School of Computing Science,
University of Newcastle upon Tyne

A Dependability Analysis of the Chaum Digital Voting Scheme

Jeremy Bryans and Peter Ryan

Technical Report Series

CS-TR-809

July 2003

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

A Dependability Analysis of the Chaum Digital Voting Scheme

Jeremy Bryans and Peter Ryan

Abstract

We present a discussion of the requirements for a voting system, then a detailed description of the Chaum digital voting scheme [4] with respect to these requirements.

This report is intended primarily to provide a detailed and, we hope fairly accessible, description of the technical aspects of digital voting requirements and the Chaum scheme. It does not seek to explore all the possible social, psychological, political or economic aspects of digital voting. This we leave for other reports.

1 Introduction

More technologically sophisticated alternatives to the traditional pen and paper methods of casting and counting votes are currently in use in many countries, and are being investigated in many others. The UK government has stated it's enthusiasm for such schemes [13] and a number of trails have been performed, e.g. [6]. The motivation for this appears to be:

- Making the casting of a vote more convenient and appealing may lead to improved turnout.
- Electronic tabulation and counting of votes may be faster and less labour intensive.
- Arguably, digital technology could provide greater accuracy and perhaps even greater anonymity than conventional, pen and paper approach.

It is not the purpose of this paper to debate the validity of these claims but to try to formalise the requirements of a voting scheme and then analyse the extent to which a particular proposed scheme (the Chaum scheme, presented in [4]) achieves these goals. Many of the schemes that have been put forward appear to have little in the way of checks on the correct recording and tallying of votes. The Chaum scheme is of interest because it provides the voter with the possibility of verifying that their vote is accurately included in the final tally, while maintaining the secrecy of the election.

Although we are interested in digital voting in general, in this paper we will focus our attention primarily on the Chaum scheme. The reason for this is that this scheme offers a number of particularly interesting features from both a dependability and an interdisciplinary point of view. It is thus as excellent candidate for a DIRC case study.

We start with a discussion of the requirements of a voting system in Section 2, and in Section 3 give a brief overview of some digital voting schemes. In Section 4 we give an intuitive overview of the Chaum scheme and then in Sections 7 and 8 a more detailed description the communication transactions between the voters, the voting machine and the trustees. The calculations that each party must perform are described, as well as the checks that are available to independent parties that ensure that these calculations and communications have been carried out correctly. We also indicate which pieces of information are secret and which are public. Where appropriate, the rationale for different parts of the scheme is discussed in terms of the attacks which they foil. In Section 9 we draw some conclusions.

Terminology

The term "e-voting" appears to have been reserved for schemes that do not require any bespoke equipment and rely solely on the standard infrastructures such as the internet. Typically, in an e-voting scheme, the voter will be able to enter their vote over their home computer. In this paper we do not confine ourselves to such schemes. We are interested in schemes in which digital technology is used to improve the voting process, but may still involve certain purpose designed equipment, typically called "Direct Recording Electronic" (DRE) voting machines. Thus, for example, the Chaum scheme requires special printing devices and so would still require the voter to be physically co-located with such a device. They cannot vote using the Chaum scheme from their home computer. We refer to such schemes as digital voting.

An *election* is defined as a question asked of the populace. We include the possibility of *referenda* here.

A *vote* is the choice recorded by a single voter. It may be *spoiled* (deliberately or accidentally). It may be a yes or no choice (as in a referendum), a single candidate (as in a first-past-the-post election) or a ranking of some or all of the possible candidates (as in a Proportional Representation election).

2 Requirements of a Voting System

In this section we discuss the requirements of a voting scheme. The primary requirements are accuracy and ballot secrecy. Of course, we must recognise that failures of the system may occur, and so an auxilliary requirement is to be able to detect failures with respect to accuracy (this is fulfilled by making a scheme auditable) and to be able to recover from them. Failures with respect to secrecy should also be detectable, but are of course irrecoverable. Ideally a voting scheme should also be usable, efficient, unbiased, scalable etc.

These requirements may be in conflict with each other. For example there is a tension between the requirement for ballot secrecy and that of auditability. A naive implementation of auditability would immediately violate secrecy. Chaum conjectures in [4] that it is impossible to achieve absolute assurances of

unconditional accuracy and secrecy simultaneously. His scheme provides provides both requirements up to certain probabilistic and computation bounds. He conjectures that the scheme may achieve an optimum with respect to these conflicting constraints.

2.1 Accuracy

What precisely we mean by accuracy will depend at what level we are working and where we are drawing the system boundaries. At the most abstract level, we would like the outcome of an election or referendum to accurately reflect the "intentions" of the eligible electorate. At this level we will need to consider social and psychological issues that might pose barriers to the participation of certain sectors of society, bias the choices made or introduce voter error.

For the purposes of this paper we will restrict ourselves to the purely technical question of ensuring that votes counted in the final tally accurately reflect the votes cast. We will assume that issues of authentication of voters and prevention of double voting have been addressed.

In this purely technical sense, accuracy will be the requirement that the final tally of votes exactly match the votes cast. In practice, absolute assurance of complete accuracy is not feasible and, arguably, too strong a requirement. A more realistic requirement for an election is that the outcome be "correct", e.g. that the candidate with the largest number of votes wins.

2.2 Ballot Secrecy and Anonymity

It will typically be a requirement that the way any individual voter voted remain secret. This may in some cases be strengthened: voters may want to keep even the fact that they voted a secret. For some forms of vote secrecy may not be required at all, for example voting in the House of Commons. Any voting scheme must therefore be clear about the flavour of secrecy that is being provided. Besides the natural desire for privacy, ballot secrecy serves to prevent coercion or vote buying.

Note that absolute assurances of total secrecy may not be realistic here. In certain exceptional circumstances secrecy will be violated in any case, for example, if all the votes went one way. More subtle effects may be possible. In the context of the Chaum scheme, some of these are discussed in [7]. To paraphrase a scenario in this paper, consider a simple two way referendum in which half the electorate vote "yes" half vote "no". Suppose further that there is only one mix and that the revealed links (see later) also happen to lead to "yes" votes. In this case privacy fails completely.

Now such a scenario is extremely unlikely in several respects[1], especially if we are considering a large electorate, but it does make the point that, certainly where Random Partial Checking protocols are employed, there is a chance of

[1] We are reminded of the character in one of the Molesworth books [18] who, asked by a teacher to consider a right angle triangle with squares on all three sides, asks "Is that really very likely Sir?".

some information leakage. In [7] it is argued that, for the Chaum scheme, the mixing will be sufficiently rapid to ensure that with just a small number of mixes the likelihood of significant leakage is sufficiently small to be neglected.

Instead of ballot secrecy we might require voter anonymity. At first glance one might suppose that these are equivalent. We follow the approach of Schneider [16] in formalising the notion of anonymity using CSP. A system S satisfies anonymity with respect to some set V and viewpoint given by the process abstraction $\mathcal{A}$ if:

$$\forall \pi : Perm(v) \bullet \mathcal{A}(S) \equiv \mathcal{A}(S[\pi])$$

where $[\pi]$ denotes the CSP renaming operator.

Thus, if we transform the system by arbitrary permutation of the set of voters, the resulting system is indistinguishable from the original, at least from the viewpoint represented by the abstraction A. The abstraction serves to hide internal details not visible to an outside observer. For the Chaum scheme, an observer would be able to see the values posted to the web but none of the internal values used by the counting processes. Care has to be taken in the definition of the abstraction and process equivalence used where the system manifests non-determinism and utilises cryptographic mechanisms. Process algebraic formulations of non-interference are appropriate here, see for example [14].

Note that, using such a definition, the scenario of everyone voting for the same candidate would still be deemed to satisfy anonymity but would fail the ballot secrecy requirement. Given that such a scenario is perfectly admissible and that the violation of ballot secrecy seems inevitable, this would seem to suggest that voter anonymity is the more appropriate requirement.

2.3 Auditability and recoverability

In true dependability fashion we recognise that absolute guarantees are not feasible. System malfunctions and compromises will occur. It is essential therefore that mechanisms be provided to detect, contain and recover from failures with respect to the requirements. These mechanisms need to be robust in the face of malicious as well as accidental threats. A voting system should therefore be auditable (or *verifiable*) by the individual voters, as well an auditing body.

