Implementação de um Sistema de Registro Prático Integrado ao Sistema de Votação CIVIS

Alberto Sobrinho[1], **Roberto Araújo**[1]

[1]Faculdade de Computação - Universidade Federal do Pará (UFPA)
Caixa Postal 479 – 66.075-110 – Belém – PA – Brazil

aclsobrinho@gmail.com, rsa@ufpa.br

Abstract. *Internet voting systems face a major problem in how to defend their voters from coercive attacks. One proposal to mitigate this problem is the use of anonymous voting credentials. Credentials are used to post valid votes and defend against malicious agents. These credentials should be generated so that their use by voters during voting is easy and practical. This work aims to create a Password Registration System, which implements a Practical Registration Protocol, and is integrated with the CIVIS Voting System.*

Resumo. *Sistemas de votação via internet enfrentam um grande problema em como defender seus votantes de ataques coercitivos. Uma proposta para mitigar esse problema é a utilização de credenciais anônimas de votação. As credenciais são utilizadas para publicar votos válidos e se defender de agentes maliciosos. Estas credenciais devem ser geradas de forma que sua utilização pelos votantes durante a votação seja fácil e prática. Este trabalho tem como objetivo criar um Sistema de Registro de Senhas, o qual implementa um Protocolo de Registro Prático, e é integrado ao Sistema de Votação CIVIS.*

1. Introdução

Apesar de suas diversas vantagens, sistemas de votação via internet enfrentam um grande desafio relacionado a coerção. A liberdade de votar de qualquer lugar e de qualquer dispositivo conectado a internet, deixa votantes vulneráveis a ataques coercitivos. Por ataques coercitivos, entende-se um agente malicioso obrigando o votante a escolher uma opção específica ou forçando ele a revelar alguma informação pertinente ao voto.

A ideia de resistência a coerção busca mitigar esse problema em protocolos de votação [Juels et al. 2010]. Protocolos resistentes a coerção utilizam credenciais de votação anônimas. Elas são compostas por uma sequência de bits e necessárias durante a votação. Votantes utilizam suas credenciais válidas para publicar votos válidos. Caso precisem se defender de um ataque coercitivo, eles podem gerar credenciais falsas e publicar votos que a princípio parecem válidos, porém são descartados posteriormente.

As credenciais de votação são peças chave na segurança do processo eleitoral. No entanto, a maioria das propostas de protocolos de votação resistentes a coerção não atentam para detalhes de uso das credenciais em cenários reais de votação. As credenciais são detalhadas no âmbito teórico, comprovando sua segurança, mas se tornam inviáveis para utilização por um votante em um contexto prático de votação.

A proposta de Protocolo de Registro Prático de [de Sá et al. 2020b] apresenta um esquema de geração de credenciais anônimas de votação. Segundo o protocolo, o votante

define uma senha e ela é utilizada para geração de credenciais válidas de votação de acordo com a estrutura matemática do protocolo de ABRTY [Araújo and Traoré 2013]. A senha é baseada nas ideias de senha de pânico [Clark and Hengartner 2008] e do dicionário de palavras do BIP-39 [Palatinus et al. 2013]. O segredo do votante deixa de ser uma longa sequência de bits e passa a ser a senha definida.

Este trabalho tem como contribuição a construção de um sistema próprio que implemente o Protocolo de Registro Prático de De Sá et al. Ele é chamado de Sistema de Registro de Senhas e permite o registro de senhas e geração de credenciais válidas de votação. O sistema é integrado ao sistema de votação CIVIS [Araujo et al. 2018] e opera em conjunto com ele. Enquanto o CIVIS permanece responsável pela configuração, votação e apuração da eleição, o Sistema de Registro assume a responsabilidade de registro dos votantes e geração de suas credenciais válidas de votação.

O trabalho é estruturado da seguinte forma: a seção 2 apresenta o protocolo de votação via internet resistente a coerção ABRTY [Araújo et al. 2010] e o sistema de votação CIVIS [Silva Neto and Araújo 2017], o qual implementa o protocolo de ABRTY; a seção 3 apresenta a proposta de De Sá et al de protocolo de registro prático de votantes [de Sá et al. 2020b]; a seção 4 detalha a implementação do Sistema de Registro Senhas; a seção 5 discute os trabalhos relacionados a essa pesquisa; e a seção 6 conclui o trabalho e apresenta os trabalhos futuros.

2. O Protocolo de ABRTY e o Sistema CIVIS

O protocolo de ABRTY é um esquema de votação resistente a coerção de complexidade linear proposto por Araújo e Traoré [Araújo et al. 2010]. A proposta depende de um criptossistema limiar com propriedades homomórficas [Araújo et al. 2010]. Para tal, os autores utilizam o criptossistema El Gamal proposto por Juels et al. [Juels et al. 2010].

