School of Computing Science, University of Newcastle upon Tyne



A Practical, Voter-verifiable Election Scheme

David Chaum, Peter Y. A. Ryan and Steve A. Schneider

Technical Report Series
CS-TR-880

December 2004

Copyright©2004 University of Newcastle upon Tyne Published by the University of Newcastle upon Tyne, School of Computing Science, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, UK.

A Practical, Voter-Verifiable Election Scheme

David Chaum, Peter Y A Ryan, Steve Schneider

Abstract. We present an election scheme designed to allow voters to verify that their vote is accurately included in the tabulation. The scheme provides a high degree of transparency whilst ensuring the secrecy of votes. Assurance is derived from close auditing of all the steps of the vote recording and counting process with minimal dependence on the system components. Thus, assurance arises from verification of the election rather than having to place trust in the correct behaviour of components of the voting system.

1 Introduction

Since the dawn of democracy, it has been recognised that the process of recording and counting votes would be the target of attempts at corruption. The Ancient Greeks investigated the use technological devices to provide trustworthy voting systems and avoid the need to trust voting officials, [1]. The challenge is to provide voters complete confidence that their vote will be accurately recorded and counted whilst at the same time guaranteeing the secrecy of their vote.

Most traditional approaches to this problem involve placing significant trust in the technology, mechanisms or processes. Thus, for the traditional paper ballot, the handling of the ballot boxes and counting process must be trusted, that the boxes are not lost or manipulated and that the counting process is accurate. Various observers can be introduced to the process which helps to spread the dependence but does not eliminate it.

With many of the touch screen, DRE, devices widely used in the recent US presidential elections, the voter at best gets some form of acknowledgement of the way they cast their vote. After that, they can only hope that their vote will be accurately included in the final tally.

2 Voter-verifiability

By contrast, in [3], Chaum presents a digital voting scheme that enables voter verification, i.e., provides each voter with the means to assure themselves that their vote has been accurately included in the vote tally. This scheme combines a number of cryptographic techniques and primitives to provide a high degree of transparency whilst at the same time preserving ballot secrecy. Rather than having to place trust in the components to perform correctly, steps of the vote recording and tallying process are closely monitored to detect any malfunction or corruption.

The key elements of the Chaum approach are:

- provide the voter with a receipt showing their vote in encrypted form.

- enable the voter to confirm in the booth that her intended vote is correctly
 encoded in the receipt, whilst preventing the vote from being revealed outside
 the booth.
- have a number of tellers perform an anonymising mix on the batch of encrypted ballot receipts with all intermediate steps of the tellers processing posted to the web bulletin board.
- perform random checks on all steps of the process to ensure that, with high probability, any attempt to corrupt the vote capture and counting will be detected.

The point of the encrypted receipt is to provide the voter with a way to check that her ballot is entered into the tallying process and indeed, if her receipt has not been included, to prove this to a third party. The fact that her vote is in encrypted form ensures that there is no way for her to prove to a third party which way she voted. Voters can visit the web bulletin board and check that their (encrypted) ballot receipt has been correctly posted. The tellers process these posted receipts and there are mechanisms in place to ensure that all posted receipts are entered into the tallying process.

The anonymising mixes performed by the tellers ensure that there is no link between the encrypted ballot receipt and the decrypted version that is finally output by the tallying process.

3 Prêt à Voter

The original scheme of [3] uses visual cryptography to encrypt the receipts and perform the decryption in the booth. The scheme presented here uses a more conventional representation of the vote, i.e., ballot forms with the candidates or voting options listed in one column, and the voter choices entered in an adjacent column. As a result, we believe that the scheme is easier to understand and implement.

In accordance with the design philosophy of minimising trust in components, it is essential that the decryption process in the booth be transparent and not depend on the intercession of any hardware or software devices, as these might be susceptible to failure or corruption.

An earlier paper, [6], introduced the idea of encoding the vote in terms of two aligned strips, one carrying the candidate or option list in randomised order (independent for each ballot form) whilst the other strip carries the voter choice. There, the voter was invited to choose between the two strips and to retain one as the receipt. This introduced a certain asymmetry with both cryptographic and psychological implications.

In this paper we introduce some further innovations: we use ballot forms that are generated and printed in advance. As before, these have two columns, one of which shows the candidate list in scrambled order. Now however, rather than choosing between columns as previously, the voter makes their choice between the two forms. They will always discard the column containing the candidate list,

and submit the column containing the marked vote. This avoids the asymmetry in the choice between left and right columns of the previous scheme.

A further innovation is to use the tellers in an oracle mode to enable the checks on the well-formedness of the ballot forms. This is in addition to the usual use of the tellers to perform the anonymising mix during the tallying phase. Besides allowing independent auditing authorities to perform random checks, this also opens up the possibility of various checking modes, including enabling the voters to cast a dummy vote and have the tellers return the decryption to them as a check on the construction of the ballot forms.

