A Coercion-free Voting Scheme

Technical Field

This application is directed to the technical field of security measures for electronically-conducted elections.

Background

Various electronic and/or digital election protocols exist that provide cryptographic privacy to voters. With many of these election protocols, the voter needs to keep certain types of information secret. An example of such secret information is a voter's private key. These existing election protocols can be problematic, however, if a person threatens, or entices a voter (e.g., financially) to give up the secret information. When this type of coercion occurs, it is possible for the person to either know how the voter voted, or vote on his or her behalf.

Similar problems arise with the use of absentee vote-by-mail systems. For example, a husband might force his wife to vote a certain way. The threat of coercion intensifies in a networked world, where people can "look over each other's shoulders" from thousands of miles away. This threat is serious enough that it is often considered a reason to not allow remote electronic voting.

Under threat models that do not include coercion, the notion of a universally verifiable election is fundamental. In the past, it has been considered important that a "computing device based" election scheme be universally verifiable in order to be considered deployable on a wide scale. During elections of this type, transcripts are published that include the final tally. Under reasonable assumptions about the safety of keys, and the intractability of some computational problems, these transcripts cannot be feasibly forged by any collection of malicious agents. Although it would be desirable to carry this property over to election schemes under the threat of coercion,

this may be difficult. Election schemes under the threat of coercion lack certain very basic properties, which have generally been taken for granted in the election protocol literature, and hence may not be practical in a large-scale implementation.

Brief Description of the Drawings

Figure 1 is a block diagram showing a suitable environment for implementing the scheme.

Figure 2 is a flow diagram showing steps typically performed in accordance with the scheme.

Description

The scheme described herein allows the voter to remain in exclusive possession of secret information that is used by a voter to cast a ballot. It allows a voter that has been pushed to reveal secret information to provide a false answer without being discovered. After providing the false answer, the voter can then proceed and cast a "real" vote on his or her own. This is achieved while still maintaining a collection of election audit properties that are characteristic of good electronic election protocols. An election scheme is coercion safe if, even in the coercion threat model, its transcript can not be feasibly forged by any collusion of authorities that, together, are unable to compute a tally. Further, in the case of a collusion that is able to compute a tally, the extent of the forgery is limited by the number of voters coerced.

At a summary level, the invention works as follows:

1. Voters participate in a secret "voter registration" process in prior to the start of the election. This process must make the voter safe from coercion by standard physical means. In practice, this means the voter must report to a county registration center, where physical privacy is guaranteed. However, the voter need only participate in this registration process once. Thereafter, the method of this invention will protect the voter against coercion through the course of multiple elections.

2. During the registration process, each voter selects a secret "confirmation code," or "confirmation pass phrase."

3. The "confirmation pass phrase" is encrypted by the voter and the encrypted form is publicly registered to that voter.

4. In order to cast a ballot, each voter must supply an accompanying (encrypted) pass phrase. The accompanying pass phrase does not have *any* effect on whether the ballot is "accepted" or not – so if the voter is being "supervised" by a coercer, the voter is still free to supply any pass phrase whether it matches the voter's registered pass phrase or not. The coercer will not be able to tell the difference. However, the accompanying pass phrase *will* have an effect on whether the ballot it accompanies is counted or not. The mechanism for this (described next) nevertheless assures that

   (a) Anyone, including the coercer, can inspect the ballot box contents and the tally to determine whether the tally is accurate or not (i.e. the election is *Universally Verifiable*).

   (b) In spite of the full availability of election data, the encryption and count mechanisms ensure that the coercer will still not be able to determine what vote, if any cast by the voter is actually included in the count.

5. The tabulation (counting) of encrypted votes is accomplished roughly by randomly mixing voted ballot – encrypted pass phrase pairs as well as the original registration data. After randomization, the appropriate data is decrypted by election authorities holding shares of the encryption key. Only when a match between a pass phrase in the randomized ballot data matches a pass phrase in the randomized registration data is the ballot counted. The matching is done without ever decrypting either of the pass phrases. Since all the randomization is done by way of a cryptographic *verifiable shuffle*, the results can still be inspected and verified by anyone for accuracy.