As credenciais de votação do protocolo são anônimas e possuem uma estrutura matemática. Dado um grupo cíclico G de ordem p, onde o problema de decisão de Diffie-Hellman é difícil, a credencial é composta pelos valores (A, r, x). O valor A é tal que

$$A = (g_1 {g_3}^x)^{\frac{1}{y+r}} \tag{1}$$

onde r e x são números aleatórios em Z_p^* , g_1 e g_2 são geradores do grupo, e y é a chave secreta da autoridade de registro, responsável por gerar as credenciais. Os valores (A, r) podem ser publicados sem problemas de segurança [Araújo and Traoré 2013], porém o valor x deve ser mantido em segredo pelo votante.

Quatro fases compõe o esquema: (1) fase de configuração; (2) fase de registro; (3) fase de votação; e (4) fase de apuração. Elas são descritas a seguir:

Fase de Configuração. As autoridades da eleição definem o grupo cíclico G de ordem p, onde o problema de decisão Diffie-Hellman é difícil, e os geradores $g_1, g_2, g_3, g_4, o \in G$. Os registradores calculam cooperativamente a chave pública $pk_R = {g_3}^y$, onde $sk_R = y$ é a chave secreta. De forma semelhante, os apuradores calculam cooperativamente a chave pública $pk_A = {g_1}^{y_1} {g_3}^{y_3}$, onde $sk_A = (y_1, y_3)$ são as respectivas chaves secretas. Todas as informações, exceto as chaves secretas, são publicadas no quadro público da eleição.

Fase de Registro. As autoridades de registro geram as credenciais de votação para os

votantes habilitados à participar na eleição. Para cada credencial, os números aleatórios $r, x \in Z_p^*$ são gerados e A é calculado (Equação 1). A credencial é então enviada para o votante através de um canal seguro e inviolável. Ressalta-se que somente é necessário manter o segredo do valor x da credencial, podendo a tupla (A, r) ser publicado sem problemas de segurança para a eleição [Araújo and Traoré 2013].

Fase de Votação. Os votantes utilizam as credenciais de votação para publicar seus votos. Um voto válido é emitido pelo votante ao utilizar sua credencial válida (A, r, x). Para publicar um voto inválido, o votante pode utilizar qualquer credencial $(A, r, x' \neq x)$. A opção do voto é publicado no quadro público de forma cifrada, junto de outras cifras necessárias para verificação de validade e duplicidade. Também são publicadas provas de conhecimento-zero não-interativas relacionadas a essas cifras. As cifras são criadas a partir da chave pública das autoridades de apuração.

Fase de Apuração. As autoridades de apuração verificam a validade do conjunto de provas enviado junto aos votos publicados, eliminando votos que não passem na verificação. Votos duplicados são descartados, permanecendo somente o voto mais recente do votante. O conjunto resultante é misturado através de uma rede de misturadores. Em seguida, em cooperação com as autoridades de registro, as autoridades de apuração verificam a validade das credenciais utilizadas nos votos misturados. Por fim, os votos válidos são decifrados, contados e o resultado é publicado.

O CIVIS [Araujo et al. 2018] é um sistema de votação via internet resistente a coerção que implementa o protocolo de ABRTY . O sistema web é escrito em Python [Python 2021], com auxílio do framework Django [Django 2021]. No entanto, os mecanismos criptográficos utilizados pelo sistema são escritos em Javascript [Mozilla 2021], pois são necessariamente executados na máquina do usuário. A manipulação dos elementos na página web também ocorre através de Javascript, por meio da biblioteca jQuery [OpenJS 2021]. Ademais, o sistema utiliza do formato JSON para disponibilização das informações públicas e comunicação cliente-servidor.

Existem três autoridades atuantes no sistema. A autoridade da eleição é responsável por gerar os parâmetros da votação, bem como definir informações como nome da eleição, disputas, período de votação e etc. A autoridade de registro (registrador) é responsável por gerar, verificar e enviar credenciais válidas de votação. Por fim, a autoridade de apuração (apurador) é responsável pela apuração de votos válidos.

Uma eleição no CIVIS possui quatro fases, seguindo as quatro fases estabelecidades pelo Protocolo de ABRTY. Na fase de registro do CIVIS, o registrador gera as credenciais válidas de votação e as envia para os votantes. Cada votante possui uma única credencial válida de votação. Não há nenhuma interação entre votante e registrador.

Durante a fase de votação, os votantes inserem no sistema a credencial (A, r, x) recebida, escolhem sua opção na disputa e publicam seus votos. Para publicar um voto inválido, os votantes podem gerar credenciais inválidas $(A, r, x' \neq x)$ e utilizá-las. Ao longo da fase de apuração, votos gerados com credenciais inválidas são descartados.

O sistema permite uma versão limiar da fase de apuração, onde a responsabilidade da autoridade de apuração é divida entre um grupo de autoridades [Silva Neto and Araújo 2017]. Nessa versão, as autoridades de registro compartilham um par de chaves El Gamal. Cada autoridade possui seu próprio par de chaves que repre-

senta uma parte do par de chaves do grupo. Para gerar e recuperar a chave do grupo, é necessário a cooperação de um número mínimo de participantes.

