financeiras: o conceito de *blockchain* pode ser aplicado em vários campos que usufruem da confiança distribuída, como por exemplo *smart contracts*, computação na nuvem, sistemas de votação, registro de propriedade intelectual, *crowdfunding*, entre outros [3].

Alguns estudos realizados em urnas eleitorais do Brasil [4] e dos Estados Unidos [5] demonstraram que a tecnologia utilizada nesses sistemas de votação tradicionais apresenta vulnerabilidades. A aplicação da tecnologia *blockchain* em sistemas de votação pode trazer os benefícios da segurança, transparência e auditabilidade, além de diminuir a dependência de confiança em autoridades centrais.

1.2 Objetivos

Os objetivos deste trabalho são:

- Explicar o funcionamento da *blockchain* do Bitcoin, evidenciando os aspectos de segurança e privacidade;
- Exibir os sistemas de votação que incorporam já contam com a *blockchain*, explicando como funcionam e de que modo usufruem da tecnologia;
- Analisar os sistemas revisados quanto à segurança, privacidade, adaptabilidade a realidade atual, entre outros aspectos.

1.3 Estrutura do documento

No capítulo 2, será apresentado o funcionamento da *blockchain* e os conceitos teóricos adjacentes, com enfoque no protocolo Bitcoin, enquanto que no capítulo 3 serão analisadas a segurança e a privacidade propiciadas pela tecnologia apresentada no capítulo anterior. No

capítulo 4, serão exibidos os sistemas de votação que utilizam tal tecnologia, bem como uma análise destes. A conclusão será apresentada no capítulo 5, junto a ideias de trabalhos futuros.

2 TECNOLOGIA BLOCKCHAIN

O conceito de *blockchain* foi introduzido por Nakamoto no ano de 2008, em um artigo [1] onde foi proposta a cripto moeda Bitcoin. Esta moeda, diferentemente das moedas digitais previamente criadas, tem como objetivo dispensar a confiança em qualquer autoridade central. Mas para atingir esse objetivo, foi preciso oferecer um novo meio de autenticação e registro de transações, de modo a evitar ações fraudulentas, e por isso todas as transações de Bitcoin são publicamente anunciadas, e organizadas em blocos. Um bloco de transações deve conter um registro do bloco anterior, de modo a encadear as transações cronologicamente. Cada transação é verificada pelos mineradores, que são usuários que emprestam recursos computacionais necessários para o funcionamento da plataforma financeira. Em troca, os mineradores que seguem o protocolo da rede podem receber bitcoins como recompensa.

Inicialmente, após o invento do Bitcoin, várias cripto moedas surgiram, seguindo o mesmo modelo descentralizado porém com suas próprias peculiaridades, e por isso ficaram conhecidas como Altcoins [2]. Algumas dessas moedas procuraram melhorar a ideia por trás do Bitcoin em algum aspecto, por exemplo: Zerocoin [6] e Dash [7], que têm o intuito de melhorar o aspecto da privacidade. O Zerocoin nunca chegou a ser lançado por ser dependente de uma integração ao Bitcoin, mas posteriormente deu origem ao Zerocash [8], que é sua versão mais eficiente e independente.

Mas a tecnologia subjacente ao Bitcoin não está intrinsicamente ligada a cripto moedas: o conceito *blockchain* tem sido aplicado em vários campos que usufruem da confiança distribuída, como por exemplo *smart contracts*, computação na nuvem, sistemas de votação, registro de propriedade intelectual, *crowdfunding*, entre outros [3].

Porém para melhor se adequar aos diferentes tópicos, a tecnologia precisou sofrer alterações, afinal o modelo de transação da *blockchain* proposta por Nakamoto [1] foi pensado para aplicações financeiras [9]. Com isso, surgiu a denominação *alternative chains* ou *altchains* [9], numa tentativa de separar o modelo original dos modelos posteriores. No entanto, o termo *blockchain* ainda é o mais utilizado para referir-se à tecnologia em qualquer que seja a área de aplicação.

Neste capítulo, é apresentada a tecnologia *blockchain*, começando por conceitos fundamentais para o entendimento do seu funcionamento e posteriormente uma análise de segurança e privacidade.

2.1 Conceitos

A seguir, serão apresentados alguns conceitos que são a base teórica para o funcionamento da *blockchain*, tais como funções *hash* criptográficas, árvores de Merkle, assinaturas digitais, servidores *timestamp* e *proof of work*.

2.1.1 Funções hash criptográficas

Uma função *hash* (ou função de dispersão) pode ser definida como uma função que recebe como entrada uma cadeia de tamanho arbitrário e dá como saída uma cadeia de tamanho fixo denominada valor de *hash* [10]. O cálculo da saída da função deve ser computável eficientemente, em tempo linear sobre o tamanho da cadeia de entrada [11].

Uma função *hash* criptográfica é uma função *hash* que é resistente a colisão, resistente a pré-imagem e resistente a segunda pré-imagem [10]. Essas três propriedades estão relacionadas e seguem suas definições:

1. Resistência a colisão: Uma função *hash* H é considerada resistente a colisão se é impraticável encontrar duas cadeias, x e y, tal que $x \neq y$, e $H(x) = H(y)$ [11]. Intuitivamente, ela será resistente a colisão se é difícil encontrar duas mensagens com o mesmo valor de *hash* [10].
2. Resistência a pré-imagem (ou propriedade de mão-única): Uma função de *hash* H é considerada resistente a pré-imagem se, dado um valor de *hash* y, é impraticável computar qualquer cadeia de entrada x tal que $H(x) = y$ [10].
3. Resistência a segunda pré-imagem: Uma função *hash* H é considerada resistente a segunda pré-imagem se, dado uma cadeia x, é impraticável encontrar uma cadeia y, tal que $x \neq y$, e $H(x) = H(y)$ [10].

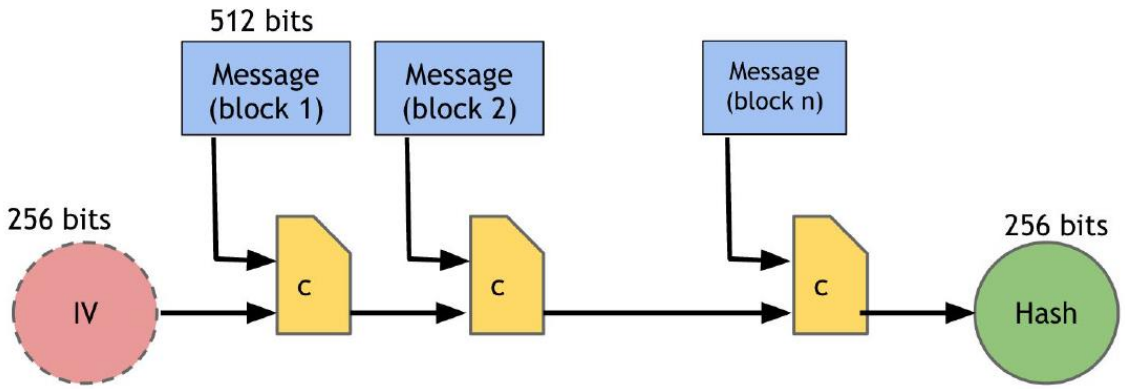
Em relação às definições acima, o termo "impraticável" significa que a computação de um cálculo pode demandar tantos passos que é considerado impossível realizá-lo num tempo aceitável. Dada uma função *hash* cujas saídas tem comprimento n, é esperado encontrar uma colisão em $O(2^{\frac{n}{2}})$ iterações, por conta do paradoxo do aniversário. A quantidade de iterações necessárias para encontrar uma pré-imagem ou uma segunda pré-imagem é $O(2^n)$, através de ataque de força bruta. Computações que requerem tempo exponencial são consideradas impraticáveis, mas é claro que isto depende diretamente de n. Para resistir a colisão, n deve ser no mínimo 128 bits, porém um valor de n maior ou igual a 160 bits é comumente preferido [10].

