Universidade Católica de Petrópolis

Centro de Engenharia e Computação

Governança de TI e Segurança da Informação

Filipe Bergara Cesar

Sistemas eletrônicos de votação baseados na tecnologia blockchain

Trabalho de Conclusão de Curso de Pós-Graduação

Volume I

Petrópolis

2018

Filipe Bergara Cesar

Sistemas eletrônicos de votação baseados na tecnologia blockchain

Trabalho de Conclusão de Curso de Pós-Graduação apresentado ao Centro de Engenharia e Computação como parte dos requisitos necessários à obtenção do título de Pós-Graduado em Governança de TI e Segurança da Informação.

Orientador: Professor Mestre Fábio Licht
Coorientador: Professor Mestre Jorge Varella

Volume I

Petrópolis
2018

Filipe Bergara Cesar

Sistemas eletrônicos de votação baseados na tecnologia blockchain

Trabalho de Conclusão de Curso de Pós-Graduação apresentado ao Centro de Engenharia e Computação como parte dos requisitos necessários à obtenção do título de Pós-Graduado em Governança de TI e Segurança da Informação.

Trabalho aprovado. Petrópolis, ____/____/2018:

_____

Professor Mestre Fábio Licht
Orientador

_____

Professor Mestre Jorge Varella
Coorientador

Petrópolis

2018

Ao meu amado filho, Filipe Gabriel.

Agradecimentos

Agradeço a minha esposa Lucrécia, por todo amor e compreensão.

Agradeço aos meus pais, Alvimar e Regina, por tudo aquilo que fizeram para que hoje eu pudesse estar escrevendo este trabalho.

Agradeço aos meus professores por me conduzirem até aqui, especialmente aos professores Mestre Fábio Licht e Mestre Jorge Varella.

Agradeço aos meus amigos por todo apoio e incentivo.

Finalmente, agradeço a Deus pelo meu filho Filipe Gabriel, personificação do Seu amor.

“Se eu não fosse imperador, desejaria ser professor. Não conheço missão maior e mais nobre que a de dirigir as inteligências jovens e preparar os homens do futuro.”
Dom Pedro II

Resumo

Sistemas criptográficos de votação ou sistemas de votação sobre *blockchain* são aderentes aos conceitos de sistema E2E (*end-to-end verifiable*) e representam, possivelmente, uma mudança no paradigma de construção de sistemas de votação eletrônica. Habilitam a verificação extensiva das transações (votos) e conferem alto nível de transparência e de integridade às eleições, visto que *blockchains* são estruturas bastante resistentes a adulterações. Este trabalho analisa representantes desta nova classe de sistemas e verifica o nível de robustez e a adequação para o emprego em grandes casos de uso, considerando os principais aspectos de segurança da informação envolvidos.

Palavras-chave: Votação eletrônica. Blockchain. E2E.

Abstract

Encrypted voting systems or blockchain voting systems are adherent to the concepts of the E2E (verifiable end-to-end) and possibly represent a change in the building of electronic voting systems paradigm. They enable extensive verification of transactions (votes) and confer a high level of transparency and response to the elections, since blockchains are structures that are quite resistant to adulteration. This work is analyzes representatives of this new class of systems and verifies the level of robustness and adequacy for implementation in cases of use, considering the main aspects of information security that are involved.

Keywords: Electronic voting. Blockchain. E2E.

Lista de ilustrações

Figura 1 – Esquema simplificado de criptografia de chave pública e de chave privada 14
Figura 2 – Esquema de funcionamento do RSA 16
Figura 3 – Estrutura geral do código de hash seguro 19
Figura 4 – Esquema de enderaçamento do bitcoin 20
Figura 5 – Esquema de uma Árvore de Merkle 21
Figura 6 – Encadeamento dos blocos 23
Figura 7 – Processo de registro e habilitação do eleitor 35
Figura 8 – Funcionamento do *BitCongress* 38

Lista de tabelas

Tabela 1 – Equivalência entre algortimos e utilização segura 18
Tabela 2 – Exemplo de alvo da prova de trabalho do bitcoin 24

Lista de abreviaturas e siglas

ANSI	American National Standards Institute
ASIC	Application-Specific Integrated Circuit
B2B	Business-to-business
DAC	Decentralized Autonomous Companhies
DPOS	Delegated Proof Of Stake
E2E	End-to-end verifiable
ECDSA	Elliptic Curve Digital Signature Algorithm
IBM	International Business Machines
IDEA	International Institute For Democracy And Electoral Assistance
MIT	Massachusetts Institute of Technology
MPOG	Ministério do Planejamento, Orçamento e Gestão
NIST	National Institude of Standards and Technology
OSCE	Organization for Security and Cooperation in Europe
P2P	Peer-to-peer
RSA	Rivest, Shamir e Adleman
SECG	Standards For Efficient Cryptografy Group
SHA	Secure Hash Algorithm
SPV	Simplified Payment Verification
URL	Uniform Resource Locator

Sumário

1 Introdução 12
1.1 Motivação 12
1.2 Objetivos do trabalho 12
1.3 Estrutura 12

2 Revisão literária das tecnologias e conceitos envolvidos 13
2.1 Criptografia assimétrica 13
2.2 Assinatura cega 15
2.3 Algoritmo RSA 15
2.4 Algoritmo de assinatura digital de curva elíptica 17
2.5 Função *hash* SHA-256 18
2.6 Árvore de Merkle 20
2.7 Mecanismo de consenso 21
2.7.1 Prova de trabalho 23
2.8 Aspectos de segurança 25
2.8.1 Ataque contra o sigilo 25
2.8.2 Ataque de gasto duplo 26
2.8.3 Ataque de 51% 26

3 Blockchain 28
3.1 Histórico 28
3.2 Conceitos básicos 29

4 Sistemas de votação eletrônica sobre blockchain 31
4.1 Follow My Vote 32
4.2 BitCongress 37

5 Conclusão 40
5.1 Dificuldades encontradas 41
5.2 Trabalhos futuros 41

Referências 42

APÊNDICES 45

1 Introdução

1.1 Motivação

A tecnologia vem alterando significativamente a forma das pessoas se relacionarem, seja uma com as outras, seja com entidades públicas ou privadas. À medida que os níveis de interação e de acesso à informação aumentam, mais a sociedade questiona a confiabilidade dos sistemas eletrônicos de votação, sobretudo quanto à transparência e à integridade dos resultados.

Sistemas criptográficos de votação ou sistemas de votação sobre *blockchain* representam uma nova classe de sistemas de votação e possuem forte aderência aos conceitos de sistemas E2E *(end-to-end verifiable)*, permitindo a verificação extensiva das transações (votos).

1.2 Objetivos do trabalho

São objetivos deste trabalho:

1) realizar uma revisão teórica sobre as principais tecnologias e conceitos envolvidos na compreensão da *blockchain* do *bitcoin*;
2) introduzir o conceito de *blockchain* e algumas de suas principais aplicações;
3) estabelecer um panorama sobre a votação eletrônica;
4) analisar os sistemas *Follow My Vote* e *BitCongress*; e
5) concluir sobre o nível de robustez dos sistemas e a viabilidade de aplicação em grandes casos de uso.

1.3 Estrutura

O capítulo 2 será dedicado à revisão literária necessária ao entendimento das características da *blockchain*. O Capítulo 3 relizará um breve histórico do *bitcoin* e demonstrará a relevância tecnológica e social da *blockchain*. O capítulo 4 estabelecerá um panorama sobre a votação eletrônica e analisará os sistemas *Follow My Vote* e *BitCongress*. Finalmente, o capítulo 5 realizará a conclusão, apresentará as dificuldades encontradas na confecção deste trabalho e indicará os trabalhos futuros.

2 Revisão literária das tecnologias e conceitos envolvidos

2.1 Criptografia assimétrica

Há aproximados 40 anos, Diffie e Hellman inventaram uma forma de criptografia que era surpreendentemente nova. A criptografia assimétrica ou criptografia de chave pública era diferente de tudo aquilo que se entendia por criptografia até aquele momento, em 1976.

Em linhas gerais, a criptografia definida por Diffie e Hellman é baseada no problema do logaritmo discreto: resolver logaritmos discretos é computacionalmente difícil, mas a operação inversa de exponenciação tem custo computacional menor (método de exponenciação por quadrados). Esta assimetria é uma importante propriedade na elaboração de um sistema criptográfico. (STINSON, 2006).

A criptografia de chave pública elimina a necessidade de um canal seguro para distribuição de chaves, uma vez que habilita o envio de mensagens cifradas com a chave pública do destinatário, assegurando que apenas este, que detém a chave privada, origem matemática da chave pública ora utilizada para cifração, possa obter a mensagem decifrada. (DIFFIE; HELLMAN, 1976)

Eles também estabeleceram as condições a que um algoritmo de chave pública precisa atender, são elas:

- Deve ser computacionalmente fácil para qualquer uma das partes gerar um par de chaves (pública e privada);
- Deve ser computacionalmente fácil para um emissor, conhecendo a chave pública do destinatário, enviar uma mensagem cifrada;
- Deve ser computacionalmente fácil para um receptor decifrar a mensagem utilizando a chave privada;
- Deve ser computacionalmente inviável que, conhecendo a chave pública, seja possível determinar a chave privada; e
- Deve ser computacionalmente inviável que, conhecendo a chave pública e o texto cifrado, seja possível recuperar a mensagem original.

Segundo Stallings (2015), existem três aplicações para o uso dos criptossistemas de chave pública:

- Encriptação (ou cifração) / decriptação (ou decifração): o emissor utiliza a chave pública do destinatário para cifrar uma mensagem;

- Assinatura digital: o emissor cifra uma mensagem com sua chave privada, permitindo a qualquer pessoa verificar a integridade e a autenticidade da mensagem através da chave pública do emissor; e
- Troca de chave: cooperação entre emissor e receptor para troca de chaves em uma sessão.

Figura 1 – Esquema simplificado de criptografia de chave pública e de chave privada

