UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Nivaldo de Oliveira Junior

Sistema de eleição seguro o suficiente

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Nivaldo de Oliveira Junior

Sistema de eleição seguro o suficiente

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Marcelo Keese Albertini

Universidade Federal de Uberlândia

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2017

Nivaldo de Oliveira Junior

Sistema de eleição seguro o suficiente

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 23 de junho de 2017:

Marcelo Keese Albertini
Orientador

Paulo Henrique Ribeiro Gabriel

Rodrigo Sanches Miani

Uberlândia, Brasil
2017

Resumo

A possibilidade de uma população eleger representantes legítimos através de um processo eleitoral transparente e que garanta o livre arbítrio dos eleitores está na essência de uma sociedade democrática. Nesse contexto, o desenvolvimento de um sistema de votação eletrônico pode minimizar práticas ilícitas, bem como tornar o processo mais confiável. O objetivo deste trabalho, foi desenvolver um sistema eleitoral que possa atender aos objetivos de segurança propostos por Karlof, Sastry e Wagner: lançar como pretendido, contabilizar como lançado, verificabilidade, um voto por eleitor, resistência à coerção e privacidade. Foram construídos quatro métodos fundamentais, em linguagem Java, que utilizam algoritmos que têm como função a geração de um identificador numérico para candidatos e eleitores, o registro do voto do eleitor, a verificação do voto e a totalização da eleição. Durante o desenvolvimento do sistema percebeu-se que há objetivos concorrente entre si, ou seja, a implementação de um compromete o estabelecimento de outro, a saber: verificabilidade e resistência à coerção.

Palavras-chave: Sistema de eleição, Eleição confiável, Voto verificável, Resistência a coerção.

Lista de ilustrações

Lista de abreviaturas e siglas

DRE	*Direct Recording Eletronic*
RSA	Algoritmo de criptografia assimétrica inventado por Ron Rivet, Adi Shamir e Leonard Adleman
SEV	Sistema Eletrônico de Votação

Sumário

1 Introdução

As democracias são construídas sobre o consentimento de sua população e um sistema de votação de confiança é crucial para esse consentimento.

Nesse contexto, as eleições populares são elementos centrais, mas não exclusivos, do processo de composição das esferas dos poderes públicos. Por conseguinte, todo o planejamento e execução do processo eleitoral é objeto de revisão e aprimoramento constantes, incrementando características essenciais, como a garantia ao sigilo e o combate à compra de votos.

Dictson e Ray (2002) ponderam que o processo eleitoral que utiliza cédulas de papel tem sido um dos mais desarticulados, ineficientes e onerosos de todos os projetos governamentais em países democráticos.

Os sistemas computacionais são cada vez mais responsáveis pela otimização e automação de tarefas manuais. A substituição da força de trabalho humana pela atividade eletrônica parece ser o futuro óbvio de determinadas atribuições e o processo eleitoral não está imune a esse movimento. Prova disso, foi a adoção para as eleições dos estados brasileiros em 1996, das urnas eletrônicas de votação, as quais trouxeram ganhos consideráveis às eleições, como a maior celeridade no escrutínio.

A urna eletrônica brasileira, mostrada na Figura 1, é um dispositivo eletrônico especificamente concebido para permitir ao eleitor escolher seus representantes políticos através da digitação de um código numérico único em um teclado. Tal código é específico para cada um dos candidatos e, após a entrada desse dado, a urna exibe em um monitor as informações e imagem do candidato. Além disso, a urna conta com um pequeno terminal utilizado pelos mesários para a identificação do eleitor e liberação da urna para o recebimento do voto do eleitor que é mostrado no dispositivo a direita na Figura 1. A arquitetura da urna eletrônica, tal qual como implementada hoje, certamente não permite ao eleitor verificar se sua escolha foi registrada corretamente. No tocante aos outros objetivos de segurança, a falta de acesso aos detalhes operacionais da urna não nos permite asseverar sobre o atendimento ou não dos mesmos.

A utilização de terminais eletrônicos ou portais *online* para a realização de eleições sempre implicará em questionamentos sobre a segurança do sistema de votação, seja de maneira a garantir o registro adequado dos votos imputados, seja para impedir qualquer adulteração nas informações registradas de forma a favorecer quaisquer candidatos.

Um sistema eletrônico de votação seguro, segundo Karlof, Sastry e Wagner (2005), deve atender a seis metas bem definidas, as quais são a referência utilizada para o desenvol-

Figura 1 – Urna eletrônica. Fonte: <http://tisaibamais.blogspot.com.br/2014/09/urna-eletronica-saiba-mais-19092014.html>

vimento de nosso sistema, a saber: Lançar-como-pretendido, contabilizar-como-lançado, verificabilidade, um voto por eleitor, resistência à coerção e privacidade. Desenvolveremos cada um deles no capítulo 2.

1.1 Objetivos

O objetivo deste trabalho é desenvolver um código-fonte capaz de atender as necessidades de segurança para uma eleição de qualquer tipo. Como necessidades de segurança, foram adotados os 6 princípios apresentados por Karlof, Sastry e Wagner (2005) como objetivos de segurança de uma eleição eletrônica.

