David Chaum's Voter Verification using Encrypted Paper Receipts

Poorvi L. Vora
Dept. of Computer Science
George Washington University
Washington DC 20052
poorvi@gwu.edu

February 20, 2005

This document is a modified version of a document with the same title and author, written in early 2004, posted at http://www.seas.gwu.edu/~poorvi/Chaum/chaum.pdf

Abstract

In this document, we provide an exposition of David Chaum's voter verification method that uses encrypted paper receipts. This document provides simply an exposition of the protocol, and does not address any of the proofs covered in Chaum's papers.

1 Introduction

By now, it is probably well-known that the voter-verifiable voting technique of Chaum [1, 2] has several uncommon and useful properties. No detailed expositions of the idea appear to exist outside the above-mentioned attempt by this author, however. It is hoped that this document will fill a need of the larger cryptographic community.

Key among the useful properties of Chaum's technique are:

1. **Voter Verification** A voter may verify that her vote was counted as cast. This is perhaps the only protocol that allows voter verification for write-in ballots.
2. **Involuntary Privacy** No voter can prove to a third party how she voted.
3. **Election Validity** It is not possible to forge a receipt or in any other way falsely call into question the validity of an election.

The count integrity of the protocol is information-theoretic, while its privacy is computational.

The outline of this manuscript is as follows. In section 2, we provide a list of participants, in section 3 a sketch of the protocol in non-technical terms, in section 4 a technical sketch, in section 5, preliminaries, and, in section 6, the protocol itself.

2 Participants

The participants in the protocol are:

1. The *Voter* who may determine that her vote is counted

2. The *Polling Machine* is assumed *untrusted* and is responsible for recording the voter's vote. The system must catch attempts by the Polling Machine to change votes[1].

3. *Trustees* are responsible for ensuring that votes are counted and anonymity maintained beyond the Polling Machine. This role is played out in physical elections by some combination of candidate representatives and government officials, depending on the country.

4. *Interested Third Parties* may verify that the system is working as it should. This role is played out by organizations such as League of Women Voters in physical elections in the US. The method described in this document requires the participation of a non-negligible fraction of voters, or, on their behalf, Interested Third Parties. This participation is the only way to detect attempts by the Polling Machine to change votes.

5. *Auditor* certifies that the election results are correct and have been determined by following the pre-specified, well-defined process. Who the Auditor is depends on who the results are being certified for. In physical elections in the US this role is played by a specified government/judicial official. In physical elections in some other countries, this role is played by a citizen who is not directly answerable to the Executive and hence is more independent of the current office bearers. In physical elections in new democracies, this role is played by organizations like Amnesty International who may also function as Interested Third Parties. In the method described in this document, only a few independent audits are possible. More audits will compromise voter anonymity.

6. The *Public*, represented by the public site that holds all receipts, decrypted receipts, and audit results, and displays them to the public, thus enabling anyone to count the votes and follow the vote verification process.

[1] Proper voter authentication is outside the scope of this paper. Hence, ballot stuffing, false electoral rolls, and the separation between voter and assigned ballot card would have to be addressed through different means. For example, ballot stuffing may be prevented by maintaining an electoral roll and a voter log independent of the Polling Machine. The security of cast ballots is also not discussed, hence other methods need to be used to ensure that the Polling Machine does not retain the entire vote and associate it with a serial number; such methods are also needed in other election schemes.

3 A Non-Technical Sketch of Protocol

Receipt Structure: The Polling Machine prints two overlaid layers, each a random-looking binary image by itself. Together, these two layers provide a visual representation of the vote - the equivalent of a filled-in paper ballot. Each layer is symmetric and contains the same amount of information on the vote: half of the pixels in each layer have been generated using a pseudo-random number generator (PRNG); the other half may be thought of as containing encrypted information on the vote. In addition to the binary image there are three numeric strings at the bottom of each receipt layer, the strings identical on both layers. These strings force the Polling Machine to commit to the seeds used to generate the random pixels, and help detect efforts by the Polling Machine to change votes.

Process in Polling Booth: Once a voter confirms ("casts") her electronic vote, the Polling Machine prints the two overlaid receipt layers. The voter checks that her votes are recorded as cast, and that the three numeric strings are identical on both layers. She then chooses the layer she wishes to take with her as a receipt. The chosen layer is an encrypted visual representation of her vote. The other layer may be thought of as the decryption key, and is destroyed by the Polling Machine (there is no way to ascertain this; however the Polling Machine cannot affect the vote count by retaining the receipt). Before the voter leaves with her receipt, the Polling Machine prints some more information. This information certifies that the receipt is authentic and allows anyone to check that the random pixels on the chosen layer were correctly generated.

Pre-count Check: The Public website displays all collected ballots by serial number. Individual voters or Interested Third Parties may check that particular receipts are among these. The following checks can also be performed for *each displayed receipt* by *anyone*: (a) the pseudo-random numbers on the chosen layer were correctly generated, (b) half of the information encrypted for the Trustees is correct, and (c) that the receipt is legitimate. For confidence in the result, a large enough fraction of the votes cast (preferably all) must be thus checked by the Election Authority to detect attempts by the Polling Machines to manipulate votes. Any anomalies would provoke further checks to determine the extent of the problem (a faulty machine, Polling Machine, District, etc.). With these checks, for each vote checked, the Polling Machine's attempt to change the vote can be detected with probability $\frac{1}{2}$.

Vote Tallying: One of the three strings printed on each receipt layer contains encrypted information for the Trustees to reproduce the original ballot images. The decryption is done sequentially: each Trustee performs his part of the decryption and passes the entire set of images on to the next Trustee after shuffling it. The shuffle prevents the linking of a final decrypted ballot image with a serial number and through that with a particular voter. The final Trustee produces ballots which

are displayed on the website and may be counted by anyone. The set of input and output images for each Trustee are publicly available. Further, all Trustees are required to retain the shuffle for the audit(s).

