Selections: Internet Voting with Over-the-Shoulder Coercion-Resistance

Jeremy Clark and Urs Hengartner

University of Waterloo
{j5clark,uhengart}@cs.uwaterloo.ca

Abstract. We present Selections, a new cryptographic voting protocol that is end-to-end verifiable and suitable for Internet voting. After a one-time in-person registration, voters can cast ballots in an arbitrary number of elections. We say a system provides over-the-shoulder coercion-resistance if a voter can undetectably avoid complying with an adversary that is present during the vote casting process. Our system is the first in the literature to offer this property without the voter having to anticipate coercion and precompute values. Instead, a voter can employ a panic password. We prove that Selections is coercion-resistant against a non-adaptive adversary.

1 Introductory Remarks

From a security perspective, the use of electronic voting machines in elections around the world continues to be concerning. In principle, many security issues can be allayed with cryptography. While cryptographic voting has not seen wide deployment, refined systems like Prêt à Voter [11,29] and Scantegrity II [9] are representative of what is theoretically possible, and have even seen some use in governmental elections [7]. Today, a share of the skepticism over electronic elections is being apportioned to Internet voting.[1] Many nation-states are considering, piloting or using Internet voting in elections. In addition to the challenges of verifiability and ballot secrecy present in any voting system, Internet voting adds two additional constraints:

- Untrusted platforms: voters should be able to reliably cast secret ballots, even when their devices may leak information or do not function correctly.
- Unsupervised voting: coercers or vote buyers should not be able to exert undue influence over voters despite the open environment of Internet voting.

As with electronic voting, cryptography can assist in addressing these issues. The study of cryptographic Internet voting is not as mature. Most of the literature concentrates on only one of the two problems (see related work in Section 1.2). In this paper, we are concerned with the unsupervised voting problem. Informally, a system that solves it is said to be coercion-resistant.

[1] One noted cryptographer, Ronald Rivest, infamously opined that "best practices for Internet voting are like best practices for drunk driving" [25].

1.1 Contributions

Coercion-resistant, end-to-end verifiable Internet voting systems have been proposed [1,4,14,24,33,34]. However, these systems all require the voter to remember cryptographic information after registration. Since the information is too long to memorize, authentication can be considered to be based on "something you have." Voters must prepare for the possibility of coercion by creating fake values, proofs, or transcripts. Our system works with passwords, "something you know," and it allows a voter to supply a panic password during ballot casting that can be created mentally in real-time by the voter. In summary, our system provides:

- Password-based authentication and cognitive coercion-resistance,
- In-person registration that can be performed bare-handed,
- Tallying that is linear in the number of voters, and
- Efficient revocation of voters from the roster during and between elections.

We compare Selections to three systems: JCJ [24], Civitas [14], and AFT [4] (see Section 1.2). Of these properties, only Selections meets each while AFT achieves the third and both JCJ and Civitas achieve the fourth.

1.2 Related Work

The field of cryptographic voting is mature. One survey in 2005 reviews 27 systems [31] and there has been no shortage of new proposals since. At this point, any new proposals for systems should be soundly motivated. Our system addresses the problem of coercion and vote selling when voters are not required to vote in a private booth. Only a small number of the most recent papers in cryptographic voting address this threat.

Coercion-resistance was first formalized by Juels *et al.* [24], who also provide a coercion-resistant system, often referred to as JCJ. JCJ was independently implemented as Civitas [14]. The main drawback of both is that tallying is quadratic in the number of voters. Aquisti [1] refined JCJ to use Paillier encryption and support write-in candidates, while both Smith [33] and Weber *et al.* [34] made the first attempts at reducing the complexity of tallying to linear. Unfortunately, both are considered broken [4,14].

Araujo *et al.* provide a linear-time system we refer to as AFT [4]. Although no proofs are provided, the system appears sound. Both JCJ/Civitas and AFT provide registered voters with anonymous credentials. A voter submits a credential along with her vote and a procedure for computing a fake credential is provided (but cannot be done without a computer). In JCJ/Civitas, the credentials of registered voters are posted and these are anonymously and blindly compared to the credential accompanying each submitted vote. In AFT, the credentials of registered voters are essentially signed and the presence of a valid signature on a credential submitted during casting is anonymously and blindly checked. Due to the difficulty of revoking a signed value, voters cannot be revoked in AFT without a change of cryptographic keys.

Some Internet systems are designed for low-coercion elections. These include Helios [2], which was used in a binding university election [3]. Other Internet voting systems concentrate on the untrusted platform issue. A common approach is "code voting," where acknowledgement codes are returned to voters upon receipt of a vote. The codes are a function of the vote and not known in advance to the network carrier. This principle can be seen in SureVote [8], CodeVoting [23], Pretty Good Democracy [30], and Heiberg *et al.* [18].

2 Preliminaries

2.1 Selections: High-Level Overview

Selections is a protocol designed to allow voters to cast ballots over the Internet during a window of time prior to traditional in-person voting. Voters can opt out of Selections at any time prior to election day and cast a ballot in-person.