Função hash SHA-256

SHA-256 é a função *hash* criptográfica primariamente utilizada na *blockchain* de Bitcoin [11]. SHA é sigla para *Secure Hash Algorithm*, que em português significa Algoritmo de Dispersão Seguro. A numeração 256 no nome da função referencia o tamanho da saída, que é de 256 bits. Para lidar com entradas de tamanho arbitrário, a SHA-256 utiliza um mecanismo chamado transformação de Merkle-Damgard, que transforma uma função hash de entrada de tamanho fixo em uma função hash de entrada de tamanho arbitrário. Isso é feito através da divisão da entrada em blocos de tamanho fixo. Cada bloco é aplicado à função de compressão (entrada de tamanho fixo) junto ao resultado do bloco anterior, e o resultado do último bloco é o resultado da função.

Um esquema da transformação de Merkle-Damgard da SHA-256 pode ser observado na Figura 2.1, e mais detalhes sobre o funcionamento interno da função estão disponíveis na padronização oficial [12].

Figura 2.1: A entrada da função SHA-256 é dividida em blocos de 512 bits (em azul). Para o primeiro bloco, é utilizado um vetor de inicialização (em vermelho). Cada bloco, junto ao resultado do bloco anterior, é entrada de uma função de compressão (em amarelo) cuja entrada tem 768 bits e saída tem 256 bits. A saída do último bloco é a saída da SHA-256 (em verde).



Fonte: [11].

2.1.2 Árvores de Merkle

Árvore de Merkle, também conhecida como árvore *hash* [13]**Error! Reference source not found.**, é uma estrutura proposta por Ralph Merkle em 1979 [14], com intuito de aprimorar esquemas de assinatura digital *one-time* [13]**Error! Reference source not found.**[14].

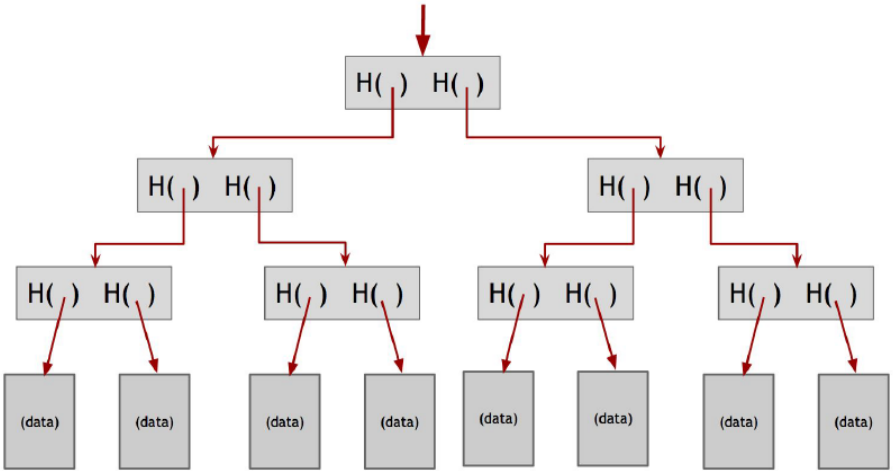
Uma árvore de Merkle é uma árvore binária com ponteiros *hash*. Um ponteiro *hash*, por sua vez, é a união de um ponteiro comum, que aponta para onde uma certa informação é

armazenada, e um *hash* criptográfico de tal informação em algum ponto fixo no tempo [11]. Com este valor *hash*, é possível saber se a informação guardada pelo ponteiro foi modificada.

A construção da árvore:

1) Os dados são divididos em blocos;
2) Cada folha é um bloco de dados [11] (ou o valor *hash* de um bloco de dados [3]);
3) Cada nó não-folha tem ponteiros *hash* para dois outros nós (filhos).

Figura 2.2: Esquema de uma árvore de Merkle com oito folhas. Autoria: [11].



Fonte: [11].

Apesar da estrutura da árvore de Merkle ter sido concebida junto ao Esquema de Assinatura de Merkle [13][14], a utilização dela na *blockchain* não está relacionada a assinaturas digitais. A raiz da árvore *hash* (também denominada raiz de Merkle) tem a propriedade de ser resistente a adulteração [15], e para isso é necessário apenas armazenar o seu ponteiro *hash* [11]. Se um bloco da árvore for alterado, o *hash* do seu novo conteúdo não terá o mesmo valor que o *hash* anteriormente armazenado no seu nó pai[1], desde que a função de *hash* utilizada seja criptograficamente segura.

Um esquema de árvore de Merkle pode ser observado na Figura 2.2.

2.1.3 Assinatura digital

Assinaturas escritas à mão possuem algumas características como a dificuldade de falsificação, a facilidade de conferência e o fato de estar atrelada a um documento e não poder

[1] O nó pai de um nó x é o nó que guarda um ponteiro *hash* para x.

ser transferida [11]. De maneira análoga, é desejável que assinaturas digitais mantenham essas características, para dessa forma manterem o nível de segurança das assinaturas escritas à mão.

Um esquema de assinatura digital é composto dos seguintes atributos [16]:

- Um espaço de mensagens purotexto M;
- Um espaço de assinaturas S;
- Um espaço de chaves de assinatura $\mathcal{K}$;
- Um espaço de chaves de verificação $\mathcal{K}'$;
- Um algoritmo de geração de chaves eficiente $Gen: \mathbb{N} \times \mathcal{K} \times \mathcal{K}'$;
- Um algoritmo de assinatura eficiente $Sign: M \times \mathcal{K} \longmapsto S$;
- Um algoritmo de verificação eficiente $Verify: M \times S \times \mathcal{K}' \longmapsto \{true, false\}$.

Observe que o espaço $\mathcal{K}$ representa as possíveis chaves privadas, utilizadas na produção de assinaturas e o espaço $\mathcal{K}'$ representa as possíveis chaves públicas, utilizadas na verificação de assinaturas. Antes de assinar qualquer documento, é preciso obter um par de chaves $(\mathcal{K}, \mathcal{K}')$ através do algoritmo de geração de chaves, que gera como saída um par de chaves do tamanho especificado pela entrada [16].

Algoritmo de Assinatura Digital de Curva Elíptica

Para assinar as transações de Bitcoin é utilizado o Algoritmo de Assinatura Digital de Curva Elíptica, em inglês *Elliptic Curve Digital Signature Algorithm* (ECDSA). Cada usuário possui uma chave privada e uma chave pública. A chave pública é derivada a partir da chave privada através de multiplicação escalar, mas é computacionalmente impraticável descobrir, através de uma chave pública, sua respectiva chave privada. Para transferir bitcoins, o usuário precisa assinar digitalmente a transação e para isso ele utiliza sua chave privada. Uma terceira parte que recebe estes bitcoins pode verificar quem os enviou através da chave pública. Mais detalhes da matemática por trás dessas operações podem ser encontrados nesse artigo [17].