Audit: A Trustee can affect the vote count in exactly one way: by not decrypting correctly. Through an audit, this may be detected with maximum probability $\frac{1}{k}$ for *each* vote cheated on, $k \geq 2$. The audit involves requiring each Trustee to demonstrate publicly the output image corresponding to specified input images. The specified images are chosen at random, and number a fraction, $\frac{1}{k}$, of the total number of input images. The correspondence between the two images may be checked by anyone using the Trustee's public key. Specified input images for consecutive Trustees are chosen so that no final ballot image can be linked to a serial number, as this would compromise voter anonymity.

On passing an audit, the vote tally can be considered final. If a Trustee fails an audit, procedures need to be in place to either declare a final count in spite of this, or to declare a revote.

4 A Technical Sketch of the Protocol

The vote is represented using two layers; each layer is pseudo-random by itself, but when overlaid the layers form an image of the ballot. Each layer is symmetric wrt the amount of information it holds about the vote itself; one-half of the pixels of each layer are pseudo-randomly generated, and the other half are pixels of the encrypted ballot. The pseudo-random pixels in the layers are staggered so that one of two pixels at a particular position is pseudo-random, the other is the ballot pixel encrypted using the pseudo-random pixel (see Figures 2 and 3).

The pseudo-random numbers are generated by summing the shares from n MIXes; the share of each MIX is generated in a well-defined manner from the serial number of the ballot, signed by the private key of the Polling Machine. The shares can only be generated if the signed value of the serial number is known, and the private keys used are distinct for the top and bottom layers.

The voter chooses one of the layers to take with her; this is also the layer retained by the Polling Machine as the encrypted ballot and is used during counting, it is the *receipt*. Before the voter chooses a layer, however, the Polling Machine commits to three strings that it prints on both layers: x, the serial number of the vote, y, a string bearing the seeds for each MIX for the top layer, and z, a string bearing the seeds for each MIX for the bottom layer. If the voter chooses the top layer, the MIXes will need z to generate the pseudo-random numbers in the bottom layer and thus decrypt half the receipt, and vice versa.

Further, if the voter chooses the top layer, y is used as a commitment to check the correct com-

munication of seeds and generation of pseudo-random numbers corresponding to the top layer. To allow the checking of this commitment, the Machine prints the signed value of the serial number for the top layer after the voter makes the choice.

After the polls, the receipts are broadcast so individual voters may check the presence of the receipts in the batch to be counted, all commitments on all broadcast receipts are checked, and the receipts are put through the MIXes after stripping the serial numbers and other strings that are now not necessary. Each MIX adds its share of the pseudo-random string to the encrypted bits, shuffles the set of receipts and passes them on to the next MIX. At the end, clear-text ballots are tallied. The input and output of all MIXes are broadcast.

An audit of the MIXing, inspired by [3], is performed.

5 Preliminaries

5.0.1 Notation

If K represents a public/private key pair, K_{pub} and K_{priv} represent the public and private keys respectively. The following denote the public key pairs used:

K_i: key pair for the i^{th} MIX, a total of n MIXes, run by $\frac{n}{2}$ or fewer Trustees

k_p: Polling Machine key pair for signing the receipt

s_t (s_b): Polling Machine key pair for generating the pseudo-random sequence embedded in top (bottom) receipt layer

s_c: represents s_t if $c = t$ and s_b if $c = b$

Additional Notation:

q: serial number

$S_K(x)$: public-keyed one-way function on x using public key pair K (e.g.: digital signature of x, or encryption of a specified digest of x using K_{priv})

t, b: top, bottom layers

$c \in \{t, b\}$: a specific one of the two layers

5.1 The receipt

The voter's receipt consists of two layers, each of which has four fields: a binary image $\mathcal{I}$, and three strings: x, y and z. The three strings on both receipts are identical, and the images, when overlaid,

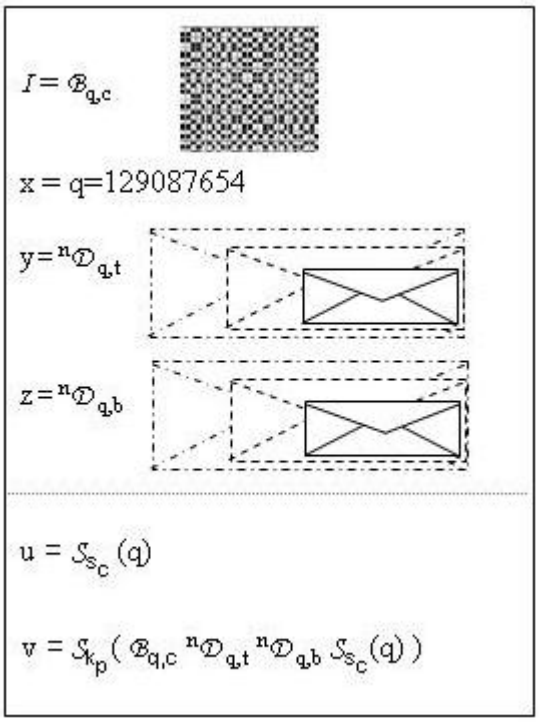

Figure 1: A Ballot Receipt (not to scale)

produce an image representing the vote. Each receipt reads as follows (see Figure 1):

$\mathcal{I} = \mathcal{B}_{q,t}$ ($\mathcal{B}_{q,b}$): the binary image for the top (bottom) layer. In the figure, c represents the chosen layer, $c \in \{t, b\}$.

String $x = q$: the serial number of the receipt

String $y = {}^{n}\mathcal{D}_{q,t}$: the encrypted string containing MIX seeds to generate the pseudo-random sequence in the top layer

String $z = {}^{n}\mathcal{D}_{q,b}$: the encrypted string containing MIX seeds to generate the pseudo-random sequence in the bottom layer

${}^{n}\mathcal{D}_{q,t}$ and ${}^{n}\mathcal{D}_{q,b}$ serve as commitments, one of which can be checked based on the voter's choice of layer. The other serves to communicate the seeds for decryption to the MIXes.

In addition, only the chosen layer, $c \in \{t, b\}$, has printed on it the following, after the Voter chooses a receipt layer:

String $u = S_{s_c}(q)$: The signature of the serial number with the Polling Machine's key for the chosen layer c. This value can be used to check the commitment ${}^{n}\mathcal{D}_{q,c}$ as we describe in section 6.2.