Figure 1 and the following discussion provide a brief, general description of a suitable computing environment in which aspects of the invention can be implemented. Although not required, aspects and embodiments of the invention will be described in the general context of computer-executable instructions, such as routines executed by a general-purpose computer, e.g., a server or personal computer. Those skilled in the relevant art will appreciate that the invention can be practiced with other computer system configurations, including Internet appliances, hand-held devices, wearable computers, cellular or mobile phones, multi-processor systems, microprocessor-based or programmable consumer electronics, set-top boxes, network PCs, mini-computers, mainframe computers and the like. The invention can be

embodied in a special purpose computer or data processor that is specifically programmed, configured or constructed to perform one or more of the computer-executable instructions explained in detail below. Indeed, the term "computer", as used generally herein, refers to any of the above devices, as well as any data processor.

The invention can also be practiced in distributed computing environments, where tasks or modules are performed by remote processing devices, which are linked through a communications network, such as a Local Area Network ("LAN"), Wide Area Network ("WAN") or the Internet. In a distributed computing environment, program modules or sub-routines may be located in both local and remote memory storage devices. Aspects of the invention described below may be stored or distributed on computer-readable media, including magnetic and optically readable and removable computer discs, stored as firmware in chips (e.g., EEPROM chips), as well as distributed electronically over the Internet or over other networks (including wireless networks). Those skilled in the relevant art will recognize that portions of the invention may reside on a server computer, while corresponding portions reside on a client computer. Data structures and transmission of data particular to aspects of the invention are also encompassed within the scope of the invention.

Referring to Figure 1, one embodiment of the invention employs a computer 100, such as a personal computer or workstation, having one or more processors 101 coupled to one or more user input devices 102 and data storage devices 104. The computer is also coupled to at least one output device such as a display device 106 and one or more optional additional output devices 108 (e.g., printer, plotter, speakers, tactile or olfactory output devices, etc.). The computer may be coupled to external computers, such as via an optional network connection 110, a wireless transceiver 112, or both.

The input devices 102 may include a keyboard and/or a pointing device such as a mouse. Other input devices are possible such as a microphone, joystick, pen, game pad, scanner, digital camera, video camera, and the like. The data storage devices 104 may include any type of computer-readable media that can store data accessible by the computer 100, such as magnetic hard and floppy disk drives, optical disk drives, magnetic cassettes, tape drives, flash memory cards, digital video disks (DVDs), Bernoulli cartridges, RAMs, ROMs, smart cards, etc. Indeed, any medium for storing or transmitting computer-readable instructions and data may be employed, including a connection port to a network such as a local area network (LAN), wide area network (WAN) or the Internet (not shown in Figure 1). Aspects of the invention may also be practiced in a variety of other computing

environments.

Figure 2 is a flow diagram showing steps typically performed in accordance with the scheme. These steps are described in more detail below. In step 201, voters are registered to add them to the list of registered voters eligible to cast votes, and to provide them with voting credentials. In step 202, the election is initialized to assign ballot choice values to candidates. In step 203, voters cast their votes by submitting encrypted ballots. In step 204, the votes cast in step 203 are tabulated, and added to the vote total only if the validity of the received ballot can be verified. After step 204, these steps conclude.

1 Coercion Implications of Partitionable Tabulation

The purpose of this section is to

1. Characterize a class of election schemes that includes the vast majority of schemes previously studied, and also seems likely to include all schemes that are "practical" for large scale, public elections.

2. Establish some bounds on what can be achieved by schemes in this class under the coercion threat model.

Definition 1 Henceforth, we call any participant in the election process, or any individual who exerts, or attempts to exert, an influence on the election process a *player*. Thus voters, election officials, and tabulators are all players, but so are all individuals who seek to influence the election outcome even though they may have no official role in it.