To be eligible for Selections, voters first complete a one-time, in-person registration protocol in a private booth without needing her own computational device. After this registration, the voter can vote in future elections over a tappable channel (see Section 2.3). The registration involves the voter choosing a password to be used for vote casting. However this password is non-traditional—it is a password from a panic password system (see Section 2.5). A semantically-secure homomorphic encryption of this password is posted on a public roster. The roster has an entry for each registered voter containing this ciphertext. The voter must be convinced that her entry is a correct encryption without being able to prove what it encrypts to anyone.

During vote submission, the voter asserts what her password is: it may be her actual password or a panic password. The voter creates a binding commitment to this asserted password. The voter then rerandomizes her entry off the roster. The voter proves in zero-knowledge the latter ciphertext is a re-encryption of some random subset of passwords off the public roster, without revealing which one. The commitment to her asserted password, re-encrypted roster entry, proof (and some additional proofs that things are well-formed), and an encryption of her vote are submitted over an anonymous channel to a public bulletin board.

When the voting period expires, a distributed group of trustees will eliminate submissions with invalid proofs, eliminate duplicate votes based on the password commitment, and then use a verifiable mix network to shuffle the order of the remaining submissions. After shuffling, voters can no longer determine where their submission is in the new permuted list. For each submission, the trustees will determine if the asserted password matches the roster entry without revealing either. If it does not, the entry is eliminated. The output of Selections is a list of encrypted votes from registered voters without duplicates. The entire protocol can be verified for soundness.

2.2 Coercion-resistance

Informally, Juels *et al.* define coercion-resistance as providing receipt-freeness, while preventing three attacks: randomization, abstention, and simulation [24].

A voting system is said to be receipt-free if the voter cannot produce a transcript that constitutes a sound argument for how they voted [6]. Adversaries should not be able to force a registered voter to cast a random vote or to abstain from voting. Finally, the system should protect against voters surrendering their credentials and allowing a coercer or vote buyer to cast their vote for them. The dominant approach to preventing such a simulation is providing voters with the ability to create fake credentials. If an adversary cannot distinguish a real credential from a fake one, he will only be willing to pay what a fake credential is worth, which is nothing.

2.3 Untappable Channels

The main challenge for coercion-resistant Internet voting is dealing with the elimination of the private voting booth, modelled as an untappable channel. One approach is to use multiple secure channels and assume that while any individual channel can be tapped, no adversary can tap all channels simultaneously. The second is to use an untappable channel just once, and bootstrap the output of this interaction into an arbitrary number of future interactions over secure (or anonymous) channels. We use the latter approach.

2.4 Registration Authority

In most coercion-resistant Internet voting systems, voters interact with a distributed registration authority [1,4,24]. To achieve coercion-resistance, it is assumed that at least one registrar is not corrupted by the adversary. Voters may be corrupted to retain a transcript, however the transcript has deniability by using a designated verifier proof [21].

While distributing trust is usually an effective approach for achieving correctness and secrecy in a protocol, it is more complex with coercion-resistance. The voter must be aware of which entity she trusts, so she can fake a proof that will not be compared to the original. If the voter discloses her private key to an adversary, it only requires a single malicious registrar to collude with the adversary and undetectably issue the voter an incorrect credential share (while retaining the correct value for potential adversarial use).

These concerns leave it unclear if the benefits of a distributed registration authority are worthwhile. While Selections is amenable to a distributed registration authority (voters would submit encryptions of shares of their password, which are homomorphically combined to create an encryption of the password), we describe the protocol using a single registrar that is assumed to not collude with a coercer (but may still misbehave in any other regard).

2.5 Panic Passwords

A panic password system [12] initializes three categories of passwords: a password, a set of panic passwords, and the residual set of inadmissible passwords.

From the user's view, submission of a password or a panic password is indistinguishable, while an inadmissible password will prompt the user to try again. If the user registers a password and one panic password, an adversary can demand two distinct admissible passwords and submit the coerced vote with each—therefore, the number of panic passwords should be arbitrarily large to prevent these "iteration" attacks. If a user registers a password and all other values are panic passwords, an accidental mistyping will result in the vote being discarded—therefore, the distance between admissible and inadmissible passwords should be maximized. Finally, with an arbitrarily large number of panic passwords distributed sparsely among inadmissible passwords, set-membership tests for panic passwords should be cognitively easy to perform.

Clark and Hengartner propose the 5-Dictionary panic password system to meet these requirements [12]. Admissible passwords consist of five words from an agreed upon dictionary: the user chooses one combination as her password and any other combination is a panic password. A typo is likely to mutate the intended word into a string not found in the dictionary. With the Unix dictionary of English words, this system offers up to 70 bits of entropy, making exhaustive search infeasible at this time.[2] The authors also propose the 5-Click alternative based on graphical passwords, and new panic password schemes could be developed based on, for example, preferences [22]. Voters would be free to choose which to use.