2.1.4 Servidor timestamp

Uma *timestamp*, ou marca temporal, é o registro de um determinado momento atrelado a uma informação. Documentos digitais eletrônicos são relativamente fáceis de adulterar [18], e para impedir que isso seja feito sem deixar evidências, é preciso que uma *timestamp* de confiança esteja atrelada à informação. Ao adicionar uma *timestamp* de confiança a um código ou a uma assinatura digital obtém-se um selo digital de integridade dos dados e uma data e

horário de confiança de quando a transação foi realizada [19]. Para que uma *timestamp* seja considerada de confiança, é necessário que:

1) Esta dependa apenas dos dados a serem marcados e não dependa de características do meio onde os dados estão inseridos [18];

2) Seja impossível marcar um documento com um horário e data diferentes dos verdadeiros [18].

O serviço de *timestamping* comumente requer confiança em terceiro, como uma autoridade de *timestamping*. De maneira simplificada, o cliente envia o valor *hash* de seus dados para a autoridade [18], e esta associa uma *timestamp* a este valor *hash* e os devolve para o cliente [19]. Marcar o valor *hash* dos dados com uma *timestamp* equivale a marcar os próprios dados, e o valor *hash* tem tamanho fixo, logo a utilização da função *hash* resolve problemas de armazenamento, largura de banda e, por conta das características das funções *hash* criptográficas, também privacidade [18]. Uma opção para aumentar a confiança no servidor seria a publicação dos *hashs* e suas respectivas *timestamps*, de modo que o usuário pode checar a validade destes [20].

Para obter descentralização no servidor de *timestamp*, a ideia de Nakamoto [1] foi incluir em cada bloco uma timestamp e o *hash* do bloco anterior, de modo a estabelecer uma ordem e cada *timestamp* reforçar as anteriores.

2.1.5 Proof of work

A ideia do que hoje é conhecido como *proof of work* (prova de trabalho, traduzido do inglês) surgiu desatrelada do nome em um artigo de Dwork e Naor em 1993 [21], com o intuito de combater o envio de spam por email e controlar o acesso a recursos compartilhados. Como o nome descreve, deve haver uma prova de trabalho, que se dá pela resolução de um problema complexo. No contexto do envio de emails, o usuário prova que o email que está querendo enviar é relevante ao gastar algum tempo de processamento na resolução de tal problema. É importante ressaltar que o problema não deve ser resolvido pelo usuário, e sim pela máquina que enviará o email.

Posteriormente, o sistema baseado em *proof of work* que se tornou mais conhecido foi o Hashcash [22], e Nakamoto utilizou um sistema similar para o Bitcoin [1]. Nestes, o problema que se quer resolver é encontrar um valor *nonce* que quando aplicado à função *hash* utilizada pelo sistema junto a outros parâmetros de entrada resulte em uma saída que tenha uma certa quantidade de zeros à esquerda. Para resolver esse problema é preciso incrementar o *nonce* e testar se ele satisfaz essa condição, até que se encontre um valor que satisfaça. Com isso, é

realizado trabalho para encontrar a solução, mas esta pode ser facilmente verificada, bastando apenas aplicar o valor encontrado à função hash e reconhecer que este satisfaz a condição desejada.

2.2 Funcionamento

A mecânica aqui explicada refere-se à *blockchain* do Bitcoin, proposta por Nakamoto [1], no qual as transações financeiras ocorrem em tempo real, mas só são confirmadas (ou seja, inseridas na *blockchain*) a cada 10 minutos, quando um bloco é montado.

Uma transação é um conjunto de dados assinado digitalmente que se for válido fará parte de um bloco da *blockchain*. O propósito de uma transação é transferir propriedade de uma certa quantia de bitcoins para um endereço Bitcoin. Sumariamente, cada transação [23] contém:

- *ID da transação*
- *Descrição e metadados*

  Inclui número de entradas, número de saídas, tamanho da transação em bytes, entre outros.
- *Lista de entradas*

  Cada entrada faz referência a uma saída de uma transação realizada anteriormente, composta do *hash* de tal transação e o índice de uma saída específica. Na entrada também é incluído um *script* denominado *signature script*. Este *script* contém a chave pública completa do usuário que está fazendo a transação atual e também sua assinatura (vide seção 2.1.3) para provar que ele é o dono de tal chave pública. Uma transação pode precisar de múltiplas entradas para totalizar o valor desejado.
- *Lista de saídas*

  Cada saída é composta por um valor de bitcoins e um *script* denominado *pubKey script* para destravar esses bitcoins. Este script inclui o *hash* da chave pública do usuário recebedor (que está incluído em seu endereço), e permite que o usuário que possui a chave pública correspondente se aproprie dos bitcoins.

A soma dos valores das entradas deve ser maior ou igual à soma dos valores de saída. O usuário pode acrescentar à transação uma saída a mais direcionada para si mesmo, que será o troco do usuário. Normalmente será cobrada uma taxa por cada transação, de modo a incetivar a rede.

Felizmente, o usuário não precisa realizar manualmente essas manipulações de chaves. Ele pode utilizar um software *wallet* que propiciará uma interface amigável. Um software *wallet* [24] (do inglês, carteira) de bitcoins é um software que funciona como uma carteira virtual. Para isso, armazena as informações do usuário necessárias para a realização e recebimento de transações, como:

- Pares de chaves do usuário, onde cada par é composto por uma chave privada e uma chave pública;
- Transações feitas a partir/para os endereços do usuário;
- Preferências do usuário;
- Chave padrão;
- Contas.

Outros dados podem também ser armazenados, dependendo da carteira virtual escolhida. Existem quatro tipos de carteiras de Bitcoin: *web wallets*, *desktop wallets*, *mobile wallets* e *hardware wallets* [25].

Figura 2.3: Lista de softwares wallet disponíveis para Bitcoin.

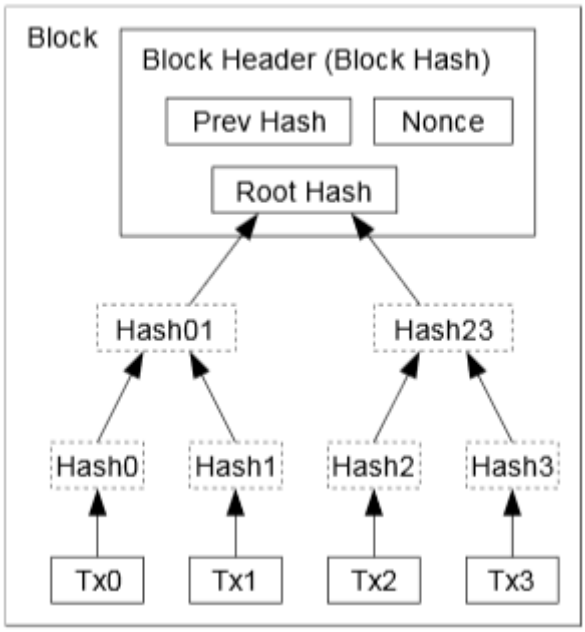
Fonte: [26].

Cada transação feita só está confirmada quando se encontra armazenada em um bloco da *blockchain*. Cada bloco da *blockchain* contém:

- *Hash* do cabeçalho do bloco anterior;
- *Timestamp* que representa o momento aproximado de criação do bloco;
- *Proof of work* (*nonce*);
- Árvore de Merkle (registro das transações).

Na montagem do bloco, as transações são armazenadas através de uma árvore de Merkle, como ilustrado na Figura 2.4, porém apenas a raiz da árvore é incluída no cabeçalho do bloco.

Figura 2.4: Componentes de um bloco da blockchain, omitindo a timestamp para simplificação.



Fonte: [1].