The scheme presented here provides a number of appealing aspects, notably:

- Voters will find the vote casting process quite familiar.
- Cryptographic commitments are generated before voter choices are known.
- Voter checks on the correct construction of the ballot forms are supplemented by random audits. Thus, voters are able to contribute to the verification of the vote capture process but we are not dependent on the voters being sufficiently diligent.
- Checks on the correct construction of the ballot forms are performed before votes are cast. This simplifies the recovery strategies.
- The vote recording devices in the booth do not learn the voters' choices. This neatly avoids any threats of such devices leaking the voters' choices.
- The scheme is conceptually much simpler than others that have been proposed. This increases the chance of voter acceptance.
- The current scheme shows considerable flexibility, suggesting that it could readily be adapted to different electoral requirements.

4 The Election Setup

A number of tellers are appointed. Each is assigned or creates two secret/public key pairs. These public keys are publicised and certified.

An authority creates a large number of ballot forms, significantly more than required for the electorate. These will have a familiar appearance: a left hand column listing the candidates or options and a right hand column into which the voter can insert her selection. This might just be an X in one cell for a single choice election or a ranking for a Single Transferable Vote (STV) system. Thus, for a four candidate race, a typical ballot form might look like:

Nihilist	
Buddhist	
Anarchist	
Alchemist	
	onion

However, the order in which the candidates are listed will be randomised for each ballot, that is, far any given ballot, the order candidate order shown should be totally unpredictable. The onion contains the information allowing the ordering to be reconstructed, buried cryptographically under the public keys of the tellers. The precise construction of the onions will be described in Section 7.2.

The exact details of the voting procedure can be varied according to the details of the election and according to the perceived nature of threats to which the system is exposed. For simplicity of presentation we outline one simple procedure. Others procedures are possible and indeed one of the advantages of this scheme is that it appears to be significantly more flexible than previous variants.

We suppose then that a suitable authority has generated and distributed a large number of printed ballot forms to the polling stations. In the spirit of the design philosophy described earlier we do not wish to place any trust in this authority to generate the forms correctly. Rather, independent auditors will be appointed whose task is to subject a random sampling of these ballots to well-formedness checks. These checks are designed to establish that the seeds buried cryptographically in the onions correctly correspond to the candidate list that appears on the form. We will describe details of these checks once the construction of the onions has been presented. The auditors might also be tasked with checking that the entropy used in the creation of the ballot forms is sufficiently random.

Further random audits could also be performed during the election. Indeed, once the election has closed, leftover forms could all be routinely audited as well. In addition to these audits, the voters are also able to perform some checks of their own, as detailed shortly. Thus, the voters are empowered to contribute to the verification of the election.

5 An Example

The scheme is best introduced by way of a simple example. We will give a more formal description later. We suppose for simplicity of presentation that we are dealing with a simple election system in which each voter selects exactly one candidate. This allows us to give the example using just cyclic shifts of the candidate ordering. Generalisations to deal with options to select more than one candidate or to rank them etc. are straightforward and discussed later. Clearly, a "none of the above" option could also be included.

5.1 Processing votes

Suppose that there are four candidates and these are given a base ordering:

Anarchist Alchemist Nihilist Buddhist

Since we are considering only cyclic shifts in this example, there are four possible candidate lists. These will be numbered from 0 to 3 according to the

offset from the base candidate list. Ballot forms will be generated with random offsets as described in detail later.

For convenience of the mathematical manipulations later, we will adopt a numbering convention for the candidates from 0 to 3 as indicated. Thus a vote for Anarchist will be encoded as 0, for Alchemist as 1 etc. This numerical representation is purely for the machine manipulations and need not trouble the voter.

Consider the following ballot form:

Buddhist	
Anarchist	
Alchemist	
Nihilist	
	Qqkr3c

This has an offset of 1. Thus the onion—Qqkr3c—encodes the value 1. Suppose the system is to process a vote for Nihilist. This would be represented by a mark in the Nihilist box:

Buddhist	
Anarchist	
Alchemist	
Nihilist	X
	Qqkr3c

Once the voter has marked their choice, the left hand column that shows the candidate ordering is detached and destroyed, to leave a ballot receipt of the form:



Such right hand strips showing the position of a X and an onion value constitute the ballot receipts.

This is now fed into the voting device, presumably an optical reader, which transmits the information on the strip, the position of the X (as a numerical value 0, 1, 2 or 3) and the value of the onion, to the tellers. The tellers use their secret keys to perform the decryption of the onion (see later), and provide the decrypted vote value corresponding to the vote in the base ordering. In this case the process yields the offset 1, so the vote value is the position of the vote (3) with the appropriate offset removed, yielding candidate 3-1=2: Nihilist. This process is illustrated in Figure 1. A more detailed description will be provided later.

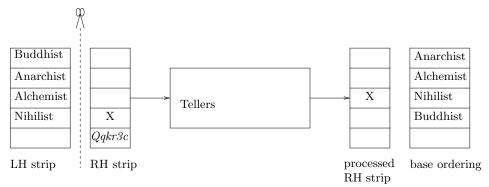


Fig. 1. Processing a vote

5.2 Introducing the Voter

Our voter, Anne, first authenticates herself and registers at the polling station. She is invited to select, at random, a pair of ballot forms. Of these, she will choose one with which to cast her vote. The other will be used for a simple check to test the veracity of the onions and the vote extraction process, after which it can be discarded.