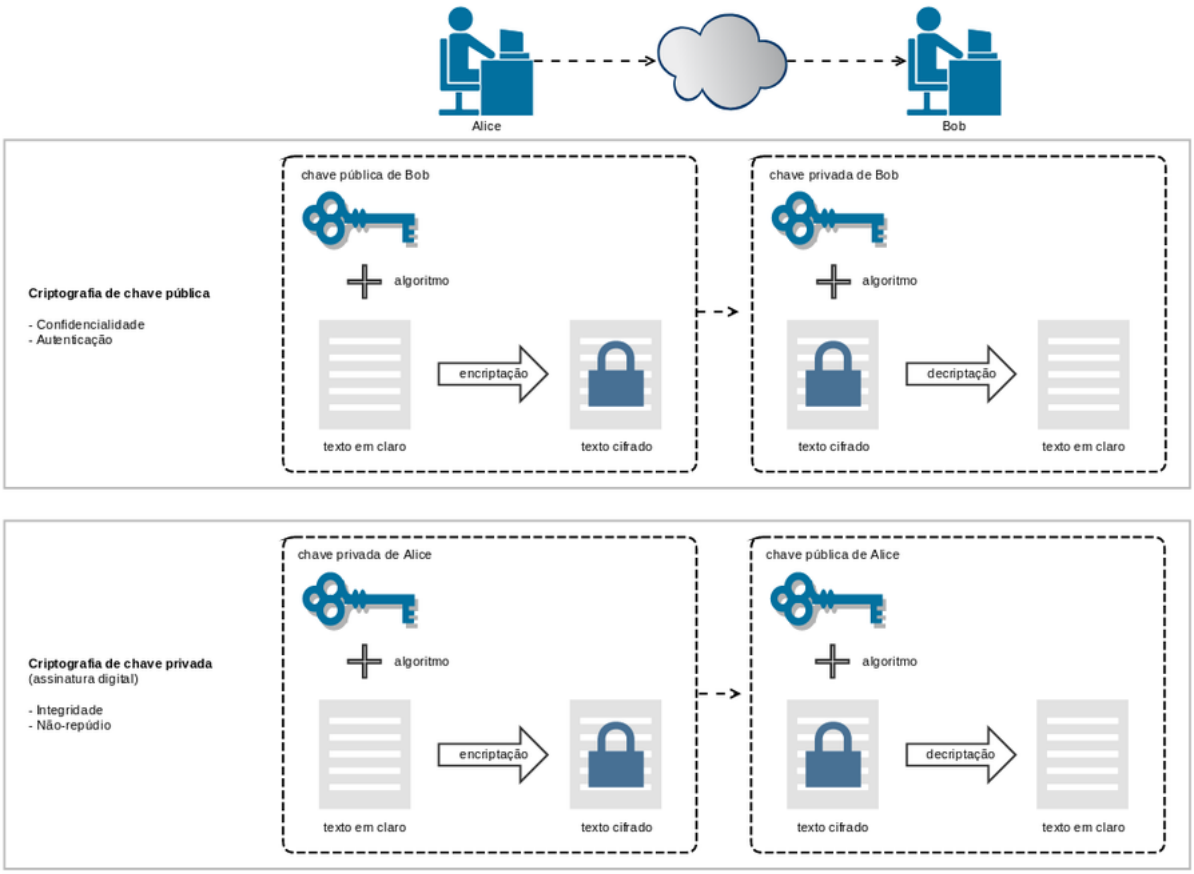


figura produzida pelo autor

Canavan (2001) destaca que as principais vantagens da criptografia de chave pública são: 1) não depender de canal seguro para distribuição de chaves, 2) suportar autenticação, 3) prover o não-repúdio e 4) ser escalável; enquanto a principal desvantagem da utilização deste tipo de criptografia consiste no grande custo computacional exigido, tornando-a consideravelmente mais lenta que a criptografia simétrica. Isto ocorre porque as chaves dos algoritmos de criptografia assimétrica precisam ser bem maiores do que as chaves dos algoritmos de criptografia simétrica para fornecerem segurança equivalente. (BLUMENTHAL, 2007)

2.2 Assinatura cega

Importante no contexto de votações eletrônicas, esquemas de assinatura cega consistem basicamente em assinar uma mensagem sem conhecer seu conteúdo.

Chaum (1982), didaticamente, ilustra a ideia básica de um esquema de assinatura cega, dando o exemplo de um envelope de papel carbono e um documento no interior deste envelope. Quando o exterior do envelope é assinado, a assinatura se transfere para o documento.

De outra maneira, Chaum (1982) demonstra:

- a função de assinatura de Bob b', conhecida apenas por ele mesmo, e correspondente ao inverso da função publicamente conhecida como b, de maneira que $b(b'(x)) = x$;
- para obter a assinatura de Bob sem conhecer o conteúdo da mensagem, Alice utiliza as funções a e a', esta inversa de a, também conhecidas apenas por ela mesma, de forma que satisfaçam a condição $a'(b'(a(x))) = b'(x)$;
- Alice envia uma mensagem m para Bob: $a(m)$;
- Bob recebe $a(m)$ e a assina $b'(a(m))$, devolvendo-a a Alice;
- Alice utiliza a' para obter b' através de $a'(b'(a(m))) = b'(m)$;
- Desta forma, Alice obtém $b'(m)$, a assinatura da mensagem realizada por Bob, sem revelar a mensagem m.

2.3 Algoritmo RSA

A fim de estabelecer uma linha comparativa com o Algoritmo de Assinatura Digital de Curva Elíptica, que é utilizado pela *blockchain* do *bitcoin*, será realizada uma breve revisão literária do Algoritmo RSA.

Desenvolvido em 1977 por Ron Rivest, Adi Shamir e Len Adelman, no MIT, o RSA utiliza chaves que são obtidas através da multiplicação de grandes números primos. A força do RSA baseia-se na dificuldade em fatorar o produto destes números, ou seja, para um computador, a multiplicação tem um custo de processamento menor se comparada à fatoração. (RIVEST; SHAMIR; ADLEMAN, 1977)

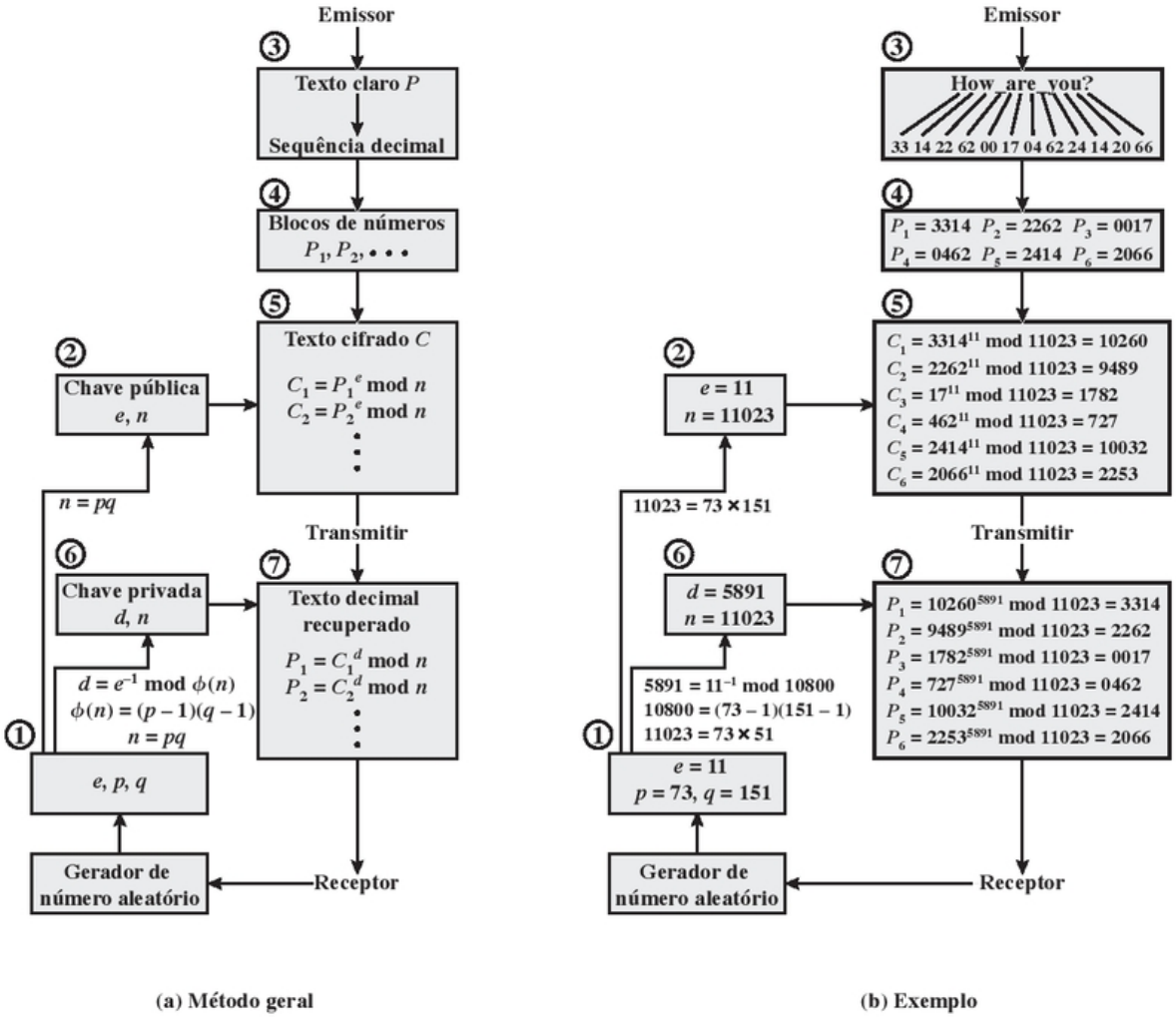
O RSA é o algoritmo de criptografia assimétrica mais utilizado no mundo. Suas forças e fraquezas estão bastante documentadas e são objeto de pesquisa constante da comunidade acadêmica. (KOSCIELNY; KURKOWSKI; SREBRNY, 2013)

Boneh (1999) documenta duas décadas de pesquisa de ataques ao RSA, concluindo que, até aquela data, o RSA continuava confiável no fornecimento de segurança para as atividades no mundo digital.

Passados 18 anos do estudo de Boneh (1999) e 40 anos da invenção do RSA, fatorar um grande número não é algo tão complexo. Segundo Bos *et al.* (2009), uma chave RSA-768 bits (232 dígitos decimais) foi fatorada em 2009 e a previsão para a fatoração de uma chave RSA-1024 bits era em 2020. No entanto, Stallings (2015) ajustou esta previsão para 2019.

O MPOG (2015) recomenda a utilização do RSA como base da infraestrutura de chaves públicas, com chaves de comprimento entre 1024 e 2048 bits, enquanto o NIST (2015) recomenda a utilização do RSA com chaves maiores ou iguais a 2048 bits, para qualquer finalidade.

Figura 2 – Esquema de funcionamento do RSA



(STALLINGS, 2015, p. 210)

2.4 Algoritmo de assinatura digital de curva elíptica

A criptografia de curva elíptica foi idealizada em 1985 por Neal Koblitz e Victor Miller como alternativa de implementação da criptografia de chave pública. A segurança deste algoritmo reside na dificuldade de resolução do problema do logaritmo discreto em curvas elípticas[1]. De forma mais acessível, detalhes sobre a matemática envolvida podem ser obtidos em Gomzin (2016, p. 89-114).

O Algoritmo de Assinatura Digital de Curva Elíptica, do inglês *Elliptic Curve Digital Signature Algorithm* (ECDSA), teve a padronização iniciada em 1998 pela ANSI e normatizado em 2000 pelo NIST, quando este também recomendou a utilização de um conjunto de curvas elípticas como parâmetro de configuração. (ANSI, 1998; NIST, 2000)

De acordo com Sangalli (2012), a parametrização das curvas é uma tarefa complexa e fundamental para segurança do criptossistema de curvas elípticas: a má escolha desses parâmetros pode inviabilizar a utilização ou a tornar insegura.

O *bitcoin* utiliza a curva secp256k1 para gerar chaves privadas de 256 bits, a partir das quais são geradas as chaves públicas por meio do processo de multiplicação escalar[2]. Esta curva foi padronizada pelo *Standards For Efficient Cryptografy Group* (SECG)[3], um consórcio de indústrias de tecnologia, cuja missão é promover a interoperabilidade e a eficiência dos criptossistemas de curva elíptica. (BITCOIN.ORG COMMUNITY, 2009; SECG, 2010)

Sobre o desempenho e a segurança dos métodos de criptografia baseados em curvas elípticas, Lara e Oliveira (2007, p. 9) concluem que:

> [...] os métodos de criptografia sobre grupos de curvas elípticas possuem segurança equivalente ao RSA, entretanto, com quantidade de bits consideravelmente menor na chave criptográfica, o que possibilita um ganho substancial de processamento. Uma vez que a chave criptográfica pode ser reduzida, podemos ter criptografia com curvas elípticas em dispositivos com pouco poder computacional [...]

Ainda sobre a segurança, o NIST (2016) estabeleceu uma relação de equivalência entre a força criptográfica das chaves do RSA e do ECDSA, onde a coluna “utilização segura” da tabela abaixo é uma previsão válida enquanto ataques quânticos não forem viáveis:

---

1 <https://www.certicom.com/content/certicom/en/ecc.html>. Acesso em: 30/10/2017.