3. Protocolo de Registro Prático

Cada valor da credencial do Protocolo de ABRTY é composto por uma longa sequência de bits, o que acaba tornando sua utilização em cenários práticos de votação muito difícil. Consequentemente, o sistema de votação CIVIS sofre do mesmo problema relacionado a utilização das credenciais de votação.

No trabalho onde se apresenta o protocolo de ABRTY, os autores citam a possibilidade de uma variante da fase de registro do esquema proposto [Araújo et al. 2010]. Nessa versão, o valor x da credencial, relativo ao segredo do votante, pode ser gerado de forma conjunta pelo votante e a autoridade de registro. A ideia é expandida por De Sá et al em sua proposta de Protocolo de Registro Prático de Votantes [de Sá et al. 2020b].

No protocolo proposto por De Sá et al, os votantes registram senhas, as quais são mapeadas para o valor x da credencial de ABRTY. Elas também são necessárias posteriormente para publicação de um voto válido durante a fase de votação. Somente uma autoridade de registro atua nesse processo, e portanto, espera-se que ela seja confiável.

As senhas empregadas na proposta são compostas por um grupo de palavras, onde cada palavra é associada a uma cor. As senhas da proposta tem como base o conceito de senhas de pânico [Clark and Hengartner 2008] e o dicionário utilizado no BIP-39 [Palatinus et al. 2013]. No entanto, o dicionário utilizado na proposta necessita manter no mínimo 128bits de segurança e, portanto, possui mais palavras que o do BIP-39.

O BIP-39 é um protocolo de geração de chaves determinísticas empregado no Bitcoin [Bitcoin 2021]. Ele define um dicionário de palavras que seguem regras específicas de diferenciação de palavras. No conceito de senhas de pânico, existe um conjunto de senhas válidas e inválidas. O delimitador desses conjuntos, por exemplo, pode ser um dicionário de palavras como o do BIP-39. Qualquer senha que seja composta por alguma palavra não presente no dicionário é considerada inválida.

Ademais, o conjunto de senhas válidas é dividido em dois subconjuntos: senhas admissíveis e senhas de pânico. A utilização de uma senha admissível indica uma ação normal (ex. publicação de um voto válido, o qual deve ser contabilizado). Por outro lado, o uso de uma senha de pânico indica uma ação anormal (ex. publicação de um voto inválido, o qual deve ser descartado posteriormente).

A proposta de Protocolo de Registro Prático utiliza das seguintes primitivas criptográficas [de Sá et al. 2020b]:

- **Provas de Conhecimento-Zero Não-Interativas (NIZKP)**. Esta primitiva permite a um verificador verificar a afirmação de um provador, sem que este tenha que revelar qualquer informação secreta a respeito da afirmação.
- **Provas de Verificação Designada (DVP)**. Esta primitiva é uma versão das provas de conhecimento-zero. Elas somente possuem relevância para uma pessoa em particular, não tendo utilidade para qualquer outra pessoa.
- **Funções de Derivação de Chave (PBKDF)**. Essas funções de hash criptográfico permitem a geração de uma chave secreta a partir de alguma entrada.

O protocolo possui somente duas fases: fase de configuração e fase de registro. Essas fases são descritas a seguir:

Fase de Configuração. As autoridades da eleição definem o grupo cíclico G de ordem p, onde o problema de decisão de Diffie-Hellman seja difícil. Em seguida, os geradores $g, g_1, g_3 \in G$ são definidos e o dicionário de palavras δ é publicado. Por fim, o registrador gera seu par de chaves El Gamal tal que

$$(sk_R = y, pk_R = g^y) \tag{2}$$

onde sk_R representa sua chave secreta e pk_R representa sua chave pública.

Fase de Registro. Os votantes interagem com o registrador afim de gerar suas credenciais válidas de votação. O protocolo de registro de credenciais é executado da seguinte forma:

1. O **votante** seleciona um conjunto de palavras aleatórias $\omega \in \delta$ e utiliza uma função de derivação de chaves para calcular a parte secreta x de sua credencial, tal que

$$x = PBKDF(\omega) \mid x \in Z_p. \tag{3}$$

2. O **votante** calcula o valor C_1 tal que

$$C_1 = g_1 g_3^x \tag{4}$$

   e uma prova de conhecimento-zero não-interativa $NIZKP1$ relacionada à C_1. A tupla $(C_1, NIZKP1)$ é enviada ao registrador.
3. O **registrador** verifica a validade da prova $NIZKP1$. Caso seja inválida, o processo é anulado. Caso contrário, o processo continua normalmente.
4. O **registrador** gera o número aleatório $r \in Z_p^*$ e calcula o expoente $\frac{1}{y+r}$, utilizando de sua chave privada (Equação 2).
5. O **registrador** calcula A tal que

$$A = {C_1}^{\frac{1}{y+r}} \tag{5}$$

   e a prova de conhecimento-zero não-interativa $NIZKP2$, relacionada ao cálculo de A. Ressalta-se que o resultado da Equação 5 é a Equação 1. A tupla $(A, r, NIZKP2)$ é enviada ao votante.
6. O **votante** verifica a validade da prova $NIZKP2$. Caso seja inválida, o processo é anulado. Caso contrário, o votante aceita a credencial de votação.