String $v = S_{k_p}(\mathcal{B}_{q,c}, q, {}^{n}\mathcal{D}_{q,t}, {}^{n}\mathcal{D}_{q,b}, S_{s_c}(q))$: : The signature of the entire receipt.

6 The Protocol

The protocol consists of four stages: in the polling booth, receipt broadcast and pre-count verifications, counting, audit.

6.1 Stage I: In the polling booth

Step 1: *Voter defines ballot image.* The voter chooses her candidates using an interface i.e. the voter defines the filled-in ballot, binary image $\mathcal{B}_q$.

Step 2: *Receipt generation.* The Polling Machine generates two binary images $\mathcal{B}_{q,t}$ (top layer) and $\mathcal{B}_{q,b}$ (bottom layer) such that:

(a) $\mathcal{B}_{q,t}$ and $\mathcal{B}_{q,b}$ are of the same size as $\mathcal{B}_q$.

(b) Pixels in alternate rows and columns of each layer are pseudo-random bits and placed so that bits from one layer are staggered by one unit from those of the other (see Figure 2).

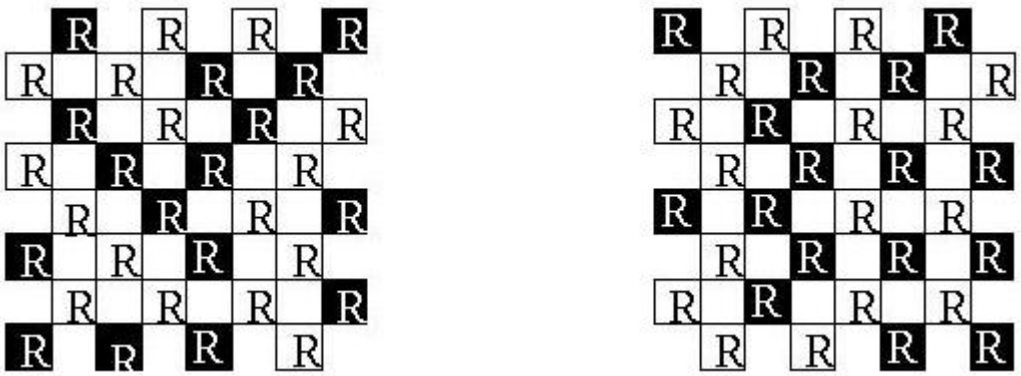

Figure 2: The two receipts, with random pixels (denoted "R") staggered. Black pixels denote "0"s, white ones "1"s, pixels without an "R" are not filled at this stage of the process.

In the image processing literature, an image is often represented as a one-dimensional array formed by stacking the pixels row by row. Let $\mathcal{I}^j$ represent the j^{th} pixel in the one-dimensional array representing image $\mathcal{I}$. WLOG, we may assume that alternate pixels in the arrays $\mathcal{B}_{q,t}$ and $\mathcal{B}_{q,b}$ are pseudo-randomly generated. With this notation,

(i) $\mathcal{B}^j_{q,t}$ is pseudo-randomly generated if and only if $\mathcal{B}^j_{q,b}$ is not and

(ii)$\mathcal{B}^j_{q,c}$ is pseudo-randomly generated if and only if $\mathcal{B}^{j+1}_{q,c}, \mathcal{B}^{j-1}_{q,c}$ are not for $c \in \{t, b\}$.

We denote the set of positions for which pixels in the top and bottom layers are pseudo-randomly generated as $\mathcal{R}_t$ and $\mathcal{R}_b$ respectively. If the image is of size n pixels (each pixel is a bit) then $\mathcal{R}_t$

and $\mathcal{R}_b$ are of size $\frac{n}{2}$ each.

(c) The two pseudo-random bit sequences are hence of size $\frac{n}{2}$ and are computed as sums of shares from each MIX, the share of the i^{th} MIX being $h'(h(i, S_{s_c}(q)))$.

where $S_{s_c}(q)$ is the digital signature of q by the Polling Machine using public key pair S_{s_c}, for $c \in \{t, b\}$ representing the layer. Further, h' and h are public functions. In particular, h is a secure one-way function which provides distinct seeds $h(i, S_{s_c}(q))$ for each MIX so that no MIX may generate another's share. On the other hand, h' is a Pseudo-Random Number Generator (PRNG) that is not necessarily one-way or secure. Its purpose is to expand the information communicated to a MIX, i.e. if it were efficient and convenient to communicate the entire pseudo-random share of the MIX on the receipt, h' would not be necessary.

Hence the two pseudo-random number sequences generated are:

$$\mathcal{W}_{q,c} = \bigoplus_{i=1}^{n} h'(h(i, S_{s_c}(q))); c \in \{t, b\} \tag{1}$$

where $\mathcal{W}_{q,c}$ is the pseudo-random number sequence for layer c.

(d) The other $\frac{n}{2}$ pixels of each layer are computed such that $\mathcal{B}^j_{q,t} \bigoplus \mathcal{B}^j_{q,b} = \mathcal{B}^j_q$.

The pixels in each image $\mathcal{I}$ are divided into two sequences $(\mathcal{I}^j)_{j\in\mathcal{R}_t}$ and $(\mathcal{I}^j)_{j\in\mathcal{R}_b}$, which may be denoted as $\mathcal{I}_{\mathcal{R}_t}$ and $\mathcal{I}_{\mathcal{R}_b}$ in short. For each image, one of these sequences is pseudo-randomly generated. The other is generated using the vote and the pseudo-randomly generated sequence of the other image.

More formally, the non-pseudo-randomly generated pixels for layer c are determined as:

$$(\mathcal{B}^j_{q,c})_{j\in\mathcal{R}_{c'}} = (\mathcal{B}^j_q)_{j\in\mathcal{R}_{c'}} \oplus (\mathcal{B}^j_{q,c'})_{j\in\mathcal{R}_{c'}} = (\mathcal{B}^j_q)_{j\in\mathcal{R}_{c'}} \oplus \sum_{i=1}^{N} h'(h(i, S_{s_{c'}}(q))) = (\mathcal{B}^j_q)_{j\in\mathcal{R}_{c'}} \oplus \mathcal{W}_{q,c'} \tag{2}$$

The two images, when overlaid, provide a pictorial representation of the voter's choices, image $\mathcal{B}_q$, see Figure 3. Image $\mathcal{B}_q$ is formed by a physical process that performs the XOR of the two binary layers, using one of a number of possible methods. The one used in [1, 2] is visual cryptography.