Much of the opposition against DRE machines in the United States has centered around a call for a "voter-verifiable audit trail", which in [1] is taken to mean

> "a permanent record of each vote that can be checked for accuracy by the voter before the vote is submitted, and is difficult or impossible to alter after it has been checked. "

The important point is the ability of the individual voter to determine whether or not their vote has been correctly included in the tally.

Although other technologies are suggested in [1] as possible future implementations of voter verifiability, the tested and favoured method is a paper audit trail.

2.4 Usability

The voting process should not be unnecessarily difficult for a voter to execute. Voting should be convenient and easy to understand. It should not be error-prone and it should be easy for the voter to detect and recover from any mistakes before they commit to their choice.

A balance may have to be struck between convenience on the one hand and the requirements of accuracy and privacy on the other. Thus, for example, it might be argued that the most convenient way to vote would be over the internet via a home computer or using text-messaging. It seems unlikely however that appropriate levels of assurance of accuracy and privacy could be provided with such mechanisms. It also appears difficult to be sure that a remote voter is not under duress and is able to cast a free vote.

Equally, any checking procedures should be convenient and easy to understand.

2.5 Absence of bias

The question of bias and how to avoid it appears at a number of places within the Chaum scheme. Most importantly, it should avoid introducing any bias into the voting process. This could happen if particular subsets of the electorate were disenfranchised. For example, those with a low level of technical know-how could be put off by an apparent need to understand the technicalities of the system. It is also possible that particular candidates may be favoured because of the order in which the choices are presented, although this would then be a difficulty of almost any voting system.

More specifically to the Chaum system, voters may show a bias for chosing one or other of the layers of the receipt. This would then make certain attacks on the system at least slightly more likely to succeed.

2.6 Efficiency and Scalability

For small scale elections the computational cost per vote is unlikely to be a limiting factor, but for elections at a national scale the millions of votes cast must be recorded, stored and counted within a day or so. The voting scheme must be designed to cope with the scale of the election in mind.

3 Overview of some Digital Voting Schemes

A very brief overview and assessment of some alternative electronic and digital voting schemes.

- Deibold

  Diebold touchscreen equipment has been in the news recently. Although they are "black-box" systems, so the internal code is supposed to be secret, a website recently appeared containing what claimed to be actual source

for one of their machines. A detailed analysis was published [9] of the code claiming to have found numerous errors. Diebold quickly produced a rebuttal of the claims [2], and the original authors have produced a rebuttal of the rebuttal [3].

- Schneier

  One of the example Schneier electronic voting schemes in [17] requires voters make their own choice of password when registering on the electoral roll. They use this password to caste their vote. All votes are posted on a web site, each vote alongside the corresponding password. Thus the voter can search on their password and check that the corresponding vote is as cast. They cannot however prove to a third party that this is their vote as there is no way for them to prove that this is their password.

- VoteHere

  Although traditionally a "black-box" manufacturer, recently VoteHere promised to make their source code publically available. At the time of writing this has just been made available.

4 Overview of the Chaum digital voting scheme

The Chaum scheme is appealing in several respects from an inter-disciplinary point of view. It strives to provide the voter with good levels of assurance that their vote will be accurately recorded and that the privacy of their vote will be guaranteed. In particular, with respect to the accuracy requirement, the scheme provides the user with a physical receipt and the means to check, that their vote is accurately represented in the final count.

One of the remarkable features of the scheme is that it side-steps the standard wisdom that it is not possible to provide a voter receipt without violating voter privacy. Voter privacy is essential to avoid the possibility of coercion and vote buying. A naive receipt that could provide proof to a third party of which way the vote was cast would allow vote buying. On the surface of it, it would appear to be impossible to devise a form of receipt that would, on the one hand allow the voter to check that their vote is accurately represented in the final tally whilst, on the other hand, not providing any evidence to a third party as to which way the vote was cast. Many commentators on the subject seem to assume that this is indeed impossible. For example, the assumption is implicit in Rebecca Mercuri's question 14 of her set of questions for evaluators of voting schemes, [11]:

"How is vote confirmation provided without ballot-face receipt?"

In this section we will attempt to give the reader an intuition as to how this is achieved within the Chaum scheme and in the later sections we provide a more detailed description of the mechanisms required.

Note that the ancient Greeks actually had a partial solution to this problem. Votes were cast using marks on shards of pottery. Thus a voter could check that

their shard was included in the final tally by recognising their shard. They could not however prove to a third party that this was their shard. One of the Schneier schemes [17], provides a electronic analogue of this: voters make their own choice of password when registering on the electoral role. They use this password to cast their vote. All votes are posted on a web site, each vote alongside the corresponding password. Thus the voter can search on their password and check that the corresponding vote is as cast. They cannot however prove to a third party that this is their vote as there is no way for them to prove that this is their password.

Most of the alternative digital voting schemes require the voter to trust the hardware and/or software that processes their vote and provide little or no means for the voter (or indeed anyone) to detect a failure in the processing of votes.

4.1 Alice casts a vote

Let us suppose that Alice is our intrepid voter. For simplicity, let us assume for the moment that there is just one booth. Alice will show up at a voting station and authenticate herself in some way. The Chaum scheme assumes that suitable safe-guards are in place to ensure that any eligible member of the electorate is able to cast a vote at most once.

Once authenticated, Alice is ready to cast her vote. The booth presents her with a choice of alternatives. See Figure 1 as an example.

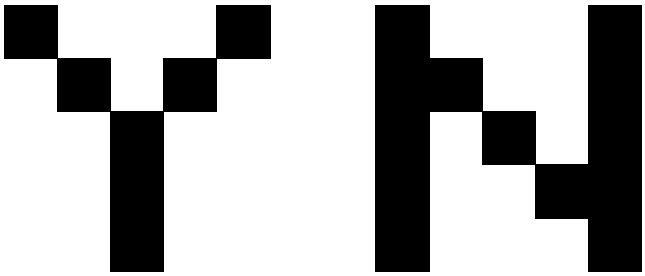
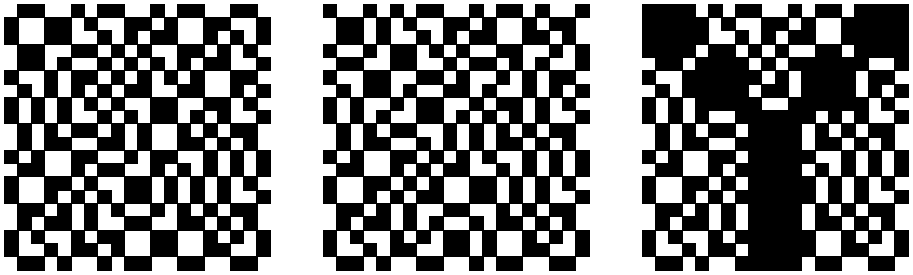


Figure 1: Screen view

Alice makes her choice, via a touch screen perhaps, and her selection is now printed out as two overlaid pixel patterns on two sheets of acetate (see Figure 2.) As long as the patterns are correctly overlaid, her selection is visible as a ballot image formed out of a pattern of opaque and semi-opaque cells. For example, suppose that she has chosen the "Yes" option, this would be constructed as in Figure 2.

The point about two layers of pixel patterns is that, although, the choice is visible when the two viewed correctly overlaid, each viewed separately reveals nothing about her choice. The details of how this is achieved are left until Section 5.1. Suffice it to say at this stage that the ballot image is obscured with random noise which is woven between the two layers. Where part of the



First Sheet Second Sheet Overlaid Image

Figure 2: Construction of the Ballot Image

encrypted ballot image appears in a region of one layer, the matching region of the other layer carries the opposite of the random noise. Thus, when the two are overlaid, the noise cancels out revealing the ballot image.

Assuming that the ballot image that emerges from the printer on the overlaid sheets is what Alice expects, she can signal her okay. At this point, the booth asks her to make a choice of either the upper or the lower of the printed layers. She will retain the chosen layer, while the other layer will be destroyed.

Once she has made this choice, some further information is printed onto both sheets alongside the pixel patterns printed earlier. Again, we leave the details of exactly what information is printed at this stage until later. For the moment we simply note that the information on the chosen layer includes information enabling certain checks to be performed on the printouts.