Os mineradores são os usuários que cedem o processamento e memória necessários para o funcionamento da *blockchain*, e em troca, têm a chance de coletar bitcoins. Estes se conectam numa rede *peer-to-peer*, onde cada nó da rede é um computador minerador. Para um usuário *payer*[2] realizar uma transação ele precisa apenas ter alguma quantia de bitcoin, e não precisa necessariamente ser um minerador da rede. Segue o funcionamento da rede:

1) Usuários *payers* fazem transações utilizando algum software *wallet* (vide exemplos na Figura 2.3).
2) Toda transação feita deve ser transmitida para todos os nós da rede.
3) A cada 10 minutos, cada nó vai utilizar as novas transações recebidas através da rede para formar um novo bloco de transações.
4) Cada nó, de posse do bloco por ele montado, vai tentar obter um valor *nonce* que satisfaça o problema mencionado na seção 2.1.5. Este valor será sua prova de trabalho;

[2] *Payer* se refere aos usuários que realizam transações, com intuito de distinguí-los dos mineradores da rede.

5) O nó que conseguir encontrar a prova de trabalho a insere no bloco e o transmite para os outros nós, que irão verificá-lo;
6) Cada nó verifica se as transações dentro do bloco são válidas e se a prova de trabalho é aceita, o bloco é adicionado à *blockchain* do nó e o minerador que gerou a prova de trabalho é recompensado com bitcoins;

Para receber a recompensa, é preciso que cada nó, durante a construção do bloco (etapa 3), inclua nele uma transação especial de criação de moeda. Esta transação distingue-se de todas as outras por possuir como único parâmetro de entrada um *coinbase*, que permitirá a reivindicação da recompensa. A saída da transação deve ser um endereço à escolha do minerador, comumente um endereço para sua própria carteira. Desse modo, o nó que conseguir adicionar o bloco que construiu à *blockchain* será o nó recompensado. A criação de moedas tende a diminuir com o tempo, pois a cada quatro anos é reduzida a quantidade de bitcoin que se ganha por cada bloco cai pela metade.

Após a descrição dos componentes de uma transação, no início dessa seção, foi mencionada uma outra forma de incentivo: uma taxa que poderia ser cobrada por cada transação. Inicialmente, essa taxa não é obrigatória, mas cada vez mais se tornará um importante incentivo para a rede. Para incluir uma taxa em uma transação, basta que a soma dos valores de entrada seja maior que a soma do valores de saída. A diferença será o valor da taxa que será recebida pelo minerador que conseguir colocar a transação na *blockchain*, seguindo o mecanismo já descrito.

Eventualmente, pode acontecer de mais de um minerador conseguir encontrar uma prova de trabalho para seu bloco e com isso mais de um bloco é transmitido pela rede. Os nós da rede podem receber portanto versões de blocos diferentes, ou mesmo mais de uma versão. O protocolo diz que a maior *blockchain* é a verdadeira, e portanto cada nó vai montar blocos sobre a versão que estiver maior. Mas os nós guardam outras versões, para o caso de uma delas posteriormente se tornar a maior.

3 SEGURANÇA E PRIVACIDADE

Vimos no capítulo anterior que o protocolo da *blockchain* não depende de nenhuma autoridade central, pelo contrário, cada nó da rede realiza exatamente o mesmo algoritmo. Mas o que previne um usuário malicioso de roubar bitcoins? Os aspectos de segurança são abordados na seção 3.1, e em seguida há uma análise sobre a privacidade do usuário.

3.1 Segurança

Dado que as contas de Bitcoin não estão associadas a identidades do mundo real, e sim a pseudônimos, não é possível punir usuários que têm comportamento malicioso. A solução para tornar a rede mais segura então é tornar muito difícil que esse tipo de comportamento seja bem sucedido.

Nessa seção não trataremos de ataques de roubo de chaves, que acontecem quando uma parte maliciosa consegue acesso a informações que estão guardadas na carteira de um usuário. Embora sejam possíveis, estes ataques não dizem respeito à segurança da *blockchain*, e sim a vulnerabilidades de sistemas operacionais e *wallets*.

Serão tratados os ataques conhecidos por *double-spending*, 51% e supressão de transações, por conta da relevância que ganham quando movidos para o contexto de votações. Num sistema de votação, vulnerabilidade a um ataque *double-spending* pode significar o cancelamento de votos, enquanto a supressão de transações pode impedir a confirmação deles.

Depois

3.1.1 Ataque double-spending

Um usuário Alice não pode simplesmente “pegar” os bitcoins de um outro usuário Bob (ou seja, transferir os bitcoins endereçados a Bob para um outro endereço), afinal, todo bitcoin que pertence a Bob só pode ser destravado através da chave pública e assinatura dele. Como o esquema de assinaturas que Bitcoin utiliza é seguro, Alice não é capaz de forjar a assinatura de Bob.

Mas Alice pode utilizar uma estratégia um pouco indireta: ela pode comprar algo a Bob e pagar através de uma transação de bitcoins. Após Bob entregar a Alice a mercadoria, ela pode apagar a transação na qual paga Bob da *blockchain*, permitindo-a gastar aqueles bitcoins novamente. Este ataque que Alice emprega se chama *double-spending* [11][27], e parece contradizer as afirmações de que a *blockchain* de Bitcoin é imutável, e que uma transação está

confirmada assim que entra na *blockchain*. Mas podemos considerar tais afirmações corretas, porque a chance de Alice conseguir apagar a transação é muito pequena.

Para entender quais são as chances de Alice de ser bem sucedida, é preciso entender o que significa apagar um bloco da *blockchain*. Um bloco eliminado é denominado bloco órfão. Como cada bloco contém o *hash* do bloco anterior, retirar um determinado bloco significa eliminar não só aquele bloco, mas todos os blocos que foram acrescentados após ele. Para isso, Alice usa as transações que recebeu através da rede e monta seu bloco, excluíndo a transação que tinha endereçado a Bob. Em lugar desta, Alice coloca uma transação diferente onde gasta os mesmos bitcoins que antes estavam endereçados a Bob. Para a montagem desse bloco, ao invés de colocar o *hash* do bloco mais recente, Alice utiliza o bloco anterior ao que guardava a transação que havia feito.

Para concluir a montagem do bloco, Alice precisa encontrar uma prova de trabalho, e se for bem sucedida nisso, transmitirá seu bloco para os nós da rede. Estes devem verificar o bloco criado por Alice, e com isso notam que esta versão de *blockchain* é diferente da que estavam trabalhando. Dependendo que quantos blocos Alice apagou, a sua versão adulterada pode ser menor que a versão que outros mineradores estavam trabalhando, e portanto não será aceita. Como mencionado no capítulo anterior, a *blockchain* mais comprida é aceita pelos nós que seguem o protocolo da rede como a verdadeira.

Mas ainda existem chances de Alice ser bem sucedida em seu ataque:

1) Se ela age rapidamente, sua transação pode estar armazenada no último bloco acrescentado à *blockchain*. Desse modo, Alice só precisa apagar este bloco, de modo que ao acrescentar o novo bloco criado por ela, obterá uma versão adulterada da *blockchain* onde todas as transações são válidas e de tamanho igual à versão original. Como entre as duas não existe uma maior, qualquer uma das duas pode ser continuada.[3]
2) Se Alice precisa eliminar vários blocos e por isso constrói uma versão adulterada da *blockchain* mais curta que a original, ela deve aumentá-la, até que fique maior que a legítima.