4. Sistema de Registro de Senhas

O Sistema de Registro de Senhas implementa a proposta de Protocolo de Registro Prático e tem como objetivo gerar credenciais válidas de votação. Seu código-fonte é separado do sistema CIVIS, porém ainda integrado a ele, operando os dois sistemas em conjunto. A construção do sistema foi feita utilizando das mesmas tecnologias do sistema CIVIS. Desse modo, toda inserção/definição de valores e informações no Sistema de Registro é feita por meio de arquivos no formato JSON.

4.1. Visão Geral

O sistema tem como principal objetivo permitir à votantes o registro de uma senha composta de palavras e cores. O acesso ao sistema é restrito a usuários com contas registradas. O administrador do sistema é responsável pelo registro dessas contas. A diferenciação entre conta de votante e registrador é feita pelo sistema em tempo de execução.

Durante o processo de registro de senhas, gera-se um token válido de votação. Este token é associado à senha definida pelo votante. A senha é mapeada para o valor x da credencial de ABRTY, enquanto que o token é composto pelos valores (A, r) da credencial. Posteriormente, a senha registrada e o token válido de votação são utilizados para publicar um voto válido na eleição.

4.2. Arquitetura do Banco de Dados

O sistema é composto por três módulos: Fase de Registro, Registro de Senhas e Usuário. O módulo Fase de Registro é responsável pela configuração, início e encerramento de uma fase de registro. Por sua vez, o módulo Registro de Senhas é responsável pela execução de uma fase de registro, ou seja, pelo registro de senhas em si. Por fim, o módulo Usuário é responsável pelos tipos de usuários presentes no sistema.

A Figura 1 apresenta as entidades do banco de dados presentes no sistema. A entidade *User* é uma classe padrão própria para usuários disponibilizada pelo framework Django. Ela é a única entidade presente no módulo Usuário. Destaca-se que tanto registradores quanto votantes utilizam da mesma entidade Usuário. A diferenciação entre os papeis dos usuários é feita em tempo de execução pelo sistema.

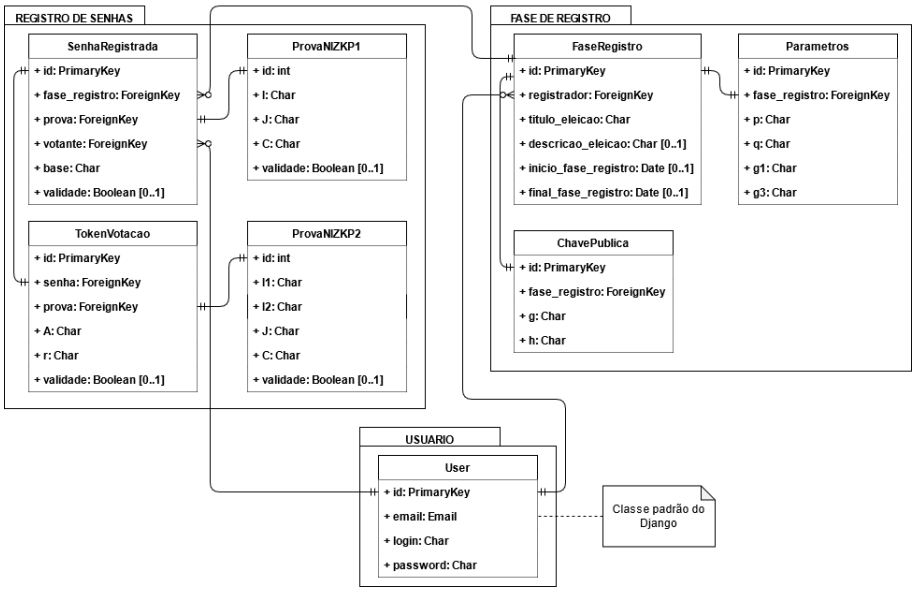

Figura 1. Diagrama de classes das entidades do banco de dados no Sistema de Registro de Senhas.

O módulo Fase de Registro possui três entidades: a entidade FaseRegistro, que armazena informações relacionadas à fase de registro; a entidade parâmetros, a qual armazena o conjunto de parâmetros necessários relacionados com a fase de configuração

do Protocolo Prático de Registro de Senhas; e a entidade ChavePublica, que armazena os valores da chave pública do registrador responsável.

Uma fase de registro (FaseRegistro) possui somente um registrador responsável (User), um conjunto de parâmetros (Parametros) e uma chave pública (ChavePublica). Um conjunto de parâmetros (Parametros) e uma chave pública (ChavePublica) são exclusivos de uma única fase de registro (FaseRegistro), porém um registrador (User) pode ser responsável por mais de uma fase de registro (FaseRegistro).