6 Checking the construction of the ballot forms

The novel technique of using the tellers as an oracle during the voting phase suggests a number of possible modes for checking the correct construction of the ballot forms:

- 1. Single dummy vote.
- 2. Multiple or ranked dummy vote.
- 3. Given the onion value, the tellers return the candidate ordering.
- 4. Return the seed and run a checking algorithm for the well-formedness.

Of these, the last is the most rigorous and the most appropriate for the auditing authorities. Any of the first three seem more suitable for the voters to perform.

In the first, Anne would cast a dummy vote in exactly the same way that she will later cast her real vote in the booth. Thus, she could put a cross against a random selection and send the receipt off to the tellers. They decrypt the onion and return what they believe was the vote cast. If the onion was correctly constructed, this should of course agree with the dummy vote Anne selected.

Psychologically this is an interesting possibility: assuming that the check succeeds, it should provide the voter with some assurance that when they come to cast their real vote, it will also be correctly counted. On the other hand it might undermine their confidence that the secrecy of their vote will be assured.

It should also be noted that the single dummy vote provides a rather weak check on the ballot form construction, checking only part of the construction.

The second modes seeks to rectify this: by allowing the voter to cast several dummy votes, either in series of in parallel by making a ranking selection. In the later case, given the receipt, the tellers should return what they believe to be the candidate ranking chosen by the voter. This provides a more complete check on the construction of the ballot form. Both of these suffer the drawback that the voter is expected to make random choices. People are notoriously bad at making random choices.

The third mode is perhaps the most satisfactory. It provides a complete check on the ballot form but does not require the voter to make any random selections. Here, given the onion value, the tellers should return what they believe to be the candidate ordering as shown on the ballot form.

We note that the first three modes are vulnerable to collusion attacks. If the authority that generated the forms is in collusion with one of the tellers there is the possibility of corrupting forms without detection by these modes. For example, the authority could flip a pair of candidates on the ballot forms. The colluding teller performs the corresponding flip during the checking phase, but not during the tallying phase.

The last checking mode is not vulnerable to such collusions and so is more rigorous. It therefore appears to be most suitable for the auditing authorities. It could also be made available to voters, but it seems less intuitive and so perhaps less reassuring to the voters. Investigating the psychological aspects of these checking modes from a voter perspective will be investigated in future work.

We stress that all the checks detailed here serve purely to probe the well-formedness of the ballot forms, i.e., serve to detect any failure of the candidate orderings on the forms to correspond to the information buried in the onions. These checks do not provide any detection of corruption during the tallying phase. For this we have quite different mechanisms that will be presented shortly.

Assuming that the checks go through okay, Anne can proceed to the booth with her "real" ballot form. If any check fails, Anne should notify an official who should then investigate and diagnose the source of the error. We will discuss the error handling and recovery strategies later.

6.1 Casting the vote

Suppose that the check on Anne's test vote succeeded. This provides confidence that the ballot forms have been correctly constructed and hence that onion on her real ballot form also corresponds correctly to the offset of the candidate list. Anne now enters a booth with her "real" ballot form. She marks her X in the usual way. Suppose that she decides to vote for the "Buddhist" candidate:

Nihilist	
Buddhist	X
Anarchist	
Alchemist	
	e1rg38

She now removes the left hand strip (for shredding), and feeds the right hand strip into the voting device, which reads the position of Anne's X, and the value of the onion. The device then returns the right hand strip to Anne for her to retain as the ballot receipt.



Note that the vote recording device will not learn which way Anne voted. Its role is merely to read the information on Anne's receipt and relay it to the the tellers via the web bulletin board. This is a significant advantage of this scheme over earlier ones in which the voting device necessarily learnt the voter's choice, which raises the possibility that the device could somehow leak this information.

The device transmits its digital record of the receipt to a central server for subsequent posting to the web bulletin board once the election has closed. Anne will later be able to visit the bulletin board and confirm that her receipt is correctly posted and hence that it is correctly entered into the tallying process. The tallying process is deliberately constructed to hide the link between specific ballot receipts and the resulting decrypted votes, in order to provide voter anonymity. Thus Anne cannot directly link her input vote strip to any specific resulting vote, and so she cannot directly verify that her vote has been correctly decrypted. However, the fact that the votes are all correctly processed can be checked to a high degree of confidence, provides Anne with the assurance that her vote will be decrypted correctly.

Observe that Anne's receipt alone does not reveal which way she voted. Unless the tellers are involved, this can only be determined if the left hand strip (now destroyed), that carries the candidate ordering, is aligned against it. Only the totality of the tellers, acting in consort, using their collection of secret keys are able to extract the seed information and so reconstruct the candidate ordering for that ballot form.

7 Construction of the Ballot Forms

The above description should have provided the reader with the key intuition. We now give some of the mathematical details.