One of the layers will be retained as a record of the vote, by the Voter and the Polling Machine. We denote this as the *chosen* layer, by the letter $c \in \{t, b\}$. The other layer, $c' \in \{t, b\}$, is destroyed, and needs to be reconstructed to decrypt the vote. The protocol ensures that this can be done only with the cooperation of the n MIXes.

The encrypted PRNG seeds for both layers are communicated to the MIXes along with the encrypted vote. The pseudo-random pixels in the destroyed layer can be reconstructed using the seeds. The non-pseudo-random pixels, however, are determined by the vote image, $\mathcal{B}_q$ and hence *cannot*

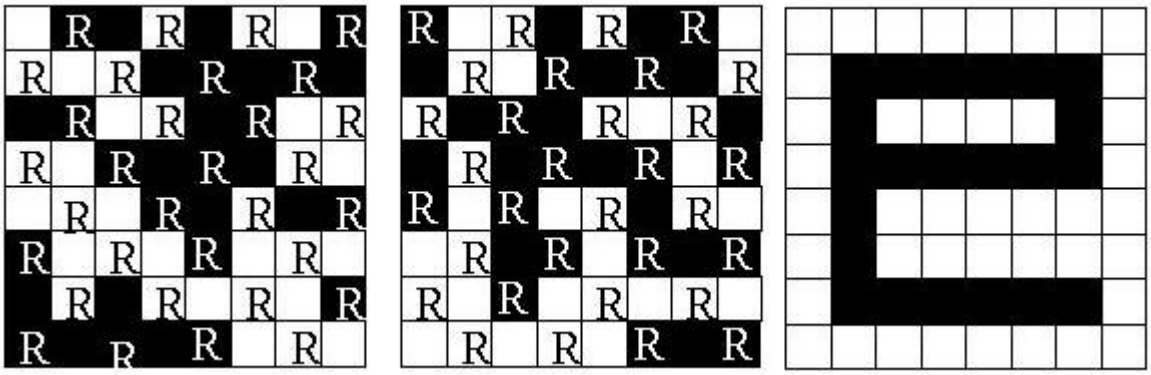

Figure 3: The receipts of Figure 2 with non-pseudo-random pixels filled in to represent the letter "e". An overlay of the two receipts forms the binary ballot image by a physical version of the XOR operation

be reconstructed independently by the Authorities. Assuming the pixels are small enough, however, and the ballots well-designed, reconstructing only the pseudo-random pixels in the destroyed layer is sufficient to determine the ballot.

Step 3: *Communicating the pseudo-random values to the MIXes for decryption* The PRNG seeds for each MIX are encrypted sequentially and printed at the bottom of both layers, along with the serial number, which serves as a commitment that can be checked later.

As the decryption of the votes is done in a MIXnet style, we use the MIXnet analogy of nested envelopes to describe these strings. A string encrypted with a public key is thought of as a sealed envelope. Anyone can seal it, only the entity holding the private key can open it. ${}^{n}\mathcal{D}_{q,t}$ and ${}^{n}\mathcal{D}_{q,b}$ are sealed envelopes for the n^{th} MIX and each contains $n-1$ other nested envelopes for the other MIXes.

The innermost envelopes, ${}^{1}\mathcal{D}_{q,t}$ and ${}^{1}\mathcal{D}_{q,b}$ are generated first and contain the PRNG seed for the first MIX. The i^{th} envelopes contain the $(i-1)^{th}$ envelope and the PRNG seed for the i^{th} MIX (see Figure 4). At decryption time, the envelopes are opened in reverse order, the i^{th} envelope

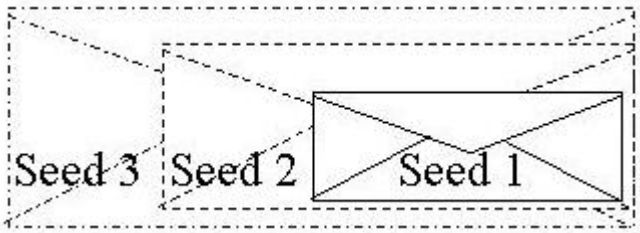

Figure 4: Serially encrypted strings as sealed envelopes

containing, when opened, the $(i-1)^{th}$ envelope and the PRNG seed for the i^{th} MIX. Only the i^{th} MIX can open the i^{th} envelope.

More formally, ${}^{n}\mathcal{D}_{q,t}$ and ${}^{n}\mathcal{D}_{q,b}$ represent the final encrypted strings. ${}^{n}\mathcal{D}_{q,c}$ contains enough information for each MIX to generate its share of the pseudo-random number sequence in layer c. ${}^{n}\mathcal{D}_{q,c}$ is computed recursively as follows:

$$ {}^{1}\mathcal{D}_{q,c} = K_{1_{Pub}}[h(1, S_{s_c}(q))] \tag{3} $$

where c represents t or b. ${}^{i-1}\mathcal{D}_{q,c}$ and the PRNG seed for the i^{th} MIX are encrypted together inside ${}^{i}\mathcal{D}_{q,c}$ which can only be decrypted by MIX i:

$$ {}^{i}\mathcal{D}_{q,c} = K_{i_{Pub}}[h(i, S_{s_c}(q)); {}^{i-1}\mathcal{D}_{q,c}] \tag{4} $$

Step 4: *Voter Check and Choice* The voter checks that the two superimposed layers provide a visual representation of her vote, i.e. that $\mathcal{B}_{q,t} \bigoplus \mathcal{B}_{q,b} = \mathcal{B}_q$. She checks that there are three numbers also printed at the bottom of both layers, and that the numbers are the same on both layers. She chooses a layer to take away, and communicates her choice to the Polling Machine. This step is the only one in which the voter *has* to participate herself to benefit from the voter verification property. She may choose not to, in which case the method is no worse than any other method that does not provide voter verification.