Definition 2 Player P_1 *coerces* player P_2 if P_1 obtains from P_2 any information that the election protocol does not require P_2 to reveal to P_1. Identical terminology is used when the coercer is actually a *group of players*. That is, no aspects of the invention limit its utility to the case were the coercer is a single individual. Therefore, henceforth, we will not endevor to make an explicit distinction between coercion by an individual and coercion by a group of individuals acting together.

Definition 3 *Coercible information* is all information whose authenticity can be "verified" by the coercer. If the authenticity can not be verified, then the voter (or individual being coerced) is free to lie about it to the coercer.

Definition 4 Recall that a *tally* is a function $\mathbf{t} : C \rightarrow \mathbf{N} = \mathbf{Z}^{+} \cup \{0\}$, where $\Gamma = \{c_1, \ldots, c_l\}$ is the "candidate slate". We write

$$|\,\mathbf{t}\,| = \sum_{i=1}^{l} \mathbf{t}(c_i)$$

The invention requires something roughly like *digital ballot box.* At very least, this is a storage device connected to a network, or otherwise openly accessible to voters. A standard web server and database application provides an embodiment of such a device. In practice, more security measures would be built into, or around this device in order to protect against damage or destruction caused by either malicious or natural forces. The invention also requires that voters be able to translate their choices into a digital representation, and further encrypt that representation by the methods presented in the remainder of this invention. A generic PC provides an embodiment of such a device.

Definition 5 Since the transmission and storage of information are the key elements of this invention rather than the particular technologies that facilitate transmission and storage, we will adopt the more generic term *bulletin board* to denote the openly accessible storage device, and we denote the act of recording information on (or in) the bulletin board as *posting.* (In the voting context, this corresponds, intuitively, to the act of "casting a ballot".) Further, we denote the strings, or records of information that are posted to the bulletin board as *posts.* (In the voting context, these correspond, intuitively, to voted ballots.)

Let us now consider a set of very general properties that characterize a broad class of election protocols. The properties are considered *in the absence of coercion*. That is, in verifying a given property with respect to a particular protocol, we consider all potential protocol executions where *the only information exchanged between players is that which is specified by the protocol.* (We number these properties sequentially as **PP-1**, **PP-2**, etc.)

PP-1. Posts are always appended to the bulletin board, $\mathcal{BB}$, that is, deletions are not allowed. And posting is an atomic transaction, that is, at any given time, $\mathcal{BB}$ will contain exactly k posts, for some non-negative integer k.

PP-2. Any player may append a post regardless of the state (contents) of $\mathcal{BB}$.

PP-3. At any given time, a tally can be formed, and it is unique. That is, it is not possible (or at least "overwhelmingly improbable"), that $\mathcal{BB}$ is in some state, $\mathbf{C}(\mathcal{BB})$ that is "invalid" for tabulation, and the tally, $\mathsf{tally}(\mathbf{C}(\mathcal{BB})) : C \rightarrow \mathbf{N}$ is well defined.

PP-4. A collection of players either can or cannot compute the tally independent of the state of $\mathcal{BB}$.

Recall that the *voter role*, $\mathcal{O}$, is essentially a public list of players (eligible voters), $\{v_1, \ldots, v_k\}$. Also, we use $\mathbf{C}(\mathcal{BB})$ to denote the contents of $\mathcal{BB}$ at an arbitrary time, that is, the sequence of posts $p_1, \ldots, p_t$. Let $\mathcal{P}$ be the the set of all players in the protocol, so $\mathcal{O} \subset \mathcal{P}$.

For simplicity of presentation, we assume that the ballot consists of one issue, that the candidate slate, Γ, is given by $\{c_1, \ldots, c_l\}$, and that each voter is allowed to choose (vote for) *"at most one"* candidate. Generalizing this setting to one that includes more general ballot types (that do not include "write-ins") is fairly straightforward.

Definition 6 Let $C = \mathbf{C}(\mathcal{BB})$ be any state of $\mathcal{BB}$ (sequence of posts). If ρ is a post, we denote by $C \oplus \rho$ the state of $\mathcal{BB}$ after appending the single post ρ. We also use the notation $\mathbf{t}_C$ to denote the tally, $\mathsf{tally}(C)$.