Para impedir que o cenário 1 ocorra, Bob deve esperar um pouco antes de entregar a mercadoria para Alice. Quantos mais blocos forem adicionados à *blockchain* após o bloco que

[3] Em realidade, os nós que seguem o protocolo da rede consideram a primeira transação que receberam a correta, e portanto devem continuar trabalhando na ramificação da *blockchain* primitiva. No entanto, dada a extensão da rede, é possível que parte dos nós ainda não tenha recebido o bloco com a transação de Alice para Bob, e receba primeiro o bloco projetado por Alice, onde há uma transação diferente que gasta os mesmos bitcoins. Essa parte dos nós, ao seguir o protocolo, escolheria estender a ramificação que contém o *double-spend*.

registra a transação de Alice para Bob, menor a chance de um ataque de *double-spending* ser bem sucedido. Há um consenso de que após seis blocos de confirmação, a chance já é bem pequena.

Caso Bob de fato espere, pode acabar no cenário 2, onde Alice apaga vários blocos e tenta fazer com que o comprimento da *blockchain* adulterada supere a legítima. Para isso, todo bloco que Alice monta é adicionado à ramificação com *double-spend*. Em realidade, Alice começa a trabalhar secretamente nese ramo da *blockchain* enquanto Bob espera por blocos de confirmação, e espera que em algum momento, após Bob entregar a mercadoria, o ramo no qual está trabalhando consiga superar o legítima em tamanho. No entanto, todos os nós honestos da rede ainda estão trabalhando sobre a maior ramificação da *blockchain*, que por hora é a primitiva. Se a maioria da rede tem um comportamento honesto, a chance de Alice atingir seu objetivo é muito pequena.

3.1.2 Ataque 51%

Na seção anterior foi dito que a chance de Alice ter seu ataque *double-spending* bem sucedido é muito pequena numa rede onde a maior parte dos nós age honestamente. Na seção atual analisaremos o que Alice pode fazer se a maior parte da rede tiver um comportamento malicioso e se aliar a ela.

Quando mais da metade da rede se alia a Alice, ela se torna capaz de subverter o consenso da rede. Esta parte da rede é capaz de gerar blocos mais rápido que o restante, de modo que em algum momento a ramificação da *blockchain* com o *double-spend* se tornará a maior.

Esse ataque recebe o nome de 51% ou maioria porque só funciona com 100% de probabilidade se Alice detiver mais da metade do *hashrate* da rede [27]. Porém com apenas 40% da rede, Alice tem uma chance de 50% de ser bem sucedida num ataque de *double-spending* a uma *blockchain* com 6 blocos de confirmação [28].

Não é possível prevenir esse ataque, mas quanto mais blocos de confirmações Bob esperar, menor a chance de Alice atacá-lo, pois ela precisaria empregar mais poder computacional no ataque. Mas com tanto poder computacional, Alice provavelmente lucraria mais bitcoins ao minerar a rede honestamente do que ao aplicar tal ataque. Isso depende não só do grau de dificuldade do ataque (quantidade de blocos de confirmação) como também de quanto ela lucraria com seu êxito. Com isso concluímos que a segurança do Bitcoin depende também dos incentivos à mineração.

3.1.3 Ataque de supressão de transações

Na seção anterior vimos que com mais da metade da rede sob seu domínio, Alice é capaz de realizar ataques *double-spending*. Nesta seção é abordado um outro ataque suscetível ao ataque 51%.

No ataque de supressão de transações [28], o grupo atacante decide ignorar transações criadas por/endereçadas a um certo usuário. Se o grupo é majoritário na rede, o ataque é bem sucedido, e as transações ignoradas não são confirmadas porque não fazem parte da *blockchain*.

No entanto, os atacantes não conseguem impedir que as transações sejam transmitidas para a maioria dos nós da rede, de modo o ataque pode se tornar aparente [11]. Para diminuir a chance de sofrer tal ataque, o usuário pode incluir uma taxa em sua transação, de modo que se o atacante não incluí-la em seu bloco, não receberá o pagamento da taxa.

3.2 Privacidade

Todas as transações de Bitcoin são anunciadas publicamente, pois essa *blockchain* é pública. Como vimos no capítulo anterior, as transações não estão associadas a identidades do mundo real, e sim a endereços. O endereço de uma conta é o valor *hash* de sua chave pública. Para realizar um pagamento a tal conta, é preciso ter conhecimento de seu endereço. Isso significa que a conta não é anônima, e sim pseudônima: o usuário é conhecido por um "apelido", que é o endereço da sua conta.

No contexto de troca financeira, é possível ligar informações do usuário ao seu pseudônimo, pois ao adquirir um bem ou serviço, o usuário pode precisar revelá-las. Por exemplo, ao pagar uma compra no supermercado utilizando bitcoins, o usuário está revelando sua imagem. Já uma compra online revelaria um endereço de entrega. Essas informações podem facilmente mapear o pseudônimo a uma identidade real.

Se no cenário acima a identidade revelada estivesse relacionada a apenas uma transação (como a compra no supermercado ou a compra online), a quebra da anonimidade não seria um grande problema, afinal, já é um costume identificar-se nesse tipo de situação. Porém ao tomar conhecimento de um determinado endereço, um atacante pode buscar na *blockchain* todas as transações que o envolvem. Essa situação é semelhante a ter um extrato bancário público, o que é indesejável.

É importante ressaltar que uma transação direta não é a única forma de identificar o usuário por trás de um endereço [11]. Quando um usuário interage com um provedor de serviços, é comum que este peça informações de identificação do usuário e as guarde. Usuários também podem publicar seus endereços ao pedir doações. E ainda que nenhum dos cenários acima seja

verdadeiro, existem ataques de desanonimização que utilizam pistas sutis como por exemplo o horário em que a transação ocorreu, a ligação que pode existir entre múltiplas transações.

Para dificultar a desanonimização, é comum que um usuário tenha vários endereços. A criação e administração deles é feita pela sua carteira de bitcoins. Mas mesmo com bitcoins distribuídos em diferentes endereços, uma transação pode revelar que eles pertencem ao mesmo usuário, bastando observar a lista de entradas da transação. Se essas entradas são bitcoins endereçados a diferentes contas e o usuário é capaz de validar a transação, significa que todas as contas o pertencem. Desse modo, múltiplos endereços dificultam o ataque, mas não o impossibilitam. Claro que tudo isso depende também da maneira que o software *wallet* administra essas múltiplas contas.

3.3 Considerações

Vimos na seção 3.1 que um ataque 51% na rede de Bitcoin tornaria possível *double-spending* e supressão de transações. Ainda assim, nenhum grupo de mineradores domina mais que 50% do *hashrate* da rede, e mesmo que algum o fizesse, com tamanho poder computacional, é normalmente mais vantajoso minerar a rede e receber as recompensas do que atacar outros usuários. Uma onda de ataques poderia causar a perda de confiança e a evasão dos usuários. Ter menos usuários realizando pagamentos com bitcoins implica em obter menos recompensas através da mineração, e desse modo a rede poderia colapsar.

É importante ressaltar que toda essa análise se aplica à *blockchain* do Bitcoin, cujo propósito é registrar transações financeiras, e qualquer variação na tecnologia ou aplicação pode acarretar em diferenças na avaliação de segurança da *blockchain*.

4 SISTEMAS DE VOTAÇÃO

Existem diferentes ideias de como utilizar o potencial da *blockchain* em uma votação. Uma das abordagens [29], dirigida inicialmente a eleições governamentais, consiste em manter o meio de recolhimento de votos tradicional (por exemplo, cédulas de votação) e utilizar a *blockchain* para verificar e guardar os votos. Outra abordagem [30][31] consiste em recolher os votos por meio digital, através da internet e aplicativos instalados em dispositivos. Existe também uma abordagem ainda mais revolucionária [32], que propõe trazer não só as votações para o meio online, mas toda a legislação.