Step 5: *Various checks printed and end of vote casting* After the voter chooses a layer, the seed for the hash *for the chosen layer* is printed. This is a commitment check. Also printed is a signature of the entire receipt to prevent forgery and false accusations of election fraud. The checks are described in more detail in section 6.2

The Polling Machine now prints, only on the chosen layer, digital signatures (a) $S_{s_c}(q)$ - of the serial number, and (b) $S_o(\mathcal{B}_{q,c}, q, {}^{n}\mathcal{D}_{q,t}, {}^{n}\mathcal{D}_{q,b}, S_{s_c}(q))$ - of the entire document, where c represents the chosen layer. It may also print the value of c, i.e. whether the layer chosen was the top or the bottom one, though in the absence of this information, it can be determined from the other information on the receipt. The voter gets the chosen layer, the other is destroyed. Even if the untrusted Polling Machine does not destroy the other layer, it cannot change the vote count.

6.2 Stage II: Receipt broadcast and validity checks prior to counting

Step 6: *Receipt broadcast* Receipts for $q \in \mathcal{Q}$ (where $\mathcal{Q}$ represents the values of q for all cast votes) are displayed in a publicly accessible place. Those with receipts (voters or Interested Third Parties) can check that their receipts (and hence votes) have been correctly retained.

Though individual voters may delegate this task to a trusted representative and do not have to do it themselves, the step itself is not optional for the voter verification property, and a "large enough"

fraction of receipts must be checked for confidence in election results. In the absence of such a check, the count integrity is no worse than in a system without voter verification.

Step 7: *Validity check* Any Interested Third Party can check that all the receipts broadcast were correctly generated. This is done by checking the commitments as follows. Let $< \mathcal{I}, x, y, z, u, v >$ represent a receipt (see Figure 5).

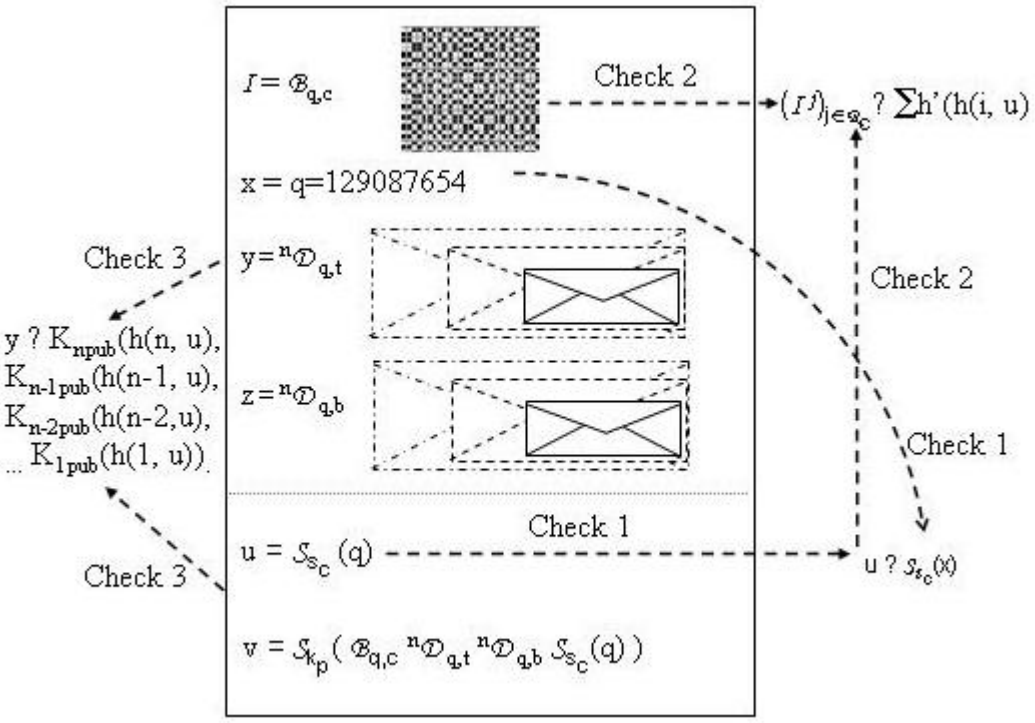

Figure 5: Three ballot checks of broadcast/displayed receipts, to be performed by anyone. This figure illustrates the case when the chosen layer is the top one, i.e. $c = t$.

Check 1:

The public key of the pair s_c should confirm that u checks as the signature of the committed serial number x, i.e. check that $S_{s_c}(x) = u$.

Check 2:

The string u should have been used to generate the committed pseudo-random numbers in the chosen layer as specified by equation (1), i.e. check that $\mathcal{I}_{\mathcal{R}_c} = \sum_{i=1}^{n} h'(h(i, u))$

Check 3:

Of the two envelopes printed on the receipt, one contains encrypted information for the Tallying

Authorities. The other should have been generated as specified by equations (3) and (4), i.e. if

$$^{1}\mathcal{X} = K_{1_{Pub}}(h(1,u))$$

and

$$^{i}\mathcal{X} = K_{i_{pub}}(h(i,u),^{i-1}\mathcal{X}))$$

check that $^{n}\mathcal{X} = y$ if $c = t$, or $^{n}\mathcal{X} = z$ if $c = b$.

Check 4:

The receipt should have been correctly signed, i.e. check that $v = S_{p_k}(\mathcal{I}, x, y, z, u)$

These checks are the only way to detect Polling Machine cheating, and are hence not "optional". Again, a large enough fraction of receipts must pass these checks for confidence in the election results. Typically the Election Authority itself, as well as Independent Third Parties, will run the check for all displayed receipts.