3 The Selections Protocol

Selections involves four participants: a set of voters, a set of election trustees, an election authority, and a registrant. The system has six main protocols: registration set-up, voter preparation, registration, election set-up, casting, and pre-tallying. Let $\langle \mathsf{DKG}, \mathsf{Enc}, \mathsf{DDec} \rangle$ be a threshold encryption scheme. Distributed key generation $\mathsf{DKG}(n, m)$ generates public key, e, and a private key share, d_i, for each of n trustees. Encryption, $\mathsf{Enc}_e(m, r)$, is semantically secure and homomorphic with respect to one operation. Distributed decryption, $\mathsf{DDec}_{d_i}(c)$, on ciphertext c can be performed with $m+1$ trustees submitting shares d_i.[3] We use threshold Elgamal [27], which is IND-CCA1 secure [26].

3.1 Registration Setup

The registration set-up protocol involves a set of n trustees: $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and the election authority. Primes p and q are chosen such that the DL-problem and DDH-problem are hard in the multiplicative subgroup $\mathbb{G}_q$ of $\mathbb{Z}_p^*$. Each T_j participates in $\mathsf{DKG}(n, m)$. Commitments are sent to the election authority, who posts them

[2] In reality, users are unlikely to choose uniformly from the entire dictionary and reach this maximum. The number of words can be increased to compensate for this.

[3] Proactive security can maintain the secrecy of the shares over time, both the number of shares and the threshold can be adjusted without a dealer, and more a complex access structure than m-out-of-n can be created.

to an append-only broadcast channel called the Bulletin Board. At the end of the protocol, each $\mathcal{T}_j$ has private key share d_j and public key e is posted. The protocol is standard and will not be described here [27].

3.2 Voter Preparation

The voter preparation procedure is performed by each voter $\mathcal{V}_i$ on a trusted computational client. Let $\langle P, I \rangle$ be the domain of a panic password system. P represents the set of admissible passwords and $I = \neg P$ is the set of inadmissible passwords. $\mathcal{V}_i$ chooses a password $\hat{\rho}$. The client runs $\mathsf{PassSubmit}(\hat{\rho})$, which tests if $\hat{\rho} \in P$. If $\hat{\rho} \in I$, $\mathsf{PassSubmit}(\hat{\rho})$ returns an error. The set of panic passwords are the remaining passwords in P: $\{\forall \hat{\rho}^* \in P | \hat{\rho}^* \neq \hat{\rho}\}$. $\mathsf{PassSubmit}(\hat{\rho^*})$ will behave identically upon submission of a panic password (otherwise an adversary could distinguish the case where he is given a panic password).

Once $\mathsf{PassSubmit}(\hat{\rho})$ accepts $\hat{\rho}$, the client encodes $\hat{\rho}$ as a bitstring and appends a non-secret salt to prevent accidental collisions with other users. This string is supplied as input to a password-based key derivation function (PBKDF) for strengthening and encoding into $\mathbb{Z}_q^*$. For brevity, we denote this entire password processing procedure as ϕ: $\rho \leftarrow \phi(\hat{\rho}) = \mathsf{PBKDF}(\mathsf{PassSubmit}(\hat{\rho}) \| \mathtt{salt})$.

Perhaps through a user-guided tutorial familiarizing the voter with the system, the voter will generate α admissible passwords: $\hat{\rho}_1, \ldots, \hat{\rho}_\alpha$. The value of α will determine the soundness of the registration protocol. An example value for α is 10. The password the voter wishes to register is in a random location in the list. Each is encrypted by the voter under the trustees' public key e. The voter prints out the list of ciphertexts on to a piece of paper, *e.g.,* with the ciphertexts encoded into barcodes. The registration protocol in Algorithm 1 includes the voter preparation protocol.

3.3 Registration

The registration protocol is completed by each voter $\mathcal{V}_i$. It is a two-party cut-and-choose protocol between a voter $\mathcal{V}_i$ and the registrar $\mathcal{R}$. The protocol is described in Algorithm 1. It is an adaptation of the Benaloh's voter initiated auditing [5], with a predetermined number of challenges. The voter enters the protocol with a list of α encrypted passwords $\{c_1, \ldots, c_\alpha\}$ and the protocol completes with a re-encryption of one of the ρ's being posted to an append-only broadcast channel, called the Roster. The protocol itself is conducted over an untappable channel which is instantiated as an in-person protocol.