Além do sistema operacional que controla o funcionamento das urnas eletrônicas, o processo eleitoral é composto por uma série de elementos, cada qual responsável por atribuições diferentes, como por exemplo a logística de movimentação das urnas eletrônicas, a seleção de cidadãos para trabalhar como mesários e etc. Tais elementos, contudo, extrapolam o âmbito da ciência da computação e não serão objeto deste trabalho.

1.2 Método

A metodologia de trabalho englobou o planejamento e desenvolvimento de um sistema web para gerenciar eleições por contagem simples de voto e pesos iguais. Tal planejamento foi realizado de modo a atender os objetivos de segurança em eleições propostos por Karlof, Sastry e Wagner (2005) em sua pesquisa, conforme apresentados no objetivo deste trabalho.

Para construção do sistema foi utilizado a linguagem de programação Java juntamente com vários *frameworks* que auxilia no desenvolvimento sendo o principal deles o Jhipster <https://jhipster.github.io/>, o Sistema de Gerenciamento de banco de dados escolhido para utilizar foi o MongoDB por acreditar que para este sistema iria trazer uma melhor performance.

Todo código construído neste trabalho se encontra disponível em <https://github.com/nivaldojunior/jasees>.

1.3 Resultados

Espera-se com este trabalho um sistema de votação capaz de assegurar sob a premissa de que o sistema será utilizado sobre controle físico de organização de um comitê eleitoral todos os 6 princípios de segurança de uma eleição apresentado na seção 1.1

1.4 Organização do Trabalho

Este trabalho está organizado em quatro capítulos, os quais estão dispostos da seguinte maneira:

O primeiro capítulo trata dos aspectos introdutórios do trabalho: objetivos, metodologia utilizada, resultados obtidos e esta própria seção sobre a organização de todo o conteúdo.

O segundo capítulo apresenta a revisão bibliográfica sobre os temas que influenciam processos eleitorais diversos e a operação eletrônica de votação: conceitos gerais sobre a eleição eletrônica, aspectos sobre segurança da informação e conceitos sobre números primos.

O terceiro capítulo apresenta o desenvolvimento do código-fonte com o detalhamento de cada um de seus métodos, considerações e explicações referentes a cada um deles. Nesse capítulo, também são apresentadas análises das garantias de segurança que o protótipo implementado é capaz de prover.

Por fim, o quarto capítulo engloba as conclusões acerca do código desenvolvido: suas potencialidades e limitações, bem como propostas de pesquisa futura.

2 Revisão Bibliográfica

Segundo Dictson e Ray (2002), uma alternativa para a eleição utilizando cédulas de papel seria a realização de eleições através da internet. A operação ocorreria através do acesso de um site ao qual os eleitores possam se conectar através de meios seguros, confirmar suas identidades e votar usando uma cédula eletrônica.

Contudo, a votação exclusiva pela *Internet* engloba uma série de reflexões e cuidados que surgem dada a utilização da rede mundial de computadores para a transmissão de dados, especialmente dados tão sensíveis como o voto de eleitores. Os debates envolvem aspectos gerais de segurança, questões sociais relacionadas à exclusão digital e capacidade da rede em suportar volumes massivos de dados. Mesmo com diversos pontos de atenção, eleições eletrônicas são realidade em algumas situações, seja com o uso direto da *internet*, seja com o uso de dispositivos eletrônicos de registro de voto, as *Direct Recording Eletronic* ou DRE, como é o caso do Brasil. Esses dispositivos são sistemas nos quais o voto é registrado através de equipamentos com interface homem-máquina, comandados pelo eleitor. Os votos inseridos no dispositivo ficam armazenados em uma memória local até o envio dos mesmos a uma autoridade central, via rede de comunicação, para a contagem e divulgação dos resultados. (COSTA, 2008).

A utilização das DREs tem recebido críticas devido a uma série de limitações não apenas do sistema desenvolvido para a operação das mesmas, mas pela dificuldade em se observar e aplicar uma série de objetivos que norteiam processos eletrônicos de segurança. Conforme apresentam Karlof, Sastry e Wagner (2005), seis objetivos de segurança devem nortear o desenvolvimento de um sistema com o qual se pretenda operacionalizar um processo eleitoral. Os autores escrevem no original:

> *Cast-as-intended: A voter's ballot on the bulletin board should accurately represent her choices;*
>
> *Counted-as-cast: The final tally should be an accurate count of the ballots on the bulletin board;*
>
> *Verifiability: The previous two properties should beverifiable. Verifiably cast-as-intended means each voter should be able to verify her ballot on the bulletin board accurately represents the vote she cast. Verifiably counted-as-cast means everyone should be able to verify that the final tally is an accurate count of the ballots contained on the bulletin board;*
>
> *One voter/one vote: Ballots on the bulletin board should exactly represent the votes cast by legitimate voters. Malicious parties should not be able to add,duplicate, or delete ballots;*

Coercion resistance: A voter should not be able to prove how she voted to a third party not present inthe voting booth;
Privacy: Ballots should be secret.

