Registro Prático Aplicado a um Sistema de Votação Resistente à Coerção

Matheus O. L. de Sá[1], **Roberto Araújo**[1], **Alberto Sobrinho**[1], **Jacques Traoré**[2]

[1]Faculdade de Computação – Universidade Federal do Pará (UFPA)
Belém/PA, Brasil

matheus.sa@icen.ufpa.br, rsa@ufpa.br, alberto.sobrinho@icen.ufpa.br

[2]Orange Labs
Caen Cedex, France

jacques.traore@orange.com

Abstract. *The recent pandemic has taken many Brazilian institutions to adopt Internet voting. However, they certainly ignored the coercion problem. Although there is no proper solution for this problem, some voting protocols can mitigate it. For this, they use the idea of anonymous credentials. Voters receive these credentials in a registration phase and use them to vote later on. Unfortunately, voting protocols as ABRTY do not consider practical aspects in this phase and this hampers the use of these proposals in real world elections. In this context, this work introduces a practical protocol for registering credentials in ABRTY's scheme. The new idea is also applied to a coercion-resistant voting system.*

Resumo. *A pandemia recente levou muitas instituições brasileiras a adotarem votações via Internet. Todavia, elas certamente ignoram o problema da coerção. Embora esse problema não possua uma solução apropriada, alguns protocolos de votação possibilitam mitigá-lo. Para isso, eles utilizam a ideia de credenciais anônimas. Votantes recebem essas credenciais em uma fase de registro e as utilizam posteriormente para votar. Infelizmente protocolos de votação como o de ABRTY não consideram aspectos práticos na fase de registro, o que dificulta o uso desses protocolos em eleições reais. Neste contexto, este trabalho introduz um protocolo prático para o registro de credenciais no esquema de ABRTY. A nova ideia é também aplicada a um sistema de votação resistente à coerção.*

1. Introdução

A pandemia causada pelo COVID-19 levou muitas instituições brasileiras (e.g. universidades como a UFMS [Amin 2020]) a realizarem eleições via Internet para a escolha de seus gestores. Tais eleições evitaram o deslocamento de votantes a locais de votação bem como o contato necessário em uma eleição presencial, o que poderia favorecer a disseminação da doença. Assim, a realização de eleições via Internet foram necessárias e contribuíram para evitar problemas maiores.

Muitas dessas eleições, no entanto, desconsideraram ou não atentaram para o problema da coerção. Esse problema está associado a qualquer eleição via Internet. Como o votante pode votar a partir de qualquer dispositivo conectado a rede mundial, opressores

podem facilmente influenciá-los a escolherem determinados candidatos. Ademais, votantes podem provar em quem votaram simplesmente permitindo que terceiros os observem durante a emissão de seus votos, o que facilitaria a venda de votos.

Em princípio, os problemas apontados não possuem solução e isso pode levar muitos gestores a desconsiderá-los ao decidirem pelo emprego de votações via Internet. Todavia, a literatura recente apresenta soluções para mitigar tais problemas. Elas são baseadas na noção de resistência à coerção introduzida por Juels, Catalano e Jakobsson (JCJ) [Juels et al. 2005].

A noção de resistência à coerção considera um forte adversário que atua no ambiente eleitoral. Tal adversário é capaz de realizar ataques que generalizam parte dos problemas relacionados a eleições via Internet. Segundo a noção, uma votação (ou mais especificamente, um protocolo de votação) é resistente à coerção se um adversário não for capaz de forçar votantes a escolherem aleatoriamente opções de votos (ataque aleatório), de utilizar segredos (e.g. chaves privadas) pertencentes aos votantes e necessários para votar (ataque de simulação) e de forçar votantes a se absterem de votar (ataque de abstenção forçada). Além disso, não deve ser possível aos votantes a emissão de qualquer recibo que possa ser utilizado para comprovar suas opções de voto (isenção de recibos).

Desde a introdução da noção de resistência à coerção, várias propostas de protocolos criptográficos para votação foram introduzidas a fim de atendê-la, como o protocolo de JCJ [Juels et al. 2005]. Para atender corretamente a noção estabelecida, tais protocolos utilizam a ideia de credenciais anônimas. Essas credenciais são compostas por uma sequência de bits e são necessárias para votar. Para isso, os votantes recebem suas credenciais em sigilo e em um ambiente livre de adversários. Isso ocorre em uma fase de registro, preliminar a de votação.

A maioria das propostas existentes (por exemplo, o trabalho de [Araújo et al. 2010]), todavia, tratam apenas dos aspectos teóricos dessa fase e não especificam detalhes relacionados ao emprego do protocolo em uma eleição real. Isso dificulta o uso desses protocolos em cenários eleitorais. Além disso, muitas propostas consideram credenciais contendo um grande número de bits (e.g. 160bits) o que torna difícil sua utilização pelos votantes. Consequentemente, a resistência à coerção, que depende das credenciais, é inviabilizada pela dificuldade de uso desses elementos.