The voter presents identification and is authorized to register. The voter is given a blank transcript card and enters a private booth that has a computer in it capable of printing and scanning barcodes. A transcript card has α rows and two columns. The second column for each row has a scratch-off surface. The voter is provided the option of downloading and printing a document from the Internet—with the intention that the voter could print her voter preparation sheet in the event that an adversary ensured she entered the registration process

Algorithm 1: Registration Protocol

Participants : Voter $\mathcal{V}_i$ and registrant $\mathcal{R}$
Public Input: Encryption parameters p, q, g, public key e, and soundness parameter $\alpha > 1$
Private Input (V_i): Ciphertexts $\{c_1, \ldots, c_\alpha\}$ as described below

Prior to the protocol, each voter should:

for *k from 1 to* α **do**
1. Choose a password $\hat{\rho}_k$.
2. Process password: $\rho_k \leftarrow \phi(\hat{\rho}_k)$.
3. Encrypt g^{ρ_k} with random r_k: $c_k \leftarrow \mathsf{Enc}_e(g^{\rho_k}, r_k)$.
4. Complete a NIZKP of knowledge of plaintext g^{ρ_k}: $\pi_k \leftarrow \text{NIZKP}_{pok}\{(\rho_k, r_k) : c_k = \mathsf{Enc}_e(g^{\rho_k}, r_k)\}$.
5. Record $\langle c_k, \pi_k \rangle$.

Registrar should:

6. Receive $\{\langle c_1, \pi_i \rangle, \ldots, \langle c_\alpha, \pi_\alpha \rangle\}$.
7. **for** *k from 1 to* α **do**
8. Check π_k.
9. Rerandomize c_k with random r'_k: $c'_k \leftarrow \mathsf{ReRand}(c_k, r'_k)$.
10. Print $\langle c'_k, (c_k, r'_k) \rangle$.

Each voter should:

11. Receive for each k: $\langle c'_k, (c_k, r'_k) \rangle$.
12. Optionally, rewind to line 7.
13. Choose $s \leftarrow [1, \alpha]$.
14. Erase (c_s, r'_s).
15. Send s to R.

Registrar should:

16. Receive s.
17. Publish $\langle \texttt{VoterID}, c'_s \rangle$ on the Roster.

Each voter should:

18. After leaving, check that $c'_k \leftarrow \mathsf{ReRand}(c_k, r'_k)$ for all $k \neq s$.
19. Check that received c'_s matches $\langle \texttt{VoterID}, c'_s \rangle$ on the Roster.

Remarks: This protocol is completed *bare-handed* [28] with pre-computations and erasures. The proof of knowledge of an Elgamal plaintext is standard. The cut-and-choose mechanism is a variant of Benaloh's voter initiated auditing [5]. The option to rewind is included to prevent coercion contracts [13].

without her sheet. The computer has a barcode scanner, which the voter uses to submit her α ciphertexts.

The computer will rerandomize each ciphertext and print the value in the first column of the transcript card. Beside this value on the scratch-off surface, it will print the original ciphertext and the randomization used. The voter chooses one password to register: for that password, the voter will erase the original ci-

phertext and randomization by scratching off the appropriate cell.[4] It is assumed the voter cannot memorize or copy the randomization (*e.g.,* it is encoded into a barcode). The voter shreds her preparation sheet and retains the transcript card. The remaining $\alpha - 1$ re-encryptions can be shown to anyone and checked for correctness at home.

3.4 Election Set-up

The Roster is a universal registration. To prepare for an election, entries from the Roster are copied to smaller lists, called ElectionRosters. An ElectionRoster is specific to a particular election, precinct or district. The trustees will also modify the encrypted message in each entry from g^ρ to g_0^ρ, where g_0 is a unique publicly-known generator for that election. This prevents information leakage across elections.

Recall that Roster entries are encrypted with ρ in the exponent: $\{c_1, c_2\} = \{g^r, g^\rho y^r\}$. For each ElectionRoster, each trustee chooses $b_i \leftarrow_r \mathbb{G}_q$. Then each trustee will in turn blind each ciphertext on the ElectionRoster as follows: output g^{b_i}, $c_1^{b_i}$ and $c_2^{b_i}$, and prove knowledge of b_i such that $g, c_1, c_2, g^{b_i}, c_1^{b_i}, c_2^{b_i}$ form a threewise DH-tuple with a NIZKP (*cf.* [10]). The next trustee will repeat the process using the previous trustee's output as input. All outputs are posted to an appendix on the ElectionRoster. Let $b_0 = \prod b_i$ and $g_0 = g^{b_0}$. The blinding sequence re-randomizes each ciphertext from r to $r' = r \cdot b_0$ and changes the encrypted message from g^ρ to g_0^ρ. The public and private key shares are the same. The public value g_0 will be used during the casting protocol.

3.5 Casting

The casting protocol involves a voter $\mathcal{V}_i$ and the election authority. The protocol is described in Algorithm 2. The communication occurs over an anonymous channel. The anonymity is to be built into the voter's client using an anonymous remailer or onion routing technology.

$\mathcal{V}_i$ submits a commitment to her asserted (*i.e.,* real or panic) password, $g_0^{\rho^*}$, and a rerandomization of her entry on the ElectionRoster, c'. If ρ^* matches the ρ encrypted in c', the pre-tallying protocol will ensure the ballot is included in the final result. Otherwise if it does not match, it will be discarded in a way that is unlinkable to the original submission.