No módulo Registro de Senhas existem quatro entidades. A entidade SenhaRegistrada armazena informações a respeito da senha definida gerado pelo votante. A prova de conhecimento-zero não-interativa relacionada a esta senha é representada na entidade ProvaNIZKP1. A entidade TokenVotacao armazena as informações da tupla (A, r) gerada pelo registrador. A prova de conhecimento-zero não-interativa relacionada a este token é representada na entidade ProvaNIZKP2.

Uma senha registrada (SenhaRegistrada) pertence a somente um votante (User), mas um votante pode ter mais de uma senha registrada. A senha registrada (SenhaRegistrada) possui somente uma prova de conhecimento-zero não-interativa relacionada (ProvaNIZKP1) e gera somente um token de votação (TokenVotacao). Este, por sua vez, possui somente uma prova de conhecimento-zero não-interativa relacionada (ProvaNIZKP2). As provas (ProvaNIZKP1 e ProvaNIZKP2) são exclusivas da senha registrada (SenhaRegistrada) e do token de votação (TokenVotacao), respectivamente.

4.3. Configuração da fase de registro

Somente registradores podem configurar uma fase de registro. A configuração inicia com a definição do título da eleição a qual a fase de registro faz parte e, opcionalmente, da descrição da eleição. O título é utilizado como identificador e, portanto, deve ser único entre as demais fases de registro. Após essa definição inicial, a fase de registro é criada, porém ainda não pode ser iniciada.

Para iniciar a fase de registro, o registrador deve inserir os parâmetros (p, q, g_1, o, g_3) da fase de registro. Também é necessário inserir no sistema os valores (g, pk_R), onde g é um gerador do grupo G, e pk_R é a chave pública correspondente a chave privada (Equação 2) do registrador responsável pela fase. Somente o responsável pela fase de registro pode inserir esses valores. A Figura 2 apresenta a visão no sistema de uma fase de registro configurada e pronta para ser iniciada.

Com todos os dados necessários definidos, o registrador responsável pela fase de registro pode programar uma data de início automático para o registro de senhas. As informações e parâmetros relativos a fase de registro se tornam públicas após inicio do registro. Após o início, o encerramento da fase de registro somente ocorre com todos os registros em progresso finalizados, seja pela anulação do registro ou aceitação da credencial. Não é necessário que todos os votantes iniciem um registro de senha para que a fase seja encerrada.

4.4. Início da fase de registro

Com a fase de registro iniciada, votantes autorizados podem começar seu processo de registro de senha. Cada votante só pode ter um processo em andamento por vez. Caso

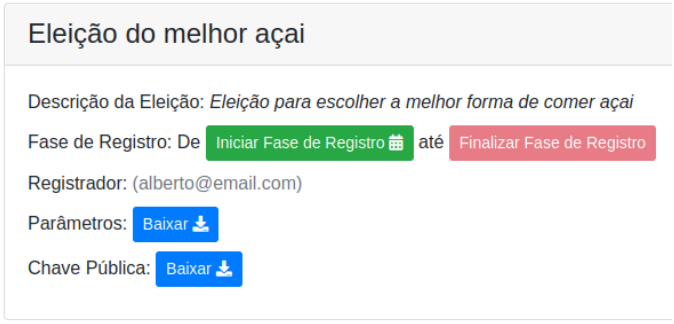

Figura 2. Visualização de uma fase de registro configurada e pronta para ser iniciada no sistema.

alguma verificação no registro seja inválida e ele seja anulado, o votante pode iniciar outro processo normalmente, desde que a fase de registro ainda não esteja encerrada. Votantes só podem ter uma única senha registrada por fase de registro, sendo impedidos de iniciar um novo processo caso já tenham se registrado.

As etapas da fase de registro do Protocolo de Registro Prático (ver Seção 3) foram resumidas nas cinco etapas apresentadas na Figura 3. As linhas contínuas apresentam a sequência de atividades no registro de uma senha. As linhas tracejadas apresentam quais objetos são criados e em quais etapas esses objetos são utilizados.