Contribuições

Tendo em vista a insuficiência de detalhes relativos a fase de registro da solução de [Araújo et al. 2010] bem como as dificuldades relacionadas ao emprego das credenciais pelos votantes, este trabalho apresenta um novo protocolo para a realização do registro de votantes. Tal protocolo torna o emprego da proposta de [Araújo et al. 2010] mais viável pois considera a sua aplicação em eleições reais. Para isso, o protocolo utiliza senhas compostas por palavras ao invés de números grandes (geralmente empregados em protocolos resistentes à coerção). O presente trabalho também demonstra a aplicabilidade do novo protocolo através de sua integração ao sistema de votação resistente à coerção CIVIS [Araújo et al. 2018].

2. Credenciais de Votação e o Protocolo de ABRTY

Credenciais anônimas são um dos principais mecanismos utilizados na construção de protocolos de votação resistentes à coerção. É por meio delas que votantes podem votar anonimamente e enganar adversários sobre sua real opção de voto. Votantes recebem credenciais legítimas, em segredo, na fase de registro. Tais credenciais indicam votos contáveis na apuração. Ao serem coagidos na fase de votação, votantes devem gerar e utilizar credenciais falsas. Essas credenciais indicam que os votos correspondentes devem ser removidos adiante. Ambas as credenciais são diferenciadas secretamente e de forma anônima na apuração. Dessa forma, não há como um adversário identificá-las nesse processo.

Uma credencial é composta por um conjunto de *bits* e sua composição pode diferir em protocolos resistentes à coerção. Em alguns protocolos, como o proposto por [Juels et al. 2005], a credencial é formada por um número aleatório grande (e.g. 160 bits). Outros, como o protocolo de [Araújo et al. 2010], utilizam credenciais baseadas em estruturas matemáticas, que também resultam em números aleatórios. Este trabalho tem como foco credenciais baseadas em estrutura matemáticas. Embora a proposta introduzida aqui possa ser adaptada a outros protocolos que utilizam esse tipo de credencial, ela tem como base o protocolo de [Araújo et al. 2010] (ABRTY).

A estrutura matemática das credenciais utilizadas no protocolo de ABRTY ajuda a garantir a segurança do mesmo. Para isso, as credenciais desse protocolo utilizam-se de problemas computacionalmente inviáveis que definem a estrutura da credencial. Tais problemas também evitam que adversários possam obter qualquer vantagem a partir da credencial, mesmo que eles obtenham muitas credenciais legítimas. As credenciais do esquema de ABRTY são baseadas no problema q-forte Diffie-Hellman (q-SDH) [Boneh and Boyen 2004] e no problema de decisão de Diffie-Hellman forte invertido (SDDHI) [Camenisch et al. 2006]. O primeiro problema garante que se um adversário tiver muitas credenciais genuínas, ele não poderá forjar uma nova credencial. O segundo problema garante que um adversário ativo não pode decidir se uma credencial é genuína ou falsa.

Uma credencial no esquema de ABRTY é composta por três valores: $\langle A, r, x\rangle$. Seja $\mathbb{G}$ um grupo cíclico de ordem p onde o problema de decisão de Diffie Hellman (DDH) [Boneh 1998] é difícil e sejam g, g_1, g_3 três geradores de $\mathbb{G}$. Para gerar uma credencial legítima, os registradores calculam cooperativamente $A = (g_1 g_3^x)^{\frac{1}{y+r}}$, onde $r, x \in \mathbb{Z}_p$ são números aleatórios e $sk_R = y$ é a chave privada compartilhada entre os registradores que corresponde a chave pública $pk_R = g^y$.

Embora uma credencial no esquema de ABRTY possua três valores $\langle A, r, x\rangle$, o votante precisa manter somente o valor x em sigilo. Como descrito em uma versão aprimorada desse protocolo proposta por [Araújo and Traoré 2013], os valores A e r podem ser disponibilizados publicamente sem afetar a segurança do protocolo. Portanto, uma credencial do esquema de ABRTY pode ser composta por uma parte pública $\langle A, r\rangle$ e uma parte privada x.

3. Um Protocolo para Registro Prático de Votantes

Esta seção apresenta a nova proposta de protocolo de registro para o esquema de ABRTY.

3.1. Primitivas Criptográficas

O novo protocolo requer as seguintes primitivas criptográficas.

Provas de Conhecimento (PoK)

Uma prova de conhecimento zero de conhecimento (ZKPK) é um protocolo interativo entre aquele que deseja realizar a prova P e o verificador V. P tenta convencer V sobre o conhecimento de algum segredo e V verifica a declaração de P. Tal verificação é realizada sem revelar qualquer informação sobre o segredo em sí.

