UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Vinicius Macelai

VERIFICAÇÃO DE ELEIÇÃO UTILIZANDO BLOCKCHAIN

Florianópolis

2019

Vinicius Macelai

VERIFICAÇÃO DE ELEIÇÃO UTILIZANDO BLOCKCHAIN

Trabalho de Conclusão de Curso submetido ao Programa de Graduação em Ciência da Computação para a obtenção do Grau de Bacharel em Ciência da Computação.
Orientador: Prof. Dr. Jean Everson Martina

Florianópolis

2019

Vinicius Macelai

VERIFICAÇÃO DE ELEIÇÃO UTILIZANDO BLOCKCHAIN

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de "Bacharel em Ciência da Computação", e aprovado em sua forma final pelo Programa de Graduação em Ciência da Computação.

Florianópolis, 01 de julho 2019.

Prof. EL FLETO, Dr.
Coordenador

Banca Examinadora:

Prof. Dr. Jean Everson Martina
Orientador

Fernando Pereira Me.
Universidade Federal de Santa Catarina

Dedicatoria top

AGRADECIMENTOS

Agradeço a todos

Citação top

RESUMO

As abordagens utilizadas nos sistemas de eleição da maioria dos países continuam a ser realizadas de forma manual, com cédulas na forma de papel. Tal modelo traz problemas enormes de logística e um alto custo para funcionamento devido aos requisitos de uma eleição segura, que deve fornecer privacidade, transparência, verificabilidade e confiabilidade. Já as abordagens eletrônicas via internet, ainda carregam desconfiança sobre manter estas propriedades. Uma possível solução para melhorar uma abordagem eletrônica seria utilizar uma blockchain para melhorar sua auditabilidade, que é o ponto mais questionado nesse esquema. A blockchain possui propriedades intrínsecas, como a imutabilidade dos dados, e com esquemas utilizando contratos inteligentes na blockchain é possível realizar a verificação dos votos de forma descentralizada e aberta ao público. Assim, cria-se um sistema que mantém as propiedades citadas anteriormente, com um baixo custo e menor necessidade de confiar em uma entidade central.

Palavras-chave: criptografia, eleições, democracia, blockchain, contratos inteligentes

ABSTRACT

As approaches used in the selection systems of most remaining countries an executed service manual with paper banknotes. Such a model brings huge logistical problems and a high cost of operation due to requirements of a safe selection, which should provide privacy, transparency, verifiability and requirements. Already the electronic approaches via internet, still carry distrust about maintaining these properties. A possible solution to improve an electronic approach would be to use a blockchain to improve your auditability, which is the most questioned point in this scheme. A blockchain has intrinsic properties such as data immutability and schemas smart contracts on blockchain it is possible to check votes decentralized and open to the public. Thus, a system is created that maintains as properties mentioned above, with a low cost and the least need to rely on a central entity.

Keywords: crypto, elections, democracy, blockchain, smart contracts

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

SUMÁRIO

1 INTRODUÇÃO

Ainda que vivamos no momento onde tudo é digital e façamos as mais diversas tarefas de maneira eletrônica e online, quando o assunto é votação no meio eletrônico, existem as mais diversas e controversas opiniões a respeito.

No Brasil, eleições eletrônicas têm sido utlizadas por mais de 20 anos e testes recentes mostram que, mesmo com o desenvolvimento durante todo esse período, o atual sistema não consegue se mostrar realmente seguro(ARANHA, 2018). O maior problema com a solução proposta pelo Governo Brasileiro é a falta de auditabilidade, em que só é possível se voluntariar para testar o sistema em um ambiente controlado. Ainda durante o processo eleitoral, é necessário confiar cegamente no sistema, não há instrumento nenhum que permita verificar se o voto foi realmente computado.

Os sistemas de votação eletrônicos atuais se baseiam em esquemas que utilizam de criptografia homomórfica, que permite que dados cifrados possam ser processados sem serem decifrados, assim garantindo propriedades importantes para o sistema.(SMART; VERCAUTEREN, 2010). Entretanto, esses esquemas são utilizados de forma centralizada, rodando apenas em um servidor central, sem a possibilidade de tais informações serem acessadas pelo o público em geral de maneira transparente.