6.3 Stage III: Counting

Step 8: *Vote batch enters MIXes* All strings except the required envelope are stripped off each vote. Also stripped off are the alternate pixel bits that were pseudo-randomly generated, $(\mathcal{B}^{j}_{q,c})_{j\in\mathcal{R}_c}$. Only the pixels representing the voter's choice, $(\mathcal{B}^{j}_{q,c})_{j\in\mathcal{R}_{c'}}$, remain. The entire collection of ballots is then passed on to the n^{th} MIX. For each vote, the n^{th} MIX gets the pixels representing the voter's choice from the voter's chosen layer (receipt) and the entire envelope for the other layer, $^{n}\mathcal{D}_{q,c'}$ (i.e. x or y depending on whether the chosen layer is the bottom - x is then the other envelope for the other layer - or the top -y). Together, the MIXes may then decrypt the vote. Thus the n^{th} MIX gets a sequence of pairs ordered by the corresponding value of $q \in \mathcal{Q}$:

$$(<^{n+1}\mathcal{T} = (\mathcal{B}^{j}_{q,c})_{j\in\mathcal{R}_{c'}}, ^{n}\mathcal{D}_{r,c'} >_r)_{r\in\mathcal{Q}}$$

where the first term is the r^{th} ballot image obtained by the n^{th} MIX, c is the chosen layer for serial number r, and $^{n}\mathcal{D}_{r,c'}$ is y if $c = b$, and z if $c = t$, (c' is the complement layer of c). Though there is no longer a serial number on the ballot, the order of the ballots can reveal information on the original serial number.

The pseudo-random values on the destroyed layer need to be generated to recreate the ballot image. If, for example, $\mathcal{B}_{q,b}$ is taken by the voter, $^{n}\mathcal{D}_{q,t}$ will be opened by the MIXes, and information obtained to compute the pseudo-random numbers in $\mathcal{B}_{q,t}$. This allows for $\mathcal{B}_q$ to be computed[2]. The other envelope, $^{n}\mathcal{D}_{q,b}$, contains the seeds required to generate the pseudo-random numbers in the bottom layer, $\mathcal{B}_{q,b}$. It will serve as a commitment to check that the Polling Machine is indeed

[2]at alternate pixel positions

communicating correctly with the MIXes. This commitment is not checked by the MIXes who do not need the envelope ${}^{n}\mathcal{D}_{q,b}$. It is checked by the Independent Third Parties and voters or their representatives, as described in the previous section.

The pseudo-random numbers in the destroyed layer are generated in parts by each MIX, and added on to the chosen layer sequentially. After the last MIX adds on its part, decrypted ballot images are obtained. These may now be counted, in public. Each MIX shuffles its output images, so the original order is not retained and voting is anonymous. See Figure 6 for a simple explanatory example.

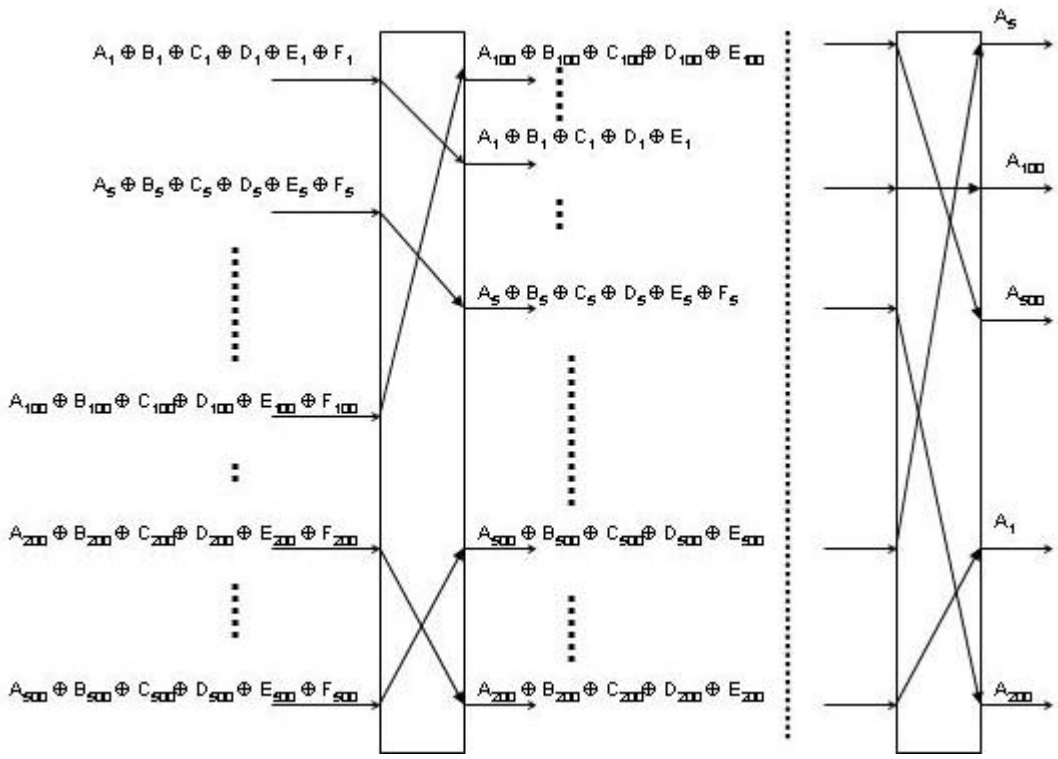

Figure 6: Message A_i is encrypted using a one time pad generated by shares from five MIXes, B_i, C_i, D_i, E_i, and F_i. It is decrypted in reverse order, by each MIX adding its contribution and shuffling the entire batch of messages

We denote by π_i the permutation applied by the i^{th} MIX, and further denote the composition of all permutations from the n^{th} MIX down to the i^{th} one by $\lambda_i = \pi_n \circ \pi_{n-1} \circ ... \circ \pi_i$. Note that the 1^{st} output of the i^{th} MIX will correspond to its $\pi_i^{-1}(1)^{th}$ input. From here onward we drop subscripts denoting position within the brackets $<,>$ and retain these only outside the bracket, WLOG.

The pairs

$$\{<^{i+1}\mathcal{T},^{i}\mathcal{D}_{c'}>_{\lambda_{i-1}^{-1}(1)},<^{i+1}\mathcal{T},^{i}\mathcal{D}_{c'}>_{\lambda_{i-1}^{-1}(2)},....<^{i+1}\mathcal{T},^{i}\mathcal{D}_{c'}>_{\lambda_{i-1}^{-1}(q)},...\}_{q\in\mathcal{Q}} \tag{5}$$

will be the input sequence of ballots for the i^{th} MIX, and

$$\{<^{i}\mathcal{T},^{i-1}\mathcal{D}_{c'}>_{\lambda_i^{-1}(1)},<^{i}\mathcal{T},^{i-1}\mathcal{D}_{c'}>_{\lambda_i^{-1}(2)},....<^{i}\mathcal{T},^{i-1}\mathcal{D}_{c'}>_{\lambda_i^{-1}(q)},...\}_{q\in\mathcal{Q}} \tag{6}$$

the output sequence; see Figure 7. The i^{th} MIX computes each output pair by opening the corresponding envelope passed on by the previous MIX. The envelope contains the seed for the MIX's PRNG share, and the next MIX's envelope.