Essas metas de segurança definidas por Karlof, Sastry e Wagner (2005) podem ser explicadas da seguinte forma:

1) Lançar-como-pretendido: Uma cédula de um eleitor registrada na urna deve representar a escolha real do eleitor;

2) Contabilizar-como-lançado: A contagem final de votos deve ser uma apuração fidedigna das cédulas da urna;

3) Verificabilidade: Os dois objetivos anteriores dever ser verificáveis. A verificabilidade do primeiro objetivo significa que cada um dos eleitores deve poder verificar se sua cédula apresentada na urna representa de maneira fiel o voto por ele lançado. A verificabilidade do segundo objetivo representa a capacidade de todos terem a oportunidade de confirmar que a apuração final é realmente uma contagem confiável de todos, e apenas dos votos na urna;

4) Um voto por eleitor: as cédulas contidas na urna devem representar exatamente os votos realizados por eleitores legítimos. Indivíduos mal intencionados não podem ter a capacidade de adicionar, duplicar, alterar ou apagar cédulas;

5) Resistência à coerção: Um eleitor não deve ter a capacidade de provar a terceiros fora do local de votação como ele votou;

6) Privacidade: As cédulas devem ser secretas.

Em sua análise, Karlof, Sastry e Wagner (2005) apresentam que para se atacar os seis objetivos de segurança, uma ameaça pode originar-se de três fontes separadas, a saber: as DREs; os agentes de apuração de votos; e os indivíduos que utilizam ações coercivas. Em uma fraude, esses três elementos podem interagir para o desenvolvimento de delitos mais elaborados.

Uma DRE pode ser comprometida de diferentes formas por exemplo: um programador que insere um código malicioso na urna, um zelador que trabalha em turno noturno e que tenha acesso às urnas para instalação de qualquer código que comprometa o funcionamento das mesmas. (KARLOF; SASTRY; WAGNER, 2005).

Além disso, elementos maliciosos, sejam eles eletrônicos ou humanos, que participam do processo de contagem dos votos podem comprometer a eleição de maneira significativa. Tomando o painel eletrônico de votos: caso seja desenhado para ações ilícitas, pode, por exemplo, apagar todas as cédulas eletrônicas cadastradas. Ainda, um membro mal intencionado da equipe de apuração pode provocar alterações nas porções privadas

das chaves de criptografia de outros responsáveis, tornando impossível a decifração das cédulas. (KARLOF; SASTRY; WAGNER, 2005).

Em outro estudo, Costa (2008) amplia os requisitos de segurança necessários a um sistema de votação eletrônica, ponderando que, dessa forma, há concordância com as regras de eleição de vários países. São eles:

1. Registro e autenticidade do eleitor:

a. Todos os eleitores, para exercer o seu direito de voto, devem estar devidamente registrados e habilitados, junto à autoridade eleitoral, a votar para determinado pleito.

b. Devem existir mecanismos que afiram a identidade do eleitor no dia de votação, visando evitar fraudes de personificação (votante que se apresenta falsamente como outro).

2. Anonimato do eleitor: o voto deve ser anônimo e não deve haver mecanismos que possibilitem associar o voto ao votante durante o processo, seja na votação, apuração, apuração ou em qualquer processo de auditoria/recontagem de votos. Esse requisito visa inibir procedimentos de coerção do eleitor por terceiros e a comercialização de voto por iniciativa do mesmo, sendo esses procedimentos caracterizados como crime eleitoral;

3. Unicidade: em consonância com as leis de cada país, o eleitor somente poderá votar uma única vez em um mesmo pleito eleitoral, devendo o total de votos refletir proporcionalmente o total de votantes;

4. Confidencialidade: a partir do término da votação e armazenamento do voto, seja depositando-o numa urna tradicional ou numa urna eletrônica, a confidencialidade do mesmo deve ser mantida. Assim, não pode haver meios de se obter, durante o pleito, a quantidade de votos recebidos por um determinado candidato. Ou seja, não devem ser permitidas apurações parciais e a apuração só pode ocorrer após o término do pleito eleitoral;

5. Integridade: todo voto proferido por um votante não pode ser alvo de alteração ou eliminação durante o processo de votação ou apuração. A apuração deve garantir o princípio da unicidade da contagem;

6. Não-coerção: nenhum votante deve ser vítima de nenhum tipo de repressão que o obrigue a votar contra a sua vontade;

7. Não-comercialização do voto: nenhum votante, depois de votar, deve usufruir de condições ou prova material que ateste a terceiros a qualidade de seu voto;

8. Materialização do voto: o Sistema Eletrônico de Votação (SEV) deve prover meios de materialização da qualidade do voto, ou seja, a impressão do voto do eleitor e seu depósito em uma urna convencional;

9. Contraprova do voto: em um SEV, a apuração é feita computacionalmente e

não por meio do procedimento de escrutínio da autoridade eleitoral sob fiscalização da sociedade, tal como é feito no sistema convencional. A contraprova deve ser fornecida pelo SEV para permitir que o votante tenha um indício de que seu voto foi computado corretamente, mas sem propiciar a coerção, venda do voto, ou revelação da identidade do eleitor;

10. Auditabilidade: um sistema de votação deve proporcionar auditabilidade, fornecendo, em todas as fases do processo, informações necessárias à obtenção de quaisquer indícios que possam representar fraude, funcionamento inadequado de *software* e *hardware* ou erro de operação humana. No entanto, as informações geradas não podem, em hipótese alguma, guardar dados que comprometam os requisitos de anonimato do eleitor, de confidencialidade e integridade do voto;

11. Usabilidade: em um SEV, a usabilidade do sistema deve contribuir para que o eleitor profira seu voto de forma amigável, ágil e sem erros. Assim, as interfaces devem ser projetadas de forma a garantir que o eleitor vote no menor tempo e com autossuficiência. Além disso, possibilitar sua utilização por eleitores com baixo nível de escolaridade e com limitações físicas, visual ou auditiva. Para os eleitores com alguma deficiência, o normal é a existência de teclados com código em Braile e instruções através de áudio, entre outros mecanismos de acessibilidade;

12. Disponibilidade: um SEV deve agregar a robustez necessária para garantir a disponibilidade do serviço, com as garantias dos demais requisitos, em todo o transcorrer do pleito eleitoral.