7.1 Construction of the Cryptographic Seeds and Offsets

For each ballot form, the authority will generate a unique, random seed. If there are k tellers (numbered 0 to k-1), this seed will be made up of a sequence of 2k values that we will call the germs:

$$seed := g_0, g_1, g_2 \dots g_{2k-1}$$

Each of these germs should be drawn from some modest size field, perhaps 2^{32} . Thus, for k=3 say, the seed values will then range over 2^{192} . These numbers can be adjusted to achieve whatever cryptographic strength is required.

The offset for the candidate list is now calculated from these germ values as follows. First a publically known cryptographic hash function is applied to each of the germs and the result taken modulo v, where v is the size of the candidate list:

$$d_i := hash(g_i) \pmod{v} \quad i = 0, 1, 2, \dots, 2k - 1$$

The cyclic offset θ that will be applied to the candidate list on this form is now computed as the $(mod\ v)$ sum of these values:

$$\theta := \sum_{i=0}^{2k-1} d_i (mod \ v)$$

7.2 Construction of the Onions

In order to facilitate auditing of the tellers while preserving anonymity of the voters (see [3] or [2] for more details), each teller performs two Chaum mixes and, accordingly, has two independent secret/public key pairs assigned to it. Teller i will have public keys $PK_{T_{2i}}$ and $PK_{T_{2i+1}}$, and corresponding secret keys. The onion is formed by nested encryption of the germs under these public keys, and is given by:

$$\{g_{2k-1}, \{g_{2k-2}, \{\dots, \{g_1, \{g_0, D_0\}_{PK_{T_0}}\}_{PK_{T_1}}\dots\}_{PK_{T_{2k-2}}}\}_{PK_{T_{2k-2}}}\}_{PK_{T_{2k-1}}}$$

We introduce a little more notation to denote the intermediate layers of the onions. D_0 will be a random, nonce-like value, unique to each onion. The further layers are defined as follows:

$$D_{i+1} := \{g_i, D_i\}_{PK_{T_i}}$$

$$Onion := D_{2k}$$

The construction of an onion is pictured in Figure 2.

8 The Role of the Tellers

The primary role of the tellers is to perform an anonymising mix and decryption on the batch of encrypted ballot receipts posted to the web bulletin board. This

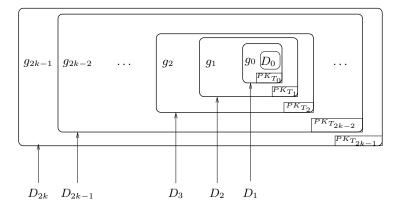


Fig. 2. An onion

ensures that the decrypted votes that emerge at the end of mix cannot be linked back to the encrypted receipts that are input to the process. Aside from some minor differences, the role of the tellers and the auditors are essentially as in the Chaum original. For completeness we give a brief overview here. More detailed descriptions can be found in [3] or [2].

The first, left hand column, of the bulletin board shows the receipts in exactly the same form as the printed receipts held by the voters. The voters can check this column to verify that their receipt has been accurately posted. An easy way to do this would be to search on the string representing the onion value and check that the X appears in the correct box, i.e., as shown on the voter's receipt.

The information in the first, left hand column of the bulletin board is then passed to the first teller, $Teller_{k-1}$, for processing. There is no shuffling of the information when it is passed to the teller. The position of the X on the voting slip is encoded as an integer r, and the correctness of this encoding can be simply and publically verified.

The tellers will subsequently manipulate the numerical representations of the receipts, i.e., pairs of the form (r_i, D_i) , where r_i is an element of Z_v and D_i is an *i*th level onion. The initial value of r_{2k} is the encoding of the position of the X as originally placed by Anne on her receipt.

Each column (apart from the first, which contains the actual receipts) shows only the simplified, digital representation: a pair (r_{2k}, D_{2k}) consisting of a value r from Z_v and the value D of the onion layer.

Each teller accepts an input column of votes (r, D) from the previous teller, and then carries out two manipulations, to produce a middle column of votes and an output column of votes. The output column produced by the teller is then passed to the next teller in the chain.

Thus for each of the (r_{2i}, D_{2i}) pairs in the batch in the input column, $Teller_{i-1}$ will:

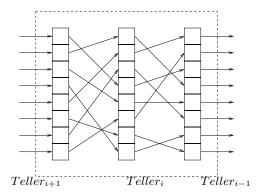


Fig. 3. A teller

– apply its first secret key, $SK_{T_{2i-1}}$ to strip off the outer layer of the onion D_{2i} to reveal the enclosed germ g_{2i-1} and the enclosed onion D_{2i-1} .

$$g_{2i-1}, D_{2i-1} = \{D_{2i}\}_{SK_{T_{2i-1}}}$$

– apply the hash function to the germ value and take the result (mod v) to recover d_{2i-1} :

$$d_{2i-1} = hash(g_{2i-1}) \pmod{v}$$

- subtract d_{2i-1} from $r_{2i} \pmod{v}$ to obtain a new r value r_{2i-1} :

$$r_{2i-1} = r_{2i} - d_{2i-1} \pmod{v}$$

- form the new pair (r_{2i-1}, D_{2i-1})

Having completed these transformations on all the pairs in the initial batch as posted in its input column, it applies a secret permutation to all the resulting pairs and posts the resulting permuted pairs to its middle column on the bulletin board.