O protocolo de registro introduzido neste trabalho utiliza a versão não interativa de ZKPKs, ou seja, prova de conhecimento zero de conhecimento não interativa (NIZKPK) para provar o conhecimento de (e entre relações de) logaritmos discretos em grupos cíclicos ou grupos de ordem desconhecida. A NIZKPK é obtida através da transformação heurística de Fiat-Shamir [Fiat and Shamir 1986]. A solução introduzida aqui requer o protocolo de assinaturas de Schnorr [Schnorr 1991] e o teste de igualdade de logaritmo discretos de Chaum e Pedersen [Chaum and Pedersen 1992]. As versões não interativas desses protocolos são descritas brevemente a seguir:

Seja um grupo cíclico $\mathbb{G}$ de ordem prima p (onde o problema de decisão de Diffie-Hellman [Boneh 1998] é difícil) e os valores g_1, g_2 são geradores do grupo. Seja H uma função criptográfica de hash segura como o SHA3-256 [NIST 2015].

Assinatura de Schnorr: Seja um valor g_1^r em que um indivíduo deseja provar o conhecimento do segredo r para um verificador. O indivíduo seleciona $t \in \mathbb{Z}_q$ aleatório, calcula $I = g_1^t \pmod q$ e o desafio $c = H(I, g_1^r) \pmod q$, calcula $J = t + rc \pmod q$ e envia (I, J, c) ao verificador. O verificador calcula e aceita a prova se $g_1^J = I(g_1^r)^c \pmod p$.

Teste de Igualdade de Logaritmo Discreto: Um indivíduo deseja provar ao verificador que os valores g_1^r e g_2^r foram gerados com o mesmo segredo r, ou seja, provar a igualdade de dois logaritmos discretos $log_{g_1} g_1^r = log_{g_2} g_2^r$ nas bases g_1 e g_2. Para isso, ele calcula seleciona $t \in \mathbb{Z}_q$ aleatório, calcula $I_1 = g_1^t \pmod q$ e $I_2 = g_2^t \pmod q$, calcula o desafio $c = H(I_1, I_2, g_1^r, g_2^r) \pmod q$ e $J = t + rc \pmod q$ e envia (I_1, I_2, J, c) ao verificador. O verificador calcula e aceita a prova se $g_1^J = I_1(g_1^r)^c \pmod p$ e $g_2^J = I_2(g_2^r)^c \pmod p$.

De forma a abstrair essas NIZKPs, aqui utiliza-se a noção introduzida por Camenisch and Stadler [Camenisch and Stadler 1997]. As provas são expressas através da notação $PoK[(\alpha, \beta, \dots) : \text{DECLARAÇÕES SOBRE } \alpha, \beta, \dots]$, onde $(\alpha, \beta, \cdots)$ são segredos (e.g. logaritmos discretos) que satisfazem a DECLARAÇÃO. P declara o conhecimento de $(\alpha, \beta, \cdots)$ e todos os outros valores no protocolo são públicos.

Provas de Verificação Designada (DVP)

Uma prova de verificação designada (DVP) [Jakobsson et al. 1996] é uma ZKPK específica. Ela visa convencer somente um verificador particular sobre uma declaração e não tem qualquer uso para qualquer outro indivíduo. Uma DVP consiste de uma prova em

que “se conhece a chave privada associada a chave pública designada de verificação" ou que uma dada declaração é verdadeira. Dessa forma, ao receber uma prova, o verificador (designado) será convencido de que a declaração é verdadeira enquanto outros não terão como verificá-la. Isso por que o próprio verificador sempre poderá provar ser ele mesmo o verificador (designado). Tudo o que ele precisa fazer é provar o conhecimento da chave privada associada com a chave pública do verificador (designado).

Funções de Derivação de Chaves

No protocolo de registro apresentado adiante, cada votante seleciona uma sequência de palavras que correspondem a uma senha. No entanto, tal senha não é utilizada diretamente para gerar uma credencial. Ao invés disso, ela é utilizada como uma chave e uma nova chave é derivada a partir dela. A nova chave é posteriormente processada para gerar uma credencial legítima. A fim de realizar tal derivação, é utilizada uma função de derivação de chave tal como o PBKDF2 [Kaliski 2000]. Como resultado, é adicionado mais entropia a chave o que torna mais difícil ataques de força-bruta.

3.2. A Nova Proposta

Baseado nas primitivas criptográficas introduzidas, a seguir é descrito o novo protocolo para registro de credenciais de votação.

Atores, Suposições de Segurança e Limitações

O novo esquema considera a existência de uma única autoridade de registro (registrador). Tal autoridade deve ser confiável de forma a emitir credenciais legítimas somente para votantes elegíveis. Além disso, ela deve manter sua chave secreta em sigilo. Cada votante interage com o registrador a fim de registrar uma única e exclusiva senha de votação. Essa senha representa a sua credencial legítima.

Para emitir a sua credencial, o votante visita um local protegido e autentica-se previamente (e.g. utilizando a sua identidade civil) para ter acesso ao ambiente de registro. Considera-se que este ambiente é livre de adversários. Por exemplo, um quiosque com acesso restrito contendo uma cabine para registro de senhas.

Adicionalmente, os computares utilizados no processo de registro devem estar livres de programas maliciosos e o canal de comunicação entre o votante e o registrador é considerado seguro. Idealmente, o canal de comunicação deve ser inviolável. Esse é um requisito teórico de qualquer protocolo de votação resistente à coerção. Para fins práticos, aqui a inviolabilidade desse canal é relaxada e considera-se que ele tenha apenas segurança computacional, e.g., através do protocolo TLS.