Definition 7 A *vote function* (on $\mathcal{BB}$) is a map

$$\chi : \mathcal{P} \times \mathbf{C}(\mathcal{BB}) \rightarrow \{0,1\}^{\Gamma} \tag{1}$$

characterized by the following

vf-1. For all $p \in \mathcal{P}$

$$|\, \chi(p, \mathbf{C}(\mathcal{BB}))\, | \in \{0,1\} \tag{2}$$

vf-2. For all $\mathbf{C}(\mathcal{BB})$, if $p \notin \mathcal{O}$, then (with "overwhelming probability")

$$\chi(p, \mathbf{C}(\mathcal{BB})) \;=\; \mathbf{0} \tag{3}$$

Intuitively, this says that the protocol *"only allows members of the voter role (eligible voters) to vote"*.

vf-3. For all $p \in \mathcal{P}$, if p posts ρ, then the following holds (with "overwhelming probability") for all $q \in \mathcal{P}$,

$$\chi(q, \mathbf{C}(\mathcal{BB}) \oplus \rho) - \chi(q, \mathbf{C}(\mathcal{BB})) = \begin{cases} \mathbf{t}_{\mathbf{C}(\mathcal{BB})\oplus\rho} - \mathbf{t}_{\mathbf{C}(\mathcal{BB})} & \text{if } q = p \\ \mathbf{0} & \text{if } q \neq p \end{cases} \tag{4}$$

Intuitively, this says that the protocol *"only allows a voter to vote on his own behalf"*. It rules out schemes that allow multiple voters to combine their votes into one or more posts.

vf-4. For all $1 \leq i \leq l$, and all $1 \leq j \leq k$, if $|\chi(v_i, \mathbf{C}(\mathcal{BB}))| = 0$, then v_i can compute (with probability 1) a post ρ such that

$$\mathbf{t}_{\mathbf{C}(\mathcal{BB})\oplus\rho}(\pi) - \mathbf{t}_{\mathbf{C}(\mathcal{BB})}(\pi) = \begin{cases} 1 & \text{if } \pi = j \\ 0 & \text{if } \pi \neq j \end{cases} \tag{5}$$

Intuitively, this simply says that if v_i has *"not yet voted"*, then v_i can append a "vote" for any candidate. However, the statement does not preclude the possibility that the protocol may allow v_i to "cast a vote" and then later "change it". (Nevertheless, the majority of protocols in the literature, which essentially allow each voter "one and only one chance to vote", do satisfy this criteria.)

vf-5. For all $1 \leq i \leq l$, if $|\chi(v_i, \mathbf{C}(\mathcal{BB}))| = 1$, then v_i can *with at most negligible probability* compute a post ρ satisfying

$$|\mathbf{t}_{\mathbf{C}(\mathcal{BB})\oplus\rho}| > |\mathbf{t}_{\mathbf{C}(\mathcal{BB})}| \tag{6}$$

Intuitively, this simply says that no voter may "vote more than once". Again, however, the statement does not preclude the possibility that the protocol may allow a voter to "change a vote" or to "retract a vote". (As before, the majority of protocols in the literature satisfy this criteria.)

Let A_{ij} be the event that v_i computes a post, ρ, satisfying

$$\mathbf{t}_{\mathbf{C}(\mathcal{BB})\oplus\rho}(c_j) - \mathbf{t}_{\mathbf{C}(\mathcal{BB})}(c_j) = -1 \tag{7}$$

Let B_{ij} be the event that $\chi(v_i, \mathbf{C}(\mathcal{BB}))(c_j) = 1$.

vf-6. There is a constant, α ($0 \leq \alpha \leq 1$) such that, for all $1 \leq i \leq l$, and all $1 \leq j \leq k$, the conditional probability, $\mathbf{P}(A_{ij}|B_{ij})$ satisfies

$$\mathbf{P}(A_{ij}|B_{ij}) = \alpha \tag{8}$$

independent of the values of i, j, and the state of the bulletin board, $\mathbf{C}(\mathcal{BB})$.