Alice now detachs the sheets from the printer and separates them before leaving the booth. On exiting the booth, she hands over the sheet marked for destruction to a voting official who should verifiably destroy the sheet in front of her. Chaum suggests that a transparently housed shredder might be suitable for this.

The vote casting stage is now over as far as Alice is concerned. As an upstanding member of a democratic society however, she should not relax completely at this point: she is encouraged to perform a number of checks.

In particular, she should run her receipt through a device that can perform some mathematical checks to establish that the receipt was correctly generated by the booth. We will come the significance of this and the other checks in Section 4.3.

4.2 The Tallying Stage

Tallying is performed by a number of trustees, all of whom have "read and append" access to a publically readable website. The votes are passed through all the trustees, with each trustee finishing their work before passing the receipts on to the next one.

To begin with the booth passes the information on the receipt to the web

site to be publicly posted along with the other receipts for the election. The full set of receipts are also passed on to the first trustee.

Each trustee must perform two tasks on the batch of receipts they receive: they strip off a layer of encryption from each of the receipts and they must perform a secret shuffle on the batch. They then post this shuffled, partially decrypted set of receipts to the web site and pass it to the next trustee. This continues through the set of trustees until the last trustee strips off the final layer of encryption to reveal the voter's original ballot images.

The overall effect then is to have posted on the web site, in the left hand column say, the batch of initial receipts posted by the booth. In the right hand column we will have the fully decrypted ballot images. There will also be a set of columns in between with the intermediate, partially decrypted sets of receipts. Each column will be some secret permutation of the previous one. Note that the encryption prevents the permutation being reconstructed by simple matching of elements.

Assuming that all the trustees have performed their transformations correctly, there will be a one-to-one correspondence between the elements of each column and the next. The exact correspondence, which receipt is the decrypt of which receipt in the previous column, will be hidden and known only to the trustee who performed the transformation between those columns. Thus, the receipts will have undergone multiple, secret shuffles between the first column as posted by the booth and the final decrypted column. This ensures that no voter can be linked to their vote, so ensuring voter privacy (ballot secrecy).

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

The decrypted votes will all be available in the final column output of the last trustee and so the overall count will be checkable by anyone. It might be a bit tedious to do by eye for a large election and any automated counting process would have to be dependable (but anyone could write such a process, allowing cross-checking).

4.3 Voter Verification

The description so far has assumed that all the players, the booth and the trustees, have behaved correctly, in accordance with the rules of the scheme. If everyone obeys the rules we can be sure that the election will be both accurate and private. If, however, the booth or any of the trustees cheat then the accuracy and privacy could be undermined. We really don't want to have to put such a level of trust in the components of the scheme. In the words of Kissinger we should "trust but verify!". We really want ways to check on the behaviour of the booth and trustees to catch any attempts to cheat (or indeed to detect innocent malfunctions).

Alice should perform two checks that serve to detect attempts to cheat by the booth. She should run her receipt through a reader device that checks that the receipt has been correctly formed by the booth. Such devices should be

readily available at the voting station for example and provided by independent organisations, such as the Electoral Reform Society or similar. Such devices will perform certain mathematical computations on the data on the receipt according to publically known algorithms. Thus, in principle, anyone could construct such a checker and make it freely available. Similarly anyone would be able to examine such a checker to establish that it was performing correctly. Indeed, if she is really enthusiastic, Alice may choose to run several such independent checks.

Once all the receipt batches have been posted to the web site, Alice should also check that her receipt is accurately recorded there.

All this sounds rather elaborate but is intended to prevent cheating by the booth. Let us consider how the booth might try to cheat. The simplest attack by the booth would be to "lose" the receipt altogether, or to alter it in favour of another candidate. But in this case the voter would either find no record of theie receipt on the website, or find an altered one. In either case they could demonstrate that the booth had cheated.

It might try to arrange for Alice to see the ballot image she expects whilst passing on data to the trustees for the decryption and tallying phase that will yield a different vote once all the decryptions have been performed.

Suppose that a more straightforward implementation of the scheme were used: the booth generates a layer of random noise which is printed on one of the layers. The booth then generates the second layer so that the overlay of the two layers reveals the voter image. This is in fact just the standard implementation of visual cryptography as described in Section 5.1. Alice retains the second layer (note: there would be no point here retaining the first layer as this is pure noise and totally independent of her vote). The booth now passes on a copy of the ballot receipt along with information on how to generate the noise.

Such an implementation is vulnerable to an easy attack by a subverted booth. The difficulty is that there is no mechanism to tie the noise used to generate the visual layer to the noise used by the trustees to reveal the image on the receipt. Thus the booth simply chooses noise on the first layer that shows the voter what they want to see whilst passing noise on to the trustees that will reveal the vote the the booth (or its controller) wants to see. More precisely, the noise passed to the trustees would be the "correct" noise as generated from the ballot serial number to ensure that it would pass any such well-formedness checks.

The physical layer to be discarded could be checked for well-formedness before destruction, but this would require an additional compulsory check between the booth and the shredder, and increase the amount of time that the two physical layers of the receipt are together. This may then open up the possibility of coercion or vote buying, as the voter would be able to demonstrate the way they had voted.

It might seem at first sight that this would be a difficult trick for the booth to pull off: to find two sets of noise such that, when combined with the ballot receipt, one yeilds the voters choice whilst the other yeilds the booth's choice. In fact it is quite trivial, all the booth has to do is to solve the 2 linear equations

in $\mathcal{Z}_2$:

$$\begin{aligned} W \oplus C &= B \\ W' \oplus C &= B' \end{aligned}$$

where W is the OTP printed in the receipt, C the cipher text and B the ballot image seen by the voter, and W' is the OTP passed to the trustees and B' the image the trustees will then reveal. This is really just a manefestation of the well-known fact that for a one-time pad every possible plain text has a corresponding cipher text that will reveal it.

The interweaving of the noise image and the ballot receipt image between the two transparency layers along with the voter choice between the two layers ensures that there is a binding between the noise used in the image and the noise passed to the trustees. More precisely, this ensures that any attempt by the booth to decouple these, along the lines suggested above, runs a 50/50 chance of being detected. The full mathematical details are presented later.

5 Elements of the Chaum Scheme

The Chaum scheme combines three pre-existing mechanisms:

- Visual cryptography
- Anonymising (Chaum) mixes
- Randomised Partial Checking

We introduce each of these in the following sections.

5.1 Visual Cryptography

Visual cryptography was introduced in [12]. Within the Chaum scheme, it provides assurance to the voter that their voting intention has been properly recorded, using an easy visual check. Visual cryptography is in fact an visual implementation of a one-time-pad (OTP). The intuition is that a message or image is encoded as a grid of $m \times n$ black or white cells. In the example in Figure 3, $m = n = 5$.

This is now encrypted using a visual one-time-pad Z, and a cipher text C. These are both composed of $m \times n$ grids of *parity cells*, each of which have one of two forms, see Figure 4.

When Z and C are printed on transparent paper and superimposed, these two possible cell patterns combine visually as shown in Figure 5. (The $\oplus_v$ operator is the visual combination operator.)

Thus, in our example, the one time pad Z might look like the first sheet in Figure 2.

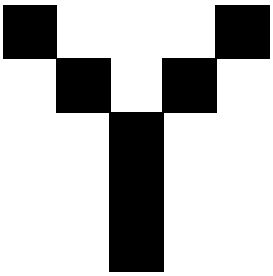


Figure 3: Screen view

Figure 4: The parity cell patterns.

The cipher-text grid C is now computed such that when it is overlaid with Z the image is restored, except that what were clear cells in the original (Figure 3) are now semi-opague in the decrypted image (the overlaid image in Figure 2).

When these cells are printed on transparent foils, then overlaid cells of the same parity give a semi-opaque cell, and overlaid pixels of opposite parity give a fully opaque cell. Thus, the parity cells obey an exclusive or ($\mathcal{Z}_2$) like algebra, under the visual operator $\oplus_v$.