O emprego de uma única autoridade de registro, que deve ser confiável, bem como a necessidade de um ambiente seguro para registro dos votantes são limitações do protocolo. Dessa forma, para que o mesmo funcione adequadamente, é necessário que tanto o ambiente de registro (isso inclui, computadores, canais de comunicação, etc.) como o registrador sejam confiáveis. Ressalta-se que um registro confiável é um requisito fundamental de esquemas baseados nas ideias de JCJ. Tais esquemas dependem dessa suposição

para garantir resistência à coerção. Do contrário, adversários podem facilmente obter as transcrições da comunicação ou mesmo observar suas vitimas enquanto elas registram suas senhas, por exemplo.

Senhas

A solução apresentada aqui utiliza senhas como credenciais. Essas senhas são transformadas em valores numéricos para que possam ser empregadas no protocolo de ABRTY. A nova proposta é baseada na ideia de senhas de pânico [Clark and Hengartner 2008] (ver Seção 4.3). Diferentemente das senhas de pânico, ela utiliza um dicionário baseado no dicionário empregado pelo BIP-39 [Palatinus et al. 2013]. Originalmente, o BIP-39 define um dicionário de palavras que segue regras específicas como diferenciar as palavras a partir da terceira letra. Esse dicionário é utilizado na geração de carteiras determinísticas em *Bitcoin*.

O protocolo descrito a seguir requer um dicionário que segue as mesmas regras do BIP-39. Esse dicionário, no entanto, contém um número de palavras maior e deve ser definido a fim de manter um mínimo de 128bits de segurança. Como exemplo, para 150bits de segurança, é necessário um dicionário de 15bits (ou 32768 palavras) e uma senha de 10 palavras.

As senhas utilizadas na nova solução são baseadas nas ideias de senhas de pânico e do dicionário utilizado no BIP-39. Tais senhas são descritas e empregadas como segue. Seja um dicionário δ composto por n palavras (e.g. $n = 32768$) e seja m (e.g. $m = 8$) o número de palavras que compõem a senha, o votante seleciona aleatoriamente uma sequência de m palavras aleatórias $\omega \in \delta$. De forma a converter a senha do votante em um valor numérico, as palavras são primeiramente concatenadas $palavra_1||palavra_2||\cdots||palavra_m$ e então é aplicada a função de derivação de chaves. O número gerado a partir da derivação de chaves é mapeado para um elemento aleatório dentro do grupo numérico utilizado. Esse elemento corresponde ao valor secreto x da credencial numérica de ABRTY (ver Seção 2). Por exemplo, se for utilizado o grupo multiplicativo dos inteiros $\mathbb{Z}_p$ (onde p, q são números primos e $p = 2q + 1$) e o elemento obtido a partir da derivação de chaves for u, o valor x da credencial numérica é obtido calculando-se: $x = u \pmod q$.

Fase de Configuração

Seguindo o protocolo de ABRTY, a fase de configuração ocorre antes da fase de registro. Nessa fase, as autoridades de eleição definem um grupo cíclico $\mathbb{G}$ de ordem prima p. O problema de decisão de Diffie-Hellman [Boneh 1998] precisa ser difícil nesse grupo. Além disso, as autoridades de votação definem três geradores do grupo $g, g_1, g_3 \in \mathbb{G}$ e publicam o dicionário δ de onde os votantes devem escolher as palavras correspondentes as suas senhas. Seja R um único registrador, R calcula a sua chave secreta $sk_R = y \in \mathbb{Z}_p$ e a chave pública correspondente $pk_R = g^y$.

Fase de Registro

Nesta fase o votante recebe sua credencial de votação após provar que é um votante qualificado para isso. Essa fase, portanto, é onde o protocolo de registro de credenciais é executado. A fim de ter sua credencial gerada, o votante interage com o registrador conforme o *Protocolo de Registro de Autoridade Única*. Seja V um votante e R um registrador, o protocolo é apresentado a seguir.

Protocolo de Registro de Autoridade Única

1: V: Selecionar a sequência de palavras aleatórias e secreta $\omega \in \delta$;
2: V: Calcular $x = \text{PBKDF}(\omega)$ e $x \in \mathbb{Z}_p$, onde x é a parte privada da credencial;
3: V: Calcular $C_1 = (g_1 g_3^x)$;
4: V: Calcular a NIZKPK PoK_V;
5: $V \rightarrow R$: $[C_1, PoK_V]$;
6: R: Verificar PoK_V;
7: R: Se $PoK_V =$ FALSO, então ABORTAR;
Senão CONTINUAR;
8: R: Calcular o valor aleatório $r \in_R \mathbb{Z}_p$;
9: R: Calcular $(\frac{1}{y+r})$ a partir de sua chave $sk_R = y$ e do valor r;
10: R: Calcular $A = (g_1 g_3^x)^{\frac{1}{y+r}}$ (a partir de C_1) e a NIZKPK PoK_R;
11: $R \rightarrow V$: $[(A, r), PoK_R]$;
12: V: Verificar PoK_R;
13: V: Se $PoK_R =$ FALSO, então ABORTAR;
Senão ACEITAR credencial $\sigma = (A, r, x)$;

O protocolo de registro apresentado requer duas provas de conhecimento (PoK) como descrito a seguir:

PoK_V: O votante V utiliza esta prova para demonstrar que ele conhece o expoente x referente ao valor $C_1 = (g_1 g_3^x)$. Mais especificamente, por meio do protocolo de Schnorr (ver Seção 3.1), o votante prova que conhece o logaritmo discreto de C_1/g_1 na base g_3: $PoK_V = \text{POK}[\mu : C_1/g_1 = g_3^{\mu}]$.