Intuitively, this says that *if* the protocol allows *"a voter to change a vote at some time"* then the protocol allows *"any voter to change a vote at any time"*. However, it does not preclude the protocol from forbidding vote changes, which is more common in the literature.

vf-7. For all $1 \leq i \leq l$, and all $1 \leq j \neq \eta \leq k$, the conditional probability, $\mathbf{P}(A_{ij}|B_{i\eta})$ satisfies

$$\mathbf{P}(A_{ij}|B_{i\eta}) \leq \epsilon \tag{9}$$

where $\epsilon \geq 0$ is negligible.

Intuitively, this says that the protocol only allows *"a voter to reduce the count for a candidate"* if *"that voter has voted for that candidate"*. Again, this does not preclude the protocol from forbidding vote changes.

PP-5. The protocol admits a vote function. (Note that this does not require that the vote function be computable by any of the players, only that it exist.)

Definition 8 An election protocol is said to have *partitionable tabulation* if and only if it satisfies **PP-1**–**PP-5**. For brevity, we will also use the term *partitionable election protocol* to describe any election protocol has partitionable tabulation.

Theorem 1 *If an election protocol has partitionable tabulation, and a coercer contains a collection of players capable of computing a tally, then for any $1 \leq i \leq l$, the value of $\chi(v_i, \mathbf{C}(\mathcal{BB}))$ is coercible.*

Proof: (Sketch) The coercer can step through the sequence of ballot box images, at each point computing the tally (see assumption **PP-4**) and requiring v_i to "add a vote" of a particular value. By re-computing the tally with v_i's post appended, the coercer can determine which posts were added by v_i and their cumulative effect on the tally.

Note that this presumes a model in which "after the fact" coercion is allowed. That is, the coercer may interact with the voter after the bulletin board has been closed. However, this assumption can be eliminated with a reasonable assumption on the computing power of voters. In particular, we can show that the coercer is able, by way of a single coercion event, to

1. Impersonate the voter during the course of the election – thereby "adding any chosen vote to the bulletin board (ballot box)", and consequently forging "part" of the election transcript.

2. Detect any attempts by the voter to independently change the vote.

Definition 9 A partitionable election protocol is *coercion resistant* if, under the assumption that there is no coercer capable of independently computing a tally:

CS-1. If $p \in \mathcal{P}$ and $v_i \in \mathcal{O}$, $v_i \neq p$, then p cannot compute $\chi(v_i, \mathbf{C}(\mathcal{BB}))$ with probability higher than "random guess $+\epsilon$".

CS-2. The election results are publicly verifiable.

Definition 10 A partitionable election protocol is *coercion safe* if, it is coercion resistant and, under all collusion scenarios,

CS-3. If t_I is the "ideal tally", then verification of the election guarantees

$$|t_{\mathbf{C}(\mathcal{BB})} - t_I| \leq M \tag{10}$$

2 A Coercion Safe Election Protocol

We assume the standard ElGamal cryptographic setting: p and q are large primes with $q|p-1$. A subgroup generator, $g \in \mathbf{Z}_p^*$ with $|g| = q$, and $h = g^s$ with s shared by a (t, n) threshold scheme among n tabulation authorities, $A_1, \ldots, A_n$.

The protocol we next describe is *coercion resistant.* We will later describe how it can be easily augmented to give a coercion safe protocol. The advantage of describing the weaker version first is that most of the difficulty lies in its construction.

2.1 Registration

Recall that we assume voters are safe from coercion during their registration session. Care must still be taken to assure that information exchanged during registration is not coercible afterwards.

We denote the voter by v_i.

R-1. v_i chooses a random $r_i \in \langle g \rangle$, and a random $\alpha_i \in \mathbf{Z}_q$, and forms

$$(U_{i0}, W_{i0}) = (g^{\alpha_i}, h^{\alpha_i} r_i) \tag{11}$$

R-2. For each $1 \leq j \leq n$

R-2.1. v_i obtains from A_j the pair (U_{ij}, W_{ij}) given by

$$(U_{ij}, W_{ij}) = (g^{\beta_{ij}}, h^{\beta_{ij}}) \tag{12}$$

where $\beta_{ij} \in \langle g \rangle$ is chosen randomly by A_j.