It is important to note that the visual `XOR` is not a closed algebra: the $\oplus_v$ operator can only be applied to the semi-opaque cell patterns, but may result in a fully opaque pixel pattern. However within the context of the Chaum scheme this is perfectly acceptable: only semi-opaque patterns are printed on the transparent sheets, and the operator is not applied more than once.

Let ψ be the mapping back from the parity cells and the fully opaque cell into the binary, represented in figure 6.

The OTP Z and the cipher text C will be internally represented as two binary strings, each mn long. If we let φ (Figure 7) be the translation from these binary strings into parity cells:

then the exact correspondence between $\oplus$ and $\oplus_v$ is

$$\psi(\varphi(x) \oplus_v \varphi(y)) = x \oplus y$$

or categorically in Figure 8.

The scheme could be implemented entirely in software (or hardware) with the XOR being performed, then the result turned into human readable form. The problem with this is that you would have to depend entirely on (trust in) this software. The visual cryptography solution allows all this to be done in a tangible way and assure the voter that the correct vote is buried in the encoding without needing to invoke any intervening software.

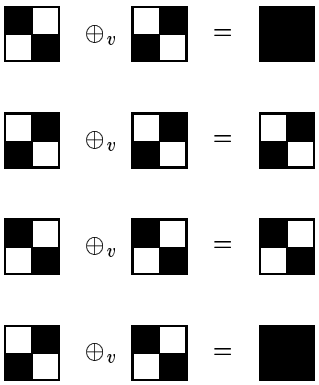


Figure 5: The $\oplus_v$ operator on parity cells.

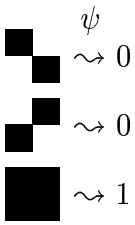


Figure 6: The ψ mapping

5.2 Chaum Mixes

Chaum mixes were originally invented primarily to provide anonymity for email, but the original paper, [5], also suggested their possible application to voting. Note that anonymity here means that an eavesdropper should not be able to determine the communications link between Anne and Bob. Typically it will still be possible for Anne and Bob to authenticate (identify) each other in these interactions. Suppose Anne wants to send a message to Bob. She determines a route via a number of mixer nodes $C_1, C_2, ...C_k$, and forms an onion around the message. The outer layer is an encryption with the public key of C_1. This onion is sent to C_1 who can strip off this outer layer of encryption to reveal the address of C_2 along with a further layer of encryption using the public key of C_2. C_1 then forwards the enclosed message to C_2, who in turn strips off the next layer of encryption to reveal the address of C_3 and another onion encrypted with $C_3's$ public key. This continues until finally, when the C_k layer of encryption is stripped off, the message and final address is revealed. The final message may be further encrypted so as to be readable only by Bob if confidentiality is also required.

If many such messages are flying around the network, it is possible to each of the mix nodes to accumulate a number of messages before outputing them in some scrambled order. This has the effect is disguising the route of any given

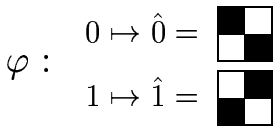


Figure 7: The φ mapping

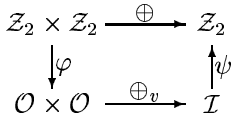


Figure 8: Categorically

message through the network so concealing the link between Anne and Bob.

This idea is readily adaptable to voting schemes, indeed particularly well suited to voting. The idea is to have the voting booth form such onions around each vote and then to have a sequence of trustees acting as mixes to progressively strip off the encryption layers and perform a secret scrambling of the order of the votes. The last of the trustees outputs the votes in clear. In the Chaum scheme, each trustee actually performs two decryption/scrambling transformations. This is for technical reasons associated with the Randomised Partial Checking.

5.3 Randomised Partial Checking

As long as at least one of the trustees can be relied on to perform a genuinely unpredictable and secret permutation on the vote packages, then we will be assured of anonymity. As it stands however this scheme provides no assurance of accuracy. Any of the players could falsify votes by dropping some, injecting fake ones or altering some. This is where the Randomised Partial Checking (RPC) comes in.

The idea of Randomised Partial Checking [8] is employed to ensure that the chance of any given trustee cheating undetected is extremely small. This ensures that the chance of the trustee cheating undetected on p votes diminishes exponentially with p.

As noted previously, each trustee performs two decryption/permutation transformations in sequence. At a later stage, each of the trustees is required to reveal a randomly selected half of the links of the first transformation. They are also required to reveal half of the links of their second transformation but these are chosen to be disjoint from the first set of links, i.e., none of the links line up, and so there can be no complete path from the encrypted to the decrypted vote. Hence it is not possible to find a vote and track it through both transformations, so ensuring that no input onion can be associated with the corresponding output onion.

On each of these revealed links we can now check that the trustee has correctly performed the decryption. We leave the details of these checks to the

mathematical description section. As with the checks peformed on the ballot receipt outside the polling station, the mathematical details of how to perform these checks is publically known so, in principle, anyone can code up the algorithms and perform the checks.

6 Assurance

The Chaum scheme can be usefully viewed in terms of the dependability conceptual model, see for example [10].

In this section, we use the Chaum scheme as a vehicle to illustrate and discuss approaches to, and challenges, to the establishment and maintenance of assurance in a complex, computer-based system.

We take the term "assurance" to mean the acquisition of confidence that the behaviour of a particular system will be in accordance with certain requirements.

Techniques for gaining assurance in the correct behaviour of a system with respect to some requirement can be positioned along a spectrum: at one end there is the pure verification approach, whilst at the other end is that of monitoring for any departures from the requirement (failures in the dependability terminology). Testing can be thought of as lying somewhere between these two extremes. of course, in practice we should not confine ourselves to employing a single approach, but should combine approaches drawn from points along the spectrum.

Diebold appears to lie at the verification end of the spectrum. More precisely, the Diebold system does not provide any means to monitor the system performance at run-time and so any assurance we may have must rely on (claims of) verification of the code. The problem with such an approach is that:

- Full verification of even modest systems is notoriously difficult and error prone. To quote Needham, speaking of security protocols: "they are 3 to 5 line pieces of code that people still manage to get wrong".
- Even supposing for a moment that we do succeed in verifying the system, we still need mechanisms to guarantee that the system that is actually fielded does exactly match the system that was verified. This calls for supervised loading of code, tamper resistant/evident devices etc.
- Verification is of necessity with respect to various assumptions, either explicit about the environment or implicit in the models used for the verification. If any of these assumptions are invalidated then the whole basis of the verification may be rendered invalid.

The Chaum scheme lies essentially at the other end of the spectrum, at least with respect to the accuracy requirement. Assurance that votes will be accurately registered does not depend on placing trust in components of the system. The behaviour of each of the components is closely monitored and any deviation from the specified behaviour is, with high probability, detectable.

Once a failure of a component is detected, various recovery mechanisms can be deployed:

- Isolation of the offending component.
- Roll-back to an earlier, correct stage of the computation and recomputation.

The situation is subtly different for the voter anonymity (secrecy) requirement. Anonymity is an information flow, and hence non-enforceable property [15]. Consequently, failures with respect to anonymity will not in general be detectable. Furthermore, even where such failures are detected, it is difficult to recover: once a secret is out, it is out. Consequently it is essential that the secrecy properties be designed in and verified. Thus, it is important that the trustees be designed to implement truly random shufflings on the ballot batches and that they can be relied on to keep these secret.

Here, rather unusually for a secrecy property, replication comes to our help. The fact that the scheme employs a number of trustees, assumed independent, means that all would have to be compromised in order to violate voter anonymity.

Note also that diversity has a role to play in the scheme. The fact that the checks are all according to publically known algorithms means that anyone can perform the checks are provide implementations of checking devices and implementations of the algorithms.

7 The Protocol for a single vote

In this section, we present in detail the Chaum protocol for a single vote, explaining the communications and calculations that must take place at each stage.

Notation

Note that our notation differs in places from [4].

- $\{t\}_e$ represents the encryption of t under an encryption key e.
- n is the number of trustees.
- k is the number of layers of encryption on each vote, and $k = 2n$.
- Information internal to the booth and trustees will frequently be stored and transmitted as mn bitstreams, but presented to the reader as $m \times n$ matrices. We perform will perform the translation between the $(m \times n)$ matrix and the mn bitstream as

  $$B^{mat}_{i,j} = B_{(j-1)m+i}$$

  where B^{mat} is the matrix form, and B the bitstream.