PoK_R: O Registrador R utiliza esta prova para demonstrar que ele calculou $A = (g_1 g_3^x)^{\frac{1}{y+r}}$ a partir do valor C_1, sua chave secreta $sk_R = y$ e o valor aleatório r. De forma a calcular essa prova, supõem-se que $A = (g_1 g_3^x)^{\frac{1}{y+r}}$ e seja $h = (g1g3^x)$. Assim, $A^{y+r} = h$ que implica que $A^y = hA^{-r}$. Como o votante conhece sua credencial $\sigma = (A, r, x)$, ele pode calcular $B = aA^{-r}$. Para calcular a prova, o registrador utiliza o teste de igualdade de logaritmo discreto de [Chaum and Pedersen 1992]. Ou seja, ele prova que o logaritmo discreto de B na base A é igual ao logaritmo discreto de sua chave pública $R = g^y$ na base g: $PoK_R = \text{POK}[\lambda : A = a^{\lambda} \wedge R = g^{\lambda}]$.

3.3. Discussão (simplificada)

Como apresentado, a fase de registro é fundamental para garantir a segurança de qualquer protocolo de votação resistente à coerção. Nessa fase, votantes recebem credenciais legítimas que indicarão os votos que devem ser contados na apuração.

A geração da credencial legítima, viabilizada pelo novo protocolo, ocorre de forma cooperativa entre o votante e o registrador. Cada votante gera a sua senha de votação e ela é transformada na parte secreta de sua credencial (i.e. o valor x) em sigilo. Assim, nem a senha nem o valor correspondente a ela são revelados ao registrador. O registrador, por sua vez, recebe do votante um valor criptografado correspondente a senha dele e calcula a parte pública da credencial a partir desse valor.

O novo protocolo de registro utiliza mecanismos criptográficos para garantir que os votantes registrem suas senhas de forma segura. Para isso, após transformar a senha em um valor numérico x, o votante calcula $(g_1 g_3)^x$ e o envia para o registrador juntamente com uma NIZKPK. A dificuldade imposta pelo problema de decisão de Diffie-Hellmann (DDH) evita que o registrador obtenha qualquer informação sobre o valor x a partir do valor recebido $(g_1 g_3)^x$. Assim, a não ser que o registrador tenha a solução para o problema de DDH, ele não conseguirá obter o valor secreto x da credencial.

Após verificar como correta a NIZKPK recebida do votante, o registrador calcula os valores A e r que correspondem a parte pública da credencial do votante. Para isso, ele utiliza a sua chave privada. De forma a provar que utilizou o valor $(g_1 g_3)^x$ recebido do votante para calcular A e r, o registrador gera uma NIZKPK. Essa NIZKPK impede que um registrador malicioso gere uma parte pública da credencial que não corresponde a parte privada gerada pelo votante. A não correspondência resultaria na remoção do voto, relacionado à credencial, na apuração. Assim, ao receber os valores $A, r,$ NIZKPK, o votante aceita os valores A e r como parte de sua credencial caso a NIZKPK seja válida.

Como apresentado por [Araújo and Traoré 2013], a publicação dos valores A, r não implica em problemas de segurança. Portanto, eles podem ser disponibilizados através de uma página *Web* pública. Dessa forma, o votante pode ter certeza que sua credencial foi corretamente registrada. É importante ressaltar que o votante precisa obter esses valores para votar, mas o fato dos valores serem públicos facilita a obtenção dos mesmos.

Uma credencial falsa é gerada através da escolha de uma senha diferente da senha legítima, ou seja, qualquer sequência de palavras (dentro do dicionário utilizado) diferente da senha legítima. Considerando x' o valor resultante da senha falsa, esse valor pode ser utilizado juntamente com qualquer valor A, r publicado. O problema de decisão de Diffie-Hellman forte invertido impede que o adversário consiga verificar os valores A, r, x' como legítimos ou não.

A proposta aqui apresentada utiliza um único registrador. Esse registrador precisa ser confiável. Caso contrário, embora ele não possa obter as credenciais legítimas dos votantes (i.e. o valor x), ele ainda poderia emitir credenciais legítimas para terceiros, comprometendo a segurança do protocolo.