2 consultar apêndice

3 <http://www.secg.org/>. Acesso em: 30/10/2017.

Tabela 1 – Equivalência entre algortimos e utilização segura

RSA (bits)	ECDSA (bits)	Utilização segura
1024	160 a 223	até 2010 (não seguro)
2048	224 a 255	até 2030
3072	256 a 383	após 2030
7680	384 a 511	—
15360	512+	—

(NIST, 2016 p. 52) - adaptada

Ficou demonstrado que a segurança do RSA está diretamente ligada ao tamanho da chave utilizada, requerendo grande poder de processamento. O ECDSA, por outro lado, com uma chave consideravelmente menor, consegue níveis equiparáveis de segurança, mas com ganhos expressivos de processamento e, portanto, recomendado: (RANDHAWA; SINGH, 2011)

1) para dispositivos com pouco poder computacional;
2) quando a alta velocidade das transações é um requisito;
3) quando o uso intenso de assinatura, verificação e autenticação é um requisito; e
4) quando a largura de banda é limitada.

2.5 Função *hash* SHA-256

A função *hash*, ou função de dispersão, é uma função matemática que converte uma entrada de dados de tamanho variado em uma saída de tamanho fixo chamada de *hash*.

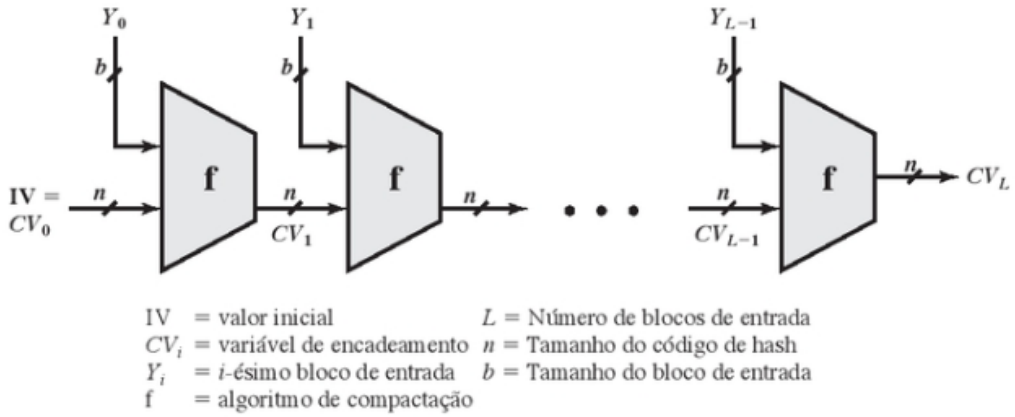
Para Smart (2003), as funções hash criptográficas (H) precisam satisfazer três propriedades:

- Resistência à pré-imagem (propriedade de mão única): deve ser computacionalmente inviável obter a mensagem original (x), dado o valor do *hash* (h), de modo que $H(x) = h$;
- Resistência à colisão (ou resistência à colisão forte): deve ser computacionalmente inviável encontrar duas mensagens (x, y) com o mesmo valor de *hash* (h), de modo que $H(x) = H(y)$; e

- Resistência à segunda pré-imagem (ou resistência à colisão fraca): deve ser computacionalmente inviável encontrar $x \neq y$, de modo que $H(x) = H(y)$.

O SHA-256 (*Secure Hash Algorithm*) utiliza o método Merkle-Damgard para converter uma entrada de dados variável em uma saída de tamanho fixo, cujo funcionamento pode ser sintetizado da forma seguinte: 1) divide-se a entrada de dados em blocos de tamanho fixo (no caso do SHA-256, o bloco tem 512 *bits*). 2) Para cada bloco, adicionado o *hash* do bloco anterior, é gerado um novo *hash*. No caso do primeiro bloco, para quem não há o *hash* do bloco anterior, é utilizado um vetor de inicialização. O bloco final inclui o tamanho total da entrada. (STALLINGS, 2015)

Figura 3 – Estrutura geral do código de hash seguro



(STALLINGS, 2015, p. 257)

Figura 4 – Esquema de enderaçamento do bitcoin

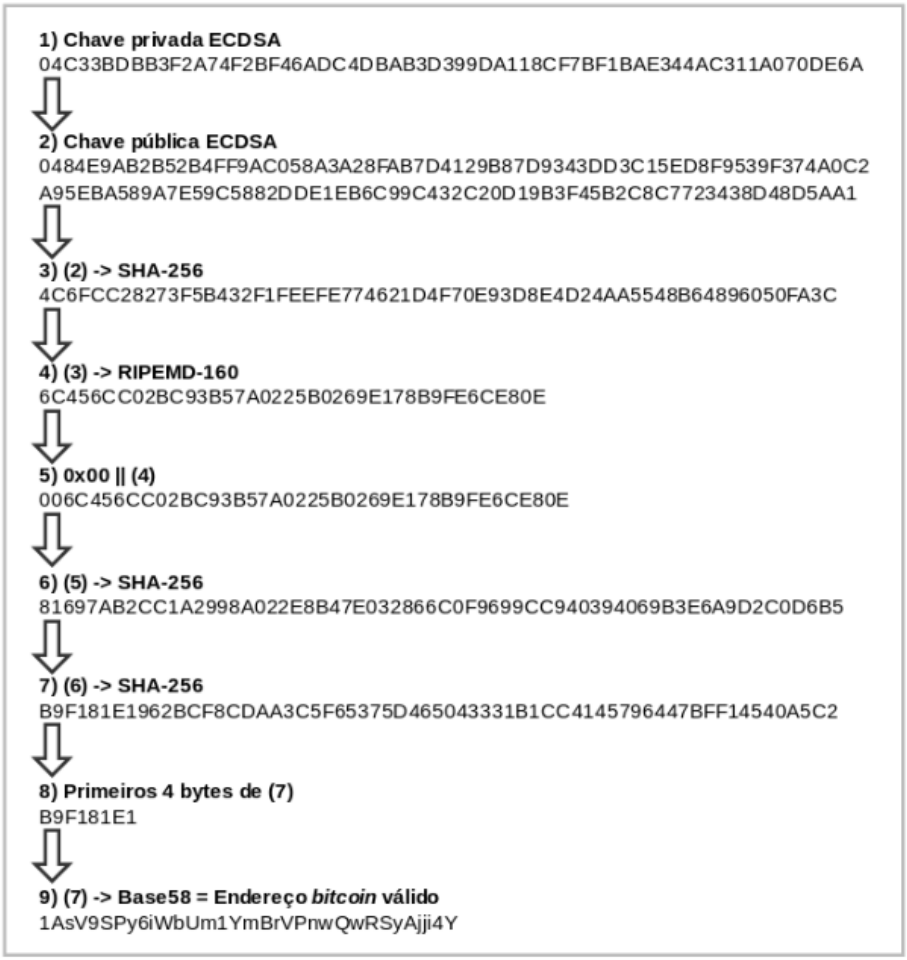


figura produzida pelo autor

O bitcoin utiliza o SHA-256 em rodadas, juntamente com outras funções, para gerar endereços válidos. O objetivo é dificultar a ocorrência de colisões e garantir o máximo de segurança quanto à unicidade do endereço. É mais provável o planeta ser destruído nos próximos 5 segundos do que uma colisão ocorrer no próximo milênio[4].

O SHA-256 também é utilizado na geração da prova de trabalho do *bitcoin*, que, sinteticamente, consiste na realização de cálculos matemáticos com o objetivo de encontrar um resultado de *hash* menor que um alvo proposto pelo algoritmo do *bitcoin* no exercício do controle do nível de dificuldade da mineração.

2.6 Árvore de Merkle

É uma estrutura de dados criada por Ralph Charles Merkle em 1982 que utiliza ponteiros *hash* em estrutura de árvore binária. Seu funcionamento consiste em agrupar

[4] <https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses>. Acesso em: 30/10/2017.

pares de blocos de dados, cujos respectivos *hashes* são submetidos a uma nova rodada para obtenção de novo *hash*. A operação se repete até alcançar o último nó do topo da árvore, gerando a raiz da árvore de Merkle.

Qualquer alteração em um bloco de dados propagará a consequente modificação do *hash* até a raiz, característica que torna a árvore de Merkle uma estrutura de dados bastante resistente a adulterações. (FRANCO, 2015)

Esta estrutura exerce um papel muito importante na *blockchain* do *bitcoin*, pois através dela é possível verificar se uma transação pertence a um determinado bloco por meio da simples conferência dos agrupamentos de *hashes*. (BITCOIN.ORG COMMUNITY, 2009)

Figura 5 – Esquema de uma Árvore de Merkle

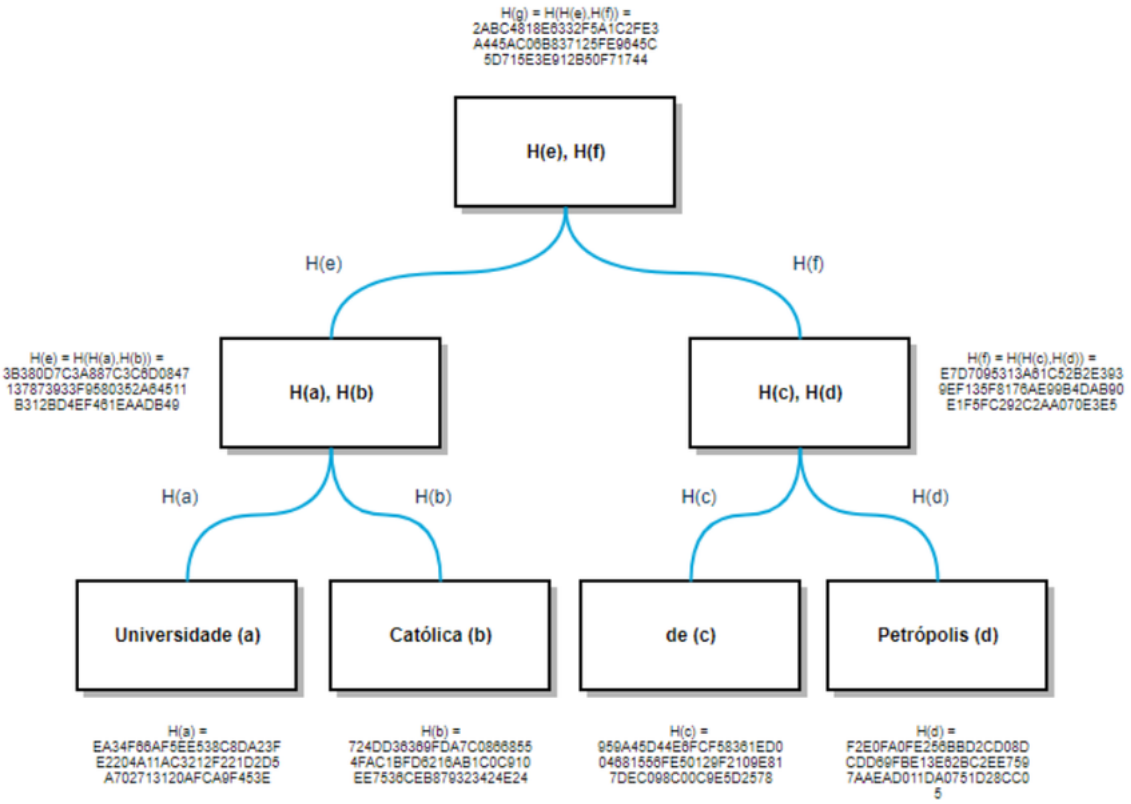


figura produzida pelo autor

2.7 Mecanismo de consenso

Os pilares do *bitcoin* são a criptografia e o consenso. A criptografia é responsável por garantir a integridade e a validade das transações enquanto o consenso é responsável pela extensão da *blockchain*.