2.1 Aspectos sobre segurança da informação

O termo informação refere-se a uma reunião de dados ou registros que, após serem processados e analisados por um sistema eletrônico ou por um agente humano, apresentam um significado mais profundo sobre determinado objeto de análise. (KLETTENBERG, 2016)

Atualmente, grande parte dos processos, sejam eles oriundos da esfera pública ou privada, são intermediados ou mesmo integralmente executados pelo trânsito de informações através da rede mundial de computadores, o que tem revolucionado a forma usuários e a tecnologia têm interagido, haja vista a velocidade de processamento dos sistemas informacionais. Não é de causar surpresa que as informações carregam hoje uma importância fundamental e, por isso mesmo, podem ser alvo de ações mal-intencionadas. (KLETTENBERG, 2016).

Nesse contexto, Sêmola (2014) apresenta os três princípios fundamentais que sustentam o funcionamento de um sistema de segurança da informação, conforme a figura

2.

Figura 2 – Princípios de segurança da informação.

1. O primeiro atributo ou princípio diz respeito à confidencialidade. Nesse sentido ferramentas de segurança atuam no sentido de proteger o sigilo, limitando o acesso à informação. Esse princípio garante que apenas as pessoas que devam ter conhecimento da informação possam acessá-la. A confidencialidade está relacionada à acessibilidade da informação aos agentes autorizados e inacessibilidade aos agentes não autorizados;

2. O segundo princípio, da integridade da informação, visa garantir a manutenção das características originais da informação, estabelecidas pelo autor, ou seja, inviabiliza as alterações do documento de origem. A integridade está relacionada com a possibilidade de que a informação somente pode ser alterada por agentes autorizados e os não autorizados estão impedidos de comprometê-las;

3. O terceiro princípio refere-se à disponibilidade, ou seja, a informação poderá ser acessada por qualquer pessoa e a qualquer tempo, assim, refere-se ao acesso da informação somente por agentes autorizados e a qualquer momento.

Os três princípios apresentados permitem o entendimento da segurança da informação como um fenômeno composto por questões técnicas. Também é possível compreender a segurança como um fenômeno social, de forma que nenhuma análise pode desconsiderar a ação humana como uma das possíveis fontes geradoras de fragilidades sobre a segurança da informação em trânsito pela rede. Dessa forma, conforme detalhado na Tabela 1, os elementos que compõem, segundo Klettenberg (2016), o ecossistema de segurança são: Organização, Atributo de segurança, Ativo, Controle, Ameaça e Vulnerabilidade.

Tabela 1 – Elementos que compõe um ecossistema de segurança. Fonte: (KLETTENBERG, 2016)

Conceito	Definição
Organização	Uma organização é uma entidade social composta por recursos materiais e humanos, a qual possui objetivos comuns, procedimentos sistemáticos para controle de seu desempenho e limites definidos que a separam do ambiente. Pode ser uma instituição pública ou privada.
Atributo de segurança	Um atributo de segurança é uma propriedade atribuída a um ativo, a qual diz respeito a requisitos de segurança. Pode ser um atributo de confidencialidade, de integridade e de disponibilidade.
Ativo	Um ativo é um bem de propriedade da organização, utilizado para alcançar seus objetivos sociais. Pode ser um equipamento, estoque e imóvel.
Controle	Um controle é um procedimento padrão sistemático implementado para atenuar vulnerabilidades, bem como para proteger ativos através de medidas preventivas e corretivas.
Ameaça	Uma ameaça é uma possibilidade de dano aos ativos da organização, que afeta os atributos específicos e explora vulnerabilidades da organização. Pode ser de origem humana ou natural e ter como fonte um evento acidental ou uma ação deliberada.
Vulnerabilidade	Vulnerabilidade é uma situação caracterizada pela falta de medidas e proteção adequadas. Uma vulnerabilidade possui um grau de severidade associado (por exemplo, crítico, moderado ou baixo). Pode ser uma vulnerabilidade administrativa, técnica ou física.

Afirma Klettenberg (2016) que o desafio de proteger os sistemas e informações contra acessos ou ações mal-intencionadas é grande quando se realiza uma análise de todos os componentes relacionados na tabela 1. Assim, quanto maior o controle e a quantidade de restrições para acessar uma informação, maior será o grau de desconforto do agente mal-intencionado. Nesse sentido, continua Klettenberg (2016), a segurança não é problematizada somente pela tecnologia, mas sim pelas pessoas, consideradas o elo mais fraco da segurança.

2.2 Conceitos sobre números primos e criptografia RSA

A criptografia, segundo Coutinho (2003), "estuda os métodos para codificar uma mensagem de modo que só seu destinatário legítimo consiga interpretá-la". Assim, qualquer procedimento que permita a codificação de uma mensagem, seja ela transmitida em papel, ondas de rádio ou mesmo pela rede mundial de computadores, é denominado criptografia. Para o âmbito da informática, conforme apresenta Coutinho (2003), o mais

conhecido método criptográfico é o RSA, desenvolvido em 1977 por três pesquisadores do *Massachussets Institute of Technology*.