De forma a simplificar o processo eleitoral, algumas versões de sistemas (baseados em protocolos criptográficos de votação) empregam uma única autoridade confiável. No sistema Helios [Adida 2008], por exemplo, considerou-se uma única autoridade de apuração, apesar desse sistema não ser resistente à coerção. As versões iniciais do sistema CIVIS (apresentado adiante) consideram um único registrador, mas é possível distribuir o trabalho dos apuradores. O protocolo apresentado, portanto, poderia ser empregado em sistemas eleitorais onde não são necessários vários registradores.

4. Aplicando o Protocolo no Sistema de Votação CIVIS

De forma a demonstrar a aplicabilidade da proposta apresentada, o protocolo foi integrado ao sistema de votação CIVIS. Tal integração é apresentada a seguir.

4.1. O Sistema de Votação CIVIS

O CIVIS [Araújo et al. 2018] é um sistema de votação via Internet baseado no protocolo criptográfico para votação digital de ABRTY (ver Seção 2). O sistema foi desenvolvido em Python [PSF 2020], através do framework Django [DSF 2020], e utiliza JavaScript [MDN 2020] para realizar cálculos nos clientes (i.e. dispositivos dos votantes) que fazem acesso a seu servidor. Graças ao protocolo de ABRTY, o CIVIS possibilita aos votantes reagirem a ataques coercivos. Tal reação é realizada através das credenciais anônimas como apresentado na Seção 2.

Uma votação no sistema possui quatro etapas: configuração, registro, votação e apuração. Na etapa de configuração é gerado o material criptográfico necessário para a votação bem como a definição de autoridades, candidatos, etc. Na etapa de registro os votantes recebem suas credenciais de votação legítimas de uma autoridade de registro (registrador). Os votantes escolhem suas opções de voto e emitem seus votos na etapa de votação. Por fim, as autoridades de apuração realizam o processo identificação dos votos legítimos, a partir das credenciais, e apresentam os resultados finais.

As fases de registro e de votação do sistema são de particular importância para o trabalho introduzido aqui. A primeira está diretamente relacionada ao protocolo apresentado na seção anterior. A segunda depende das credenciais geradas na fase de registro. Essas fases funcionam originalmente no CIVIS da seguinte forma.

Na fase registro, após a identificação do votante, o registrador gera uma única e exclusiva credencial para ele. Tal credencial é composta pelos valores $\langle A, r, x\rangle$ e não há separação entre a parte pública A, r e a parte privada x. As duas partes são codificadas em conjunto gerando um único arquivo no formato JSON. O votante deve manter esse arquivo em sigilo. Na fase de votação, cada votante insere o seu arquivo JSON, em um campo apropriado, após escolher suas opções.

4.2. Integrando o novo Protocolo ao Sistema CIVIS

De forma a integrar o CIVIS ao protocolo introduzido na Seção 3.2 são necessárias modificações na fase de registro. Diferentemente da fase original do sistema, agora o votante deve selecionar uma sequência de palavras e a credencial é gerada a partir dela. Para isso, o novo protocolo requer a interação entre o votante e o registrador. Ademais, a credencial agora possui duas partes: a pública contendo os valores A, r e a privada, relativa a senha, que corresponde ao valor x.

A nova fase de registro do sistema foi dividida em duas partes: votante e registrador. Em cada uma dessas partes foi adicionado um mapa. É por meio deles que o registrador e o votante podem acompanhar e realizar todos os passos para geração de uma credencial. Ambos os mapas contém os mesmos passos, mas a utilização deles somente é permitida de acordo com o papel (e.g. votante ou registrador). Enquanto o mapa do votante contém somente os passos relativos a emissão de sua credencial, o mapa do registrador contém informações de todos os votantes que solicitaram registro. A fim de abstrair

detalhes do protocolo, várias etapas foram unidas em um único passo dentro do mapa. As Figuras 1 e 2 ilustram os mapas referentes ao registrador e ao votante, respectivamente.

Figura 1. Mapa referente ao registrador. Este mapa contém a solicitação de registro de todos os votantes e as etapas correspondentes.

Figura 2. Mapa referente ao votante. Este mapa exibe as etapas necessárias para o votante registrar sua credencial.

Dentro do passo 1. **Registro Enviado**, o votante informa a sua sequência de palavras (i.e. a sua senha) como ilustrado na Figura 3. Ela é transformada em um número aleatório x dentro do grupo multiplicativo utilizado pelo sistema. Após isso, esse número é criptografado (i.e. $g_1 g_3^x$) e é gerada uma prova de conhecimento sobre o valor x. O valor $g_1 g_3^x$ é codificado em JSON juntamente com a prova gerada. O JSON é enviado ao registrador.