Uma maneira de se realizar a autenticação no sistema seria utilizando certificado digital, que são arquivos digitais, que permite que uma pessoa seja identificada virtualmente, com garantia de autenticidade(ADAMS, 2002). Sendo no Brasil, o modelo adotado para gerenciar o sistema, a Infraestrutura de Chaves Públicas Brasileira (ICPBrasil). Já na área da educação, há a Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPEdu), que pode ser utilizada em votações no âmbito acadêmico.

Para realizar a auditoria das votações, é possível utilizar da tecnologia blockchain, que são bases de registro de dados distrubuídos e compartilhados, desta forma criando um consenso e confiança sobre o estado atual do sistema.(NAKAMOTO, 2009). Garantir essas propriedades intrínsecas, como a imutabilidade dos dados, é um bom sistema para manter o registro dos votos, além da possiblidade de rodar contratos inteligentes que podem processar os votos de maneira descentralizadas junto com a criptografia homomórfica para garantir o anonimato.

Neste trabalho, optou-se por enfatizar o estudo e a utilização da blockchain e protocolo Ethereum, a qual fornece contratos inteligentes de alto nível(BUTERIN, 2014). Apresenta-se um esquema que utiliza esses contratos para garantir as propriedades já citadas, além do estudo de seu impacto financeiro.

Ainda, visa implementar uma solução que integre todas estas partes, um sistema de eleição eletrônico que permite ter informações sobre o votante de forma confiável. Além disso, almeja posibilitar a realização da verificação da eleição em uma blockchain de forma descentralizada e pública.

1.1 MOTIVAÇÃO

Os modelos de eleição utilizados nos dias de hoje são em sua maioria em cédulas de papel. Dependendo da dimensão da votação, isso externaliza grandes problemas, no que concerne à logística, que tem como consequência um aumento de custos. Já as abordagens que utilizam do meio eletrônico e online, geram grande desconfiança para a maioria das partes interessadas, com receio que o resultado seja hackeado e alterado. Consequentemente há uma demanda por um modelo de votação que seja mais auditável e aberto para sanar esse problema de desconfiança.

1.2 JUSTIFICATIVA

Este tema foi escolhido devido sua grande importância, uma uma vez impacta basicamente todos os setores da sociedade, pois existem votações nas mais diversas esferas, desde a governança de empresas, até consultas de opinião sobre assuntos delicados. Além dos pontos anteriores citados, a utilização da recente tecnologia blockchain para a solução destes problemas é uma grande inovação na área. Este trabalho visa contribuir com modelos mais eficientes e transparentes, que beneficiarão a sociedade como um todo.

1.3 PERGUNTA DE PESQUISA

Este trabalho visa responder se é possível construir um modelo de eleição eletrônica utlizando a blockchain para garantir uma maior auditabilidade do sistema. Além de responder até qual volume de dados seria possível processar em contratos inteligentes na blockchain, juntamente com o cálculo ecônomico.

1.4 HIPÓTESES

- É possível criar um modelo de eleição eletrônica utilizando blockchain para forncer maior auditabilidade.
- É viável utilizar a blockchain Ethereum até qual volume dados.
- É economicamente viável esse modelo.

1.5 OBJETIVO GERAL

Estudar e criar a implementação de um sistema online de eleição, com foco principal na parte de realizar auditoria e verificação dos votos em blockchain com auxílio de contratos inteligentes, utilizando um sistema já desenvolvido. Além disso, analisar as implicações que esse sistema teria no funcionamento e custos de uma eleição.

1.6 OBJETIVO ESPECÍFICO

i. Analisar o estado da arte: estudar as principais soluções já propostas na literatura com o objetivo de identificar problemas e oportunidades para melhorar o trabalho.

ii. Implementar possibilidade de verificação na blockchain: Criação de um módulo para tornar o sistema mais auditável e verificável para o público em geral.

iii. Comparar e analisar as consequências do esquema.