Independentemente da abordagem escolhida, a principal vantagem da votação com tecnologia *blockchain* é trazer segurança contra fraudes, transparência de votos, auditabilidade, sem sacrificar a privacidade do votante. Em abril de 2014, foi divulgada a notícia de que a primeira votação *blockchain* foi experimentada pela Aliança Liberal da Dinamarca, em uma eleição interna [33].

A seguir é feita uma breve descrição da proposta dos sistemas de votação Blockchain Apparatus[4], FollowMyVote[5], V iniciative[6] e BitCongress[7]. É importante ressaltar que, até a data de publicação desse documento, esses sistemas ainda não são utilizados na prática.

4.1 Blockchain Apparatus

Blockchain Apparatus é uma plataforma da Blockchain Technologies Corp[8] cujo propósito é a criação de aplicações que utilizam o poder da *blockchain*. O sistema de votação assegurado pela *blockchain* da Blockchain Apparatus é de código aberto, e portanto qualquer pessoa pode conferir seu funcionamento [29].

Ele diferencia-se dos outros sistemas revisados nesse trabalho por propor que seja mantida a interface familiar ao usuário, ou seja, o recolhimento dos votos aconteceria de maneira usual, mas para gravar cada voto seria utilizada uma *blockchain*. Portanto, o votante deve se registrar normalmente, e no dia da eleição, ele preencherá uma cédula de votação muito parecida com a atualmente utilizada (nos Estados Unidos), a diferença é que nela estão impressos três pequenos códigos QR: o primeiro representa o endereço de uma *blockchain*, o segundo representa o identificador único da cédula e o terceiro representa o identificador da eleição [34].

[4] http://blockchainapparatus.com/
[5] https://followmyvote.com/
[6] http://www.v-initiative.org/
[7] http://www.bitcongress.org/
[8] http://blockchaintechcorp.com/

Após preenchida, a cédula será escaneada por uma máquina, que utilizará os dados contidos para transferir cada unidade de voto ao candidato apropriado. Para isso, cada candidato deve ter um endereço único para onde os votos podem ser transferidos. É possível acompanhar a votação utilizando uma ferramenta chamada "explorador de *blockchain*", que pode mostrar quantos votos cada candidato recebeu utilizando seu endereço único [34].

Para reforçar a segurança, a máquina fica desconectada da internet enquanto as cédulas são depositadas e grava as cédulas em um DVD antes de ser conectada. As informações também estarão registradas nas cédulas, de modo que se as contagens da *blockchain* e do DVD não coincidirem, é possível recontar os votos [34].

A organização da votação fica encarregada de decidir qual *blockchain* será utilizada. Os desenvolvedores alegam que é possível utilizar a *blockchain* de Bitcoin, assim como também é possível utilizar a *blockchain* desenvolvida pela Blockchain Technologies Corp, chamada VoteUnit. Esta última funciona de maneira semelhante à primeira, porém não requer taxas de transação [34].

4.2 FollowMyVote

A proposta da plataforma FollowMyVote [31] é tornar online todo o processo de votação que acontece atualmente num dia de eleição, com o intuito de adicionar facilidade e segurança e diminuir custos. Ela é desenvolvida sobre a *blockchain* de BitShares[9], uma plataforma de *smart contracts* financeiros de nível industrial que promete em seu lançamento ser 100 vezes mais rápida que Bitcoin [35]. O sistema FollowMyVote tem código aberto, que modo que qualquer pessoa pode auditá-lo [31].

Para utilizar esse sistema de votação, o votante deve instalar um *software* (em um computador pessoal, *tablet* ou *smartphone*), que atuará como cabine de votação. Lá, o votante deve prover informações de identificação, que deverão ser aprovadas pela organização a frente da eleição. Uma vez verificada sua identidade, o usuário ganha direito ao voto através de uma cédula de votação, a qual ele preenche e submete para a urna baseada em *blockchain*. Se permitido pela entidade que organiza a eleição, o votante poderá alterar seu voto a vontade, até o encerramento da votação. Uma vez que termine o prazo, o voto considerado de cada votante será o depositado mais recentemente. O votante pode utilizar sua conta para acessar a urna e verificar se seu voto foi depositado da maneira esperada. Ele pode ainda auditar cada voto

[9] https://bitshares.org/

presente na urna para saber se o resultado da contagem de votos está correto, porém ele não poderá ver a identidade de quem preencheu cada cédula, por questões de privacidade.

Para manter a privacidade do votante, o FollowMyVote utiliza criptografia de curva elíptica, como segue:

- Durante a criação do registro, o usuário cria dois pares de chaves de criptografia de curva elíptica;
- O votante envia sua identidade para o verificador de identidades, que certifica que o primeiro par de chaves, denominado *par de chaves de identidade*, pertence ao votante.
- O Verificador de Identidades revisa as informações pessoais do votante, certifica que sua identidade ainda não foi registrada no sistema, determina que tipo de cédula de votação ele deve receber, e assegura que ele só está autorizado a receber uma cédula por eleição;
- O votante registra o segundo par de chaves, denominado *par de chaves de votar*, como pertencendo a uma das chaves de identificação;
- O Arquivista (em inglês, *Registrar*) é responsável por preparar os tipos de cédula e certificar que cada votante receba o tipo de cédula correto, atrelado a sua chave de votar.
- Ao preencher e submeter sua cédula de votação, o votante estaria na verdade criando transações que com seus votos e assinando-as com sua chave secreta de votar.

4.3 V Initiative

V Initiative é uma iniciativa que visa encontrar uma solução para votação online que seja a prova de fraude e garanta o anonimato do votante [30]. A ideia inicial é utilizar a *blockchain* para descentralizar a segurança, assim como nos sistemas mencionados anteriormente. Mas para manter o anonimato do usuário, é proposta a utilização do Zerocash [8] que utiliza provas *zero knowledge*[10]. Também é proposto o uso de um software para mascarar endereços IP, de modo que nenhum votante seja identificado através do rastreamento do seu voto.

[10] Um protocolo *zero knowledge* consiste em provar conhecimento sobre um certo segredo sem revelá-lo [30].

Infelizmente, a V Initiative ainda não formalizou os detalhes do funcionamento do sistema de votação, de modo que parecem existir lacunas na proposta, por exemplo, como ocorre a adaptação das transações financeiras de Zerocash para uma transação de voto, ou mesmo a mecânica da votação online.

4.4 BitCongress

O BitCongress é uma plataforma de legislação e votação *blockchain* descentralizada [31], que combina a *blockchain* de Bitcoin, Counterparty[11], Smart Contracts e uma ferramenta chamada Axiomity para formar um Congresso completo e funcional na *blockchain*.

- Counterparty é uma uma plataforma cujo objetivo é estender a funcionalidade do Bitcoin, proporcionando operações financeiras mais robustas que o simples pagamento [36].

  Para isso ser possível, Counterparty trata de armazenar dados extras em outras transações Bitcoin, de modo que todas as transações Counterparty são transações Bitcoin, e por isso sua segurança recai sobre a segurança da rede Bitcoin, que é considerada a *blockchain* mais segura do mundo [36].
- Smart Contracts são contratos programáveis que neste caso são mantidos em uma nuvem *blockchain* descentralizada [31].
- Axiomity é uma aplicação descentralizada que funciona como *wallet* do BitCongress [31]. A comunicação das três redes (Bitcoin, Counterparty e Smart Contract) é realizada pela ferramenta Axiomity, que é o *front-end* do sistema de votação. Qualquer usuário pode criar uma nova votação com facilidade através dessa aplicação.

Cada eleição do BitCongress é guardada numa blockchain de *smart contract*, e possui um tempo de vida, um conjunto de regras, candidatos, legislação, orçamento e um endereço URL para que seja acessada pelo público. Para cada opção de voto será criado um endereço através da *blockchain* Counterparty.