É importante mencionar que, embora o *bitcoin* faça uso intensivo de funções criptográficas, as informações contidas na *blockchain* não são criptografadas. A criptografia é utilizada para garantir a integridade das informações, que permanecem visíveis e sob escrutínio constante dos utilizadores da rede.

Sem prejuízo de uma melhor explicação adiante, a *blockchain* pode ser entendida como um conjunto descentralizado de nós especializados que executam as funções definidas pelo algoritmo do *bitcoin*. São estes nós os responsáveis por alcançar o consenso. Antonopoulos (2014) os classificou como:

- Nós completos *(full nodes)*: realizam a cópia de toda *blockchain*, aplicam as regras de consenso e realizam a conferência de *hashes*.
- Nós leves *(light nodes)*: não possuem a cópia integral da *blockchain*, mas apenas um subconjunto de dados para realizarem a verificação das transações, executando um método chamado *Simplified Payment Verification* (SPV). Este método consulta um nó completo para obter as informações.
- Nós mineradores *(miner nodes)*: escutam a rede à espera de transações e as inserem em um bloco, após a execução da prova de trabalho. Funcionam mediante incentivo, tanto pela criação de novos blocos quanto pela validação de transações. Exibem as características de nós completos quando mantêm a cópia da *blockchain* e de nós leves quando estão associados a um *pool* de mineração.

Alcançar o consenso em uma rede de nós não confiáveis não é uma tarefa trivial, posto que é imperativo considerar a existência de nós e processos maliciosos que tentam subverter o funcionamento regular da rede.

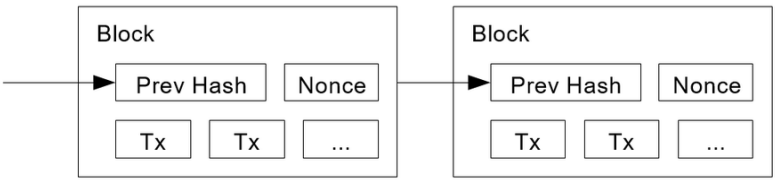
Dado que não há entidade central que verifique as identidades ou as credenciais de nós ingressos na rede, ataques *Sybil* devem ser considerados. Estes ataques são viabilizados através da criação de nós que são controlados por usuários maliciosos, com a finalidade de alteração do consenso, o que, dentro do escopo de sistemas eletrônicos de votação, assume elevada criticidade (NEMATI, 2011). Felizmente, os ataques *Sybil* têm pequena probabilidade de sucesso no *bitcoin*. Isto se deve à aleatoriedade do algoritmo na escolha dos nós que realizarão a propositura dos blocos candidatos. (NARAYANAN *et al.*, 2016)

O funcionamento do algoritmo pode ser sintetizado da seguinte maneira: quando um usuário precisa transferir determinada quantia de *bitcoins* para outro usuário, ele propaga esta transação para todos os nós da rede. Os nós mineradores *(miner nodes)* recebem as transações e as validam, conferindo se as assinaturas estão corretas e

se as moedas transacionadas foram gastas anteriormente, com posterior registro de todas as operações. Após validadas pelos mineradores, um nó completo *(full node)*, aleatoriamente escolhido, agrupa as transações em um bloco candidato a compor a cadeia consensual, estendendo-se a partir do último bloco válido da cópia local atualizada da *blockchain*. Como um nó minerador *(miner node)* também pode ser um nó completo *(full node)*, eles também podem gerar seus próprios blocos candidatos caso sejam aleatoriamente selecionados.

O trabalho computacional mais intenso começa a partir deste momento, com a procura do *nonce*[5] para realizar a validação do bloco candidato. Apesar de todo este esforço, não há certeza de que o bloco candidato estenderá a cadeia consensual. Isto dependerá de outros nós aceitarem o bloco candidato como antecessor válido e passarem a trabalhar referenciando-o, situação expressa pela inclusão do *hash* do bloco propagado no cabeçalho do próximo bloco a ser criado, ou seja, os nós precisam entrar em acordo sobre quais transações realmente ocorreram e em que ordem elas ocorreram (tarefa do mecanismo de consenso). Todo o fluxo se repete a cada dez minutos, com cada nó propagando suas transações para comporem um novo bloco.(NARAYANAN *et al.*, 2016; FRANCO, 2015; ANTONOPOULOS, 2014; NAKAMOTO, 2008)

Figura 6 – Encadeamento dos blocos



(NAKAMOTO, 2008, p. 3)

2.7.1 Prova de trabalho

Para que um novo bloco seja adicionado à cadeia consensual, é necessário que um nó minerador, ou um conjunto desses nós, resolva um desafio: encontrar um *hash* numericamente menor que um alvo.

Observe o seguinte exemplo:

[5] ver seção 2.7.1

Tabela 2 – Exemplo de alvo da prova de trabalho do bitcoin

Entrada de dados	*Hash* SHA-256
Universidade Católica de Petrópolis**01**	CAD9FDC01B5A27581C63DB7E2AA2A9B62CF274A 85460CE22D85374BD3F458458
Universidade Católica de Petrópolis**02**	806CC08C3F6639DCD106848378D441F10894617 688AE94B2A58437606A1F4821
Universidade Católica de Petrópolis**03**	A440F49CF237D55379C2E2027928C7AD35B6838 EB2E51B76990B35D6832A7B6A
...	...
Universidade Católica de Petrópolis**10**	**0**A1E39CEA7D02FA345CDE08B43D79631AA45D1E 9A30B6A8657D86DDEB78861AF

Observe que cada sequência de caracteres resulta em um *hash* completamente diferente. Mesmo uma alteração sutil na entrada de dados provoca uma alteração completa no resultado da função *hash*.

O número usado como variável neste exemplo é chamado de *nonce*, cuja finalidade é modificar a saída da função *hash* SHA-256.

O desafio no exemplo acima é encontrar um *hash* hexadecimal que inicie com zero. Percebe-se que foram necessárias 10 tentativas para encontrar o alvo proposto. Em termos de probabilidades, se a saída da função *hash* estiver distribuída uniformemente, esperaríamos um resultado com um 0 como o prefixo hexadecimal uma vez a cada 16 *hashes* (de 0 a F). Em termos numéricos, isso significa encontrar um valor de hash menor que:

```
0x1000000000000000000000000000000000000000000000000000000000000000
```

Objetivo da prova de trabalho *(proof-of-work)* é encontrar um *hash* que seja numericamente menor que o alvo. Se diminuirmos o alvo, a tarefa de encontrar um *hash* se torna cada vez mais difícil. Portanto, a dificuldade e o tamanho do alvo são inversamente proporcionais.

De forma mais didática, considere jogadores que lançam um par de dados repetidas vezes para atingir um valor menor do que um número qualquer definido como alvo. Imagine que o alvo seja 12. Neste caso, apenas aquele que lançar os dados e

obter um duplo 6 perderá. Todas as outras combinações serão aceitas como vitoriosas. Agora imagine o alvo como 3. Percebe-se que atingir uma combinação vitoriosa torna-se mais difícil. Neste caso existe apenas uma combinação possível, 1 + 1 ou 1/36, algo em torno de 2,78%, ou seja, é possível estimar a quantidade trabalho requerida a partir da dificuldade imposta pelo alvo.

No *bitcoin*, o cálculo da prova de trabalho *(proof-of-work)* acontece praticamente da mesma maneira. O nó minerador calcula o *hash* SHA-256 de um bloco candidato e verifica se este *hash* é menor do que o alvo corrente. Se o *hash* calculado for maior do que alvo, o nó minerador incrementa o *nonce* e calcula um novo *hash* até encontrar algum que seja menor do que o alvo.

A dificuldade na rede do *bitcoin* é calculada e recalculada, dinamicamente, de forma a adequar o tempo de criação do bloco, que ocorre a cada dez minutos. Se há aumento do poder computacional na rede, o algoritmo descentralizado do bitcoin aumenta a dificuldade para manter a criação de um novo bloco a cada 10 minutos. O algoritmo realiza o cálculo tendo como base os últimos 2016 blocos, cujo tempo esperado de criação deverá ser 20160 minutos. Ou seja, exatamente a cada duas semanas um novo cálculo da dificuldade é realizado. (ANTONOPOULOS, 2014)

2.8 Aspectos de segurança

Será realizada uma breve análise dos principais ataques aos quais o *bitcoin* está sujeito e, analogamente, qualquer aplicação descentralizada que utiliza uma *blockchain* baseada em nós não confiáveis.

As questões de segurança no âmbito das votações que utilizam uma *blockchain* como plataforma tecnológica assumem grande relevância, especialmente o ataque contra o sigilo e os ataques contra a integridade, tais como os ataques de gasto duplo e de 51%.

2.8.1 Ataque contra o sigilo

No *bitcoin* não há correlação entre a identificação no mundo real e a identificação na rede, uma vez que não há a figura do terceiro de confiança responsável por este gerenciamento. Logo, em tese, usuários não podem ser punidos por comportamentos maliciosos nem serem associados à posse de determinada quantia da criptomoeda. Todavia, como não há anonimato no *bitcoin*, mas “pseudonimato”, é possível a verificação de padrões que possibilitem algum grau rastreamento.

No caso das votações sobre *blockchain*, é razoável admitir a necessidade de formas adicionais de proteção a fim de assegurar o sigilo do voto e evitar a coerção do

votante, especialmente através da utilização de esquemas de assinatura cega.

2.8.2 Ataque de gasto duplo

Suponha-se que Alice realizou a transferência de determinada quantia para Bob devido à aquisição *online* de uma licença de *software*. Bob, inadvertidamente, cedeu o *link* de acesso à licença logo após verificar que a transação havia acontecido, porém sem estar inserida em nenhum bloco da cadeia consensual, existindo apenas em bloco candidato gerado pelo nó aleatório escolhido pelo algoritmo.

Considerando que o próximo nó aleatório escolhido pelo algoritmo seja um nó controlado por Alice e que ela tenha percebido a situação gerada por Bob, Alice então submete um novo bloco candidato de transações que ignora a transferência de fundos outrora realizada, reutilizando-os em outra transação.

Após algum tempo, o bloco que continha a transação de transferência de fundos de Alice para Bob recebeu a primeira confirmação (bloco B), juntamente com o bloco contendo o gasto duplo (bloco C), devido a problemas de latência da rede.

Ressalte-se que o bloco C, gerado pelo nó controlado por Alice aponta para o mesmo bloco A, que também é referenciado pelo bloco B. A política do algoritmo do *bitcoin*, seguida por nós honestos, define que um novo bloco será aposto à cadeia de maior comprimento (cadeia consensual). Sobre qual bloco a cadeia se estenderá, visto que ambas as cadeias AB e AC têm o mesmo comprimento? Neste exemplo, qualquer uma das cadeias poderá ser estendida e Alice poderá ter sucesso no ataque.

A proteção contra ataques de gasto duplo é puramente consensual e depende que a maioria da composição da rede seja de nós honestos, pois quanto mais confirmações um bloco tiver, menor a chance de um atacante ter sucesso.

2.8.3 Ataque de 51%

O *bitcoin* está sujeito a ataques de mineradores (ou *pools* de mineração) no que diz respeito ao uso desonesto/destrutivo do poder de mineração. Como visto anteriormente, para atingir o consenso em um ambiente de nós não confiáveis, o *bitcoin* recorre a uma política de incentivos para manter o nós agindo de forma honesta. No entanto, é possível que um grupo de mineradores possuidores de grande parte do poder computacional da rede possa atacar o consenso de forma a estender outra cadeia que não seja a principal, na tentativa de criar uma nova cadeia consensual.