1.7 METODOLOGIA

O trabalho será desenvolvido utilizando a infraestrutura e recursos do Laborarório de Segurança em Computação (LabSEC/UFSC), em que será estudada a bibliografia referente aos assuntos abordados nesta pesquisa, visando encontrar uma abordagem para um sistema de eleição eletrônica utilizando certificação digital juntamente com blockchain e contratos inteligentes. Frisando suas vantagens e desvantanges e seus custos.

1.8 RESULTADOS ESPERADOS

Espera-se contribuir para o estado da arte em eleição eletrônica utilizando blockchain de forma que aumente a transparência e a auditabilidade do processo. É esperado também que tal abordagem tenha um gargalo no volume de dados processados, visto que a tecnologia blockchain por ser descentralizada, não conseguirá processar diversas transações por segundo. Além de que, como há muito processamento de dados, devido a criptografia homomórfica, tende a ser inviável economicamente para casos onde não há necessidade de tanta segurança.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ELEIÇÕES DIGITAIS

2.2 CRIPTOGRAFIA HOMOMÓRFICA

A ideia de usar homomorfismo na criptografia surgiu em um trabalho científico publicado com o objetivo de propor um criptossistema em um cenário de aumento da utilização de terminais remotos. Neste caso, não seria ideal ter acesso a todo o banco de dados cifrados e então decifra-los para trabalhar os dados. Neste contexto, surge o conceito de se operar com dados cifrados e obter-se o mesmo resultado caso se estivesse operando em texto plano.(RIVEST; ADLEMAN; DERTOUZOS, 1978)

Criptografia homomórfica inclui diversos tipos de esquemas de criptografia que podem ser executados em diferentes classes de dados cifrados. Os tipos mais comuns de homomorfismo em criptografia são parcialmente homomórfico, relativamente homomórfico e completamente homomórfico.(ACAR et al., 2018)

O nome criptografia homomórfica é derivado do conceito de homomorfismo em algébra abstrata. Um homomorfismo é uma aplicação que preserva a estrutura entre duas estruturas algébricas X e Y.

$$f : X \longrightarrow Y \tag{2.1}$$

Seja α uma função de ciframento e β uma função de desencriptação correspondente. Sejam $x1$, $x2$ dados em texto plano. A tupla (α, β) é uma cifra homomórfica com o operador $\star$ se a proprieade for satisfeita:

$$\beta(\alpha(x1)) \star (\alpha(x2)) = x1 \star x2 \tag{2.2}$$

As operações principais de um esquema homomórfico de criptografia são: *KeyGen*, *Encryption*, *Decryption*, *Eval*. *KeyGen* é a operação para criar uma chave pública e outra privada em uma versão de criptografia assimétrica e a criação de uma chave única em modelo simétrico. *KeyGen*, *Encryption*, *Decryption* não são diferentes de suas funções em seus modelos tradicionais. Entretanto, *Eval* é uma operação específica de sistemas homomórficos, que tem como entrada e saída textos cifrados, adicionalmente, o dado cifrado resultante não deve aumentar de tamanho, caso contrário, haveria um limite de operações possíveis.(ACAR et al., 2018)

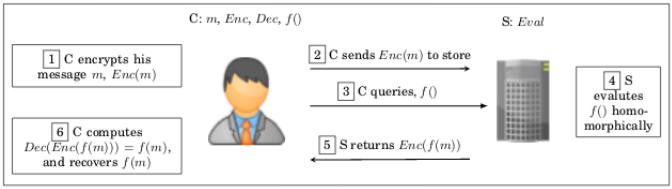

Figura 1 – Imagem tirada do paper ACAR et al., 2018.

2.3 BLOCKCHAIN

Da sua tradução literal, blockchain significa cadeia de blocos. De maneira simplista, pode ser definido como um bloco ligado com o anterior, gerando assim uma cadeia. Estes blocos carregam as informações que são importantes para a rede. No caso de uma criptomoeda como *Bitcoin*, cada bloco teria dados sobre as transações realizadas naquele dado instante, ou seja, uma definição do estado atual do sistema.