- We use the conventional protocol notation

$$Anne \rightarrow Bob \quad : \quad M$$

to mean that Anne sends message M to Bob.

Step 1

The voter enters a vote, B, into the machine. This will be chosen from a list presented by the machine to the voter. Each possible vote will be presented as a $m \times n$ matrix on the screen in order to be readable by the human voter.

We write the first step as

$$(1) \quad voter \rightarrow machine \quad : \quad B$$

Step 2

The next communication step is the voting machine printing the first portion of the receipt. However before this can happen, the machine must do the following calculations.

7.1 Constructing the dolls

Before the voter chooses to take either the top or the bottom layer of the receipt, the machine must define the two "dolls"[2] ${}^{t}D$ and ${}^{b}D$. (${}^{t}D$ is the doll associated with the top layer, and ${}^{b}D$ is the layer associated with the bottom layer.) One of these dolls will later be used by the trustees to decrypt the chosen layer. The one not chosen will have its seed revealed in the receipt, and an independent checker will be able to verify that the doll was correctly formed, (see Check 4.3), but without revealing any information about the vote. This provides some assurance of integrity: A malicious machine which wanted to produce a corrupt doll to send to the trustees would have a one-in-two chance of having the corrupt doll checked in Check 4.3.

The two decryption dolls ${}^{t}D$ and ${}^{b}D$ are each made up of k layers, where k is the number of partial decryptions that will take place. Each layer is "wrapped" with a different public key/private key decryption function. The public keys are known to everyone; each private key is known only to the relevant trustee. When that trustee "unwraps" their layer enough information is available for the trustee to calculate their part of the decrypting key.

To build the dolls(Figure 9) the voting machine requires

- q: a sequence number (which can be consecutive or linked to the voter) There is a unique sequence number for each voter.

[2]Note, the terminology of "onions" tends to be used in the context of anonymising mail. These are essentially the same as the "dolls" used in the voting scheme.

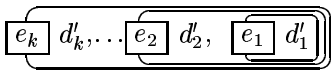


Figure 9: The Layers of a Doll.

- h, h': these are pseudo-random sequence functions which when composed yield a pseudo-random binary sequence of length $mn/2$;
- ${}^{t}s$ and ${}^{b}s$: private signature functions known only to the voting machine.
- e_l, where $1 \leq l \leq k$: these are public encryption keys of the trustees. For each key, the private counterpart is known only by the trustee in charge of that part of the decryption (see Section 7.3).

For both dolls, the machine prepares k variables d'_l, by seeding the function h as below. For $1 \leq l \leq k$,

$$
\begin{aligned}
{}^{t}d'_l &:= h(\{q\}_{{}^{t}s}, l) \\
{}^{b}d'_l &:= h(\{q\}_{{}^{b}s}, l)
\end{aligned}
$$

and then prepares the two k-layered dolls, by encrypting each layer with a different encryption key e_l. For $1 \leq l \leq k$,

$$
\begin{aligned}
{}^{t}D &= e_k({}^{t}d'_k, \ldots e_2({}^{t}d'_2, e_1({}^{t}d'_1))) \\
{}^{b}D &= e_k({}^{b}d'_k, \ldots e_2({}^{b}d'_2, e_1({}^{b}d'_1)))
\end{aligned}
$$

7.2 Constructing the layers of the printed receipt

${}^{t}L$ and ${}^{b}L$ (see Figure 10) are what is printed on the top and bottom layers of the receipt. They are printed so that when aligned directly on top of each other the voter can see the chosen ballot image B, but when observed individually they offer no information to the observer.

This relies on the "visual cryptography" of [12], explained in Section 5.1. The calculation proceeds by forming top and bottom "white" matrices ${}^{t}W$ and ${}^{b}W$, each of size $m \times n/2$. Each of these is built by `XOR`ing k $mn/2$ bitstrings (the ${}^{t}d_i$ and ${}^{b}d_i$ of Figure 10) together, then converting the resulting bitstrings into $m \times n/2$ matrices.

The $mn/2$ bitstreams are generated pseudo-randomly using the $2k$ d' variables generated previously as seeds for h'.

For $1 \leq l \leq k$,

$$
\begin{aligned}
{}^{t}d_l &:= h'({}^{t}d'_l) \\
{}^{b}d_l &:= h'({}^{b}d'_l)
\end{aligned}
$$

We produce the ${}^{t}W$ and ${}^{b}W$ matrices from all these bitstrings by `XOR`ing them together and forming the matrices as follows (arrows 1 and 2 in Figure 10):

$$
\begin{aligned}
{}^{t}W_{i,j} &:= ({}^{t}d_k \oplus {}^{t}d_{k-1} \oplus \cdots \oplus {}^{t}d_1)_{(j-1)m+i} \\
{}^{b}W_{i,j} &:= ({}^{b}d_k \oplus {}^{b}d_{k-1} \oplus \cdots \oplus {}^{b}d_1)_{(j-1)m+i}
\end{aligned}
$$