 $Teller_{i-1}$ now repeats this process on the contents of the middle column using its second secret key, $SK_{T_{2i-2}}$ to obtain a new set of (r_{2i-2}, D_{2i-2}) pairs. It will apply a second secret shuffle, independent of the previous one, to this batch of new pairs. The resulting transformed and shuffled (r_{2i-2}, D_{2i-2}) pairs are now posted to the output column on the bulletin board, and passed on to the next teller, $Teller_{i-2}$. This process is illustrated in Figure 3.

This process is repeated by all the tellers in sequence, as illustrated in Figure 4 for a sequence of three tellers. The value of any of the intermediate r values is thus given by:

$$r_{2k-i} = r_{2k} - \sum_{i=1}^{i} d_{2k-i} \pmod{v}$$

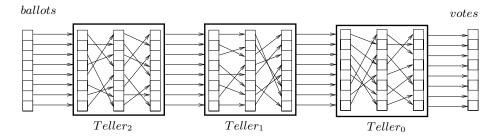


Fig. 4. Three tellers anonymising mix

When the last teller performs the final transformation it outputs a batch of pairs which comprise a final r value r_0 and the inner onion value D_0 . The final r_0 values are the values of the original votes in the canonical, base ordering. Figure 5 illustrates the effect of the process on a single vote.

To see this, observe that the candidate list on each form is shifted by the $(mod\ v)$ sum of the d values, i.e., θ . Thus the initial r value is the candidate value plus θ modulo v. For each ballot pair, the tellers will have subtracted out the d values from the initial r value, thus cancelling the original shift of the candidate list and so recovering the original candidate value. Thus:

$$r_0 = r_{2k} - \sum_{j=1}^{2k} d_{2k-i} \pmod{v} = r_{2k} - \theta \pmod{v}$$

Consider the example of Anne's vote again (illustrated in Figure 5). The form she used to cast her vote had an offset of 2 and her X was in the second box, value 1. Hence the initial value of r_{2k} was 1 in her case. The tellers will in effect compute:

$$r_0 = r_{2k} - \sum_{j=1}^{2k} d_i \pmod{4} = 1 - 2 \pmod{4} = 3$$

Thus the final r value $r_0 = 3$ does indeed translate to a vote for "Buddhist" in the base ordering. The encryption of the vote can thus be thought of as a (co-variant) transformation of the frame of reference, decryption to the corresponding (contra-variant) transformation.

The overall effect then, is to have posted on the bulletin board, in the left hand column, the batch of initial receipts as posted by the voting devices. In the right hand column we will have the fully decrypted votes. In between there will be a set of columns with the intermediate, partially decrypted (r, D) pairs. Each column will be some secret permutation of the previous one, and the permutation will not be published. This is illustrated in Figure 6. Note that the decryptions at each mix stage prevent the permutation being reconstructed by simple matching of onions or r values.

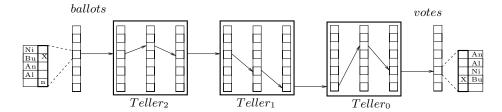


Fig. 5. A vote processed by three tellers

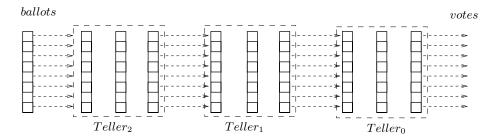


Fig. 6. Information posted by the sequence of three tellers

The purpose of using the hash of the germ values buried in the onion layers to transform the r values is to foil guessing attacks on the mixes. Without these hashes it would be possible to guess links through the mixes and check the guess by performing the appropriate computations (with the knowledge of the teller's public keys). With the hash functions, these checks would require the computation of pre-images of the hashes, thus rendering such attacks intractable. We will see later that, for audited links the tellers are required to reveal not only the link but also the associated germ. The computations performed by the auditors are thus perfectly tractable.

Assuming that all the tellers perform their transformations correctly, there will be a one-to-one correspondence between the elements of each column and the next. The exact correspondence, which (r,D) pair in one column corresponds to which pair in the next column, will be hidden and known only to the teller who performed the transformation between those columns. Thus, the receipts will have undergone multiple, secret shuffles between the first column as posted by the voting devices and the final decrypted column. This ensures that no voter can be linked to her vote, so ensuring ballot secrecy.

The fact that several tellers are used gives several layers of defence with respect to voter privacy: even if several of the tellers, but not all, are compromised, the linkage of voters with their votes will remain secret.

The decrypted votes are posted in the final column so the overall count can be verified by anyone.

9 Checking on the Authority

The description so far has assumed that all the players: the authority, the voting device and the tellers, have behaved correctly, i.e., in accordance with the rules of the scheme. If we could be sure that everyone would obey the rules, we could be sure that the election will be both accurate and private. However, should any of these

players cheat, the accuracy and/or privacy would be undermined.