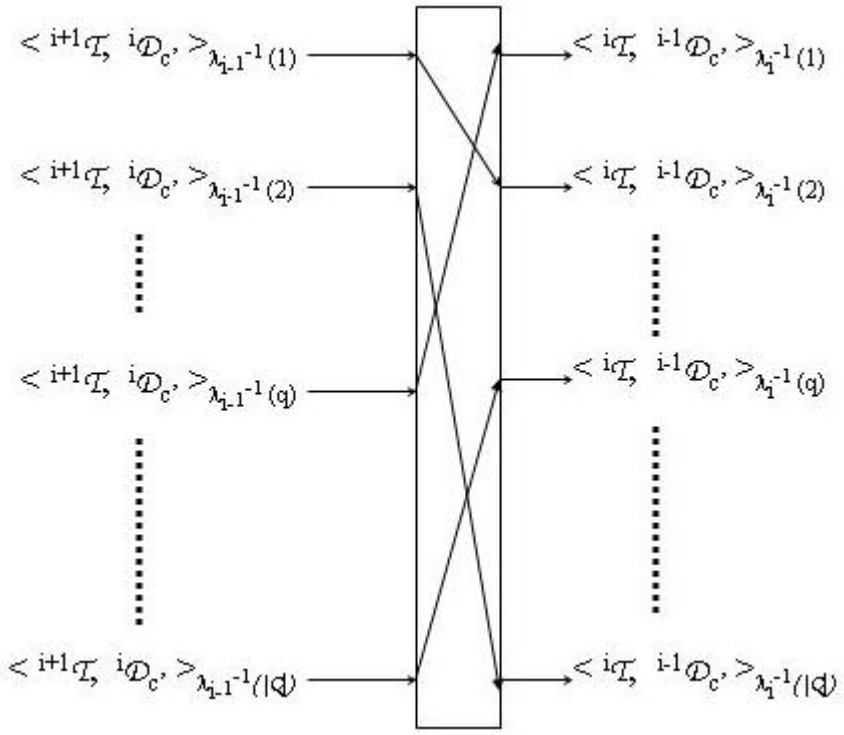

Figure 7: Input and output of i^{th} MIX

The following two steps, 9 and 10, are performed by each MIX in sequence.

Step 9: *Envelope Opening and Decryption for each receipt.*

Step 9a *Envelope Opening* (Compare the equation below with equation (4)). The MIX decrypts the string to obtain a PRNG seed ${}^{i}\alpha_r$ and an envelope ${}^{i-1}\mathcal{D}_{r,c'(r)}$ to pass on to the next MIX.

$$K_{i_{Priv}}[{}^{i}\mathcal{D}_{r,c'(r)}] = ({}^{i}\alpha_r,{}^{i-1}\mathcal{D}_{r,c'(r)})$$

Step 9b: *Add pseudo-random contribution* The MIX generates the pseudo-random bit sequence and adds it to the pixel values.

This step may be compared to equation (1), and $h'({}^{i}\alpha_r)$ seen to be the i^{th} MIX's contribution to the decryption key of the encrypted receipt.

$$^{i-1}\mathcal{T} =^{i} \mathcal{T} \oplus h'(^{i}\alpha_r) \tag{7}$$

where $^{i}\mathcal{T}$, $^{i-1}\mathcal{T}$, and $h'(^{i}\alpha_r)$ are sequences.

Step 10: *Shuffle* Each MIX shuffles the entire set of votes it receives. The shuffle is retained for the audit. Shuffled output pairs are broadcast to Public from where the next MIX obtains them. The shuffled output pairs of the i^{th} MIX are as in (6).

It can be seen that, at the very end, $\mathcal{W}_{q,c'(q)}$ is reconstructed by all MIX contributions:

$$^{1}\mathcal{T}_r = (\mathcal{B}^{j}_{r,c(r)})_{j\in\mathcal{R}_{c'}} \oplus \sum_{i=1}^{N} h'(^{i}\alpha_r)$$

(The right hand side of the above equation can be seen to be the sum of two sequences).

If the envelopes are correctly constructed, $^{i}\alpha_r = h(i, s_{c'}(r))$, and $^{1}\mathcal{T}_r = (\mathcal{B}^{j}_{r})_{j\in\mathcal{R}_{c'}}$.

Step 11: *Tallying* The final decrypted ballots are samples of the original ballots at alternate grid points, resulting in binary images with half the original number of pixels. It is assumed that these smaller images represent all the information necessary to tally the votes, i.e. the image is smaller, but not different in "content". They may be restored to images that look more "pleasant" through typical interpolation techniques known in the image processing literature. The ballots are tallied by Public.

6.4 Stage IV: Audit and Certification

The only way for the i^{th} MIX to affect vote count is by generating false values of $^{i}\mathcal{T}_r$; any cheating or colluding during shuffling only affects voter privacy. The probability of affecting vote count can be decreased exponentially with the number of votes cheated on by performing the following audit, inspired by [3].

Step 14: *MIX audit* For each MIX, check that the sequence of pairs of (6) was correctly constructed from (5) with high probability.

For the i^{th} MIX, this is done as follows:

A. If $i = n$, this is the first MIX to decrypt. A fraction, $\frac{1}{k}$, of the output positions for this MIX are chosen at random, so that specific input serial numbers may not be targeted. For these output positions, MIX n is required to "open" the corresponding inputs, i.e. to provide the corresponding input ballot positions and the decrypted seeds. These values are then checked to be consistent with the public values of the input and output of the n^{th} MIX. The value of $\frac{1}{k}$ cannot be larger than one half for privacy reasons. The probability of detecting a cheating MIX increases with a larger

value of $\frac{1}{k}$, but the number of times an audit can be performed decreases. The maximum number of possible audits is unity when $k = \frac{1}{2}$.