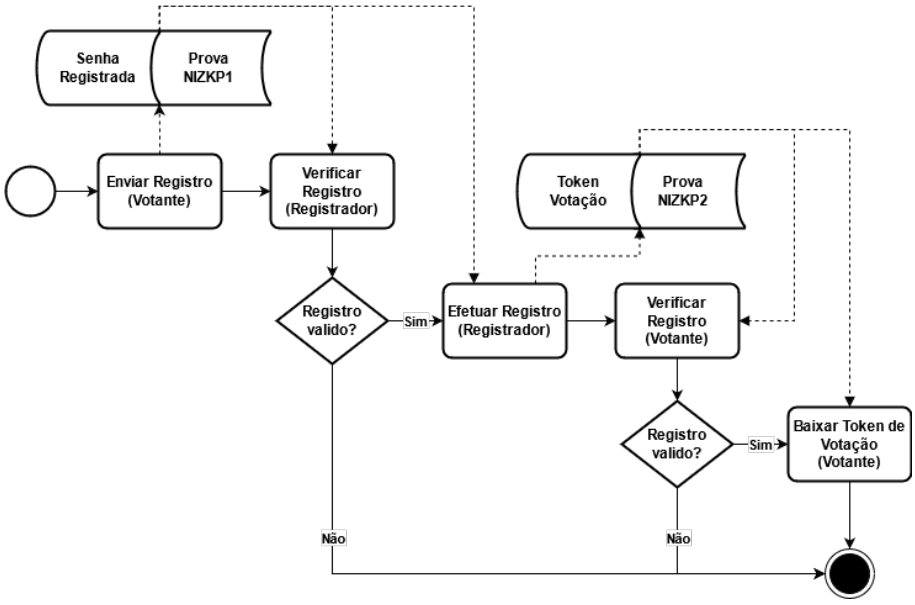

Figura 3. Diagrama de atividades do registro de senhas. São apresentadas as etapas do registro, bem como os objetos gerados e onde são utilizados.

O registro é iniciado exclusivamente pelo votante na Etapa **Enviar Registro (Votante)** da Figura 3. A Figura 4 apresenta a visão principal da página de criação de senhas no Sistema de Registro de Senhas. As palavras do dicionário de palavras são dispostas de forma aleatória na página web. O votante obrigatoriamente deve escolher cinco palavras e uma cor para cada palavra.

Com a senha definida, o conjunto de palavras/cores é concatenado na forma

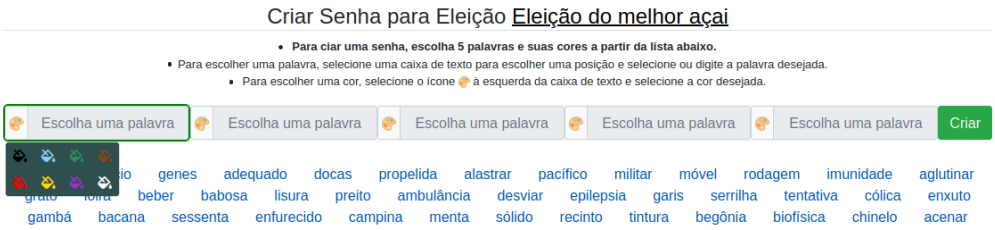

Figura 4. Visão principal da página de criação de senhas.

$palavra1||cor1||palavra2||cor2||...||palavra5||cor5$. O elemento x da credencial é calculado de acordo com a Equação 3, onde w é a concatenação das palavras/cores. Em seguida, o votante calcula C_1 de acordo com a Equação 4. Ele também calcula a prova de conhecimento-zero não-interativa $NIZKP1$, comprovando que conhece o expoente de g_3^x. Por fim, a tupla $(C_1, NIZKP1)$ é enviada ao sistema e o processo de registro avança.

Na Etapa **Verificar Registro (Registrador)** da Figura 3, o registrador verifica a validade da prova enviada pelo votante e o resultado é enviado ao sistema. Em caso negativo, o registro é anulado e o votante pode iniciar um novo processo. Caso contrário, o processo avança para a próxima etapa. Ao fim da etapa, o votante e o registrador podem verificar a prova $NIZKP1$ novamente a qualquer momento. Desse modo, é possível atestar o resultado previamente enviado ao sistema.

A Etapa **Efetuar Registro (Registrador)** da Figura 3 calcula a parte pública da credencial, chamada de token de votação no sistema. O registrador gera o valor $r \in Z_p^*$ e utiliza ele e sua chave secreta (Equação 2) no cálculo do valor A (Equação 5). Ele também calcula a prova de conhecimento-zero não-interativa $NIZKP2$, comprovando a utilização do C_1 enviado pelo votante, a chave secreta correspondente a chave pública da fase de registro e o valor r. A tupla $(A, r, NIZKP2)$ é enviada ao sistema e o processo segue para a próxima etapa.

Na Etapa **Verificar Registro (Votante)** da Figura 3, o votante verifica a validade da prova enviada pelo registrador, enviando o resultado para o sistema. Caso a prova seja inválida, o votante rejeita o registro, o processo é anulado e ele pode iniciar um novo processo de registro. Caso contrário, o votante aceita o registro e o token de votação gerado. Após essa etapa, é possível verificar a prova $NIZKP2$ novamente a qualquer momento, permitindo atestar o resultado previamente enviado.

Por fim, na Etapa **Baixar Token de Votação (Votante)** da Figura 3 o votante baixa as informações do token de votação gerado. O token é necessário para publicar um voto válido durante a fase de votação no sistema CIVIS. A Figura 5 apresenta a página web correspondente a essa etapa. Ao atingi-la, o votante não é permitido iniciar novos processos de registro de senha na fase de registro.