A medida que o tempo passa, novos blocos são adicionados à cadeia consensual, tornando cada vez mais difícil e complexa a alteração de blocos antigos. No máximo, um ataque bem sucedido, afetará o consenso futuro e alguns dos blocos

recentemente adicionados. (ANTONOPOULOS, 2014)

O ataque de 51% baseia-se nessa premissa e seus objetivos são: realizar ataques de gasto duplo ou promover ataques de negação de serviço.

À primeira vista, sem prejuízo da análise conclusiva, como a proteção da rede é consensual e baseada em incentivos monetários, parece razoável admitir a ideia de que apenas os incentivos do algoritmo não sejam suficientes para manter o comportamento honesto da rede, uma vez que a interferência no processo eleitoral possa constituir um incentivo maior.

3 Blockchain

3.1 Histórico

Com a crise do *subprime* em agosto de 2007, o sistema financeiro global foi levado ao limite devido às condutas de assunção de riscos excessivos por parte de entidades norte-americanas na concessão de empréstimos a famílias de baixa renda, garantidos pelas hipotecas dos imóveis e, em última instância, pelo Tesouro norte-americano. Em pouco tempo, bancos do mundo inteiro haviam investido em títulos lastreados em hipotecas americanas.

Os tomadores dos empréstimos começaram a ter dificuldade em realizar o pagamento da dívida inicial, dando início a um ciclo de inadimplência que, posteriormente, gerou uma grave crise de liquidez, levando bancos à insolvência no mundo inteiro. Esta crise gerou milhões de desempregados, tornado-se um problema político, social e econômico de dimensões globais. (ROQUE, 2013)

Stiglitz (2009, p. 326) creditou as condutas improvidentes do sistema financeiro à estrutura de incentivos utilizada:

> [...] estruturas de incentivos para a maioria dos altos executivos e muitos dos agentes de crédito desses bancos foram desenhadas para encorajar um comportamento míope e de riscos excessivos [...]

O resultado desta crise foi a quebra da confiança entre as pessoas e o sistema financeiro, o que levou Satoshi Nakamoto[1], a propor a criação da moeda digital *bitcoin* e de um sistema de pagamentos.

O que Nakamoto provavelmente não esperava era que a infraestrutura tecnológica proposta para apoiar o funcionamento do *bitcoin* fosse viabilizar a implementação de demandas tão importantes da sociedade.

A *blockchain* é um marco tecnológico e, para alguns, a descoberta mais importante desde a Internet[2] [3] [4], com profundas implicações no relacionamento das pessoas, seja uma com as outras, ou com entidades públicas ou privadas.

---

[1] provavelmente um pseudônimo referente a uma pessoa ou a um grupo de pessoas. Até hoje não se sabe ao certo quem é Satoshi Nakamoto.

[2] <https://fortunascorner.com/2017/09/24/bitcoin-important-invention-decade-block-chain-important-invention-since-birth-internet-says-silicon-valley-visionary/>. Acesso em: 30/10/2017.

[3] <http://www.nasdaq.com/article/why-the-bitcoin-blockchain-is-the-biggest-thing-since-the-internet-cm608228>. Acesso em: 30/10/2017.

[4] <http://fortune.com/2016/05/08/why-blockchains-will-change-the-world/>. Acesso em: 30/10/2017.

3.2 Conceitos básicos

Entendida a motivação de Nakamoto, fica evidente que a principal característica da tecnologia proposta é a descentralização com o fim de esvaziar o papel do terceiro de confiança, neste cenário, substituído pela criptografia e pelo consenso.

As regras de segurança da *blockchain* são consensuais e tecnológicas. No contexto do *bitcoin*, tem como principais funções evitar o gasto duplo e garantir a auditabilidade, tradicionalmente exercidas por instituições bancárias. As transações são agrupadas em blocos criptográfica e cronologicamente encadeados, cuja persistência é distribuída em rede P2P, característica que lhe garante alta tolerabilidade a falhas. (WINTER, 2016)

Segundo Brakeville e Perepa (2016), a *blockchain* é um livro razão descentralizado para troca de ativos que realiza o armazenamento seguro e verificado de dados, tornando-a adequada a suportar uma gama de aplicações: armazenamento em nuvem, emissão de documentos (identidades, passaportes, certidões etc), contratos inteligentes, votação eletrônica, governo eletrônico dentre outras. [5]

A simplicidade e a robustez desta tecnologia tem gerado grande interesse junto a governos, empresas de tecnologia e empresas do setor financeiro. Para citar algumas iniciativas, a IBM, em parceria com a *Linux Foundation*, criou o projeto *Hyperledger*, que consiste no desenvolvimento de um *framework blockchain* de nível empresarial para criação de aplicações descentralizadas. (IBM, 2017)

A Google, de acordo com o relatório da CBInsights[6], alcançou em 2017 o posto de segunda maior investidora na tecnologia *blockchain* através dos serviços/empresas *Storj*[7] (serviço de armazenamento em nuvem baseado em *blockchain*), *LedgerX*[8] (primeira casa de câmbio de *bitcoin* com regulamentação federal nos Estados Unidos) e *Veem*[9] (plataforma de pagamentos globais B2B).

Citi, Nasdaq, Visa e outras grandes instituições estão investindo em serviços como o *chain.com*, que permite a criação e operação de *blockchains* privadas[10] [11].

Na esfera governamental, a Estônia[12] foi o primeiro país no mundo a utilizar uma *blockchain* privada como infraestrutura tecnológica de serviços públicos em escala

[5] <https://www.huffingtonpost.com/ameer-rosic-/5-blockchain-applications_b_13279010.html>. Acesso em: 30/10/2017.

[6] <https://www.cbinsights.com/research/report/blockchain-trends-opportunities/> Acesso em: 30/10/2017.

[7] <https://storj.io/index.html> Acesso em: 30/10/2017.

[8] <https://ledgerx.com/about-ledgerx/> Acesso em: 30/10/2017.

[9] <https://www.veem.com/our-story/>. Acesso em: 30/10/2017.

[10] <https://chain.com/press-releases/chain-raises-30-dollars-million-from-financial-industry-leaders/>. Acesso em: 30/10/2017.

[11] <http://www.nasdaq.com/article/nasdaq-and-citi-announce-pioneering-blockchain-and-global-banking-integration-cm792544>. Acesso em 30/10/2017.

[12] <https://e-estonia.com/>. Acesso em: 30/10/2017.

nacional, tais como o *e-Health*, que armazena todos os registros médicos dos cidadãos de forma distribuída; o *e-Law*, que armazena toda a produção legislativa do país; o *e-Court*, sistema de processos judiciais eletrônicos e recentemente o *e-Residency*[13], que emite identidades digitais para não nacionais serem cidadãos eletrônicos da Estônia, possibilitando-lhes a abertura de negócios *online* e lhes fornecendo acesso a todos os serviços públicos necessários para operar o empreendimento remotamente.

[13] <https://e-resident.gov.ee/>. Acesso em: 30/10/2017.

4 Sistemas de votação eletrônica sobre blockchain

A evolução dos sistemas criptográficos nas últimas décadas viabilizou a criação de sistemas eletrônicos de votação, cuja adoção visa tornar o processo eleitoral mais rápido e menos sujeito à interferência humana, esta última, característica que, paradoxalmente, torna-se alvo de críticas: se por um lado, com utilização dos sistemas eletrônicos de votação obtêm-se a rápida apuração dos resultados e a menor interferência humana no escrutínio, por outro, esta interferência passa a ser exercida pelo corpo técnico responsável em confeccionar e manutenir tais sistemas, o que pode prejudicar a sua adoção, uma vez que a complexidade natural do software se antagoniza com a segurança, com a correção do código, com vulnerabilidade a hackers e, em última instância, com a possibilidade do comprometimento da integridade do resultado das eleições, pois é técnica ou/e economicamente inviável encontrar todos os erros de um sistema complexo. (FELDMAN; HALDERMAN; FELTEN, 2006; SOMMERVILLE, 2007)

Segundo a OSCE (2008), eleições que utilizam sistemas eletrônicos são menos transparentes do que as eleições baseadas em sistemas de cédulas de papel. A falta de transparência dos sistemas eletrônicos de votação é o maior desafio para que sua implementação ocorra em maior escala.

Considerando a problemática da falta de transparência, em 2009, a Suprema Corte Alemã considerou o voto eletrônico inconstitucional por entender que havia o cerceamento do direito à verificabilidade dos resultados por pessoas sem conhecimento especializado[1].

Segundo Vegas e Barrat (2017), há de se considerar a natureza pública das eleições quanto à verificabilidade dos resultados: enquanto a recontagem de cédulas de papel pode ser entendida e monitorada por qualquer pessoa, a recontagem de votos em um sistema eletrônico de votação só pode ser entendida e monitorada por pessoal especializado.

Adicionalmente, a eleição é uma atividade cuja complexidade abrange aspectos legais, políticos, culturais, ambientais e logísticos, onde pequenas alterações em qualquer um destes aspectos podem afetar profundamente o processo eleitoral, contexto que pode dificultar ainda mais a adoção da sistemática eletrônica de votação (GOLDSMITH, 2017).

De acordo com pesquisa realizada pelo IDEA *(International Institute for Democracy and Electoral Assistance)*[2], com 123 países participantes, verificou-se que 57% nunca realizaram eleições sob a sistemática eletrônica, 21% utilizam o voto eletrônico em eleições nacionais e/ou estaduais e/ou municipais, 16% ainda não utilizam o voto

[1] <https://www.ndi.org/e-voting-guide/examples/constitutionality-of-electronic-voting-germany>. Acesso em: 30/10/2017.

[2] <https://www.idea.int/news-media/media/use-e-voting-around-world>. Acesso em: 30/10/2017.

eletrônico, embora estudos de viabilidade estejam sendo realizados e 6% já utilizaram o voto eletrônico, mas o abandonaram.

Rivest e Virza (2017) definiram que um sistema de votação é dito *software*-independente quando uma alteração ou erro identificados neste *software* não conseguem promover, de forma indetectável, a alteração no resultado da eleição. Ou seja, é inaceitável que uma alteração (intencional ou não) ou erro de *software* cause mudança no resultado de uma eleição sem deixar nenhuma evidência disponível.

Por este motivo, é necessário que haja uma trilha de auditoria robusta, a fim de que seja realizada a validação do resultado, que não pode depender exclusivamente do nível de correção do software.

Pouco tempo depois da definição do termo *software*-independente, novos estudos expandiram este conceito, culminando na ideia dos sistemas de votação eletrônica E2E *(end-to-end verifiable).* (RIVEST; VIRZA, 2017)

Um sistema de votação eletrônica E2E permite que os eleitores verifiquem o resultado final sem necessariamente confiar na estrutura (*hardware*, *software* e pessoal) utilizada na execução das eleições. Para ser considerado E2E, um sistema deve atender os seguintes requisitos (RIVEST; VIRZA, 2017):

- Realizado como pretendido: um eleitor deve poder verificar se seu voto foi realizado da forma como foi pretendido;
- Gravado como realizado: qualquer pessoa deve poder verificar se seu voto foi gravado da forma como foi realizado;
- Apurado como gravado: qualquer pessoa deve poder verificar se seu voto foi apurado da forma como foi gravado;

Sistemas de votação eletrônica baseados em *blockchain*, ou sistemas criptográficos de votação, possuem grande aderência aos conceitos de independência de *software* e são considerados fortes exemplos desta categoria, uma vez que habilitam a verificação extensiva das transações (votos), que se encontram imutavelmente registradas e publicizadas na cadeia de blocos. (RIVEST; VIRZA, 2017)

4.1 Follow My Vote

Na Federação norte-americana, os Estados possuem ampla autonomia legislativa, inclusive na esfera eleitoral, característica que torna o processo eleitoral complexo e heterogêneo. São várias as formas admitidas de registro de eleitores: via postal, *online* e presencial. Há, ainda, 17 Estados da Federação que não exigem o registro

do eleitor para o exercício do voto[3]. Além disso, convivem: a votação tradicional em cédulas, máquinas de cartão perfurado, urnas eletrônicas dentre outras formas de votação[4].