$\mathcal{V}_i$ must prove that c' is from the ElectionRoster. Simply including her entry without rerandomizing it reveals that she submitted a vote. To prevent abstention attacks, she instead rerandomizes it, draws an additional $\beta - 1$ entries randomly from the ElectionRoster, and proves in zero-knowledge that c' is a rerandomization of one of these β entries (her entry plus the additional ones). β acts as an anonymity set. Most voters will use a small value of β, however

[4] Under each scratch-off could be a pre-committed code in the form of a barcode, which the voter could scan to prove to the system that she scratched off the correct cell. We leave the details for such an augmented transcript card for future work.

Algorithm 2: Casting Protocol

Participants : Voter $\mathcal{V}_i$ and election authority
Public Input: Encryption parameters g, p, q, election parameter g_0, public key e, ElectionRoster, and anonymity parameter β
Private Input (V_i): Password (either real or panic) $\hat{\rho}^*$

Each voter should:

1. Find c for her `VoterID` from ElectionRoster.
2. Rerandomize c with random r: $c' \leftarrow \mathsf{ReRand}(c, r)$.
3. Randomly select β-1 other c_k from the ElectionRoster.
4. Form set $\mathcal{C} = \{c, c_1, \ldots, c_{\beta-1}\}$ in order of appearance on ElectionRoster.
5. Generate a NIZKP that r rerandomizes 1-out-of-β of $\mathcal{C}$.
   $\pi_1 \leftarrow \text{NIZKP}_{pok}\{(r) : c' = (\mathsf{ReRand}(c, r) \vee \mathsf{ReRand}(c_1, r) \vee \ldots)\}$.
6. Encode asserted password into $\mathbb{Z}_q^*$: $\rho^* \leftarrow \phi(\hat{\rho}^*)$.
7. Commit to ρ^*: $g_0^{\rho^*}$.
8. Complete an NIZKP of knowledge of ρ^*:
   $\pi_2 \leftarrow \text{NIZKP}_{pok}\{(\rho^*) : g_0, g_0^{\rho^*}\}$.
9. Complete a ballot and retain ballot information $\mathbf{B}$.
10. Send $\left\langle g_0^{\rho^*}, c', \mathbf{B}, \pi_1, \pi_2 \right\rangle$ to A.

Authority should:

11. Publish $\left\langle g_0^{\rho^*}, c', \mathbf{B}, \pi_1, \pi_2 \right\rangle$ on AllVotes.

Remarks: Rerandomization proofs are formed with a knowledge of a DDH-tuple proof due to Chaum and Pedersen [10]. 1-out-of-m proofs are due to a heuristic by Cramer, Damgard and Schoenmakers [15]. Proof of knowledge of a discrete log is due to Schnorr [32]. Parameter β represents the voter's anonymity set.

privacy-conscious voters can also (at extra computational cost) cast a stealth vote where β includes all the entries on the ElectionRoster.

Selections is designed to be versatile with different options for capturing and tallying the votes themselves. Thus we leave the information the voter submits with regard to their vote abstractly as $\mathbf{B}$ while only requiring that $\mathbf{B}$ is submittable to a mix-network. For example, $\mathbf{B}$ could be an encryption of the preferred candidate(s) or a tuple of cryptographic counters for each option, accompanied by proofs of validity as appropriate. Note that our coercion-resistance guarantee extends only to the delivery of valid, eligible, and unique $\mathbf{B}$ values, and care should be taken to ensure that tallying these values does not break coercion-resistance.

Each ZKP uses the Fiat-Shamir heuristic to make it non-interactive, and each uses the values $\left\langle g_0^{\rho^*}, c', \mathbf{B} \right\rangle$ in creating the challenge. This prevents an adversary from replaying any of the proofs individually. The submission is posted to an append-only broadcast channel called AllVotes.

If the voter is under coercion, she makes up a panic password and follows the rest of the protocol as specified. She can later cast a stealth vote with her

Algorithm 3: Pre-Tallying Protocol

Participants : Authorized set of trustees $\mathcal{T}_1, \ldots, \mathcal{T}_m$ and election authority
Public Input: AllVotes
Private Input (T_i): Share of private key, d_i
Authority should:
1 For each entry, check π_1 and π_2.
2 Remove all tuples with invalid proofs to form list ProvedVotes
3 Find all entries in ProvedVotes with duplicate values for g_0^ρ.
4 Remove all but the most recent to form list UniqueVotes.
Each participating trustee should:
5 Participate in verifiable mix network for shuffling UniqueVotes.
Note: the initial $g_0^{\rho^*}$ is treated as $c_\rho = \mathsf{Enc}_e(g_0^{\rho^*}, 0)$.
6 Output is AnonUniqueVotes.
Each participating trustee should:
7 **for** *each entry in AnonUniqueVotes* **do**
8 Read entry $\langle c_\rho, c', \mathbf{B} \rangle$.
9 Participate in a plaintext-equality test of c_ρ and c':
$\{\mathtt{T}, \mathtt{F}\} \leftarrow \mathrm{PET}_{d_i}(c_\rho, c')$.
Authority should:
10 Remove all tuples with PET outcome of `False` to form list ValidVotes.
Each participating trustee should:
11 **for** *each entry in ValidVotes* **do**
12 Participate in threshold decryption of $\mathbf{B}$.