A criptografia RSA, segundo Coutinho (2003), consiste em uma multiplicação aritmética simples entre dois números primos bastante grandes, p *e* q. O produto resultante dessa multiplicação será utilizado para a codificação da mensagem a ser transmitida e tal codificação só pode ser desfeita com o conhecimento de p *e* q.

Um elemento fundamental do método RSA é o número primo, pois a decifração depende da decomposição de números inteiros em primos. Assim, segundo Coutinho (2003), um número inteiro p é primo se $p \neq \pm 1$ e os únicos divisores de p são ± 1 e $\pm p$. Portanto 2, 3, 5 e 7 são primos, mas $45 = 5 \cdot 9$ não é primo. Um número inteiro, diferente de ± 1, que não é primo é chamado de composto.

Além disso, um teorema importante da teoria dos números é o chamado teorema da fatoração. Apresenta Coutinho (2003) que o teorema da fatoração toma um inteiro positivo qualquer, $n \geq 2$. Sempre é possível escrevê-lo na forma

$$n = p_1^{e_1} ... p_k^{e_k},$$

na qual $1 < p1 < p2 < p3 < ... < p_k$ são números primos e $e_1, ..., e_k$ são inteiros positivos. Os expoentes $e_1, ..., e_k$ na fatoração acima são chamados de multiplicidades.

O entendimento sobre números primos é especialmente importante para a área da segurança da informação, pois tais números estão na essência da criptografia RSA. Segundo Coutinho (2003), a operação do método utiliza um número n, tal que $n = p \cdot q$, com p e q sendo números primos distintos entre si.

Segundo Coutinho (2003), a RSA é composta por três fases: pré-codificação, codificação e decodificação. A primeira fase (pré-codificação) refere-se a uma preparação para a criptografia propriamente dita, que, de maneira simplificada, é um cálculo de uma potência módulo n relativamente a um expoente especialmente escolhido. Entretanto, para que isto seja viável, a mensagem deve ser um número inteiro.

Ao transformar a mensagem em um bloco numérico, o método utiliza um número $n = p \cdot q$ como parâmetros da criptografia, em que p e q são números primos distintos. O número n é a chave de codificação da mensagem pode ser tornada pública sem maiores consequências, por isso ele recebe o nome de chave pública. A regra de codificação de um bloco b qualquer é $C(b) =$ resto da divisão de b^e por n. Tomando como exemplo um bloco $b = 102$ e um n escolhido de 391:

$$102^3 \equiv 102^2 \cdot 102 \equiv 238 \cdot 102 \equiv 24276 \equiv 34 \quad (mod \quad 391)$$

Assim, o bloco 102 é criptografado gerando como código o número 34.

3 Desenvolvimento

Os objetivos de segurança propostos por Karlof, Sastry e Wagner (2005) são, conforme apresentados na seção 1.1, os referenciais para o desenvolvimento do sistema de votação. Por isso, a sua arquitetura foi pensada de modo a atender ao maior número daqueles objetivos, o que foi sintetizado nas três características abaixo:

1. O sistema a ser desenvolvido deve possuir um algoritmo que gere um código identificador único e de difícil reprodução para um voto registrado;
2. Deve fornecer um procedimento seguro da verificação de voto registrado, ou seja, o eleitor deve ser capaz de verificar se sua escolha foi corretamente computada pelo sistema;
3. O sistema deve ser capaz de minimizar o ato de coerção praticado por agentes mal-intencionados contra eleitores.

A fim de atender às três características acima, o código desenvolvido possui quatro módulos:

1. Um módulo gerador de números primos para candidatos e eleitores;
2. Um módulo registrador do voto do eleitor;
3. Um módulo que permite a verificação do voto realizado;
4. Um módulo que realiza a contagem final de votos.

A essência de operação do sistema está na utilização de números primos como identificadores de pessoas, sejam elas eleitores ou candidatos. Dessa forma, o sistema pode ser construído de modo a usar operações matemáticas simples para fornecer seus serviços: multiplicações e divisões de números primos são os fundamentos do registro de voto e de sua verificação.

O primeiro algoritmo (Algoritmo 3.1) tem como objetivo a geração de um número primo que ainda não tenha sido utilizado na eleição em questão. A característica específica de um número primo (ter apenas ele mesmo e a unidade como divisores, conforme exposto no item 2.3 da revisão bibliográfica) foi essencial na escolha dos mesmos para serem os números identificadores de eleitores e candidatos. Essa escolha foi fundamental para o cumprimento da primeira meta apresentada no início desta seção, bem como para atender ao objetivo de segurança "lançar-como-pretendido" (*Cast-as-intended*) proposto por

Karlof, Sastry e Wagner (2005). O número gerado será utilizado pelo segundo algoritmo (Algoritmo 3.2) para o registro definitivo do voto do eleitor.