O *Follow My Vote*[5] é um projeto iniciado em 2014 por Adam Kaleb Ernest e Nathan Hourt, baseado em Blacksburg, Virgínia, EUA. A motivação para o desenvolvimento foi oferecer uma alternativa segura e de baixo custo aos atuais sistemas utilizados nos Estados Unidos, nos quais foram detectados casos potenciais de fraude, manipulação de cédulas, vulnerabilidade tecnológica e processos falhos de registro. (VARSHNEYA; POUDEL; VYAS, 2015)

Uma dessas vulnerabilidades pôde ser verificada em 2012, quando um eleitor do Estado do Novo México conseguiu registrar com sucesso um cachorro para votar na eleição presidencial de 2012, caso que foi amplamente repercutido pela imprensa norte-americana[6] [7].

O *Follow My Vote* utiliza a *blockchain BitShares*[8], um *fork* da *blockchain* do *bitcoin*, que se baseia no conceito de DACs *(Decentralized Autonomous Companhies)*. Basicamente, uma DAC é uma companhia que funciona sem controle humano centralizado. Sob esta ótica, o *bitcoin* é semelhante a uma DAC:

- recompensa - o *bitcoin* cobra taxas pelas transações realizadas, assim como empresas cobram pela prestação de serviços;
- empregados - os mineradores são responsáveis pela validação e segurança das transações que ocorrem na rede;
- custos - os mineradores recebem incentivos que provêm das taxas cobradas pelas transações realizadas na rede;
- acionistas - são aqueles que possuem a criptomoeda que, neste caso, são as ações da DAC. Se as "ações" se valorizam, os acionistas têm lucro, caso contrário, têm prejuízo;
- direito a voto - os acionistas possuem direito a voto nas decisões da empresa. Mudanças no código precisam ser aprovadas por toda a comunidade;
- produto - o *bitcoin* é um serviço de pagamentos e de transferência de valores; e

[3] <http://www.ncsl.org/research/elections-and-campaigns/voter-id.aspx>. Acesso em: 30/07/2017.
[4] <https://ballotpedia.org/Voting_methods_and_equipment_by_state>. Acesso em: 30/10/2017.
[5] <https://followmyvote.com/>. Acesso em: 30/10/2017.
[6] <https://www.huffingtonpost.com/2012/03/01/dog-registered-to-vote-thomas-tolbert_n_1314963.html>. Acesso em 30/10/2017.
[7] <https://theweek.com/articles/477605/dog-whos-registered-vote-democrat>. Acesso em 30/10/2017.
[8] <https://bitshares.org/>. Acesso em: 30/10/2017.

- consumidores - qualquer pessoa que o utilize o *bitcoin* como produto.

O *Follow My Vote* é uma DAC de votação eletrônica, cuja ideia geral é utilizar um esquema de assinatura cega com *tokens* transacionados, representando os votos. Didaticamente, seu funcionamento pode ser sintetizado da seguinte maneira: (ERNEST, 2014)

1) inicialmente, o eleitor solicita o credenciamento a uma entidade centralizada, que verifica a documentação submetida, confirmando ou não a identidade do eleitor. Caso a identidade seja confirmada, o eleitor recebe um par de chaves;
2) no próximo passo, o eleitor submete um *token* ofuscado juntamente com a chave pública recebida pela entidade verificadora. Esta entidade verifica se o eleitor está devidamente registrado e se já tentou registrar algum outro *token* previamente;
3) se tudo estiver correto, o *token* ofuscado é assinado pela entidade verificadora, garantindo-se a unicidade e a validade do *token;*
4) na etapa seguinte, o eleitor recebe o *token* assinado pelo verificador e emite um novo par de chaves a partir de um pseudônimo que é desvinculado do primeiro par de chaves e, portanto, desassociado de sua identidade real;
5) após, o eleitor encaminha o *token* em claro devidamente assinado, juntamente com a nova chave pública, para a entidade verificadora;
6) ao recebê-los, a entidade checa se o *token* em claro, previamente assinado, já foi utilizado;
7) se tudo estiver correto, a entidade verificadora assina o *token* em claro e o envia novamente para o eleitor; e
8) neste ponto do processo, o eleitor, anônimo, possui um *token* (cédula) válido e está apto a votar.

Figura 7 – Processo de registro e habilitação do eleitor

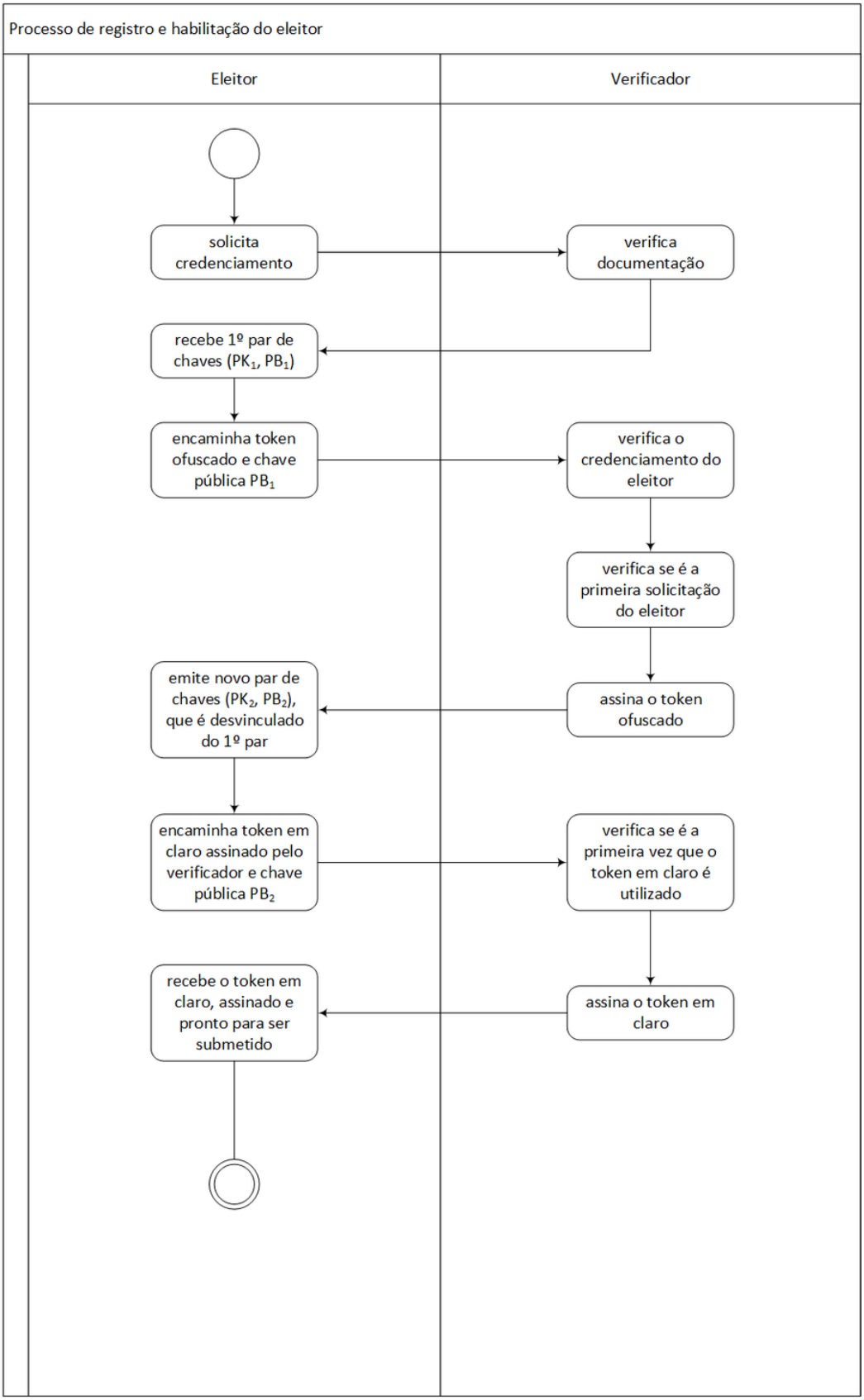


figura produzida pelo autor

O algoritmo de consenso utilizado pelo *Follow My Vote* é o DPOS *(Delegated Proof Of Stake)*, uma forma modificada do algoritmo de prova de participação, que delega a usuários considerados confiáveis as tarefas de verificar e validar as transações nos blocos.

Em relação ao desempenho, o DPOS elimina a necessidade de esperar as confirmações de um bloco realizadas por vários nós não confiáveis, como o ocorre no algoritmo de prova de trabalho do *bitcoin*. Essa característica confere maior velocidade no processamento das transações e, de forma geral, suaviza o problema de escala existente, típico das *blockchains* que utilizam a prova de trabalho como algoritmo de atingimento do consenso.

É importante ressaltar que, mesmo utilizando o DPOS, o problema da centralização continua ocorrendo, pois há uma concentração dos poderes de verificação e validação nas mãos dos delegados eleitorais. No entanto, cada nó da rede tem a faculdade de decidir a qual nó delegado irá confiar sua representação. Há, portanto, uma dupla preocupação: a primeira é afeta à segurança, pois, em se tratando de prova de trabalho, é possível haver concentração excessiva do poder computacional nas mãos de poucos, o que fragiliza a estrutura de crescimento da cadeia consensual, sujeitando-a a ataques; a segunda é concernente ao desempenho, pois há, de fato, uma limitação de escala na *blockchain* do *bitcoin*, cuja capacidade de processamento é de apenas 7 transações por segundo, enquanto a *blockchain BitShares* 2.0, que utiliza DPOS, pode processar até 100.000 transações por segundo. (SCHUH, 2018)

O algoritmo de verificação utilizado é o *momentum*[9]. Em contraponto, o *bitcoin* utiliza o SHA-256 baseado em prova de trabalho. Não há muitas informações publicadas sobre o *momentum*. Sabe-se apenas que ele é parecido com o algoritmo *scrypt*[10], que necessita de memórias de alta velocidade para sua execução, reduzindo a viabilidade de construção e, consequentemente, de utilização de chips específicos denominados ASIC *(application-specific integrated circuit)* para realização dos cálculos, visando aumentar a segurança da rede, especialmente em relação à centralização do poder de processamento.

Embora o *Follow My Vote* tenha sido projetado para ser descentralizado, fica evidente a necessidade de centralização em alguns aspectos do sistema, especialmente em relação ao credenciamento dos eleitores e à emissão e verificação dos *tokens* de votação. Sem a centralização destas etapas, o sistema pode se tornar vulnerável a ataques *Sybil*.

[9] <http://www.hashcash.org/papers/momentum.pdf>. Acesso em 30/10/2018.
[10] <https://www.tarsnap.com/scrypt/scrypt.pdf>. Acesso em 30/10/2018.

4.2 BitCongress

O *BitCongress* é um sistema de governança democrática idealizado por Morgan Rockwell com a finalidade de criar e gerenciar eleições e todo o processo legislativo. Tendo escopo mais amplo que o do *Follow My Vote*, o *BitCongress* aspira a se tornar uma grande plataforma para todos os propósitos democráticos.

O *BitCongress* realiza a integração entre *bitcoin*, *counterparty*, e *contratos inteligentes* através do front-end *axiomity*, encadeando cronologicamente transações (votos), orçamentos, candidatos, legislações e eleições através do *hash* baseado em prova de trabalho e em prova de registro, a fim de possibilitar a verificação extensiva em um modelo de rede distribuída.