R-2.2. v_i and A_j execute an interactive Chaum-Pedersen proof of validity for the relation $\log_g U_{ij} = \log_h W_{ij}$. That is, the challenge is generated randomly by v_i rather than via a hash function (Fiat-Shamir heuristic). This allows v_i to later produce simulated proofs in the face of coercion.

R-3. After checking each Chaum-Pedersen proof, v_i computes

$$(U_i, W_i) = (\prod_{\mu=0}^{n} U_{i\mu}, \prod_{\mu=0}^{n} W_{i\mu}) \tag{13}$$

R-4. For each $1 \leq j \leq n$, v_i obtains a signature on (U_i, W_i) from A_j as a receipt.

R-5. (U_i, W_i) is added to the voter roll, $\mathcal{O}$. When the registration period ends, each authority should sign $\mathcal{O}$.

Remark 1 As long as v_i knows that one specific authority, A_J, is not a coercer, and fewer than t authorities (the number necessary to compute a tally) are colluding to coerce (though v_i may not explicitly know their identities), the value of r_i is not coercible. This is because v_i can justify the validity of any r_i and α_i by lying to the coercer about the value of (U_{iJ}, V_{iJ}) and presenting a forged (i.e. simulated) Chaum-Pedersen proof.

The requirement that v_i knows a specific honest A_J may be relaxed if we assume that it is acceptable for v_i to be caught lying to the coercer. Alternatively, if $n \gg t$, then v_i can pick an J at random, $1 \leq J \leq n$, assume that A_j is honest, and then know that the chance of being caught lying is at most $(t-1)/n$.

2.2 Election Initialization

EI-1. For each $1 \leq j \leq n$, and for each $1 \leq i \leq l = |\mathcal{O}|$, authority A_j generates randomly and independently a pair of elements in $\langle g \rangle$, (ξ_{ij}, η_{ij}). The quantities

$$(\xi_i\,,\,\eta_i) \;=\; (\prod_{j=1}^{n} \xi_{ij}\,,\,\prod_{j=1}^{n} \eta_{ij}) \tag{14}$$

are publicly computed. These are all published (and signed).

EI-2. The ballot choices $\gamma_\mu \in \langle g \rangle$, $1 \leq \mu \leq k = |\Gamma|$, are assigned by some public random process, or by sharing. (The value γ_μ will indicate a vote for candidate c_μ.)

2.3 Voting

V-1. v_i chooses random $\nu_{i1} \in \mathbf{Z}_q$ and encrypts her vote as the ElGamal pair

$$(A_i, B_i) \;=\; (g^{\nu_{i1}}\,,\,h^{\nu_{i1}}\gamma(i)) \tag{15}$$

V-2. v_i then chooses random $\nu_{i2} \in \mathbf{Z}_q$, computes $s_i = r_i/\gamma(i)$ and encrypts it as

$$(C_i, D_i) \;=\; (g^{\nu_{i2}}\,,\,h^{\nu_{i2}}s_i) \tag{16}$$

V-3. v_i then constructs non-interactive proofs of knowledge, Q_i^{AB} and Q_i^{CD}, for the pairs (A_i, B_i) and (C_i, D_i) respectively. These proofs show that the pairs are of the required form, and that v_i knows the values of s_i and $\gamma(i)$.

V-4. v_i submits the encrypted voted ballot

$$E_i \;=\; \left((A_i, B_i)\,,\,(C_i, D_i)\,,\,Q_i^{AB}\,,\,Q_i^{CD}\right) \tag{17}$$

V-5. Though not necessary in the "append only" bulletin board model, in practice, v_i would be issued a receipt for E_i.

2.4 Tabulation

In this section, we assume a subset of t authorities has been fixed. Without loss of generality, we may assume these are $A_1, \ldots, A_t$.