A geração dos números primos foi implementada utilizando-se o método *probablePrime* da classe *BigInteger* do Java, o qual utiliza o algoritmo de Miller-Rabin para gerar um número primo aleatório. O teste de Miller-Rabin (HURD, 2003) é um teste probabilístico da primalidade de um dado número n. Se n não passar pelo teste, n com certeza é um número não-primo. Se n passar no teste, ele é primo, com uma probabilidade $P(n \in P) \geq 0,75$, sendo que P denomina o conjunto de todos números primos. A margem de erro pode ser diminuída, aplicando-se o teste várias vezes para o mesmo número n, no caso do método *probablePrime* essa probabilidade é de 2^{-100}, o que é pequeno o bastante para ser desconsiderada no desenvolvimento do código. O método *probablePrime* tem como um de seus parâmetros a necessidade de definição do tamanho do número primo a ser gerado. Esta determinação é feita através da *constant PRIME_NUMBER_BIT_LENGTH*, a qual foi definida como 100, ou seja, o número primo gerado possui 100 bits.

Uma vez declaradas as constantes e instanciada a classe *Random*, o laço *do ... while* repetirá o método *probablePrime* até que a condição de não existência do número primo gerado, apresentada pela instrução *primeList.contais(primeNumber)*, seja satisfeita.

Por fim, encontrado um número primo ainda não utilizado, o mesmo é armazenado na lista *primeList* e também retornado para a declaração que chama o método *generatePrimeNumber()*.

Algoritmo 3.1 – Gerar número primo

```
1  private BigInteger generatePrimeNumber() {
2      Random random = new Random();
3      //PRIME_NUMBER_BIT_LENGTH = 100
4      Integer bitLength = Constants.PRIME_NUMBER_BIT_LENGTH;
5      BigInteger primeNumber;
6      do {
7          primeNumber = BigInteger.probablePrime(bitLength, random);
8      } while (this.primeList.contains(primeNumber)); //primo utilizado
9      this.primeList.add(primeNumber);
10     return primeNumber;
11 }
```

O algoritmo 3.2 é responsável pela contabilização de um voto, que basicamente consiste na multiplicação do número resultante do candidato e o número primo gerado aleatoriamente para o eleitor. O número resultante dessa multiplicação, na maior parte das vezes, terá como tamanho a soma dos bits dos números primos utilizados na operação, ou seja, se o número do candidato tiver 100 bits e o número do eleitor também possuir

100 bits o número resultante na maior parte das vezes vai conter 200 bits. Contudo, essa multiplicação pode ter como resultado um número com 198 bits ou 199 bits de tamanho. Por conta disso, é contabilizado um erro para que a contagem de votos no final seja exata, esse erro também é utilizado para descontar na contagem final de votos se o voto for falso para isso é utilizado uma variável *booleana* chamada *bias*, que se for *true* o voto é contabilizado na contagem final, se o *bias* for falso, não.

Algoritmo 3.2 – Contabilizar os votos

```
1  /*
2  * Metodo para um usuario "user" efetuar seu voto no candidato "cand"
3  * O argumento "bias" serve para indicar a veracidade do voto
4  */
5  public BigInteger vote(String cand, String user, Boolean bias) {
6      BigInteger result = generatePrimeNumber();
7      //registro de votos atual
8      BigInteger before = this.resultList.get(cand);
9      BigInteger after = before.multiply(result);
10     Integer bitLength = Constants.PRIME_NUMBER_BIT_LENGTH;
11     resultList.put(cand, after
12     //erro atual
13     Integer error = this.voteError.get(cand);
14     if (after.bitLength() < beforeLength + bitLength) {
15         this.voteError.put(cand, error + ((beforeLength + bitLength) -
               after.bitLength()));
16     }
17     if (!bias) {
18         this.voteError.put(cand, (error - bitLength));
19         //usuario votou falso
20         this.votedList.put(user, 2);
21     }else{
22         //usuario votou
23         this.votedList.put(user, 1);
24     }
25     return result;
26 }
```

Conforme os objetivos de segurança de "verificabilidade" (*Verifiability*) e "lançar-como-pretendido" (*Cast-as-intended)* apresentado por Karlof, Sastry e Wagner (2005), o terceiro método (Algoritmo 3.3) foi desenvolvido para atender tais objetivos e, portanto, é a porção do código que fornece ao eleitor a funcionalidade de verificação do voto efetuado. O método recebe o número primo inserido pelo eleitor e realiza divisões desse por todos

os resultados previamente armazenados para cada candidato até que o resto dessa divisão seja zero. Nesta operação, fica evidente que a dinâmica só é possível devido à utilização de números primos para a identificação de eleitores e candidatos: ao se configurar um *loop* para executar uma divisão até o momento em que esta divisão retorne resto zero, o quociente encontrado certamente será um dos números gerados para eleitores ou candidatos. Quando o laço (linha 3), satisfizer a condição apresentada na (linha 4), ou seja, quando a divisão do número recebido pelo método *verifyVote()* resultar em resto zero, o método adiciona o quociente encontrado, que se trata da numeração do candidato escolhido, à lista *results*.

A lista *results* foi criada (Linha 2) para armazenar o número do candidato escolhido pelo eleitor no momento em que esse está verificando o registro de seu voto. Caso o tamanho desta lista seja diferente de 1 significa que algum erro aconteceu ou que aquele número primo não foi utilizado na eleição. Por outro lado, se o tamanho da lista for 1 significa que o resultado armazenado na lista é exatamente o código identificador do candidato no qual o eleitor votou.