Neste sistema, a *blockchain* do *bitcoin*, que é a mais antiga *blockchain* em operação, é utilizada como sistema de armazenamento distribuído, altamente tolerante a falhas.

O *counterparty* é um sistema/plataforma financeira utilizado pelo *BitCongress* para criar os *tokens* de votação *(vote)*. O *counterparty* utiliza sua própria moeda para a aquisição de *tokens*, o XCP[11], que é criada consumindo *bitcoins*. Portanto, uma transação de votação é uma transação BTC[12] para todos os efeitos, inclusive no que se relaciona à política de incentivos para os nós mineradores.

As eleições e as legislações são contratos inteligentes multi-assinatura mantidos entre eleitores e candidatos, persistidos na *blockchain* do *ethereum*, que é uma plataforma que se diferencia do *bitcoin* justamente por permitir a utilização de contratos inteligentes. Contratos inteligentes são programas armazenados distribuidamente, responsáveis por garantir o cumprimento de regras lógicas. Didaticamente, a *blockchain* está para o armazenamento confiável, assim como o contrato inteligente está para o código confiável.

O *axiomity* é o *front-end* da aplicação, que realiza a integração entre todas as partes do *BitCongress*. Ele permite ao usuário encaminhar votos, criar eleições, criar legislações, discutir emendas às legislações e visualizar o histórico de votação.

No caso das eleições, os contratos inteligentes definem a duração da eleição, os candidatos, a legislação, o orçamento, a URL de acesso pelo público e uma chave pública utilizada para aceitar, registrar e processar os votos dos eleitores.

Os contratos inteligentes são ativados por um *token* especial chamado *congress*. Ele também é responsável por ativar o contrato do candidato vencedor, registrar o candidato vencedor e, ao final da eleição, fazer retornar o *token vote* para o eleitor.

[11] XCP é o símbolo da criptomoeda *counterparty*.
[12] BTC é o símbolo da criptomoeda *bitcoin.*

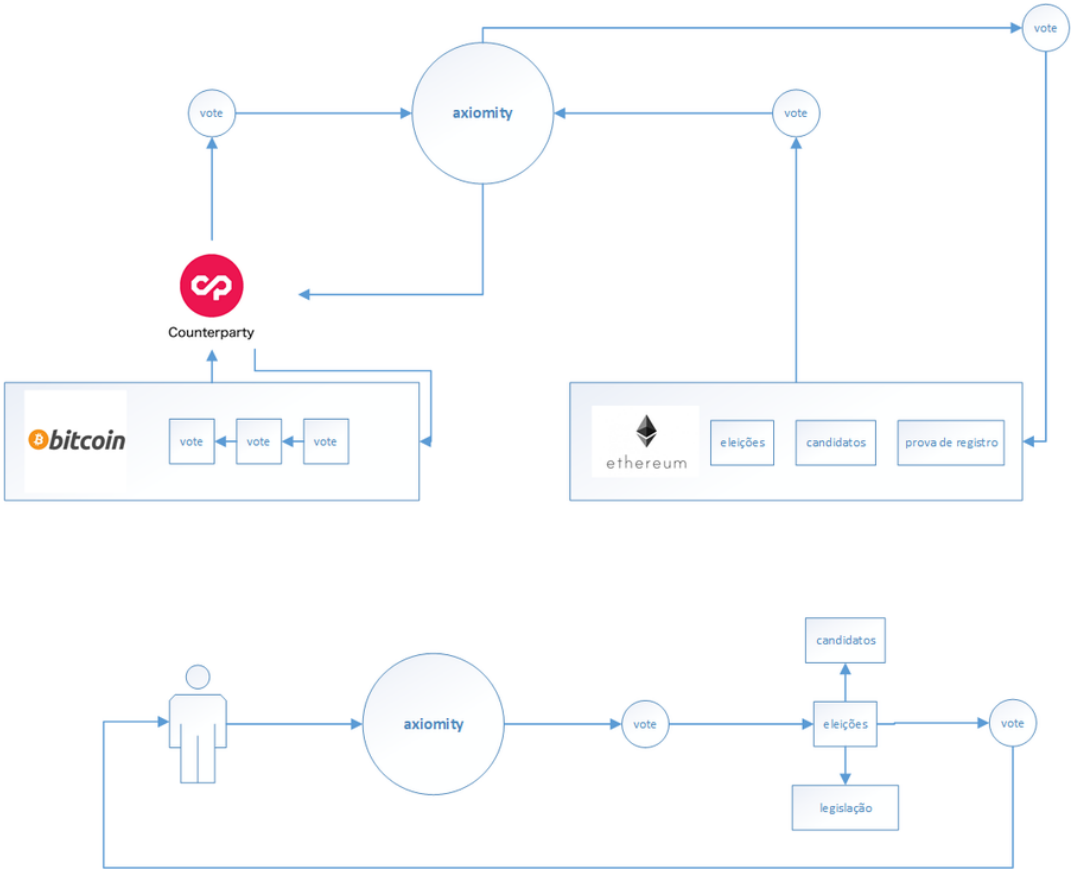
Cada eleitor interage com o *BitCongress* através de um endereço de *bitcoin*. Para receber o *token vote*, o eleitor precisa submeter um endereço de *bitcoin* e se registrar no *BitCongress*, fase que requer uma entidade centralizada capaz de verificar a identidade do eleitor.

No *BitCongress*, um eleitor, aqui representado por um endereço de bitcoin, possui um único *token* para ser utilizado durante todo ciclo de vida de utilização do sistema.

Para evitar fraudes, é gerada uma prova de registro dos endereços de *bitcoin* dos eleitores, dos candidatos e das legislações. O conceito é simples: um *hash* do endereço de *bitcoin* previamente verificado, é persistido na *blockchain* do contrato inteligente da eleição para possibilitar ao *axiomity* a verificação e a habilitação do eleitor.

A prova de registro também gera um histórico de votação completo para cada endereço de *bitcoin*. O sigilo do voto e, neste caso, de toda a história de votação do eleitor, depende da desassociação completa entre a identidade real do eleitor e o endereço de *bitcoin* utilizado. (ROCKWELL, ; VARSHNEYA; POUDEL; VYAS, 2015)

Figura 8 – Funcionamento do *BitCongress*



<http://bitcoin-class.org/presentations/Blockchain_Voting.pdf> p. 12 e 18 - [adaptada]

Alguns pontos importantes sobre o *BitCongress*:

- Assim como no *Follow My Vote*, também não foi possível descentralizar com segurança a etapa de credenciamento de eleitores e candidatos.
- Dado que cabe ao *axiomity* integrar sistemas bastante heterogêneos, além de executar diversas funções críticas para o *BitCongress*, ele se constitui como ponto central de falha de toda aplicação.
- Em relação à escala, é relevante registrar que um eleitor pode participar apenas de uma eleição por vez. Além disso, tendo em vista que cada usuário pode submeter eleições ou legislações e não há entidade central de gerenciamento, condições de corrida são esperadas. Adicionalmente, a capacidade de processamento de transações pela *blockchain* do *bitcoin* (7 por segundo) pode não ser suficiente para suportar sua utilização em cenários de grandes eleições.

5 Conclusão

Verificou-se que a descentralização completa no contexto dos sistemas eletrônicos de votação é inviável, especialmente no que tange ao credenciamento de eleitores, candidatos e na manutenção do cadastro eleitoral. Caso não haja centralização destes processos, ataques *Sybil* tem razoável probabilidade de serem executados com sucesso.

Em relação à natureza pública das eleições, a verificabilidade dos resultados nos sistemas de votação baseados em *blockchain*, semelhantemente ao que ocorre nos sistemas atualmente utilizados, também possui elevado custo de compreensão, o que gera dificuldades para que pessoas sem formação específica realizem a verificação dos resultados.

No que se refere ao sigilo da votação e à coerção do votante, é relevante destacar que ambos os sistemas são utilizados de maneira *online* através de qualquer dispositivo. Desta forma, permitem a coerção do votante e a quebra do sigilo da votação.

Sobre o mecanismo de consenso, embora a política de incentivos inserida no *bitcoin* para manter a segurança da rede tenha funcionado exitosamente, ao transportá-la para contexto da votação eletrônica, percebe-se que, em um ambiente de nós não confiáveis, ela pode não ser suficiente para incentivar o comportamento honesto dos nós da rede, ou seja, o incentivo em interferir em determinada eleição pode sobrepujar o incentivo financeiro da mineração na rede. Dessarte, descentralizar completamente o voto é difícil e não recomendável.

A escalabilidade das *blockchains* é outro fator de importante consideração, em especial daquelas que alcançam consenso através de prova de trabalho. A *blockchain* idealizada por Nakamoto claramente privilegiou a segurança em detrimento da escalabilidade, uma vez que somente a cada 10 minutos um novo bloco de transações é efetivamente adicionado à *blockchain*, tempo necessário para que nós globalmente distribuídos, assíncronos e sujeitos a altas latências consigam processar e consolidar, com segurança, as transações realizadas na rede.

Embora não sejam suficientemente robustos para substituírem os atuais sistemas de votação eletrônica, o desenvolvimento contínuo dos sistemas de votação sobre *blockchain* pode resultar em um paradigma mais adequado ao atendimento de demandas sociais por mais transparência e segurança dos sistemas de votação eletrônica.

A utilização de um modelo híbrido, inicialmente aplicável a casos de uso menores, que aproveite as qualidades de uma *blockchain* privada em um ambiente de nós confiáveis, aliada ao gerenciamento de uma entidade centralizada confiável pode realizar a convergência das melhores práticas e dar origem a uma nova geração de

sistemas eletrônicos de votação mais seguros, transparentes e auditáveis.

5.1 Dificuldades encontradas

Há pouca documentação sobre as soluções analisadas devido ao estágio incipiente de desenvolvimento em que se encontram.

Também não foi possível ter acesso às versões de teste do *Follow My Vote* e do *BitCongress* para uma observação mais detalhada do funcionamento de ambos os sistemas.

5.2 Trabalhos futuros

Realizar a pesquisa e o desenvolvimento de aplicações sobre *blockchains* privadas utilizando o *HyperLedger Fabric*, da IBM.

Referências

ANSI. *X9.62-1998 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.202.2977&rep=rep1&type=pdf>. Acesso em: 09/11/2017.

ANTONOPOULOS, A. M. *Mastering Bitcoin*. 1. ed. [S.l.]: O'Reilly, 2014. ISBN 978-1-449-37404-4.

BITCOIN.ORG COMMUNITY. *Bitcoin Developer Guide*. 2009. Disponível em: <https://bitcoin.org/en/developer-guide#block-chain-overview>. Acesso em: 02/11/2017.

BLUMENTHAL, M. *Encryption: Strengths and Weaknesses of Public-key Cryptography*. 2007. Disponível em: <http://www.csc.villanova.edu/~mdamian/Past/csc3990fa08/csrs2007/01-pp1-7-MattBlumenthal.pdf>. Acesso em: 07/01/2017.

BONEH, D. *Twenty Years of Attacks on the RSA Cryptosystem*. 1999. Disponível em: <https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf>. Acesso em: 07/01/2017.

BOS, J. W. *et al. On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography*. 2009. Disponível em: <https://eprint.iacr.org/2009/389.pdf>. Acesso em: 07/01/2017.

BRAKEVILLE, S.; PEREPA, B. *Informações básicas sobre blockchain: Introdução a livros razão empresariais*. 2016. Disponível em: <https://www.ibm.com/developerworks/br/cloud/library/cl-blockchain-basics-intro-bluemix-trs/index.html>. Acesso em: 31/07/2017.