To have to place such dependence on the components runs counter to the design philosophy of the scheme. In this section we discuss the checks that are performed on the ballot forms provided by the authority. The following checks are performed by the auditors and the voters themselves to provide assurance that the forms are correctly constructed (and hence result in the correct decryption of the votes).

Firstly, auditors select a random sample of forms to check. This can be done before, during and after (on unused forms) the election period. For each selected ballot form they perform the following mode 4 check:

- a digital copy of the onion is sent to the tellers.
- the tellers strip off the layers of encryption using their private keys to reveal the germs.
- these germ values are returned to the auditors.
- given the germ values, and knowing the public keys of the tellers, the auditors are able to reconstruct the value of the onion and can check that this agrees with the value printed on the form.
- they now recompute the offset value as the $(mod \ v)$ sum of the hashes of the germs.
- they can now check that the offset applied to the candidate list shown on the form agrees with the value obtained above.

If all these checks are successful, it is safe to conclude that the ballot form in question was correctly constructed. Checked ballot forms, for which the seed has been revealed, are then discarded.

Note that the algorithms for these checks are publicly known, so in principle, anyone could construct such a checker and make it freely available. Similarly anyone could examine such a checker to establish that it was performing correctly. Note also that any interested party could volunteer to perform some of the auditing. Thus, for example, the Electoral Reform Society could act as auditors. Representatives of the political parties could act as auditors. Furthermore, any results produced by an auditor can be double checked by independent parties.

Besides these checks performed by the auditors, the voters get to perform checks of their own on their dummy ballot forms. This serves as a further check that the authority and tellers are behaving correctly, and should help promote confidence in the electorate that their votes will count.

As noted earlier, care has to be taken in assessing the assurance provided by the voter checks as these are vulnerable to collusion attacks. Various countermeasures could be adopted to limit the likelihood of such collusions. One possibility is to use an l out of k threshold scheme for the onion encryptions. The l

cardinality subsets of the k tellers could then be chosen randomly for the dummy voting process. If the colluding tellers were omitted when a corrupted dummy vote was decrypted, an error would be flagged.

In any case, the random audits should catch such manipulated ballot forms. The auditor checks described earlier are not vulnerable to such collusion attacks as they check directly the well-formedness of the ballot forms. The tellers might return incorrect germ values but this will of course throw up a mismatch between the recomputed onion value and the value on the form. It might be that a teller malfunctions, or is loaded with the wrong keys. In this case the checks serve a useful role in debugging such configuration errors.

Note that the encryptions are all bijective, hence the germ values are uniquely determined by the onion value. The tellers cannot therefore find alternative germ values that would give the same onion value but a different offset.

Together, these checks ensure that if a malicious or corrupted authority tried to corrupt votes by providing a candidate ordering that does not correspond to the seed information buried in the onion, they stand a high chance of being detected. The chance of corruption going undetected falls off exponentially with the number of ballots they try to corrupt.

10 Checking on the vote recording devices

We need to ensure that ballot receipts are faithfully transmitted and entered into the tallying process. This is where the web bulletin board comes into play. Once voting has closed, all ballot receipts should be posted to the bulletin board. The material posted to the bulletin board should be publically available in read only mode. Thus any voter can visit the board and confirm that their receipt appears correctly in the input column.

If their receipt does not appear, or appears in corrupted form (in particular, if the position of the X is incorrect), then this should be reported. The voter has their receipt to prove to an official if their receipt does not appear correctly. In practice all ballot forms would be printed on anti-counterfeiting paper and would probably have a digital signature to prevent attempts to fake receipts.

Assuming that voters are reasonably diligent in performing these checks, any failures to faithfully post receipts to the bulletin board, and hence to enter them into the tallying, should be detected. Precautions would also be needed to prevent anyone inserting additional, invalid receipts. One simple precaution would be to ensure that the number of posted receipts matched the number of cast ballots. Digital signatures applied by the voting devices could also be used to help prevent fake ballots being introduced.

11 Checking on the Tellers

Checks must also be performed to detect any inaccuracy in the transformations performed by the tellers on the real ballot receipts. That is, we need to detect any attempt by the tellers to alter, remove, inject or corrupt votes.

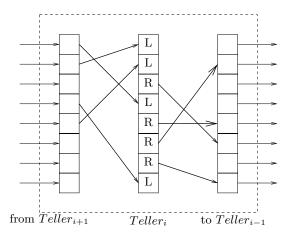


Fig. 7. Auditing $Teller_i$

As in the original Chaum scheme, the auditing of the tellers is based on the notion of partial random checking proposed in [4]. This takes place after the teller processing has finished, and is applied to the information committed to by the tellers on the web bulletin board.

For each teller an auditing authority goes down the middle column and randomly assigns R or L to each (r, D) pair. For pairs assigned an R, the auditor requires the teller to reveal the outgoing link (to the right) to the corresponding pair in the next column along with the corresponding germ value. For all pairs assigned an L, the auditor requires the teller to reveal the incoming link (from the left) along with the germ value.

This way of selecting links ensures that, for any given teller, no complete route across the two shuffles performed by that teller are revealed by the audit process. Hence no ballot receipt can be traced across the two mixes performed by any given teller. Each ballot transformation has a 50/50 chance of being audited.