A concepção da blockchain foi feita para ser um encadeamento de registros imutáveis, distribuídos e públicos. Os registros são imutáveis devido ao tipo de encadeamento que é feito com os blocos, em que o ponteiro para o bloco anterior é projetado para garantir a imutabilidade dos dados. Como é um protocolo distribuído, todas as informações não estão armezenadas em um sevidor central e não há um nodo mestre que coordene a rede. Justamente o oposto disso, a blockchain está replicada em todos os nodos participantes da rede, que podem estar espalhados pelo mundo inteiro. Além de ser um esquema distribuído, o referido também é público, pois não há como censurar uma parte de participar da rede, basta o interessado ter acesso a internet que ele poderá realizar a sua cópia da base de dados.(UNDERWOOD, 2016)

2.3.1 Cadeia de blocos

A estrutura de um bloco é basicamente a seguinte:

- Constante de valor 0xD9B4BEF9.
- Tamanho do bloco em *bytes*.
- Cabeçalho do bloco, que consiste em 6 itens.
- Quantidade de transações no bloco.

- Transações em si.

Vale notar que as transações detêm uma estrutura de dados que permite a criação de *scripts*, além de permitir a inserção de dados arbitrários.

Já a estrutra do cabeçalho do bloco é composta por:

- Versão do bloco.
- *Hash* do bloco anterior.
- *Hash* do bloco atual, baseado em todas as transações do bloco.
- *Timestamp* em que o bloco foi criado.
- *Nonce*, número aleatório que é utilizado na mineração dos blocos.
- *Bits*, objetivo atual da mineração em formato compactado.

Esta estrutura de dados permite que qualquer participante da rede possa validar o bloco de maneira rápida. Existe o desafio matemático para a criação de blocos, isto é, para criar um novo bloco, ele deve calcular um *Hash* com uma pseudo colisão de acordo com a variável *Bits* do cabeçalho. Resolver esse desafio matemático é chamado de mineração, e a cada bloco que passa se torna mais difícil, entretanto, uma vez que sua solução é conhecida, é extremamente fácil de validar sua corretude. Há diversas soluções possíveis para determinado bloco, porém é necessário que apenas uma seja encontrada.(ANTONOPOULOS, 2014)

Como a mineração de blocos se torna cada vez mais difícil, a probabilidade de algum atacante conseguir reescrever um bloco anterior é mínima, visto que ele teria que concluir o desafio matemático para o bloco que deseja modificar e ainda todos os sucessores dele.

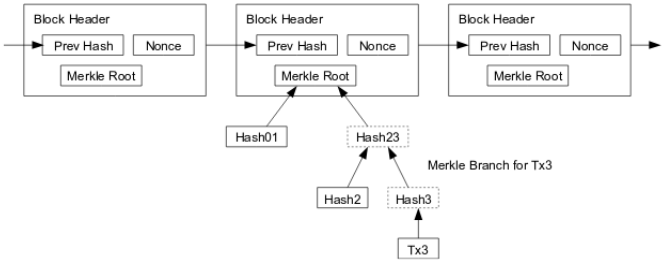

Figura 2 – Imagem tirada do whitepaper do Bitcoin.

2.4 CONTRATOS INTELIGENTES

A ideia de contrato sempre esteve presente durante todo o curso da humanidade, com a evoluação da socidade foi se criando novos conceitos para se formalizar uma relação. O contrato é o pilar das relações econômicas modernas. Entretanto esse mecanismo tem um custo elevado, durante o processo de criação e utilização de contratos, normalmente baseados em lei comum, que muitas vezes torna-se inviável para determinados casos.

Ideliazado no ano de 1997 por Nick Szabo, os contratos inteligentes foram pensados como uma substituição natural dos contratos realizados no cotidiano. Em analogia, a ideia é utilizar de software e hardware para a tomada de decisões, como em uma máquina automática de vendas de refrigerante. Os contratos inteligentes vão além da máquina de vendas, uma proposta de utilização de contratos em todos os tipos de propriedade e controlado do meio digital. Eles possibilitam a verificação desse contrato de maneira dinâmica e proativa.(SZABO, 1997)