CANAVAN, J. E. *Fundamentals Of Network Security*. 1. ed. [S.l.]: Artech House, 2001. ISBN 1-58053-176-8.

CHAUM, D. *Blind Signatures For Untraceable Payments*. 1982. Disponível em: <http://sceweb.sce.uhcl.edu/yang/teaching/csci5234WebSecurityFall2011/Chaum-blind-signatures.PDF>. Acesso em: 01/09/2018.

DIFFIE, W.; HELLMAN, M. E. New Directions in Cryptography. In: IEEE, 1976. *IEEE Transactions On Information Theory, Vol. IT-22*. 1976. Disponível em: <https://www-ee.stanford.edu/~hellman/publications/24.pdf>. Acesso em: 06/11/2017.

ERNEST, A. K. *The Key To Unlocking The Black Box*: Why the world needs a transparent voting dac. 2014. Disponível em: <https://followmyvote.com/wp-content/uploads/2014/08/The-Key-To-Unlocking-The-Black-Box-Follow-My-Vote.pdf>. Acesso em: 31/10/2017.

FELDMAN, A. J. *et al. Security Analysis of the Diebold AccuVote-TS Voting Machine*. 2006. Disponível em: <https://s3.amazonaws.com/citpsite/publications/ts06full.pdf>. Acesso em: 04/01/2018.

FRANCO, P. *Understanding Bitcoin*. 1. ed. [S.l.]: Wiley, 2015. ISBN 978-1-11901-915-2.

GOLDSMITH, B. Guidelines for Trialling E-Voting In National Elections. In: ____. *Real-World Electronic Voting*: Design, analysis and deployment. [S.l.]: CRC Press, 2017. cap. 2. ISBN 978-1-4987-1469-3.

GOMZIN, S. *Bitcoin for Nonmathematicians*: Exploring the foundations of crypto payments. 1. ed. [S.l.]: Universal-Publishers Boca Raton, 2016. ISBN 978-1-62734-071-7.

IBM. *IBM Blockchain based on Hyperledger Fabric from the Linux Foundation*. 2017. Disponível em: <https://www.ibm.com/blockchain/hyperledger.html>. Acesso em: 04/11/2017.

KOSCIELNY, C. *et al. Modern Cryptography Primer*: Theoretical foundations and practical applications. 1. ed. [S.l.]: Springer-Verlag, 2013. ISBN 978-3-642-41386-5.

LARA, P. C. da S.; OLIVEIRA, F. B. de. Curvas Elípticas: Aplicação em Criptografia Assimétrica. 2007. Disponível em: <http://www.lncc.br/~borges/doc/CurvasElipticas-AplicacaoemCriptografiaAssimetrica.pdf>. Acesso em: 08/11/2017.

MPOG. Guia de Interoperabilidade: Cartilha Técnica. 2015. Disponível em: <https://www.governoeletronico.gov.br/Plone/documentos-e-arquivos/Guia_de_Interoperabilidade_Cartilha_Tecnica_2015.pdf>. Acesso em: 07/11/2017.

NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 13/11/2017.

NARAYANAN, A. *et al. Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction*. 1. ed. [S.l.]: Princeton University Press, 2016. ISBN 978-0-691-17169-2.

NEMATI, H. R. *Security and Privacy Assurance in Advancing Technologies*: New developments. 1. ed. [S.l.]: IGI Global, 2011. ISBN 978-1-60960-202-4.

NIST. *FIPS PUB 186-2 Digital Signature Standard (DSS)*. 2000. Disponível em: <https://csrc.nist.gov/csrc/media/publications/fips/186/2/archive/2000-01-27/documents/fips186-2.pdf>. Acesso em: 09/11/2017.

NIST. *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*. 2015. Disponível em: <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf>. Acesso em: 08/11/2017.

NIST. *Recommendation for Key Management Part 1: General, Revision 4*. 2016. Disponível em: <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf>. Acesso em: 08/11/2017.

OSCE. *Discussion Paper in Preparation of Guidelines for the Observation of Electronic Voting*. 2008. Disponível em: <http://www.osce.org/odihr/elections/34725?download=true>. Acesso em: 23/12/2017.

RANDHAWA, N.; SINGH, L. *A Systematic Way to Provide Security for Digital Signature Using Elliptic Curve Cryptography*. 2011. Disponível em: <http://www.ijcst.com/vol23/1/navneet.pdf>. Acesso em: 09/11/2017.

RIVEST, R. L. *et al. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. 1977. Disponível em: <https://people.csail.mit.edu/rivest/Rsapaper.pdf>. Acesso em: 07/11/2017.

RIVEST, R. L.; VIRZA, M. Software Independence Revisited. In: ____. *Real-World Electronic Voting*: Design, analysis and deployment. 1. ed. [S.l.]: CRC Press, 2017. cap. 1. ISBN 978-1-4987-1469-3.

ROCKWELL, M. *BitCongress - Process For Blockchain Voting & Law*. Disponível em: <http://bitcoininc.co/BitCongress_Whitepaper.pdf>. Acesso em: 31/10/2017.

ROQUE, L. *Como ocorreu a crise financeira americana*. 2013. Disponível em: <http://www.mises.org.br/Article.aspx?id=1696>. Acesso em: 31/10/2017.

SANGALLI, L. A. *Criptossistemas baseados em curvas elípticas e seus desafios*. 2012. Disponível em: <http://www.dca.fee.unicamp.br/portugues/pesquisa/seminarios/2012/artigos/217.pdf>. Acesso em: 09/11/2017.

SCHUH, F. *Graphene Documentation*. 2018. Disponível em: <https://media.readthedocs.org/pdf/docsbitsharesorg/master/docsbitsharesorg.pdf#page=35&zoom=100,0,625>. Acesso em: 01/09/2018.

SECG. *SEC 2: Recommended Elliptic Curve Domain Parameters*. 2010. Disponível em: <http://www.secg.org/sec2-v2.pdf>. Acesso em: 09/11/2017.

SMART, N. P. *Cryptography*: An introduction. 3. ed. [S.l.]: McGraw-Hill, 2003. ISBN 978-0-07709-987-9.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. [S.l.]: Pearson, 2007. ISBN 978-85-88639-28-7.

STALLINGS, W. *Criptografia e Segurança de Redes*: Princípios e práticas. 6. ed. [S.l.]: Pearson, 2015. ISBN 978-85-430-0589-8.

STIGLITZ, J. *Lessons from the Global Financial Crisis of 2008*. 2009. Disponível em: <https://www8.gsb.columbia.edu/faculty/jstiglitz/sites/jstiglitz/files/2010_Lessons_Global_Financial_Crisis_Seoul.pdf>. Acesso em: 31/10/2017.

STINSON, D. R. *Cryptography Theory and Pratice*. 3. ed. [S.l.]: CRC Press, 2006. ISBN 978-1-58488-508-5.

VARSHNEYA, A. J. *et al. Blockchain Voting*. 2015. Disponível em: <https://bitcoin-class.org/projects/voting.pdf>. Acesso em: 31/10/2017.

VEGAS, C.; BARRAT, J. Overview of Current State of E-Voting Worldwide. In: ____. *Real-World Electronic Voting*: Design, analysis and deployment. [S.l.]: CRC Press, 2017. cap. 3. ISBN 978-1-4987-1469-3.

WINTER, J. *Blockchains: Embedding Integrity*. 2016. Disponível em: <https://infospectives.co.uk/2016/01/05/blockchains-embedding-integrity/>. Acesso em: 31/10/2017.

Apêndices

Código 1 – Código em Python exemplificando a geração de chave pública ECDSA a partir de uma chave privada na curva elíptica secp256k1.

```
"""
Example of how calculations on the secp256k1 curve work.
secp256k1 is the name of the elliptic curve used by bitcoin
see http://bitcoin.stackexchange.com/questions/25382
fonte: https://gist.github.com/nlitsme/c9031c7b9bf6bb009e5a
"""
p = 2**256 - 2**32 - 977

def inverse(x, p):
    """
    Calculate the modular inverse of x ( mod p )
    the modular inverse is a number such that:
    (inverse(x, p) * x) % p  ==  1
    you could think of this as: 1/x
    """
    inv1 = 1
    inv2 = 0
    while p != 1 and p!=0:
        inv1, inv2 = inv2, inv1 - inv2 * (x / p)
        x, p = p, x % p

    return inv2

def dblpt(pt, p):
    """
    Calculate pt+pt = 2*pt
    """
    if pt is None:
        return None
    (x,y)= pt
    if y==0:
        return None

    # Calculate 3*x^2/(2*y)  modulus p
    slope= 3*pow(x,2,p)*inverse(2*y,p)

    xsum= pow(slope,2,p)-2*x
    ysum= slope*(x-xsum)-y
    return (xsum%p, ysum%p)

def addpt(p1,p2, p):
    """
    Calculate p1+p2
    """
    if p1 is None or p2 is None:
        return None
    (x1,y1)= p1
    (x2,y2)= p2
    if x1==x2:
        return dblpt(p1, p)

    # calculate (y1-y2)/(x1-x2)  modulus p
```

```
    slope=(y1-y2)*inverse(x1-x2, p)
    xsum= pow(slope,2,p)-(x1+x2)
    ysum= slope*(x1-xsum)-y1
    return (xsum%p, ysum%p)

def ptmul(pt,a, p):
    """
    Scalar multiplication: calculate pt*a
    basically adding pt to itself a times
    """
    scale= pt
    acc=None
    while a:
        if a&1:
            if acc is None:
                acc= scale
            else:
                acc= addpt(acc,scale, p)
        scale= dblpt(scale, p)
        a >>= 1
    return acc

def isoncurve(pt,p):
    """
    returns True when pt is on the secp256k1 curve
    """
    (x,y)= pt
    return (y**2 - x**3 - 7)%p == 0

# (Gx,Gy)  is the secp256k1 generator point
Gx=0
   x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798

Gy=0
   x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8

g= (Gx,Gy)
g2=dblpt(g, p)
print "    2*G= (%x,%x)" % g2
print "  G+2*G= (%x,%x)" % addpt(g, g2, p)
print "2*G+2*G= (%x,%x)" % addpt(g2, g2, p)

privkey= 0
   x18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725

print " -> pubkey= (%x,%x)" % ptmul(g, privkey, p)
```

Código 2 – Implementação simplificada de um algoritmo Proof-Of-Work

```
#!/usr/bin/env python

"""
(ANTONOPOULOS, 2014, p. 314-317)
"""
```

```
import hashlib
import time

try:
    long
    xrange

except NameError:
    long = int
    xrange = range

max_nonce = 2 ** 32

def proof_of_work(header, difficulty_bits):

    target = 2 ** (256 - difficulty_bits)

    for nonce in xrange(max_nonce):
        hash_result = hashlib.sha256(str(header)+str(nonce)).
           hexdigest()

        if long(hash_result, 16) < target:
            print("Success with nonce %d" % nonce)
            print("Hash is %s" % hash_result)
            return (hash_result, nonce)

    print("Failed after %d (max_nonce) tries" % nonce)
    return nonce

if __name__ == '__main__':
    nonce = 0
    hash_result = ''

    for difficulty_bits in xrange(32):
    difficulty = 2 ** difficult_bits
    print("Difficulty: %ld (%d bits)" % (difficulty,
       difficulty_bits))
    print("Starting search...")

    start_time = time.time()

    new_block = 'test block with transactions' + hash_result

    (hash_result, nonce) = proof_of_work(new_block,
       difficulty_bits)

    end_time = time.time()

    elapsed_time = end_time - start_time
    print("Elapsed Time: %.4f seconds" % elapsed_time)

    if elapsed_time > 0:

        hash_power = float(long(nonce)/elapsed_time)
```

```
        print("Hashing Power: %ld hashes per second" % hash_
           power)
```