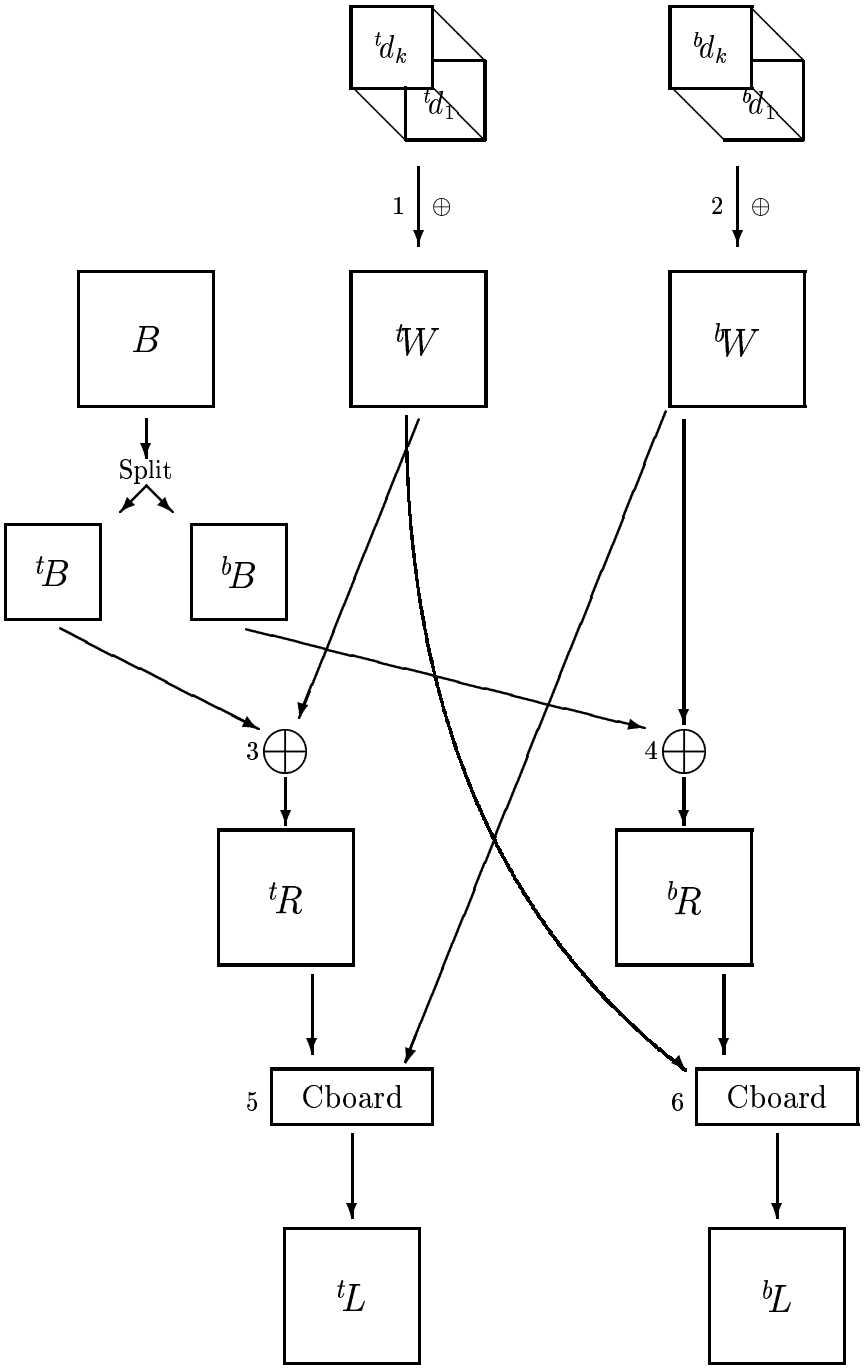


Figure 10: Diagram of step 2: Constructing the receipt

The two parts ${}^tR$ and ${}^bR$ are constructed using ${}^tW$ and ${}^tW$ and the ballot image B supplied by the voter. Consider the ballot image in the matrix form B^{mat}. This matrix is broken into ${}^tB$ and ${}^bB$ (each of size $m \times n/2$) as follows. Each row of B^{mat} is m bits long. Alternate pixel symbols on each row of B^{mat} are extracted and coalesced to form the matrices ${}^tB$ and ${}^bB$ (operations 3 and 4 in Figure 10) This is the "checkerboarding" process mentioned in the original paper [4]. So in Figure 11 the squares coloured white are used to form ${}^bB$, and the squares coloured black are used to form ${}^tB$.

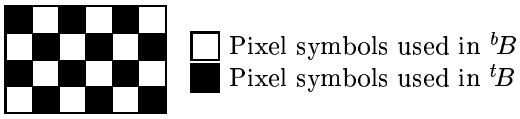


Figure 11: The checkerboarding process

More formally, this is achieved as

$$\begin{aligned} {}^tB_{i,j} &:= B^{mat}_{i,2j-(i \bmod 2)} \\ {}^bB_{i,j} &:= B^{mat}_{i,2j-(i+1 \bmod 2)} \end{aligned}$$

The "red" matrix R is designed so that when it is aligned with the white matrix W the ballot image is produced. In the same way that the white matrix is "checkerboarded" into two matrices ${}^tW$ and ${}^bW$, the matrix R is checkerboarded into two matrices ${}^tR$ and ${}^bR$. Each of ${}^tL$ and ${}^bL$ (the matrices which will be printed on the receipt) will be made from a meshing together of a red and white matrix.

The two parts ${}^tR$ and ${}^bR$ are constructed using the operator $\oplus_v$, the visual combination operator, which obeys the rules in Figure 5.

Formally, the matrices ${}^tR$ and ${}^bR$ are determined so that the two equations

$$\begin{aligned} {}^tR \oplus_v {}^bW &= {}^tB \\ {}^bR \oplus_v {}^tW &= {}^bB \end{aligned} \tag{1}$$

hold. The ${}^tR$ pixels are derived from both the ${}^bW$ and the ${}^tB$ pixels, and similarly ${}^bR$ pixels are derived from both the ${}^tW$ and the ${}^bB$ pixels. Because the W and B matrices are already determined, there is a unique solution for both ${}^tR$ and ${}^bR$.

The printed layers ${}^tL$ and ${}^bL$ are now constructed as follows. The top printed layer (${}^tL$) is the checkerboarding of the ${}^tR$ matrix and the ${}^tW$ matrix, and the bottom printed layer (${}^bL$) is the checkerboarding of ${}^bR$ and ${}^bW$ (operations 5 and 6 in Figure 10). Formally,

$$\begin{aligned} {}^tL_{i,2j-(i+1 \bmod 2)} &:= {}^tW_{i,j} \\ {}^tL_{i,2j-(i \bmod 2)} &:= {}^tR_{i,j} \end{aligned}$$

and

$$\begin{array}{rcl} {}^{b}L_{i,2j-(i+1 \bmod 2)} & := & {}^{b}R_{i,j} \\ {}^{b}L_{i,2j-(i \bmod 2)} & := & {}^{b}W_{i,j} \end{array}$$

The machine prints the first parts of each layer of the receipt as

$$\begin{array}{lrcl} (2a) & machine \rightarrow voter & : & {}^{t}L, \overline{q}, \overline{{}^{t}D}, \overline{{}^{b}D} \\ (2b) & machine \rightarrow voter & : & {}^{b}L, q, {}^{t}D, {}^{b}D \end{array}$$

where $\overline{q}$, $\overline{{}^{t}D}$ and $\overline{{}^{b}D}$ are the complemented forms of q, ${}^{t}D$ and ${}^{b}D$.[3]

Checks 2.1 & 2.2. The voter checks (visually) that ${}^{t}L \oplus_{v} {}^{b}L = B$. This means checking that the chosen ballot image is readable through the two transparent layers of the receipt. The voter also checks (again visually) that second, third and fourth components of the two layers are true complements. This means that the part of the receipt on which these components are printed is a solid black.

Steps 3 & 4.

The voter now decides which layer of the receipt to retain at the end of the transaction, and communicates this choice to the voting machine.

$$(3) \quad voter \rightarrow machine \quad : \quad x, \text{ where } x \in \{t, b\}$$

The machine now encrypts the five-tuple $\{{}^{x}L, q, {}^{t}D, {}^{b}D, \{q\}_{x_S}\}$ with ${}^{x}o$ and prints it and the previously calculated $\{q\}_{x_S}$, on the top layer of the receipt. The complements of these components are printed on bottom layer.

$$\begin{array}{lrcl} (4a) & machine \rightarrow voter & : & \{q\}_{x_S}, \{{}^{x}L, q, {}^{t}D, {}^{b}D, \{q\}_{x_S}\}_{x_O} \\ (4b) & machine \rightarrow voter & : & \overline{\{q\}_{x_S}}, \overline{\{{}^{x}L, q, {}^{t}D, {}^{b}D, \{q\}_{x_S}\}_{x_O}} \end{array}$$

Check 4.1 The voter checks visually that the remainder of the two layers are true complements, (again by observing a uniform black) then takes the receipt and leaves the booth. Outside the booth, the voter surrenders the layer not chosen to the poll worker. The voter now has a single sheet, comprising the six-tuple

$$\langle {}^{x}L, q, {}^{t}D, {}^{b}D, \{q\}_{x_S}, \{{}^{x}L, q, {}^{t}D, {}^{b}D, \{q\}_{x_S}\}_{x_O} \rangle$$

(We assume without loss of generality that the last two components of the voters sheet are not in complemented form.) We will call this the retained layer.

Three checks outside the booth are made using a barcode reader. For the first two, the barcode reader must know

- ${}^{t}s^{-1}$, ${}^{b}s^{-1}$, ${}^{t}o^{-1}$, and ${}^{b}o^{-1}$: The public keys corresponding to ${}^{t}s$, ${}^{b}s$, ${}^{t}o$, and ${}^{b}o$, the private keys of the booth.

[3] Choosing the layer on which to print the complemented forms is an arbitrary design decision not made in [4].

Check 4.2. The voter checks (using barcode reader) that $\{\{q\}_{x_S}\}_{x_S{}^{-1}} = q$.
Check 4.3. The voter checks (using barcode reader) that

$$\{\{{}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}\}_{x_O}\}_{x_O{}^{-1}} = {}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}$$

Check 4.4. The voter checks (using barcode reader) that

$${}^xD = e_k({}^xd'_k, \ldots e_2({}^xd'_2, e_1({}^xd'_1)))$$

where

$${}^xd'_l = h(\{q\}_{x_S}, l) \text{ for } 1 \leq l \leq k$$

To make check 4.3, the bar code reader must know $\{q\}_{x_S}$ (available from the retained layer), the pseudo-random sequence function h and each of the trustee's public encryption keys e_l. With this information the doll ${}^xD$ can be entirely reconstructed (by publically available software within the barcode reader) and compared with the copy on the receipt. This ensures that if a malicious voting machine shows the voter one image, then creates a false doll which will produce the wrong image, it has a one in two chance of being caught out before the decrypting process begins.

Steps 5 & 6.

The entire retained layer is posted by the voting machine onto the election website. It can therefore be considered publically available. Because each trustee in fact does two decryptions, we will let the number of trustees be n, where $k = 2n$, and k is the number of decryptions performed. The voting machine also passes the retained layer onto the the nth trustee.