Já no surgimento da blockchain descrito no protocolo do Bitcoin, havia espaço para criação de *scripts* nas transações. Entretanto, estes *scripts* foram concebidos para serem simples, com a ideia de não sobrecarregar a rede com a execução de códigos complexos. Um exemplo de *script* trivial seria o congelamento de fundos até certa data futura. Como não são permitido *loops* em sua estrutura, isto impacta na não Turing completude da linguagem de *script*.(NAKAMOTO, 2009)

Almejando-se ter *scripts* mais potentes, foi criado o conceito de contratos inteligentes, que são Turing-completos e ainda podem guardar o estado atual do sistema. Assim, possibilita a criação de aplicações extremamente mais complexas.(BUTERIN, 2014)

A *blockchain Ethereum* foi criada juntamente com sua *Virtual Machine*, a EVM como é chamada, que utiliza de um vasto conjunto de instruções para executar tanto tarefas específicas quanto gerais. Para armazenar os conjuntos de instruções de forma eficiente, a EVM os codifica em *bytecodes*.(WOOD, 2014)

Consequentemente com a criação do conjunto de instruções, houve a criação da uma linguagem de programação de alto nível que é compilada para esses *bytecodes* especifícicos. *Solidity* foi criada para ser uma linguagem simples de ser entendida e reproduzida para quem já têm os conhecimentos de programação de outras linguagens padrões de alto nível. (ZAKRZEWSKI, 2018)

Desta forma, é possível criar aplicações que rodam de forma descentralizada e que podem ter seu estado atual verificado por qualquer participante do sistema em qualquer momento além de garantir as propriedades da block-

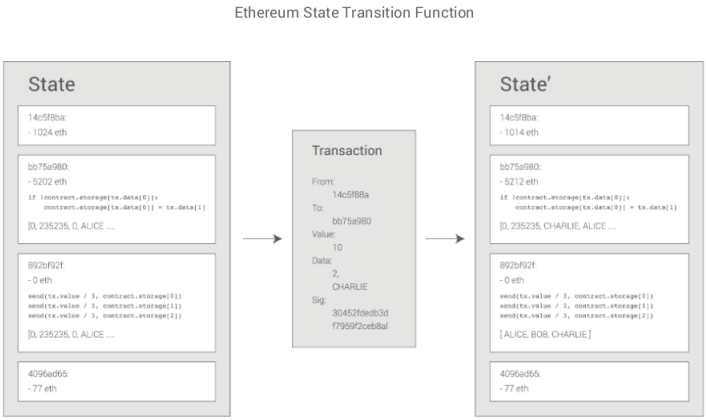

Figura 3 – Imagem tirada do whitepaper Ethereum

chain.

3 DESENVOLVIMENTO

Este capíutlo será abordado a criação do esquema e desenvolvimento do projeto proposto.

3.1 MODIFICAÇÃO DO SOFTWARE HELIOS

Para ter acesso as informações relevantes para o esquema foram criados alguns *endpoits* que disponibilizam essas informações, sendo essas:

- GET ‘elections/<election_uuid>/’ retorna todas as informações públicas da eleição
- GET ‘elections/<election_uuid>/voters’ retorna todas as informações públicas dos eleitores
- GET ‘elections/<election_uuid>/result’ retorna todas as informações sobre o resultado