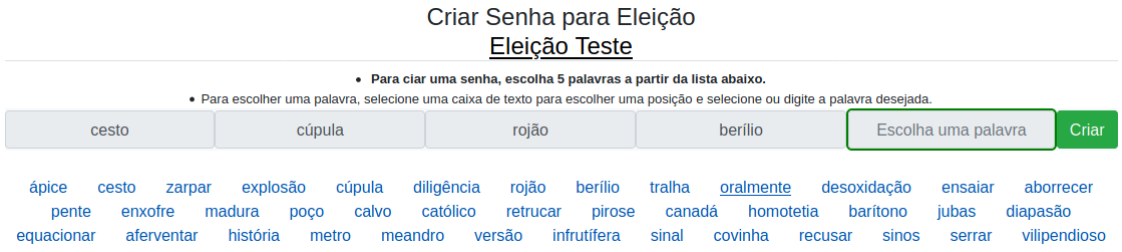

Figura 3. Seleção de uma senha de 5 palavras pelo votante a partir de um dicionário (para fins de ilustração apenas parte do dicionário é apresentado).

Após o votante realizar o passo 1, o registrador pode iniciar o passo 2. **Registro Aceito**. Nesse passo, o registrador verifica a prova do votante. Se a prova for válida, o registrador realiza o passo 3. **Registro Efetuado**. Nesse passo, o registrador gera um número aleatório r, calcula do valor A utilizando sua chave privada e gera uma prova de conhecimento de que utilizou o valor gerado por A e sua chave privada. O valores A, r bem como a prova são codificados em JSON e enviados ao votante. A Figura 4 apresenta o diagrama de atividades do registrador.

Ao fim do passo 3, o votante pode realizar o passo 4. **Verificar Registro**. Nesse passo, o votante verifica a prova e, se ela for válida, ele aceita a credencial legítima e pode iniciar o passo 5. **Baixar Token de Votação**. Esse é o passo final. O votante realiza o *download* do arquivo JSON contendo a parte pública de sua credencial, i.e., os valores A, r. Isso finaliza o processo de registro para esse votante.

Ao fim da etapa de registro, o registrador publica (em uma página *Web*) os valores A e r, codificados em JSON, de todos os votantes registrados. Ressalta-se que todos os cálculos realizados (e.g. geração do valor A) e geração de segredos (e.g. números aleatórios) utilizados no processo são realizados localmente no computador de seus respectivos participantes (votante ou registrador).

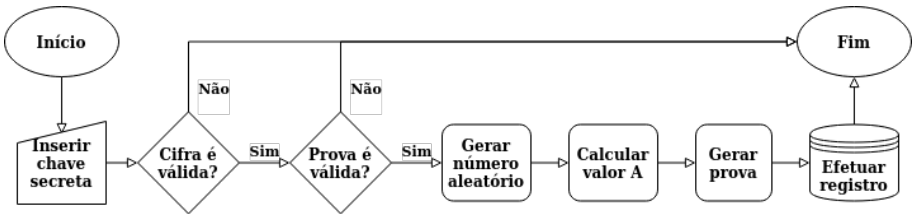

Figura 4. Diagrama de atividades realizadas pelo registrador.

4.3. Trabalhos Relacionados

A proposta aqui apresentada detalha um protocolo de registro de votantes para o esquema de ABRTY. Ela considera o uso desse esquema em cenários práticos de votação. A literatura não apresenta trabalhos, como o proposto aqui, considerando o registro prático em protocolos de votação que utilizam credenciais baseadas em estruturas matemáticas. Todavia, a nova proposta está relacionada a outros trabalhos.

A nova solução é baseada nas senhas de pânico proposta por [Clark and Hengartner 2008]. Nessa solução existe um conjunto de senhas válidas e de senhas inválidas. As senhas válidas são aceitas pelo sistema em uso enquanto as inválidas são rejeitadas. As senhas de pânico fazem parte do conjunto de senhas válidas e devem ser utilizadas para indicar uma ação anormal (e.g. coerção). As senhas admissíveis também fazem parte do conjunto de senhas válidas, mas elas indicam uma ação normal (e.g. a emissão de um voto válido). Enquanto um adversário não consegue distinguir entre as duas senhas, o sistema pode realizar a distinção e prosseguir de acordo com a senha (e.g. aceitar um voto como legítimo na contagem final). As senhas admissíveis assim funcionariam como credenciais válidas e as senhas de pânico como credenciais falsas.

As senhas de pânico foram adotadas no protocolo de votação resistente à coerção de [Clark and Hengartner 2011]. Apesar de tomar emprestado a ideia de senhas de pânico, o protocolo aqui descrito considera um novo dicionário de palavras. Tal dicionário é baseado no dicionário utilizado no BIP-39.

Embora a literatura não apresente soluções como a introduzida, [Leite and Araújo 2019] propuseram um esquema que poderia potencialmente ser integrado a diversos protocolos de votação. Tal solução é baseada no BIP-39. Infelizmente a solução deles não foi integrada ao protocolo de ABRTY. Além disso, como

discutido por eles no trabalho, a solução apresenta alguns desafios que precisariam ser resolvidos antes do seu emprego em protocolos de votação.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou um esquema prático de registro de votantes para o protocolo de votação de ABRTY. A nova proposta considera o emprego do protocolo de ABRTY em cenários práticos de eleição. Para isso, são utilizadas senhas como credenciais ao invés de valores numéricos. Essas senhas tornam mais fácil a utilização das credenciais legítimas pelos votantes pois facilitam a sua memorização. Além disso, elas facilitam a geração de credenciais falsas. O trabalho também demonstrou a aplicabilidade da nova proposta através do sistema de votação CIVIS.