Algoritmo 3.3 – Verificar um voto

```
1  public String verifyVote(BigInteger primeNumber) {
2      ArrayList<String> results = new ArrayList<>();
3      resultList.forEach((k, v) -> {
4          if (v.divideAndRemainder(primeNumber)[1].equals(BigInteger.ZERO)) {
5              results.add(k);
6          }
7      });
8      if (results.size() == 1) {
9          return results.get(0);
10     } else {
11         return null;
12     }
13 }
```

O algoritmo 3.4 realiza a contagem final de votos o que consiste em pegar a quantidade de bits do resultado de cada candidato e dividir pelo tamanho de bits padrão menos a quantidade de bits inicial e depois acrescentar o erro contabilizado para cada candidato.

Algoritmo 3.4 – Contagem final de votos

```
1  public Map<String, Integer> countVotes() {
2      Map<String, Integer> result = new HashMap<>();
3      resultList.forEach((k, v) -> {
4          Double bitLength = Constants.PRIME_NUMBER_BIT_LENGTH.doubleValue();
5          Double error = voteError.get(k) / bitLength;
```

```
6           Double partial = (v.bitLength() - bitLength) / bitLength;
7           Integer votes = (int) (partial + error);
8           result.put(k, votes);
9       });
10      return result;
11  }
```

3.1 Análise de metas de segurança

O sistema desenvolvido apresenta elementos concebidos especificamente para atender aos objetivos de segurança propostos por Karlof, Sastry e Wagner (2005). Vamos desenvolver uma avaliação acerca de cada um deles.

É importante ressaltar que, por se tratar de um sistema monolítico, existe uma fragilidade referente ao administrador do sistema, pois com acesso direto ao banco de dados é possível haver diversas fraudes na eleição. Por isso, para o sistema desenvolvido nesse trabalho, o administrador deve ser considerado um ponto de falha.

Posto isso, seja confiando no administrador do sistema ou desenvolvendo mecanismos para supervisioná-lo, inibindo fraudes oriundas de suas ações, é possível afirmar que os objetivos foram, com algumas ressalvas, atingidos conforme descrito a seguir.

3.1.1 Lançar como pretendido

A meta de segurança "lançar como pretendido" tem como objetivo garantir que o voto foi registrado para o candidato efetivamente escolhido pelo eleitor e que não haja nenhum tipo de adulteração da escolha, ou seja, que o usuário registre sua opção para um candidato, mas a contabilização seja para outro.

É possível afirmar que esse objetivo foi cumprido, pois a utilização de números primos únicos para identificar e relacionar o eleitor e seu respectivo candidato garantem o registro fidedigno do voto. Ao operar matematicamente os identificadores numéricos de eleitor e candidato, cria-se uma chave única para a operação, o que atende o objetivo de "lançar como pretendido".

3.1.2 Contabilizar como lançado

A meta de segurança "contabilizar como lançado" tem como objetivo garantir que o voto foi contabilizado conforme foi lançado e também é possível afirmar que foi cumprida, dada a mesma justificativa do item 3.1.1, com apenas uma ressalva: o mecanismo anti-coerção, por se tratar de um voto falso, deve contabilizar somente o voto verdadeiro, mas permitir a verificação de ambos votos para permitir ao eleitor se proteger de qualquer

coerção que por ventura venha a sofrer. Nesse caso, nem o candidato ou o administrador do sistema pode conseguir distinguir entre o voto falso ou o voto verdadeiro.

3.1.3 Verificabilidade

A meta de segurança "verificabilidade" tem como objetivo a verificabilidade das duas metas anteriores. Este objetivo de segurança foi alcançado com as ressalvas citadas anteriormente.

Uma característica sobre a verificabilidade, é que ela pode acontecer até mesmo *offline*, necessitando apenas de cada eleitor anotar o seu número primo e o número resultante de cada candidato ser disponibilizado. Assim, ao final da eleição, o próprio eleitor pode realizar a divisão e verificar se seu voto foi contabilizado, sem a necessidade de envio eletrônico de seu número primo.

3.1.4 Um voto por eleitor

A meta de segurança "um voto por eleitor" tem como objetivo garantir que cada eleitor vote apenas uma vez. Como o sistema criado utiliza sua própria base de dados de usuários, esse objetivo depende única e exclusivamente da administração de criação de usuários, ou seja, o sistema garante que um usuário só irá poder votar uma única vez simplesmente guardando qual eleitor já votou na eleição em questão.

3.1.5 Resistência a coerção

A meta de segurança "resistência a coerção" tem como objetivo garantir que exista algum mecanismo para restringir o voto coercitivo e a compra de votos.

Para garantir tal objetivo foi implementado neste sistema o voto falso, que se trata da emissão de um voto falso emitido após o voto verdadeiro, o qual pode ser conferido normalmente no sistema, mas não é contabilizado. Assim, o voto coercitivo é evitado, pois o agente intimidador não terá certeza se o voto apresentado será o voto verdadeiro ou o falso.

Porém foi observado que a votação utilizando apenas usuário e senha para a realização da operação inviabiliza o mecanismo anti coerção dado que ainda há a possibilidade de o candidato mal intencionado solicitar os dados do eleitor para que ele mesmo registre o voto.

3.1.6 Privacidade

A meta de segurança "privacidade" tem como objetivo garantir que o voto do eleitor é secreto. O sistema foi desenvolvido para impedir que mesmo o administrador

do sistema possa associar uma pessoa ao seu voto, já que não é registrado no sistema nenhuma informação referente a qual o número primo gerado é para cada eleitor.