Remarks: Various protocols exist for verifiable mix networks. An efficient technique with statistical soundness is randomized partial checking [20]. The plaintext equality test (PET) is due to Juels and Jakobsson [19]. The output of this protocol is the ballot information for unique and registered voters in an order that is unlinkable to the order of submission.

real password. If a voter wants to overwrite a previous vote submitted under password ρ^*, the inclusion of the same $g_0^{\rho^*}$ will indicate in cleartext that it is an overwrite. Therefore, she should use the same β entries from the ElectionRoster as her anonymity set. Also note that the inclusion of the same $g_0^{\rho^*}$ across multiple elections would also be linkable if the value g_0 was not changed in each election.

3.6 Pre-tallying

The pre-tallying protocol involves an authorized subset of the N election trustees. The protocol is described in Algorithm 3. The protocol takes AllVotes and produces a shorter list of only the most recently cast votes for voters that supply the correct, registered password. Checking the validity of each vote is linear in β. For these voters, the list includes just the ballot information, $\mathbf{B}$, in an order that is unlinkable to the order of submission. How this list is further processed to produce a tally is dependent on the voting system our system interfaces with

		Civitas	AFT	Selections
Registration	Registrar	7	9	2α
	Voter	11	10	4α-1
Casting	Voter	10	24	$(2\beta + 9)$
Pre-Tally	Check Proofs	$4V_0$	$20V_0$	$(4\beta + 6)V_0$
	Remove Duplicates	$(1/2)(V_1^2 - V_1)(8T + 1)$	—	—
	Check Removal	$(1/2)(V_1^2 - V_1)(8T + 1)$	—	—
	Mix	$8V_2T + 4RT$	$20V_2T$	$12V_2T$
	Check Mix	$4V_2T + 2RT$	$10V_2T$	$6V_2T$
	Remove Unregistered	$(8A + 1)V_2R$	$(16T + 8)V_2$	$(8T + 1)V_2$
	Check Removal	$(8A + 1)V_2R$	$(16T + 10)V_2$	$(8T + 1)V_2$

Table 1. Comparison of the efficiency of the main protocols in Civitas, AFT, and Selections, measured with modular exponentiations.

(which is why this is called a pre-tally). In a simple case, **B** is an encryption of the voter's selections (with a proof of knowledge) and the final step is jointly-decrypting each **B** from the list.

3.7 Voter Revocation

Between elections, Selections offers a way of choosing which registered voters are eligible or not to vote in a particular election. In Selections, it is also possible to revoke a voter at any point before the pre-tallying protocol. This could arise because the voter forgot their password (and is issued a new one) or registered to vote online but decides to vote in person. For every submitted vote that includes the revoked voter among its β registered voters in its anonymity set (which will include any potentially valid vote by the revoked voter herself), the submitted password is checked against the revoked voter's entry on the ElectionRoster using a plaintext-equality test. Revocation of this type is the same in Civitas and is not possible in AFT. Coercion-resistance does not necessarily extend to all types of revocation.

4 Performance

We compare the performance of Selections to JCJ as implemented in Civitas [14] and to AFT [4]. We make a number of standardizing assumptions to facilitate a better comparison. We assume a single registrar, T trustees, R registered voters, and V_0 submitted votes. We do not use the "blocking" technique of Civitas, which could improve the performance of all three systems. Of the V_0 submitted votes, $V_1 \leq V_0$ have correct proofs, $V_2 \leq V_1$ are not duplicates, and $V_3 \leq V_2$ correspond to registered voters. Recall that for Selections, α are the number of submitted ciphertexts in registration and β is the size of the voter's anonymity set during casting.

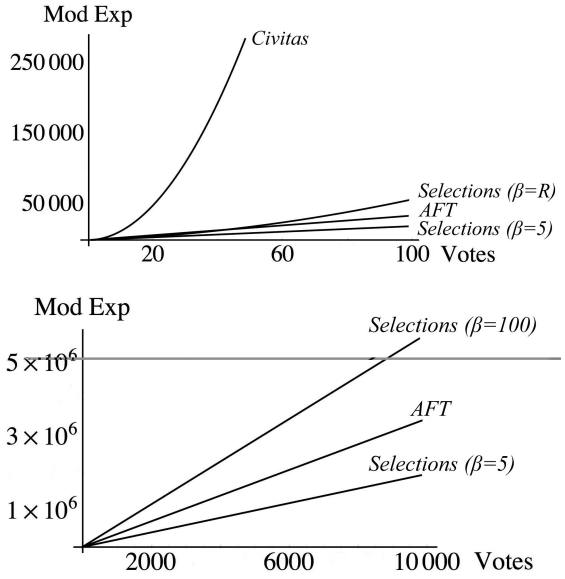

Fig. 1. Pre-tallying efficiency in modular exponentiations with $T = 5$ and variable $R = V_0 = V_1 = V_2$.