This is illustrated in Figure 7, with the selected links included. The remaining links are not revealed.

For each teller the auditor performs such a random audit. Given the property that there are no full links revealed across any teller's mixes, the L/R selection can be made quite independently for each teller. This is the rationale for making each teller perform two mixes.

Suppose that for a revealed link the pair has been transformed thus:

$$r_i, D_i \longrightarrow r_{i-1}, D_{i-1}$$

Knowing this and the corresponding germ value g_{i-1} (which the teller is required to provide for each revealed link), it can be checked that the following hold:

$$D_i = \{g_{i-1}, D_{i-1}\}_{PK_{T_{i-1}}}$$

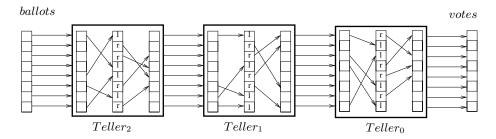


Fig. 8. Auditing the three tellers

and

$$r_{i-1} = r_i - hash(g_{i-1})(mod \ v)$$

If these equalities hold on a link we can conclude that the teller executed the correct transformation on this ballot pair. Some additional reasoning is required to show that it is not possible for a teller to perform a corrupted mix and be able to reveal false links in such a way as to pass any audit.

Figure 8 illustrates the audit across the sequence of three tellers.

12 Error Handling and Recovery Strategies

So far we have only described the checks that can be performed. A full description of the scheme requires detailing error handling and recovery modes. Due to lack of space we will not attempt to give an exhaustive description here.

Let us just consider the error handling strategy for a failed voter check. The first step for the official is to confirm that there is a real disagreement. Anne will have both parts of the dummy ballot form so she can prove which way she cast her dummy vote and she has the printout for the tellers. The official can thus establish that the problem is genuine and not just a case of voter error.

If the problem is real, the official should now run a further, mode 4 check: use the tellers as an oracle to extract the seed value and use this value to reconstruct the onion value and candidate list offset. If these values agree with those shown on the ballot, then it is fair to conclude that the form was correctly constructed by the authority. The error must then lie with the decryption of the vote performed by the tellers.

If this check fails, it can mean one of two things: the form was incorrectly constructed by the authority, or the form was perhaps actually correctly formed but the seed value returned by the tellers is incorrect.

Clearly, errors have to be diagnosed and collated. Strategies for dealing with patterns of errors must be specified. Thus, if a significant number of ballot forms were found to be mal-formed, doubt would be cast on the integrity of the authority charged with generating the forms. Note another pleasing feature of the

scheme: any significant corruption on the part of the authority generating the ballot forms would almost certainly be detecting by random audits before the election opens.

A full description of error handling and recovery strategies will be given in a forthcoming paper.

13 Generalising ballots

This paper has so far considered ballots which allow a vote against a single candidate. More generally, elections may allow votes or preferences to be cast against a number of candidates. In this case a right hand strip may contain a number of X's, or perhaps a list of numbers against candidates.

In this case, in order to avoid leaking information about votes, it is necessary to allow any permutation of the candidate list on the left hand strip, rather than just a cyclic permutation.

In order to achieve this, the germs could be used as keys for a cryptographic permutation function. The overall permutation applied to the candidate list as shown on the ballot form would then be a composition of the 2k separate permutations obtained from the 2k germs.

We use a publically known hash function h that maps germs to permutations, so that $p_i = h(g_i)$ is a permutation of names on ballots. The overall permutation is given by the composition of the permutations for all the germs:

$$\pi = p_{2k-1} \circ p_{2k-1} \circ \ldots \circ p_0$$

(where $f \circ g(x) = f(g(x))$). If the base candidate ordering is *base*, then the candidate list on the ballot is given by $\pi(base)$. Thus a corresponding vote r on the right hand strip corresponds to a vote of $\pi^{-1}(r)$ against the base ordering.

The steps in the tellers take (r_{i+1}, D_{i+1}) to (r_i, D_i) , where each step reverses one permutation comprising π . Here, the r values will encode either a ranking or an element of the power set of candidates as appropriate. The onion is unpeeled as previously to extract the associated seed g_i and the inner onion D_i . In this case the computation of r_i is given by:

$$r_i := (h(g_i))^{-1}(r_{i+1}) = p_i^{-1}(r_{i+1})$$

Given the initial vote r provided to the tellers is the initial vote r_{2k} , we obtain that

$$r_i = (p_i^{-1} \circ p_{i+1}^{-1} \circ \dots \circ p_{2k-1}^{-1})(r_{2k})$$

and thus the final vote r_0 posted by $Teller_0$ is $\pi^{-1}(r)$, which is indeed the vote cast.

14 Related work and conclusions

A large number of cryptographic voting schemes have been proposed over the past 20 or so years. These use a variety of cryptographic techniques, ranging from blind signatures to cryptographic homomorphisms etc. The idea of providing the voter with an encrypted receipt goes back the original scheme proposed by Chaum. Another scheme, that also uses encrypted receipts and has similar goals, is the VoteHere scheme of Adler and Neff, [5]. The cryptographic primitives used there are quite different from those of this paper and appear to be significantly more complex.