More formally, $\mathcal{R}_n \subset \mathcal{R}$ - where $\mathcal{R}$ is the set of output votes, a permutation of $\mathcal{Q}$ the original set of ballots - is chosen uniformly at random such that $|\mathcal{R}_n| = \frac{1}{k}|\mathcal{Q}|$. MIX n is required to provide the triplets $\{< r, \pi_n^{-1}(r),^n \alpha(\pi_n^{-1}(r)) >\}_{r \in \mathcal{R}_n}$, where, recall that ${}^n\alpha_r = h(n, s_{c'}(r))$ is the seed used by the n^{th} MIX to generate its PRNG share.

B. For $i \neq n$, the MIX is required to provide the same information for a random set of *input ballots* not contained in the "opened" ones[3] so that a single ballot cannot be traced all the way to the original serial number. More formally, $\mathcal{R}_i \subset \overline{\pi_{i+1}(\mathcal{R}_{i+1})}$ is chosen uniformly at random[4] such that $|\mathcal{R}_i| = \frac{1}{k}|\mathcal{Q}|$, and MIX i is required to provide the triplets $\{< r, \pi_i(r),^i \alpha_r >\}_{r \in \mathcal{R}_i}$.

ii. It is checked that the values of the triplets $\{< r, \pi_i(r),^i \alpha_r >\}_{r \in \mathcal{R}_i}$ are correct wrt equations (4) and (7), i.e. that

$$^i\mathcal{D}_{r,c'} = K_{i_{Pub}}[^i\alpha_r,^{i-1} D_{\pi_i(r),c'}] \ \forall r \in \mathcal{R}_i$$

and

$$^i\mathcal{T}_{\pi_i(r)} =^{i+1} \mathcal{T}_r \oplus h'(^i\alpha_r)$$

The maximum number of independent election audits that may be performed without violating privacy is $\frac{k}{2}$. In a single audit, each vote a MIX cheats on is detected with probability $\frac{1}{k}$. If the MIX cheats on p votes, the probability that he is not detected is $(\frac{1}{k})^p$.

For privacy requirements, with an audit, it is no longer enough to have at least one honest MIX that does not collude with others. With only one honest MIX, half the votes can be connected to serial numbers. One needs that at least one pair of contiguous MIXes is honest and does not collude. This may be addressed by assuming that Trustees run the MIXEs, that each Trustee runs at least 2 contiguous MIXES, and that at least one Trustee is honest.

7 Acknowledgements

The author would like to thank Rahul Simha for much input and for suggesting the writing of this document; David Chaum, Jeroen van de Graf and Peter Ryans for numerous discussions; David Chaum for much encouragement; Ben Hosp for working on an implementation that clarified numerous issues as well as required the writing of this document; and David Wagner for suggesting the e-print archive. Of course, any errors in this document reflect errors made by the author -

[3] if k=2, the choice is not random

[4] if k=2, $\mathcal{R}_i = \overline{\pi_{i+1}(\mathcal{R}_{i+1})}$

whether in understanding, writing or notation; hence the author takes all responsibility for any errors in the document.

References

[1] David Chaum. Secret Ballot Receipts and Transparent Integrity - Better and less-costly electronic voting at polling places. http://www.vreceipt.com/article.pdf.

[2] David Chaum. Secret-Ballot Receipts: True Voter-Verifiable Elections. *IEEE Security and Privacy*, Jan/Feb. 2004, pp. 38-47.

[3] M. Jakobsson, A. Juels, and R. Rivest. Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. USENIX Security '02

A Notation

K_{pub}: The public key of key pair K
K_{priv}: The private key of key pair K
K_i: key pair for the i^{th} MIX
n: number of MIXes
c: one of top or bottom layers
c': the layer other than layer c
k_p: Polling Machine key pair for signing the entire receipt s_t: Polling Machine key pair for generating $\mathcal{W}_{q,t}$
s_b: Polling Machine key pair for generating $\mathcal{W}_{q,b}$
q: ballot serial number
$S_K(x)$: a public-keyed one-way function of x, using public key pair K, example: digital signature of x using public key pair K, or encryption of a specified digest of x using K_{priv}
$\mathcal{B}_q$: the filled-in ballot, binary image, with serial number q
$\mathcal{W}_{q,t}$: pseudo-random sequence in top layer for serial number q
$\mathcal{W}_{q,b}$: pseudo-random sequence in bottom layer for serial number q
$\mathcal{I}_q^j$: the j^{th} pixel in image $\mathcal{I}_q$
$\mathcal{R}_c$: pixel positions that are filled with pseudo-random values in layer c
$\mathcal{I}_{\mathcal{R}_c}$: the pixel values in image $\mathcal{I}$ at pixel positions which are filled with pseudo-random values in layer c.
h: secure one-way function
h': pseudo-random number generator (PRNG)

i: MIX number

j: pixel number in image

${}^{n}\mathcal{D}_{q,t}$: envelope for n^{th} MIX; encrypted information to generate $\mathcal{W}_{q,t}$

${}^{n}\mathcal{D}_{q,b}$: envelope for n^{th} MIX; encrypted information to generate $\mathcal{W}_{q,b}$

${}^{i}\mathcal{T}_{r}$: the image output in position r for i^{th} Trustee

${}^{n+1}\mathcal{T}_{r}$: the image input in position r to MIX n, the first MIX to decrypt

${}^{i}\alpha_{r}$: value such that ${}^{i}\mathcal{T}_{r} =^{i+1} \mathcal{T}_{r} \oplus h'({}^{i}\alpha_{r})$

π_i: shuffle used by i^{th} MIX

$\lambda_i = \pi_n \circ \pi_{n-1} \circ ... \circ \pi_i$: the composition of shuffles from MIX n down to MIX i

$\frac{1}{k}$: fraction of votes "opened" by a single MIX in a single audit

$\mathcal{Q}$: the set of values of serial number q for all cast ballots

$\mathcal{R}_i$: the set of input values for which the i^{th} MIX "opens" pairs of input and output ballots