T-1. For each $1 \leq i \leq l$, the quantities

$$(\bar{U}_i\,,\,\bar{W}_i) \;=\; (\xi_i U_i\,,\,\eta_i W_i) \tag{18}$$

are publicly computed.

T-2. The authorities execute a *verifiable shuffle* of the sequence of pairs of ElGamal pairs, $(\bar{U}_i, \bar{W}_i), (\xi_i, \eta_i)$, resulting in output set of pairs of ElGamal pairs

$$\left\{(\Phi_i\,,\,\Psi_i)\,,\,(\bar{\xi}_i, \bar{\eta}_i)\right\}_{i=1}^{l} \tag{19}$$

where $\Phi_i, \Psi_i, \bar{\xi}_i, \bar{\eta}_i \in \langle g \rangle$. The properties of this mix are that the set of decrypted value pairs, (a_i, b_i) of the output sequence are exactly the same as the set of decrypted value pairs of the input sequence, but in randomly permuted order. Executing such a verifiable shuffle is discussed in greater detail in U.S. Patent Application No. 09/816,869, entitled "VERIFIABLE, SECRET SHUFFLES OF ENCRYPTED DATA, SUCH AS ELGAMAL ENCRYPTED DATA FOR SECURE MULTI-AUTHORITY ELECTIONS," filed March 24, 2001, and PCT Application No. WO02/77929, entitled "VERIFIABLE SECRET SHUFFLES AND THEIR APPLICATION TO ELECTRONIC VOTING," filed March 25, 2002, each of which is hereby incorporated by reference in its entirety.

T-3. Let $\{((A_m, B_m)\,,\,(C_m, D_m))\}_{m=1}^{M}$ be the set resulting from all voted ballots with verified validity proofs. The authorities execute another verifiable shuffle of the sequence of these M ElGamal pair *pairs*, with resulting output set

$$\{((\bar{A}_m, \bar{B}_m)\,,\,(\bar{C}_m, \bar{D}_m))\}_{m=1}^{M} \tag{20}$$

T-4. For each $1 \leq m \leq M$, the l ElGamal pairs

$$(\Theta_{mi}\,,\,\Omega_{mi}) \;=\; (\bar{A}_m\bar{C}_m\bar{\xi}_i\Phi_i^{-1}\,,\,\bar{B}_m\bar{D}_m\bar{\eta}_i\Psi_i^{-1}) \tag{21}$$

$1 \leq i \leq l$ are publicly computed.

T-5. The authorities *jointly* decrypt all of the pairs $(\bar{A}_m, \bar{B}_m)$, and $(\Theta_{mi}, \Omega_{mi})$, $1 \leq i \leq l$, $1 \leq m \leq M$. Let these be, respectively, a_m, and x_{mi}.

T-6. For each $1 \leq m \leq M$, a_m is added to the tally if and only if

T-6.1. $a_m \in \{\mu_1, \ldots, \mu_k\}$

T-6.2. For some $1 \leq i \leq l$, $x_{mi} = 1$.

2.5 Tabulation – Alternate Embodiment

In this section, we assume a subset of t authorities has been fixed. Without loss of generality, we may assume these are $A_1, \ldots, A_t$.

T2-1. For each $1 \leq i \leq l$, the quantities

$$(\bar{U}_i\,,\,\bar{W}_i) \;=\; (\xi_i U_i\,,\,\eta_i W_i) \tag{22}$$

are publicly computed.

T2-2. The authorities execute a *verifiable shuffle* of the sequence of ElGamal pairs, $(\bar{U}_i, \bar{W}_i)$, resulting in output set of ElGamal pairs

$$\{(\Phi_i\,,\,\Psi_i)\}_{i=1}^{l} \tag{23}$$

where $\Phi_i, \Psi_i \in \langle g \rangle$. The properties of this mix are that the set of decrypted values of the output sequence are exactly the same as the set of decrypted values of the input sequence, but in randomly permuted order.

T2-3. For each voted ballot, E_m, $1 \leq m \leq M$, with verified validity proofs, the l ElGamal pairs

$$(\Theta_{mi}\,,\,\Omega_{mi}) \;=\; (A_m C_m \Phi_i\,,\,B_m D_m \Psi_i) \tag{24}$$

are publicly computed.