A Figura 6 apresenta as etapas descritas anteriormente nas visões do (a) registrador e do (b) votante. Um esquema de cores é utilizado para informar se uma etapa: já ocorreu (cinza claro); é a atual e o usuário é o responsável por executá-la (verde); é a atual e o usuário não é o responsável por executá-la (cinza escuro); falhou (vermelho); ou ainda vai ocorrer (amarelo). A medida que as etapas avançam, é possível baixar as informações relacionadas ao registro, bem como verificar a validade das provas novamente.

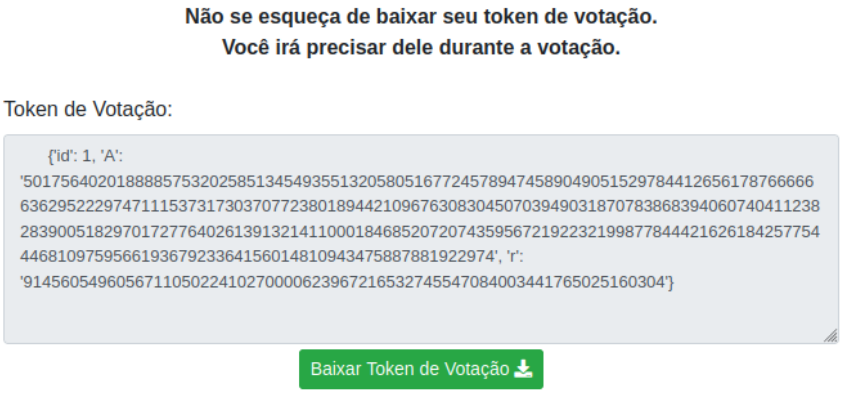

Figura 5. Visão principal da página para baixar o token de votação aceito. O votante pode tanto copiar os valores, quanto baixar o arquivo JSON.

Não é permitido a nenhum usuário voltar a etapas anteriores à etapa atual do processo. O acesso as Etapas 2 e 3 é exclusivo ao registrador responsável pela fase de registro. De forma semelhante, as Etapas 4 e 5 são exclusivas ao votante dono do registro. Na Etapa 1, somente votantes podem acessá-la.

Figura 6. Vista das etapas do processo de registro no sistema pelo (a) registrador e (b) votante.

4.5. Encerramento da fase de registro

O registrador responsável pela fase de registro somente pode encerrá-la com todos os registros em progresso finalizados, seja pelo votante aceitar o registro ou pelo registro ser anulado, tanto pelo votante quanto pelo registrador. Essa funcionalidade é exclusiva a ele. A fase de registro pode ser encerrada mesmo sem todos os votantes autorizados iniciarem o registro de uma senha. O encerramento é imediato e, a partir dele, não é permitido a nenhum votante iniciar um novo registro de senha na fase de registro.

Por fim, o sistema gera um relatório final da fase de registro. O relatório deve ser utilizado posteriormente no Sistema de Votação CIVIS para registro dos votantes. Ele contém todos os tokens de votações válidos gerados durante o período de registro. O relatório não possui qualquer valor que identifique quais votantes geraram quais tokens de votação, bem como informações específicas sobre as senhas do votante.

4.6. Adaptações ao Sistema CIVIS

O Sistema de Registro de Senhas faz parte do sistema CIVIS e deve ser utilizado em conjunto com ele. Para a integração dos dois sistemas, foram necessárias mudanças na fase de registro e na fase de votação do Sistema CIVIS. Também foi necessário introduzir alguns requisitos no Sistema de Registro.

Em relação as mudanças no sistema CIVIS, na sua implementação da fase de registro, o registrador gerava as credenciais válidas de votação e envia aos votantes. Posteriormente, eles inseriam a credencial válida no sistema e publicavam seus votos. Com a utilização do Sistema de Registro de Senhas, essa fase necessita de alterações. A partir de então, na fase de registro no CIVIS, o sistema somente necessita da inserção de um arquivo JSON contendo a lista de tokens de votação válidos. A lista inserida é disponibilizada no CIVIS no quadro público da eleição.

Ademais, para a publicação de um voto durante a fase de votação, a inserção da credencial de votação (A, r, x) foi substituída pela inserção do token de votação (A, r) e da senha correspondente. A senha inserida é novamente mapeada para o valor x através do mesmo processo de registro de senhas. Forma-se então a credencial de votação (A, r, x). Para publicar um voto inválido, o votante pode utilizar qualquer senha diferente da senha registrada, resultando em uma credencial inválida de votação $(A, r, x' \neq x)$.

A nova página de votação do Sistema de votação CIVIS pode ser visualizada na Figura 7. Agora é apresentado ao votante um campo para inserção do seu token de votação. Também é disponibilizado cinco campos para digitação das palavras e seleção da cor de cada palavra. Para evitar o uso de palavras de fora do dicionário, sugestões de palavras são apresentadas a medida as letras são inseridas nos campos.

Figura 7. Visão da nova tela de votação do Sistema CIVIS.