(5) $\quad machine \rightarrow website \quad : \quad {}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}, \{{}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}\}_{x_O}$
(6) $\quad machine \rightarrow trustee\, n \quad : \quad {}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}, \{{}^xL, q, {}^tD, {}^bD, \{q\}_{x_S}\}_{x_O}$

7.3 The work of the trustees

Let y be the discarded layer. (i.e. set $y \in \{t, b\}$ and $y \neq x$.) Recall ${}^xL$ is formed by the checkerboarding of ${}^xW$ and ${}^xR$, and that ${}^yL$ is formed by the checkerboarding of ${}^yW$ and ${}^yR$. Recall also (from Equation 1) that ${}^xB$ is the visual composition of ${}^yW$ and ${}^xR$. The task of the trustees is therefore to progressively subtract ${}^yW$ from the ${}^xR$ component of ${}^xL$. This will leave ${}^xB$, which, although it is not the entire ballot image, is enough of the ballot image (because of the pixel redundancy at the font level) to determine the original vote cast.

Trustee n uses its first private decryption key e_k^{-1} (recall $k = 2n$) to decrypt ${}^yD_k$ and calculate the (partial seed, doll) pair.

$$\{{}^yD_k\}_{e_k^{-1}} := {}^yd'_k, {}^yD_{k-1}$$

then uses h' (which is known by all the trustees and auditors) to calculate

$$h'({}^{y}d'_k) := {}^{y}d_k$$

Recall that ${}^{x}R$ is the checkerboarded bit of ${}^{x}L$ which is read using ${}^{y}W$. Let ${}^{x}R_k = {}^{x}R$. Trustee n performs the first part of the decoding of the ballot image by `XOR`ing ${}^{y}d_k$ and ${}^{x}R_k$ as below.

$$ {}^{y}d_k \oplus {}^{x}R_k = {}^{x}R_{k-1}$$

and posts the ${}^{x}R_{k-1}$ on the website, along with the ${}^{y}D_{k-1}$.

$$(7) \quad trustee\, n \rightarrow website \quad : \quad {}^{x}R_{k-1}, {}^{y}D_{k-1}$$

To facilitate the revealing of links (see Section 8) each trustee must perform the above sequence twice. Trustee n therefore performs the above sequence again, this time using its second secret decryption key e_{k-1}^{-1}

$$\{{}^{y}D_{k-1}\}_{e_{k-1}^{-1}} := {}^{y}d'_{k-1}, {}^{y}D_{k-2}$$

The same function h' is used to generate the next part of ${}^{y}W$:

$$h'({}^{y}d'_{k-1}) := {}^{y}d_{k-1}$$

and the result used to perform the second decryption

$$ {}^{y}d_{k-1} \oplus {}^{x}R_{k-1} := {}^{x}R_{k-2}$$

${}^{x}R_{k-2}$ and ${}^{y}D_{k-2}$ are then posted on the website.

$$(8) \quad trustee\, n \rightarrow website \quad : \quad {}^{x}R_{k-2}, {}^{y}D_{k-2}$$

This time, trustee n passes ${}^{x}R_{k-2}$ and ${}^{y}D_{k-2}$ on to trustee $n-1$. (or, equivalently, trustee $n-1$ reads them from the website.) Trustee $n-1$ decrypts, decodes, and posts

$$(9) \quad trustee\, n-1 \rightarrow website \quad : \quad {}^{x}R_{k-3}, {}^{y}D_{k-3}$$

and

$$(10) \quad trustee\, n-1 \rightarrow website \quad : \quad {}^{x}R_{k-4}, {}^{y}D_{k-4}$$

Each trustee repeats this pattern, and when trustee 1 has finished, the result posted on the website is ${}^{x}R_0 (= {}^{x}R)$, which is human readable.

The posting are permuted, as explained below.

8 Mixing the votes, and providing link information

In the previous section, we described the progress of an individual vote through the system. In this section, we describe the mechanism for ensuring that a particular decrypted vote cannot be traced back to its original encrypted form. As explained earlier, the encrypted votes ${}^{x}L$ and their corresponding dolls ${}^{y}D$ are kept together during the decryption process. In the language of [4] they are formed into *duos*.

At the start of the decrypting process, before the duos are first published on the web, they are grouped into sets of duos, called batches. These batches will be treated separately, and votes will be shuffled around within these batches at each stage of the decrypting process (see Figure 12.)

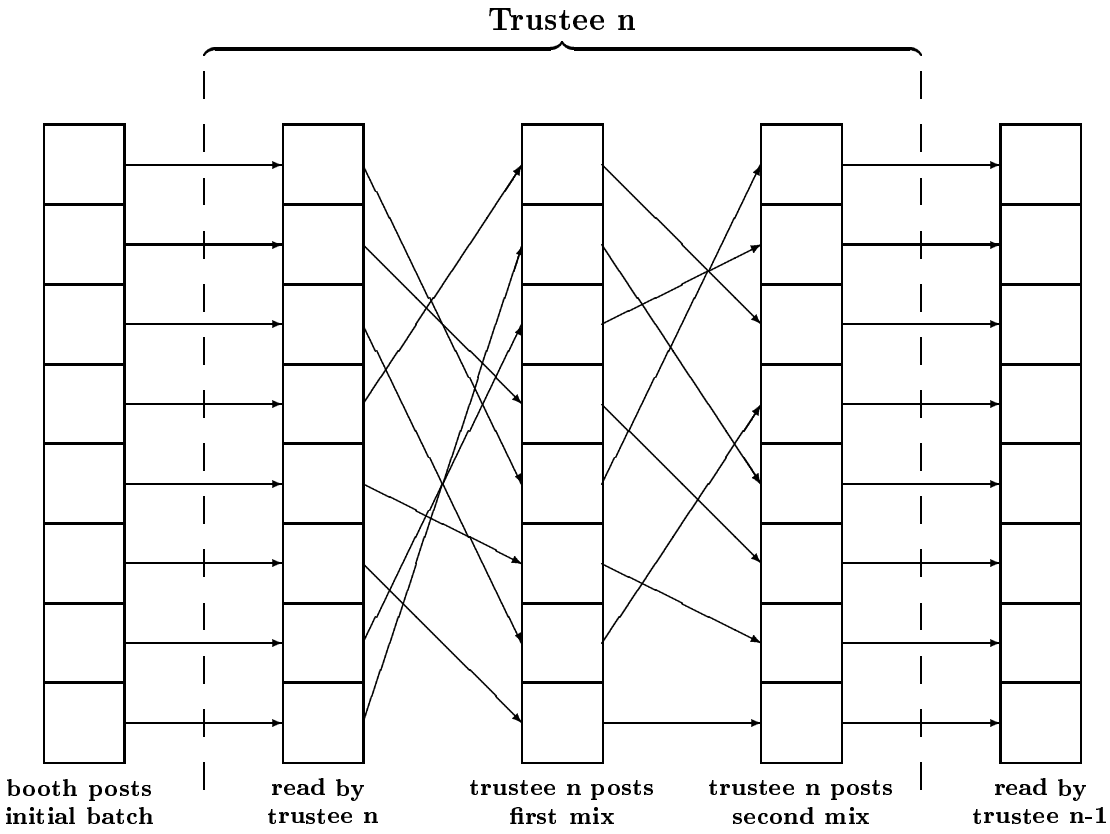


Figure 12: All links

We follow the progress of one such batch.

The (fully-encrypted duos) are published on to the web by the voting machine(Figure 12, initial posting). After the trustee has opened the outside of

each of the dolls, and done the partial decryption of all the votes in the batch, they are again published on the web, in a different, randomised, order. (This posting is also called a *mix* in [4].) The order is chosen by, and known only to, the trustee doing the decrypting. A record of the permutation used is retained by the trustee, but not published.

Recall that each trustee performs two decryptions on each duo, and therefore each trustee performs two permutations on each batch. After trustee n has finished, trustee $n-1$ reads the results from the website and performs two (decryption, permutation) pairs, publishing the results on the website as trustee n did (again see Figure 12.)