We have presented a new voter-verifiable election scheme based on the original Chaum scheme. This variant preserves the essential features of the original whilst sidestepping the complexity of the visual cryptography of the original. The presentation of the encoding on the vote is quite intuitive and familiar. A pleasing spin-off is that the randomisation of the candidate order counters any tendency to bias the voter choice that might arise from a fixed order.

The new scheme provides some interesting advantages over previous variants:

- The format of the ballot forms and the process of casting a vote is quite familiar.
- The cryptographic commitments are generated before the voter choices are revealed, even before the election period starts.
- The vote recording devices do not learn the voter choices. This avoids the
 possibility of such devices leaking this information.
- Voters get to perform some their own checks on the correct construction of their dummy ballot forms. This should help instil confidence that their real votes will ultimately be correctly decrypted during the tallying process.
- The checking performed by the voters is supplemented by audits performed by various auditing agencies.
- The problem of storing and selectively revealing seed information is solved by the novel use of the tellers during the voting period as oracles to reveal the seeds for ballot forms used for auditing.
- Voters get to run their checks before casting their vote. This avoids some of the messiness in the recovery mechanisms of earlier variants when a voter discovers a mal-formed receipt after casting their vote.

Precautions need to be taken to prevent double voting. In particular, care needs to be taken to ensure that ballot forms used for checking cannot be reused to cast real votes. These details of such mechanisms will be discussed in a future paper.

For the purposes of illustration we have described how the scheme can be used for a single vote system, i.e., in which voters get to choose just one of a set of options or candidates. Where voters can rank the candidates in order of preference (or indeed where they can vote for more than one candidate), full permutations in place of the simple cyclic shifts presented here. In practice, full permutations would probably be used even for single selection elections.

15 Future Directions

The destruction of the left hand strips of the ballot forms is essential to prevent coercion. An issue that requires careful consideration then is how to best enforce the destruction and ensure that it is not possible for the voter to exit the booth with both parts of the ballot form. Mechanical devices that enforce the destruction when the vote is cast are a possibility. Another interesting possibility is, rather than trying to enforce destruction of this strip, to ensure that plenty of dummy left hand strips are available in the booth. If a voter is threatened with coercion they can simply select an appropriate strip that will keep the coercer happy.

Another issue is that, as presented, the scheme entails the authority knowing the association of all onions and candidate lists. Thus, if the authority were compromised, it could jeopardise the secrecy of the election. Various measures can be envisaged to counter or at least minimise this risk. Ballot forms could be generated in some distributed fashion using various sources of entropy. Alternatively, ballot forms could be generated and printed on demand. An intriguing possibility is to use entropy derived from the paper used to print the forms, for example using optical fibres stirred into the paper during manufacture. Ballot forms could be supplied in sealed envelopes to prevent the information being garnered in transit. The problem remains that there is still a point at which the onion and candidate list must be presented to the voter.

For the checking modes 1 though 3, the germ values do not have to be revealed. This suggests the possibility of reusing a "dummy" ballot form to cast a real vote. This has the advantage that the form used for the real vote will itself have been tested. Ballot forms could come equipped with two onion values, both of which should yield the candidate ordering shown. One could be used for checking, the other to cast the real vote. This possibility may however open up vulnerabilities and would need to be subjected to careful analysis. We will pursue this in a forthcoming paper.

This scheme would appear to be readily adapted to remote voting. The simplest adoption is to distribute ballot forms by post. Votes could then be cast by providing the onion value along with suitable indicators of the voter selection in the right hand column. Alternatively, protocols could be used for on-line, authenticated distribution of the crypto material. Of course, the threat of coercion that plagues remote voting systems rears its head again, but there may be ways to offset this.

These avenues are the subject of current research.

16 Acknowledgements

The authors would like to thank Ben Adida, Jeremy Bryans, Jeroen van der Graaf, Michael Jackson, Cliff Jones, Rene Peralta, Brian Randell, Ron Rivest, Fred Schneider and Poorvi Vora for many helpful discussions.

This work was partially funded by the EPSRC DIRC project, www.dirc.org.uk.

References

- Robert S. Brumbaugh. Ancient Greek Gadgets and Machines. Thomas Y. Crowell, 1966.
- Jeremy W. Bryans and Peter Y. A. Ryan. A Dependability Analysis of the Chaum Voting Scheme. Technical Report CS-TR-809, Newcastle University School of Computing Science, 2003.
- 3. David Chaum. Secret-Ballot Receipts: True Voter-Verifiable Elections. *IEEE Security and Privacy*, 2(1):38–47, Jan/Feb 2004.
- 4. M. Jakobsson, M. Juels, and R. Rivest. Making Mix Nets Robust for Electronic Voting by Randomised Partial Checking. In *USENIX'02*, 2002.
- 5. C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In ACM-CCS-2001, 2001.
- Peter Y. A. Ryan. A Variant of the Chaum Voter-Verifiable Scheme. Technical Report CS-TR 864, University of Newcastle, October 2004. To appear in WITS 2005.