Ao se registrar no BitCongress, cada usuário recebe um token de votação, denominado *vote*, criado através de Counterparty. Para participar de uma votação, o votante deve enviar o seu *vote* para o endereço da opção escolhida. Quando a eleição expira, os tokens *vote* são devolvidos aos votantes, a fim de serem utilizados em outras eleições.

[11] http://counterparty.io/

As identidades dos usuários são guardadas numa autoridade central e as chaves públicas são mantidas em segredo para manter o sigilo dos votos.

4.5 Análise

Nesta seção é feita uma análise dos sistemas acima descritos em relação aos seguintes aspectos:

- ***Blockchain***: Qual a blockchain utilizada pelo sistema?
- **Segurança**: O sistema pode ser considerado seguro?
- **Privacidade**: O usuário tem sua privacidade assegurada?
- **Custo**: Comparado ao custo financeiro de sistemas de votação atualmente utilizados, há economia?
- **Adequação do usuário**: O votante se adequaria facilmente ao sistema ou existe alguma barreira?
- **Adequação da *blockchain***: Como a *blockchain* utilizada se adequa ao sistema de votação?
- **Problemas**: Possíveis problemas conceituais.

Blockchain Apparatus

- ***Blockchain***: Escolhida pelo cliente.
- **Segurança**: Depende da *blockchain* escolhida. Se utilizar a *blockchain* de Bitcoin, o registro dos votos seria assegurado por conta da extensão da rede, mas seria preciso pagar taxas de transação. Ao utilizar uma *blockchain* própria, pode haver perda de segurança, dependendo de ser privada ou pública, da quantidade de nós da rede, do incentivo que eles recebem para minerar a rede, entre outros fatores. Mas não foram apresentados esses detalhes.
- **Privacidade**: Depende de como é atribuído o endereço ao votante. Não está especificado.
- **Custo**: Dado que utiliza urnas físicas, e precisa de pessoas organizando a eleição da mesma maneira que as eleições atualmente, o custo é semelhante ao atual. Pode ser maior dependendo da *blockchain* escolhida, caso se apliquem taxas de transação.
- **Adequação do usuário**: Fácil adaptação, pois quase não há mudança na maneira de interagir com o votante.

- **Adequação da *blockchain***: Não especificada.
- **Problemas**:
    - A contagem de votos da *blockchain* tem que bater com a dos DVDs, mas não especifica quem computa os votos a partir dos DVDs. Se for uma parte maliciosa, pode manipular o resultado para não com o resultado da *blockchain*, podendo injustamente arruinar a reputação da votação com *blockchain*.
    - É dito que a votação é offline, então infere-se que o usuário não pode acompanhar a votação em tempo real, apenas após o fim da votação e com o início da conexão com a internet seria possível para o usuário ver seu voto ser confirmado na *blockchain*.

FollowMyVote

- ***Blockchain***: BitShares.
- **Segurança**: Mais detalhes sobre a interação do sistema com BitShares e mais detalhes sobre a *blockchain* de BitShares seriam necessários para avaliar a segurança do sistema.
- **Privacidade**: O mecanismo pode manter o voto secreto, desde que o Verificador de Identidade e o Arquivista não tenham comunicação.
- **Custo**: Por utilizar apenas um dispositivo pessoal do usuário, diminui o custo em relação a votações presenciais.
- **Adequação do usuário**: Pode haver dificuldade de adaptação por parte dos votantes. Ou mesmo acontecer de nem todos terem acesso a um dispositivo.
- **Adequação da *blockchain***: Não especificada.
- **Problemas**: O voto pode ser alterado pelo votante até o fim da eleição, o que implica que ele só é registrado na blockchain após o fim da eleição, logo o usuário não poderia acompanhar a eleição em tempo real.

V Iniciative

- ***Blockchain***: Zerocash.
- **Segurança**: Dado que Zerocash estende o protocolo Bitcoin e adiciona mais segurança através de zero knowledge, pode-se considerar a proposta segura sob este ponto de vista.

- **Privacidade**: A proposta do uso de provas *zero knowledge* é consistente, então podemos dizer que sim, o usuário pode ter privacidade assegurada.
- **Custo**: Por ser online, diminui o custo em relação a votações presenciais. Zerocash segue o esquema de taxas de transação de Bitcoin, ou seja, é importante para o incentivo da rede. Logo, pode ser um custo adicional.
- **Adequação do usuário**: Pode haver dificuldade de adaptação por parte dos votantes. Ou mesmo acontecer de nem todos terem acesso a um dispositivo.
- **Adequação da *blockchain***: Não especificada.
- **Problemas**: Não há muito especificado sobre o processo de votação, então os problemas podem existir mas não estão aparentes.

BitCongress

- ***Blockchain***: Bitcoin, Counterparty e Ethereum[12].
- **Segurança**: Dada a confiança que existe nas *blockchains* utilizadas, pode ser considerado seguro.
- **Privacidade**: Como é utilizada uma autoridade central para guardar as identidades dos usuários, a privacidade torna-se dependente da segurança desses dados e da honestidade dessa autoridade.
- **Custo**: Por ser online, diminui o custo em relação a votações presenciais. No entanto, há o custo das taxas de transação, que é proporcional ao número de transações e consequentemente ao número de votantes.
- **Adequação do usuário**: Pode haver dificuldade de adaptação por parte dos votantes. Ou mesmo acontecer de nem todos terem acesso a um dispositivo.
- **Adequação da *blockchain***: A teoria está bem detalhada e tem muito potencial para funcionar na prática.
- **Problemas**: O BitCongress não é apenas um sistema de votação, é também uma plataforma de legislação descentralizada. Para que tome dimensões globais, serão necessárias muitas mudanças sociais e políticas.

[12] https://www.ethereum.org/

4.5.1 Considerações finais

Ao fim dessas análises, torna-se evidente que, apesar do poder de descentralização do registro dos votos trazido pelo uso de uma *blockchain*, não foi possível dispensar completamente uma autoridade central. No BitCongress e no FollowMyVote, o papel da autoridade está relacionado à privacidade do votante. No Blockchain Apparatus existe uma autoridade implícita, que é em parte responsável pela contagem de votos. A V Iniciative não explicita a necessidade de uma autoridade central, mas como não há muitos detalhes da proposta disponíveis, é possível que algumas lacunas do sistema sejam preenchidas com a presença de uma autoridade.

Ainda que a *blockchain* proporcione ao Bitcoin e a outras criptomoedas independência de qualquer autoridade financeira, existe uma limitação quando a tecnologia é trazida para o contexto de votações. A diferença é que, quando trata-se de criptomoeda, para um usuário realizar uma transação ele precisa possuir alguma quantia, caso contrário a transação não será aceita pelos mineradores da *blockchain*. Essa quantia da criptomoeda foi obtida através de transações prévias e/ou atráves da atividade de mineração.

Já tratando-se de um sistema de votação, teoricamente não deve ser permitido que um voto seja passado de um usuário a outro através de uma transação, como ocorre com a criptomoeda, pois só quem pode receber votos são endereços de candidatos; também não deve ser possível minerar votos; é preciso determinar quem pode votar, limitar quantos votos cada usuário tem direito, determinar quando a eleição ocorrerá, quem são os candidatos, e é preciso normalmente pagar os custos da votação. Isso tudo fica sob responsabilidade de uma autoridade de votação, que administra o processo. Autoridades já estão presentes em votações tradicionais. Um exemplo de autoridade de votação no Brasil é a Justiça Eleitoral[13], que organiza, fiscaliza e realiza as eleições brasileiras.