However, if no further information about the process was revealed, trustees could alter votes without fear of detection. For example, if the last trustee simply published all the final decrypted votes as being in favour of a particular candidate, it would be difficult to prove that anything illegal had taken place. A solution proposed in [4] is derived from [8]. Trustees are required to provide *some* linkage information between the votes: enough to make the possibility of successful corruption acceptably small, but not enough to allow any of the final decrypted votes to be traced back to the original votes cast.

The partial decryption postings for each batch are grouped into contiguous sequences of four postings. For each posting half of the input links and half of the output links are revealed, in the following way. Take the first posting of a sequence of four. Some external authority choses a random half of the duos in the first posting. For each chosen duo (R, D) the responsible trustee must reveal: (1) the d' extracted from the D, and (2) the target duo in the subsequent posting (i.e. the *link*).

In the second posting all the duos not pointed to by those opened in the first batch are opened. (An example first and second opening is shown in Figure 13.)

Crucially, in the third posting, exactly *half* of the duos pointed to by the second set of links are opened, and *half* of the duos not pointed to by the second set of links are opened.

In the fourth posting, *all* the duos not pointed to by the third set of links are opened.

No full link from a single final vote to a single original vote can therefore be drawn.

In this way the choice of which links are to be revealed is independent for each trustee, and a rapid mixing of duos is ensured.

8.1 Checks on the revealed links

If the duo has been transformed correctly then each link should be of the form:

$$R_l, D_l \quad \longrightarrow \quad R_{l-1}, D_{l-1}$$

where

$$D_l \quad := \quad \{d'_l, D_{l-1}\}e_l$$

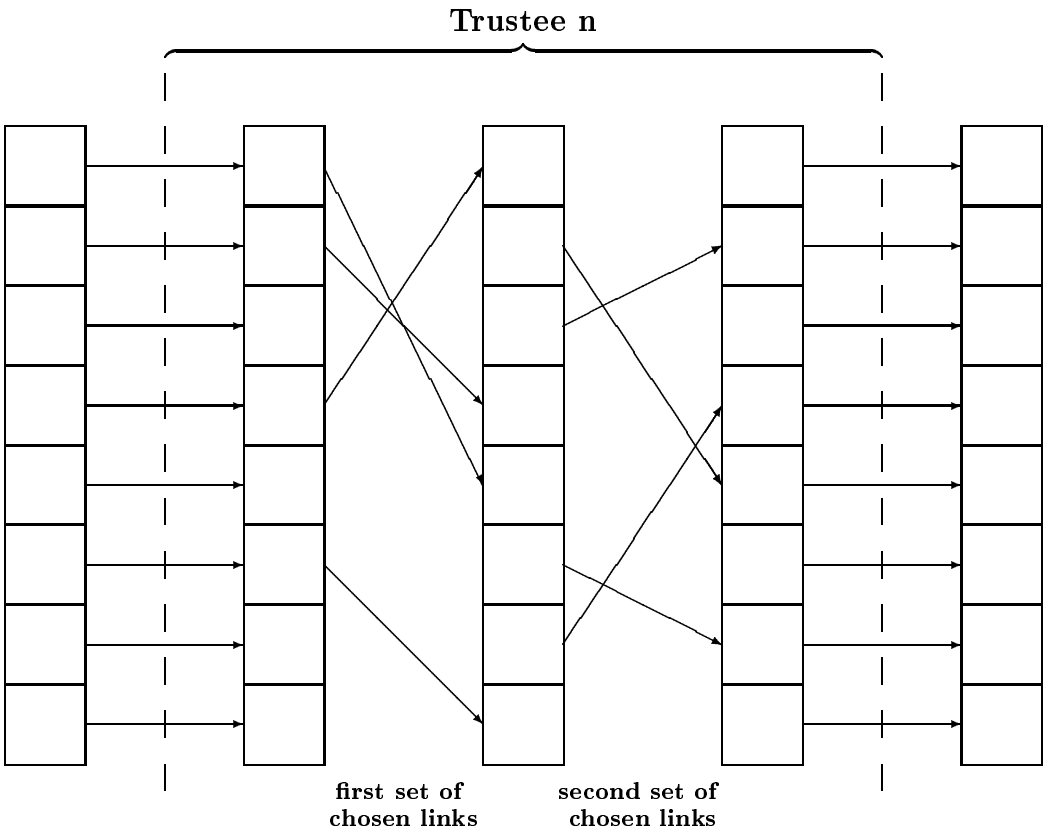


Figure 13: Revealed links from trustee n's permutations

$$\begin{aligned} D_{l-1} &:= \{d'_{l-1}, D_{l-2}\}e_{l-1} \\ R_{l-1} &:= R_l \oplus d_l \\ d_l &:= h'(d'_l) \end{aligned}$$

Note that, when a link is revealed, the corresponding d'_l is revealed.

Auditors can therefore compute

$$h'(d'_l)$$

and check that this equals

$$R_l \oplus R_{l-1}$$

The key e_l is public so the auditor can also calculate

$$e_l(d'_l, D_{l-1})$$

and check that this equals D_l.

8.2 The role of h'

Using the pseudorandom function pair h and h' is significant in that it foils a potential attack on the secrecy. Knowledge of the d'_l value is essential when the auditor performs the check outlined above. As this is the pre-image of h' it should be intractable to compute this from knowledge of the final output, the d_l value.

If the checks could be performed knowing only the putatively linked duos then the scheme would be vulnerable to a guessing attack. This would proceed as follows: Given a putative link, compute the checks. If the checks work you have, with high probability, identified a valid link. In this fashion you could, in principle, reconstruct as much of the secret permutation as needed.

Without knowledge of the d'_l pre-images, such an attack is intractable.

9 Conclusions

In this document we have laid out in detail our understanding of David Chaums digital voting scheme. We have discussed the unique combination of requirements that a digital voting scenario necessarily poses, and we believe that the Chaum voting scheme comes closer than any other scheme we are currently aware of to meeting all of these requirements.

We believe, however, that providing a system to implement (digital) voting is not purely a technical question. The larger socio-technical system encompassing the technical system must also be considered, and the social and socio-technical issues raised must be addressed. The resulting system must not only be trustworthy but be believed to be trustworthy by the electorate.

10 Acknowledgements

We are grateful to all our DIRC colleagues who took time to work with us on this project. We are also in debt to David Chaum, for his patient answering of our many questions.

References

[1] http://www.verifiedvoting.org.

[2] http://www.diebold.com.

[3] http://avirubin.com/vote/.

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

[5] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb 1981.

[6] The Electoral Commission. Modernising Elections: A strategic evaluation of the 2002 pilot schemes. http://www.electoralcommission.gov.uk/about-us/modernisingelections.cfm.

[7] Marcin Gomulkiewicz, Marek Klonowski, and Miroslaw Kutylowski. Rapid mixing and security of Chaum's visual electronic voting. In *ESORICS*, 2003. To appear.

[8] M. Jakobsson, M. Juels, and R. Rivest. Making Mix Nets Robust for Electronic Voting by Randomised Partial Checking. In *USENIX'02*, 2002.

[9] Tadayoshi Kohno, Adam Stubblefield, Aviel Rubin, and Dab Wallach. Analysis of an electronic voting machine. http://avirubin.com/vote/.

[10] http://www.newcastle.research.ec.org/maftia.

[11] Rebecca Mercuri. Questions for voting systesm vendors. http://www.notablesoftware.com/checklists.html.

[12] M. Noar and A. Shamir. Visual Cryptography. In A. De Santis, editor, *Advances in Cryptography - Eurocrypt'94*, volume 950 of *LNCS*, pages 1–12, Berlin, 1995. Springer Verlag.

[13] Office of the E-envoy. http://www.edemocracy.gov.uk, July 2002.

[14] Peter Ryan. Mathematical models of computer security. In Riccardo Focardi and Roberto Gorrieri, editors, *Foundations of Security Analysis and Design*, volume 2171 of *LNCS*, pages 1–62, 2000.

[15] F. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, February 2000.

[16] Steve Schneider and Abraham Sidiropoulos. CSP and Anonymity. In *ESORICS*, volume 1146 of *LNCS*, 1996.

[17] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code*. John Wiley and Sons, New York, 1996. Second edition.

[18] Geoffrey Willans and Ronald Searle. *Compleet Molesworth*. Pavilion Books, 1984.