Em relação aos requisitos introduzidos ao Sistema de Registro, o sistema CIVIS disponibiliza as informações da eleição como título, descrição, parâmetros utilizados, chaves públicas das autoridades envolvidas e etc., em diversos arquivos no formato JSON.

Cada um desses arquivos estrutura seu conteúdo de um determinado modo. O Sistema de Registro espera que os arquivos inseridos durante a configuração da fase de registro contenham essas mesmas estruturas.

5. Trabalhos Relacionados

O trabalho de [de Sá et al. 2020a] apresenta uma nova abordagem no contexto de credenciais de votação. Neste trabalho, De Sá et al propõem a utilização de senhas compostas por palavras e cores. O objetivo é fornecer um mecanismo que proporcione usabilidade e segurança. A proposta é uma continuação do trabalho de [Neto et al. 2018] sobre a usabilidade de credenciais de votação em protocolos de votação resistentes à coerção.

A abordagem é expandida pelo autor na proposta de Protocolo de Registro Prático [de Sá et al. 2020b]. No trabalho, o autor apresenta um protocolo para definição das senhas e geração de credenciais válidas de votação de acordo com o protocolo de ABRTY. A implementação do protocolo, no entanto, só ocorre a partir da escrita deste trabalho.

6. Conclusão e Trabalhos Futuros

Esse trabalho apresentou a implementação de uma proposta de Protocolo de Registro Prático de Votantes. O sistema gerado recebeu o nome de Sistema de Registro de Senhas. Ele permite a geração de credenciais anônimas de votação de acordo com o protocolo de votação de ABRTY. As credenciais são geradas a partir de senhas definidas pelos próprios votantes. O sistema também foi integrado ao Sistema CIVIS. Desse modo, credenciais geradas no Sistema de Registro podem ser utilizadas em votações acontecendo no CIVIS.

A utilização dessas credenciais é uma alternativa mais prática em cenários reais de votação, pois o segredo do votante deixa de ser uma grande sequência de bits e passa a ser a senha definida durante o processo de registro. No entanto, ainda se faz necessário um estudo mais detalhado sobre a usabilidade do sistema em um cenário de votação real.

Deixa-se como trabalho futuro a utilização do Sistema de Registro em um cenário real de votação. Desse modo, será possível analisar não só os detalhes da sua integração ao Sistema CIVIS, mas também os aspectos de utilização das credenciais pelos votantes. Principalmente em relação a composição (quantidade de palavras e cores) e acessibilidade das credenciais de votação, e clareza e transparência do processo de registro de senhas.

Referências

Araujo, R., Neto, A., and Traoré, J. (2018). Civis-a coercion-resistant election system. In *Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 29–42. SBC.

Araújo, R., Rajeb, N. B., Robbana, R., Traoré, J., and Youssfi, S. (2010). Towards practical and secure coercion-resistant electronic elections. In *International Conference on Cryptology and Network Security*, pages 278–297. Springer.

Araújo, R. and Traoré, J. (2013). A practical coercion resistant voting scheme revisited. In *International Conference on E-Voting and Identity*, pages 193–209. Springer.

Bitcoin (2021). Readme wiki. *Disponível em https://github.com/bitcoin/bips.*

Clark, J. and Hengartner, U. (2008). Panic passwords: Authenticating under duress. *HotSec*, 8:8.

de Sá, M. O., Araujo, R., Sobrinho, A. C. L., Neto, A. S., Maximino, G. S., and Traoré, J. (2020a). How colored passwords can improve the usability of coercion-resistant internet voting systems. In *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–6.

de Sá, M. O., Araujo, R., Sobrinho, A. C. L., and Traoré, J. (2020b). Registro prático aplicado a um sistema de votação resistente à coerção. In *SBSeg-Main Track*. SBC.

Django, D. S. F. (2021). Django - the web framework for perfectionists with deadlines. `https://www.djangoproject.com/`. Acessado em: Junho/2021.

Juels, A., Catalano, D., and Jakobsson, M. (2010). Coercion-resistant electronic elections. In *Towards Trustworthy Elections*, pages 37–63. Springer.

Mozilla (2021). Mdn web docs - javascript. `https://developer.mozilla.org/en-US/docs/Web/JavaScript`. Acessado em: Junho/2021.

Neto, A. S., Leite, M., Araújo, R., Mota, M. P., Neto, N. C. S., and Traoré, J. (2018). Usability considerations for coercion-resistant election systems. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–10.

OpenJS, O. F. (2021). jQuery - write less, do more. `https://jquery.com/`. Acessado em: Junho/2021.

Palatinus, M., Rusnak, P., Voisine, A., and Bowe, S. (2013). Mnemonic code for generating deterministic keys. *Disponível em https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki*.

Python, P. S. F. (2021). Python language reference. `https://www.python.org/`. Acessado em: Junho/2021.

Silva Neto, A. A. and Araújo, R. (2017). A integração do criptossistema el gamal limiar ao sistema de votação via internet CIVIS. In *SBSeg-WTICG*, pages 697–706. SBC.