T2-4. The authorities execute a verifiable shuffle of the sequence of $M \times l$ ElGamal pair *pairs*, $((A_m, B_m), (\Theta_{mi}, \Omega_{mi}))$, resulting in the output set

$$\{((\bar{A}_m, \bar{B}_m), (\bar{\Theta}_{mi}, \bar{\Omega}_{mi}))\}_{m=1,i=1}^{m=M,i=l} \tag{25}$$

T2-5. The authorities jointly decrypt all of the pairs (Φ_i, Ψ_i), $(\bar{A}_m, \bar{B}_m)$, and $(\bar{\Theta}_{mi}, \bar{\Omega}_{mi})$ Let these be, respectively, ϕ_i, a_m, and x_{mi}.

T2-6. For each $1 \leq m \leq M$, a_m is added to the tally if and only if

T2-7. $a_m \in \{\mu_1, \ldots, \mu_k\}$

T2-8. For some $1 \leq i \leq l$ and $1 \leq j \leq l$, $x_{mi} = \phi_j$.

2.6 Making the Protocol Coercion Safe

The protocol, as presented is clearly not coercion safe. If t or more authorities collude, they can decrypt the original voter secrets, r_i, and this allows them to impersonate all the voters. The problem can be fixed by adding an anonymous signature requirement to the ballot casting operation. (See aforementioned patent applications for a detailed description of an anonymous signature protocol that is "authority free".) In this case, even if a malicious agent has access to a secret, r_i, it can not affect the tally without the corresponding private signing key, which can not be obtained without coercion. The reason for this should be clear. An *authority free, anonymous signature* on the voted ballot prevents the authorities (even in collusion) from linking the original encrypted ballot (input to the verifiable shuffle, or mix) to an individual the way they can with a standard digital signature. A standard digital signature explicitly links signed data to a registered individual. An anonymous signature only links signed data to a member of a set, or group, of individuals.

Claims

I claim:

1. A method for conducting a coercion-resistant electronic election, comprising:

receiving from the voter a first voter confirmation value;

after receiving the first voter confirmation value, receiving from the voter an encrypted ballot and a second voter confirmation value;

without regard for the value of the received second voter confirmation value, adding the received ballot to a publicly-available list of cast ballots, such that members of the public are able to verify the addition of the received ballot to the list without being able to determine whether the ballot will be counted; and

counting the ballot if and only if the second voter confirmation value received with the ballot matches the received first voter confirmation value.

2. A computer-readable medium whose contents cause a computing system to conduct a coercion-resistant electronic election by:

receiving from the voter a first voter confirmation value;

after receiving the first voter confirmation value, receiving from the voter an encrypted ballot and a second voter confirmation value;

adding the received ballot to a publicly-available list of cast ballots, such that members of the public are able to verify the addition of the received ballot to the list without being able to determine whether the ballot will be counted; and

counting the ballot if and only if the second voter confirmation value received with the ballot matches the received first voter confirmation value.

3. One or more generated data signals collectively conveying a cast ballot data structure constructed by a voter, comprising:

an encrypted ballot choice; and

a ballot confirmation value,

such that the ballot may be added to a publicly-accessible list of cast ballots without regard for the ballot confirmation value, and

such that it can be determined whether to count the cast ballot based upon whether the ballot confirmation value matches an original ballot confirmation value earlier-stored for the voter.

4. One or more computer memories collectively containing a ballot box data structure storing cast ballots, the ballot box data structure containing, for each stored ballot,

data reflecting a ballot choice selected by a voter who generated the ballot; and

data reflecting a confirmation value selected by the voter,

such that the contents of the data structure can used by a vote tabulation authority to determine whether to count the ballot,

such that the contents of the data structure can used by a third party distinct from the vote tabulation authority to determine that the ballot has been stored in the ballot box, and

such that the contents of the data structure cannot used by the third party to determine whether the ballot will be counted.