We use Elgamal encryption in each system, with proofs of knowledge of plaintexts where appropriate. We assume each trustee participates in decryption (*i.e.,* distributed instead of threshold). We assume that ballot material is encrypted with only a proof of knowledge (no additional proofs of well-formedness). The pre-tallying protocol ends with a list of V_3 encrypted ballots. Finally, we assume mixing is done with a re-encryption mixnet and randomized partial checking [20], where each authority produces two mixes and half of these re-encryptions are checked. The complete details of our comparison are in the full paper.

Table 1 shows the efficiency in terms of modular exponentiations and Figure 4 shows a comparison of the pre-tallying protocols. With full forced-abstention, Selections is quadratic like Civitas but with a smaller constant. When β is a constant, Selections is linear in the number of submitted votes like AFT. The exact value of β dictates which is exactly faster. Recall our goal was not to improve the efficiency of AFT but rather to create a password-based system with similar performance to AFT. To this end, we are successful.

5 Security Analysis (Abstract)

5.1 Soundness of Registration

In the full paper,[5] we show that the Registration protocol is a cut-and-choose argument for $\{(c, r) : c' = \mathsf{ReRand}_e(c, r)\}$. It takes soundness parameter α (*e.g.,*

[5] `http://www.cs.uwaterloo.ca/~j5clark/papers/2011_fc.pdf`

$\alpha = 10$). It is complete and has statistical soundness of $1 - \alpha^{-1}$ for a single run. After k runs, soundness increases to $1-\alpha^{-k}$. Designing a bare-handed argument with stronger soundness (*e.g.,* $1 - 2^{-\alpha}$ for a single run) is open. With erasures, the protocol has deniability for c and computational secrecy for r.

The protocol does not protect against covert channels. This has been addressed in the literature with verifiable random functions [17] or pre-committed randomness [16]. The protocol protects against coercion contracts [13] with rewinds. Rewinds can be eliminated if the voter commits to their choice of password at the beginning of the protocol.

5.2 Coercion-Resistance

In the full paper,[6] we show several results concerning the coercion-resistance (cr) of Selections. Juels *et al.* define an experiment $\mathbf{Exp}^{\text{cr}}_{ES,\mathcal{A}}$ for non-adaptive adversary $\mathcal{A}$ in election system ES, as well as an ideal $\mathbf{Exp}^{\text{cr}-\text{ideal}}_{ES,\mathcal{A}}$. The critical component in $\mathbf{Exp}^{\text{cr}}_{ES,\mathcal{A}}$ is a coin flip $b \leftarrow_r \{0,1\}$ defining a corrupted voter's behaviour. If $b = 0$, the voter provides (in Selections) a panic password to the adversary and casts a vote with her real password. If $b = 1$, the voter complies with the adversary and provides her real password. In both cases, the adversary can use the supplied password to submit a vote. We define the advantage of $\mathcal{A}$, where an output of 1 is the adversary correctly stating b, as,

$$\mathbf{adv}^{\text{cr}}_{ES,\mathcal{A}} = |\mathbf{Pr}[\mathbf{Exp}^{\text{cr}}_{ES,\mathcal{A}}(\cdot) = 1] - \mathbf{Pr}[\mathbf{Exp}^{\text{cr}-\text{ideal}}_{ES,\mathcal{A}}(\cdot) = 1]|.$$

Case 1: $\beta = R$. We show that when β is the full roster R, $\mathbf{adv}^{\text{cr}}_{ES,\mathcal{A}}$ for Selections is negligible. Setting $\beta = R$ does impact performance. Vote casting is linear in the size of the ElectionRoster and Pre-Tallying is quadratic. However the only quadratic component is checking the 1-out-of-β rerandomization proof, where the proof length is linear in the size of the roster. These proofs can be pre-checked, while voters submit votes.

Case 2: $\beta = \text{const}$. We show that when β is constant (*e.g.,* 5 or 100), $\mathbf{adv}^{\text{cr}}_{ES,\mathcal{A}} < \delta$, where δ is small but non-negligible. Recall there are V_2 votes with valid proofs and R entries on the ElectionRoster. Let $\mathbf{F}(k; p, n)$ be the cumulative distribution function of a Binomial distribution with n trials, success probability p, and k successes. We show that δ for this case is,

$$\delta = \frac{1}{2}(F(\frac{\beta V_2}{R}; V_2, \frac{\beta}{R}) + 1 - F(\frac{\beta V_2}{R} - 1; V_2 - 1, \frac{\beta}{R})).$$

Case 3: $\beta \geq \text{const}$. Finally we consider the case where β is required to be at least a constant value (*e.g.,* 5 or 100) but voters can submit stealth votes where $\beta = R$. We show that if a corrupted voter's coercion-resistant strategy is to submit their real vote as a stealth vote, $\mathbf{adv}^{\text{cr}}_{ES,\mathcal{A}}$ is negligible. We do make one small change

[6] `http://www.cs.uwaterloo.ca/~j5clark/papers/2011_fc.pdf`

to $\mathbf{Exp}^{\mathrm{cr}}_{ES,\mathcal{A}}$: instead of the corrupted voter's real vote being appended to the cast ballots, it is inserted at a random place (*i.e.,* she votes her real ballot at some arbitrary time after being coerced).