O protocolo proposto pode ser utilizado para o registro seguro de credenciais legítimas desde que o ambiente de realização do registro seja confiável. Em outras palavras, é necessário que o ambiente de registro seja livre de adversários, permitindo somente a interação entre votante e registrador. Do contrário, não há como garantir a segurança do registro de senhas pois votantes podem ser facilmente manipulados por adversários.

Ademais, é necessário que o registrador seja confiável. Embora procedimentos adicionais possam ser adotados para reduzir essa confiança (e.g. observar o registrador enquanto ele emite credenciais), um protocolo com um número maior de autoridades de registro é recomendável. Os registradores, assim, emitiriam credenciais de forma cooperativa. Um protocolo desse tipo é deixado como trabalho futuro bem como avaliar a usabilidade do protocolo proposto por meio de votações simuladas.

Referências

Adida, B. (2008). Helios: Web-based open-audit voting. In van Oorschot, P. C., editor, *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*, pages 335–348. USENIX Association.

Amin, V. (2020). Servidores e estudantes têm até 21h para participar de consulta pública. Disponível em `https://www.ufms.br/servidores-e-estudantes-tem-ate-21h-para-participar-de-consulta-publica/`. Acesso em 30/07/2020.

Araújo, 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, Natal, RN, Brasil. Sociedade Brasileira de Computação (SBC).

Araújo, R., Rajeb, N. B., Robbana, R., Traoré, J., and Yousfi, S. (2010). Towards practical and secure coercion-resistant electronic elections. In Heng, S., Wright, R. N., and Goi, B., editors, *Cryptology and Network Security - 9th International Conference, CANS 2010, Kuala Lumpur, Malaysia, December 12-14, 2010. Proceedings*, volume 6467 of *Lecture Notes in Computer Science*, pages 278–297. Springer.

Araújo, R. and Traoré, J. (2013). A practical coercion resistant voting scheme revisited. In Heather, J., Schneider, S. A., and Teague, V., editors, *E-Voting and Identify - 4th International Conference, Vote-ID 2013, Guildford, UK, July 17-19, 2013. Proceedings*, volume 7985 of *Lecture Notes in Computer Science*, pages 193–209. Springer.

Boneh, D. (1998). The decision diffie-hellman problem. In Buhler, J., editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer.

Boneh, D. and Boyen, X. (2004). Short signatures without random oracles. In Cachin, C. and Camenisch, J., editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer.

Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., and Meyerovich, M. (2006). How to win the clonewars: Efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, page 201–210, New York, NY, USA. Association for Computing Machinery.

Camenisch, J. and Stadler, M. (1997). Proof systems for general statements about discrete logarithms. Technical report, Institute for Theoretical Computer Science, ETH Zurich.

Chaum, D. and Pedersen, T. P. (1992). Wallet databases with observers. In Brickell, E. F., editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer.

Clark, J. and Hengartner, U. (2008). Panic passwords: Authenticating under duress. In Provos, N., editor, *3rd USENIX Workshop on Hot Topics in Security, HotSec'08, San Jose, CA, USA, July 29, 2008, Proceedings*. USENIX Association.

Clark, J. and Hengartner, U. (2011). Selections: Internet voting with over-the-shoulder coercion-resistance. In Danezis, G., editor, *Financial Cryptography and Data Security - 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28 - March 4, 2011, Revised Selected Papers*, volume 7035 of *Lecture Notes in Computer Science*, pages 47–61. Springer.

DSF, D. S. F. (2020). Django - The web framework for perfectionists with deadlines. `https://www.djangoproject.com/`. Acesso em Agosto/2020.

Fiat, A. and Shamir, A. (1986). How to prove yourself: Practical solutions to identification and signature problems. In Odlyzko, A. M., editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer.

Jakobsson, M., Sako, K., and Impagliazzo, R. (1996). Designated verifier proofs and their applications. In Maurer, U. M., editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer.

Juels, A., Catalano, D., and Jakobsson, M. (2005). Coercion-resistant electronic elections. In Atluri, V., di Vimercati, S. D. C., and Dingledine, R., editors, *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005*, pages 61–70. ACM.

Kaliski, B. (2000). PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898.

Leite, M. and Araújo, R. S. (2019). Credenciais de votação baseadas em bip para protocolos de votação resistentes à coerção. In *Anais do XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - IV Workshop de Tecnologia Eleitoral*, São Paulo, SP, Brasil. Sociedade Brasileira de Computação (SBC).

MDN (2020). MDN Web Docs - Javascript. `https://developer.mozilla.org/en-US/docs/Web/JavaScript`. Acesso em Agosto/2020.

NIST (2015). FIPS PUB 202 – SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Disponível em `https://csrc.nist.gov/publications/detail/fips/202/final`.

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`. GitHub repository.

PSF, P. S. F. (2020). Python language reference. `http://www.python.org/`. Acesso em Agosto/2020.

Schnorr, C. (1991). Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174.