3.2 CONTRATOS INTELIGENTES

3.2.1 Descrição do contrato

```
1
2  pragma solidity ^0.5.11;
3  pragma experimental ABIEncoderV2;
4
5  contract Result {
6
7      struct Question {
8          string question;
9          int result;
10     }
11
12     Question[] public questions;
13     string public voters_hash;
14     uint256 public relesed_at;
15
16     function addQuestion(string memory _question, int
           _result) public {
17         questions.push(Question(_question, _result));
18     }
```

```
19
20      function setVotersHash(string memory _voters_hash)
            public {
21          voters_hash = _voters_hash;
22      }
23
24      function setReleaseAt(uint256 _relesed_at) public {
25          relesed_at = _relesed_at;
26      }
27  }
28
29  contract Votes {
30
31      struct Vote {
32          string vote_hash;
33          uint256 cast_at;
34      }
35
36      Vote[] public votes;
37
38      function addVote(string memory _vote_hash, uint256 _cast
            ) public {
39          votes.push(Vote(_vote_hash, _cast));
40      }
41
42  }
43
44
45  contract Voters{
46
47      struct Voter {
48          string uuid;
49          string name;
50          string email;
51      }
52
53      Voter[] public voters;
54
55      function addVoter(string memory _uuid, string memory
            _name, string memory _email) public {
56          voters.push(Voter(_uuid, _name, _email));
57      }
58  }
59
60  contract Election {
61
62      string public uuid;
63      string public name;
64      string public cast_url;
65      uint256 public created_at;
66      uint256 public voting_starts;
67      uint256 public voting_ends;
```

```
68
69      struct Public_Key {
70          string p;
71          string q;
72          string g;
73          string y;
74      }
75
76      Public_Key public public_key;
77
78      struct Question {
79          string question;
80          string[] anwsers;
81      }
82
83      Question[] public questions;
84
85      Voters public voters_contract;
86      Votes public votes_contract;
87      Result public result_contract;
88
89      constructor(string memory _uuid, string memory _name,
            string memory _cast_url, uint256 _created_at,
            uint256 _voting_starts, uint256 _voting_ends,
90                  string memory _p, string memory _q, string
                        memory _g, string memory _y) public {
91          uuid = _uuid;
92          name = _name;
93          cast_url = _cast_url;
94          created_at = _created_at;
95          voting_starts = _voting_starts;
96          voting_ends = _voting_ends;
97          public_key = Public_Key(_p, _q, _g, _y);
98          voters_contract = new Voters();
99          votes_contract = new Votes();
100         result_contract = new Result();
101     }
102
103     function addQuestion(string memory _question, string[]
            memory _anwsers) public {
104         questions.push(Question(_question, _anwsers));
105     }
106
107 }
```

REFERÊNCIAS

ACAR, A. et al. A survey on homomorphic encryption schemes: Theory and implementation. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 51, n. 4, p. 79:1–79:35, jul. 2018. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/3214303>.

ADAMS, S. L. C. **Understanding PKI: Concepts, Standards, and Deployment Considerations**. [S.l.: s.n.], 2002.

ANTONOPOULOS, A. M. **Mastering Bitcoin: Unlocking Digital Crypto-Currencies**. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2014. ISBN 1449374042, 9781449374044.

ARANHA, Y. **Execução de código arbitrário na urna eletrônica brasileira**. 2018. Disponível em: <https://www.researchgate.net/publication-/326261911_Execucao_de_codigo_arbitrario_na_urna_eletronica_brasileira>.

BUTERIN, V. **Ethereum: A next-generation smart contract and decentralized application platform**. 2014. Accessed: 2016-08-22. Disponível em: <https://github.com/ethereum/wiki/wiki/White-Paper>.

NAKAMOTO, S. **Bitcoin: A peer-to-peer electronic cash system**. 2009. Disponível em: <http://www.bitcoin.org/bitcoin.pdf>.

RIVEST, R. L.; ADLEMAN, L.; DERTOUZOS, M. L. On data banks and privacy homomorphisms. **Foundations of Secure Computation, Academia Press**, p. 169–179, 1978.

SMART, N. P.; VERCAUTEREN, F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In: NGUYEN, P. Q.; POINTCHEVAL, D. (Ed.). **Public Key Cryptography – PKC 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 420–443. ISBN 978-3-642-13013-7.

SZABO, N. Formalizing and securing relationships on public networks. **First Monday**, v. 2, n. 9, 1997.

UNDERWOOD, S. Blockchain beyond bitcoin. **Communications of the ACM**, v. 59, p. 15–17, 10 2016.

WOOD, G. **Ethereum: a secure decentralised generalised transaction ledger**. 2014. `http://gavwood.com/paper.pdf`.

ZAKRZEWSKI, J. Towards verification of ethereum smart contracts: A formalization of core of solidity: 10th international conference, vstte 2018, oxford, uk, july 18–19, 2018, revised selected papers. In: ____. [S.l.: s.n.], 2018. p. 229–247. ISBN 978-3-030-03591-4.