6 Concluding Remarks

Selections has many benefits: users can evade coercion without computations, registration does not require a computer, tallying the votes is linear in the number of voters, and voters can have their registration efficiently revoked. Future work includes providing protection against untrusted platforms, perhaps by merging Selections with existing work on code voting.

7 Acknowledgements

We acknowledge Richard Carback for suggesting that panic passwords could be employed in an Internet voting system to prevent undue influence. We thank Aleks Essex, Michael Clarkson, various participants of SecVote 2010, and the reviewers for useful feedback on the protocol and paper. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)—the first author through a Canada Graduate Scholarship and the second through a Discovery Grant.

References

1. Acquisti, A.: Receipt-free homomorphic elections and write-in ballots. Tech. rep., IACR Eprint Report 2004/105 (2004)
2. Adida, B.: Helios: web-based open-audit voting. In: USENIX Security Symposium (2008)
3. Adida, B., de Marneffe, O., Pereira, O., Quisquater, J.J.: Electing a university president using open-audit voting. In: EVT (2009)
4. Araujo, R., Foulle, S., Traore, J.: A practical and secure coercion-resistant scheme for remote elections. In: Frontiers of Electronic Voting (2007)
5. Benaloh, J.: Simple verifiable elections. In: EVT (2006)
6. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: ACM STOC (1994)
7. Carback, R.T., Chaum, D., Clark, J., Conway, J., Essex, A., Hernson, P.S., Mayberry, T., Popoveniuc, S., Rivest, R.L., Shen, E., Sherman, A.T., Vora, P.L.: Scantegrity II election at takoma park. In: USENIX Security Symposium (2010)
8. Chaum, D.: Surevote: Technical overview. In: WOTE (2001)
9. Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R.L., Ryan, P.Y.A., Shen, E., Sherman, A.T.: Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In: EVT (2008)
10. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: CRYPTO (1992)
11. Chaum, D., Ryan, P.Y.A., Schneider, S.: A practical voter-verifiable election scheme. In: ESORICS (2005)

12. Clark, J., Hengartner, U.: Panic passwords: Authenticating under duress. In: Usenix HotSec (2008)
13. Clark, J., Hengartner, U., Larson, K.: Not-so-hidden information: optimal contracts for undue influence in E2E voting systems. In: VOTE-ID (2009)
14. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: IEEE Symposium on Security and Privacy (2008)
15. Cramer, R., Damgard, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: CRYPTO (1994)
16. Feldman, A.J., Benaloh, J.: On subliminal channels in encrypt-on-cast voting systems. In: EVT (2009)
17. Gardner, R.W., Garera, S., Rubin, A.D.: Coercion resistant end-to-end voting. In: Financial Cryptography (2009)
18. Heiberg, S., Lipmaa, H., van Laenen, F.: On e-vote integrity in the case of malicious voter computers. In: ESORICS (2010)
19. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In: ASIACRYPT (2000)
20. Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In: USENIX Security Symposium (2002)
21. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: EUROCRYPT (1996)
22. Jakobsson, M., Stolterman, E., Wetzel, S., Yang, L.: Love and authentication. In: CHI (2008)
23. Joaquim, R., Ribeiro, C.: Codevoting: protection against automatic vote manipulation in an uncontrolled environment. In: VOTE-ID (2007)
24. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: WPES (2005)
25. Kane, C.: Voting and verifiability: interview with Ron Rivest. RSA Vantage Magazine 7(1) (2010)
26. Lipmaa, H.: On the CCA1-security of Elgamal and Damgard's Elgamal. In: Inscrypt (2010)
27. Pedersen, T.P.: A threshold cryptosystem without a trusted party. In: EUROCRYPT (1991)
28. Riva, B., Ta-Shma, A.: Bare-handed electronic voting with pre-processing. In: EVT (2007)
29. Ryan, P.Y.A., Bismark, D., Heather, J., Schneider, S., Xia, Z.: Pret a voter: a voter-verifiable voting system. IEEE TIFS 4(4) (2009)
30. Ryan, P.Y.A., Teague, V.: Pretty good democracy. In: Workshop on Security Protocols (2009)
31. Sampigethaya, K., Poovendran, R.: A framework and taxonomy for comparison of electronic voting schemes. Computers & Security 25 (2006)
32. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptography 4 (1991)
33. Smith, W.D.: New cryptographic election protocol with best-known theoretical properties. In: Frontiers in Electronic Elections (2005)
34. Weber, S.G., dos Santos Araujo, R.S., Buchmann, J.: On coercion-resistant electronic elections with linear work. In: ARES (2007)