Uma outra ressalva é que, como visto no capítulo anterior, a segurança da blockchain de Bitcoin em parte depende da relação entre o lucro obtido através da mineração e do ganho através de ataques. Se um minerador sabe que ganhará mais bitcoins sendo honesto, ele não tem razões para realizar os ataques descritos. No entanto, quando as transações que circulam na rede são decisivas para o resultado de uma votação, o ganho do atacante vai além da criptomoeda, pois seus ataques podem controlar o resultado da eleição.

Este poder pode incentivar a entrada de grandes grupos de mineração na *blockchain* utilizada pelo sistema de votação, a fim de suprimir ou cancelar votos endereçados a um

[13] http://www.justicaeleitoral.jus.br/

determinado candidato. Uma possível solução seria ter um maior controle sobre quem pode minerar a rede, através de *blockchains* privadas, mas isso pode acabar centralizando mais ainda a votação.

5 CONCLUSÃO

Neste trabalho foram estudados o funcionamento, segurança e privacidade da tecnologia *blockchain* e os sistemas de votação que a utilizam. Esses sistemas ainda não são utilizados na prática, e por isso foram analisadas suas propostas a fim de evidenciar aspectos de segurança, privacidade, adaptabilidade, custo, entre outros.

A maior dificuldade encontrada durante esse trabalho diz respeito à limitação de informações sobre os sistemas de votação que foram analisados no capítulo 4. Mas isso é compreensível porque o tema é recente e os sistemas também.

Um possível trabalho futuro é estudar o potencial de variações da *blockchain* de Bitcoin (como Sidechain, Multichain, *blockchains* privadas) a fim de esboçar possíveis aplicações em sistemas de votação.

Outra possibilidade é utilizar todo o conhecimento adquirido neste trabalho para produzir um novo sistema de votação baseado em *blockchain*.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] POELSTRA, Andrew. A Treatise on Altcoins. 2014.

[3] KOSBA, Ahmed et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. **Cryptology ePrint Archive**, Report 2015/675, 2015. http://eprint. iacr. org, 2015.

[4] ARANHA, Diego F. et al. Vulnerabilidades no software da urna eletrônica brasileira. **Relatório Técnico**, v. 18, p. 19, 2012.

[5] FELDMAN, Ariel J.; HALDERMAN, J. Alex; FELTEN, Edward W. Security analysis of the Diebold AccuVote-TS voting machine. 2006.

[6] MIERS, Ian et al. Zerocoin: Anonymous distributed e-cash from bitcoin. In: **Security and Privacy (SP), 2013 IEEE Symposium on**. IEEE, 2013. p. 397-411.

[7] DUFFIELD, Evan; DIAZ, Daniel. Dash: A Privacy-Centric Crypto-Currency. 2015.

[8] BEN SASSON, Eli et al. Zerocash: Decentralized anonymous payments from Bitcoin. In: **Security and Privacy (SP), 2014 IEEE Symposium on**. IEEE, 2014. p. 459-474.

[9] Alternative chain. In: Bitcoin Wiki. Disponível em: <https://en.bitcoin.it/wiki/Alternative_chain>. Acesso em: Novembro de 2015.

[10] SMART, Nigel Paul. **Cryptography: an introduction**. 3. ed. New York: McGraw-Hill, 2003. p. 153-154.

[11] NARAYANAN, Arvind; BONNEAU, Joseph; FELTEN, Edward; MILLER, Andrew; GOLDFEDER, Steven. Bitcoin and Cryptocurrency Technologies. Draft, 2015.

[12] GALLAGHER, Patrick. Secure hash standard (shs). **FIPS PUB**, p. 180-4, 2012.

[13] BECKER, Georg. Merkle signature schemes, merkle trees and their cryptanalysis. 2008.

[14] MERKLE, Ralph Charles. Secrecy, authentication, and public key systems. 1979.

[15] ELBAZ, Reouven et al. Hardware mechanisms for memory authentication: A survey of existing techniques and engines. In: **Transactions on Computational Science IV**. Springer Berlin Heidelberg, 2009. p. 10.

[16] MAO, Wenbo. **Modern cryptography: theory and practice**. Prentice Hall Professional Technical Reference, 2003.

[17] RYKWALDER, Eric. The Math Behind Bitcoin. In: CoinDesk. Disponível em: <http://www.coindesk.com/math-behind-bitcoin/> Acesso em: Dezembro de 2015.

[18] HABER, Stuart; STORNETTA, W. Scott. **How to time-stamp a digital document**. Springer Berlin Heidelberg, 1991.

[19] GlobalSign, Trusted Timestamp Service. Disponível em: <https://www.globalsign.com/en/timestamp-service/> Acesso em: Novembro de 2015.

[20] Safe Creative, Timestamping Authority. Disponível em: <https://tsa.safecreative.org/> Acesso em: Novembro de 2015.

[21] DWORK, Cynthia; NAOR, Moni. Pricing via processing or combatting junk mail. In: **Advances in Cryptology—CRYPTO'92**. Springer Berlin Heidelberg, 1993. p. 139-147.

[22] Hashcash. Disponível em: <http://hashcash.org/> Acesso em: Dezembro de 2015.

[23] Bitcoin Developer Guide. Disponível em: <https://bitcoin.org/en/developer-guide> Acesso em: Dezembro de 2015.

[24] Wallet. In: Bitcoin Wiki. Disponível em: <https://en.bitcoin.it/wiki/Wallet> Acesso em: Dezembro de 2015.

[25] DESJARDINS, Jeff. How Secure are Bitcoins? In: Visual Capitalist. Disponível em: <http://www.visualcapitalist.com/secure-bitcoins/> Acesso em: Dezembro de 2015.

[26] Bitcoin, Choose your wallet. Disponível em: <https://bitcoin.org/en/choose-your-wallet> Acesso em: Dezembro de 2015.

[27] Double Spending. In: Bitcoin Wiki. Disponível em: <https://en.bitcoin.it/wiki/Double-spending> Acesso em: Dezembro de 2015.

[28] Weaknesses. In: Bitcoin Wiki. Disponível em: <https://en.bitcoin.it/wiki/Weaknesses> Acesso em: Dezembro de 2015.

[29] Blockchain Apparatus. Voting, smart contracts and more on the blockchain. Disponível em: <http://blockchainapparatus.com/> Acesso em: Dezembro de 2015.

[30] V-Iniciative. Disponível em: <http://www.v-initiative.org/> Acesso em: Dezembro de 2015.

[31] FollowMyVote, The Online Voting Platform of The Future. Disponível em: <https://followmyvote.com/> Acesso em: Dezembro de 2015.

[32] BitCongress, BitCongress Whitepaper. Disponível em: <http://www.bitcongress.org/BitCongressWhitepaper.pdf> Acesso em: Dezembro de 2015.

[33] DOTSON, Kyt. Bitcoin Weekly 2014 May 7: First political party to use blockchain for e-voting is from Denmark, The Rise of Digital Currency, Bitcoin's niche in developing economies. In: SiliconANGLE. 2014.

[34] HERTIG, Alyssa. The First Bitcoin Voting Machine Is On Its Way. In: Motherboard. Disponível em: <http://motherboard.vice.com/read/the-first-bitcoin-voting-machine-is-on-its-way> Acesso em: Dezembro de 2015.

[35] BitShares. Disponível em: <https://bitshares.org/> Acesso em: Dezembro de 2015.

[36] Counterparty, About Counterparty. Disponível em: <http://counterparty.io/platform/> Acesso em: Dezembro de 2015.