4 Conclusão

O sistema construído cumpre parcialmente os objetivos de segurança propostos por Karlof, Sastry e Wagner (2005). Além disso, alguns elementos importantes não foram analisados nesta pesquisa, como por exemplo a arquitetura da rede. Sua avaliação é importante para a construção de um sistema de votação mais resistente à fraudes.

Em eleições eletrônicas atuais, observa-se uma concentração de atividades, ações e elementos de rede em poucos nós de modo a manter os processos sob o controle das autoridades responsáveis. Contudo, essa prática torna todo o sistema mais vulnerável, pois a desestruturação de um nó central coloca em risco todas as partes da rede conectadas a ele. Uma alternativa, conforme apresentada pelo movimento *Follow my vote* <www.followmyvote.com>, é a utilização de uma estrutura descentralizada de nós auto gerenciáveis utilizando algoritmos complexos para gerenciar a eleição de forma distribuída, eliminando assim também o problema de confiabilidade no administrador do sistema.

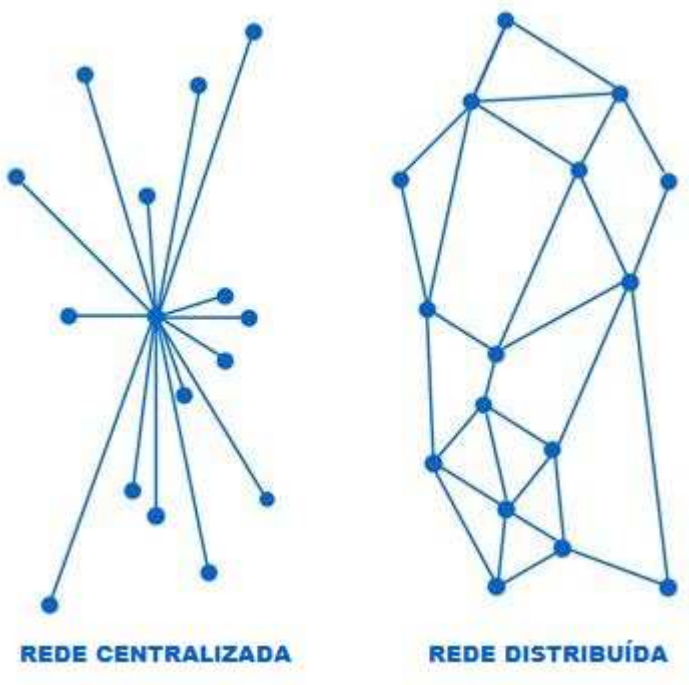

Figura 3 – Arquitetura centralizada e distribuída. Fonte: <https://followmyvote.com>

Para se pensar em um sistema funcional de forma distribuída deve-se pensar também em abandonar a arquitetura monolítica, ou seja, um único sistema que provenha todos os serviços necessários para a realização do pleito, e pensar em uma abordagem de micro serviços. Desta maneira pode-se escalar somente o módulo que precisa ser escalado de modo a minimizar os impactos de um possível ataque. Além disso, observa-se

o ganho de performance do sistema, pois os serviços serão solicitados apenas para sua atividade-fim, sem a necessidade de ação dos outros módulos.

Um problema de segurança que surge em um sistema de eleição online é a utilização de apenas um usuário e senha como forma de autenticação para realização do voto, podendo haver vários tipos de fraudes decorrentes deste fato, uma alternativa é utilizar a câmera para identificar o usuário e seu devido documento em mãos.

Outra observação importante está relacionada ao tamanho do número primo gerado, que é determinante na eleição, pois um número com quantidades pequenas de bits limita diretamente a quantidade de votos possíveis na eleição e um número muito grande também pois este número pode ficar tão grande que pode impossibilitar seu armazenamento em banco de dados convencionais deve se pensar em maneira mais eficientes de armazenar este número a fim de ocupar menos espaço.

Referências

COSTA, R. G. Sistema seguro de votação eletrônica multi-células. 2008. Citado 2 vezes nas páginas 10 e 12.

COUTINHO, S. C. *The mathematics of ciphers: number theory and RSA cryptography*. [S.l.]: Universities Press, 2003. Citado 2 vezes nas páginas 15 e 16.

DICTSON, D.; RAY, D. A moderna revolução democrática: uma pesquisa objetiva sobre as eleições via internet. *Internet e Política–Teoria e Prática da Democracia Eletrônica. EISENBERG, José*, 2002. Citado 2 vezes nas páginas 7 e 10.

HURD, J. Verification of the miller–rabin probabilistic primality test. *The Journal of Logic and Algebraic Programming*, Elsevier, v. 56, n. 1-2, p. 3–21, 2003. Citado na página 18.

KARLOF, C.; SASTRY, N.; WAGNER, D. Cryptographic voting protocols: A systems perspective. In: *USENIX Security*. [S.l.: s.n.], 2005. v. 5, p. 33–50. Citado 10 vezes nas páginas 7, 8, 10, 11, 12, 17, 18, 19, 21 e 24.

KLETTENBERG, J. *Segurança da informação*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2016. Citado 3 vezes nas páginas 13, 14 e 15.

SÊMOLA, M. *Gestão da segurança da informação*. [S.l.]: Elsevier Brasil, 2014. v. 2. Citado na página 13.