Security and Trust in the Chaum Voting Scheme

Jeremy Bryans and Peter Ryan
School of Computing Science, Newcastle University, UK
{Jeremy.Bryans|Peter.Ryan}@newcastle.ac.uk

Abstract

We describe the dependability requirements accuracy, ballot secrecy, recoverability, of digital voting schemes. We then give an overview of the Chaum digital scheme [2] and describe the extent to which the scheme achieves the requirements. In particular we describe the threat model, assumptions and the role of the scheme's mechanisms in countering threats. Finally we discuss how public trust in such a system might be engendered and maintained.

1 Introduction

More technologically sophisticated alternatives to the traditional pen and paper methods of casting and counting votes are currently being investigated. The UK government has stated it's enthusiasm for such schemes [9] and a number of trials have been performed, e.g. [3]. 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 the Chaum scheme [2] achieves these goals.

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 vote over a home computer. In this paper we are interested in schemes in which digital technology is used to improve the voting process, but may still involve certain purpose-designed equipment. 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.

In Section 2 we consider the requirements of a digital voting scheme would have to meet, then in Section 3 we present a brief overview of the Chaum voting scheme. Sections 4 and 5 detail the checks which can be performed at different stages of the voting process, In Section 6 we discuss the threat and trust model for the scheme, in Section 7 we look at the issue of public trust in digital voting, and we draw some conclusions in Section 8.

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. A secondary requirement is for failures with respect to the primary requirements to be detectable and recoverable. Ideally a voting scheme should also be usable, efficient, unbiased and scalable but we will not discuss these aspects here. These aspects are being addressed in the context of a broader, interdisciplinary study of digital voting schemes being conducted by the DIRC project, [5].

Note that there is a tension between the requirement for ballot secrecy and that of traceability and recoverability. A naive implementation of traceability would immediately violate secrecy. In [2], Chaum conjectures that it is impossible to achieve absolute assurances of unconditional accuracy and privacy 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 a election, referendum etc to accurately reflect the "intentions" of the eligible electorate. At this level we will need to consider social, psychological issues that might favour certain sectors of society, might bias choices or encourage voter error and so on. These aspects are being investigated in the DIRC project.

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 votes cast. We will assume that issues of authentication and prevention of double voting have been addressed.

In practice, absolute assurance of complete accuracy is not feasible and, arguably, too strong a requirement. The real requirement is that the outcome be "correct", e.g. that the candidate with the largest number of votes wins. The Chaum scheme provides good probabilistic assurances: the chance of p votes being corrupted undetected diminishes exponentially with p. As a result, tight statistical bounds can be placed on the probability of the outcome of an election being swung by corruption of votes.

2.2 Ballot Secrecy

It will typically be a requirement that the way any individual voter voted remain secret. For some forms of vote this might not be required, for example voting in the UK House of Commons. For this paper we will assume that ballot secrecy is a requirement. 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 [4]. 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 some information leakage. In [4] 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 [11] 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 [10]. Details of the application of these ideas to the Chaum scheme are given in [1].

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 Traceability and Recoverability

Absolute guarantees are not feasible. System malfunctions and compromises will occur. It is essential therefore that mechanisms be provided to detect, contain

[1] We are reminded of the character in one of the Molesworth books [12] who, asked by a teacher to consider a right angle triangle with squares on all three sides, asks "Is that really very likely Sir?".

and recover from failures. These mechanisms need to be robust in the face of malicious as well as accidental threats.

Many of the schemes that have been proposed for digital or electronic voting have little or nothing in the way of mechanisms to detect corruption or falsification of votes. The Chaum scheme, by contrast, provides the voter with the possibility of verifying that their vote is accurately included in the final tally. Auditing mechanisms are also in place to detect irregularities in the counting of votes, and stages of the process can easily be rerun using independent machines.

Several experts in the field have argued that any digital voting system must be backed up by a form of paper audit trail. Although this is a reasonable interpretation of the traceability requirement, it is not necessarily the most appropriate. The real requirements should be framed at a higher level of abstraction. Paper audit trails can then be seen as one, but by no means the only, implementation of this more abstract requirement.

3 Overview of the Chaum digital voting scheme

The Chaum scheme 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, in principle, 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 truism 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 coercion and 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, [7]:

"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 in the Chaum scheme. Due to space constraints we cannot give full details of the scheme but refer the reader to [2, 1].

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.

3.1 Alice casts a vote

Let us suppose that Alice is our intrepid voter. She will show up at a voting station and authenticate herself in some way. The details of how voters authenticate themselves and what measures are taken to prevent double voting etc will be not be addressed here, we will simply assume 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 and registered, Alice is ready to cast her vote. The booth presents her with a choice of alternatives, displayed on a screen for example (for a simple yes/no referendum) in Figure 1.

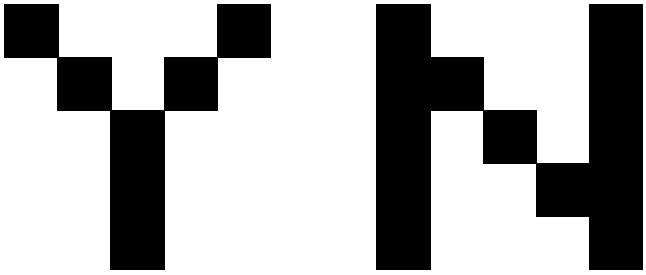


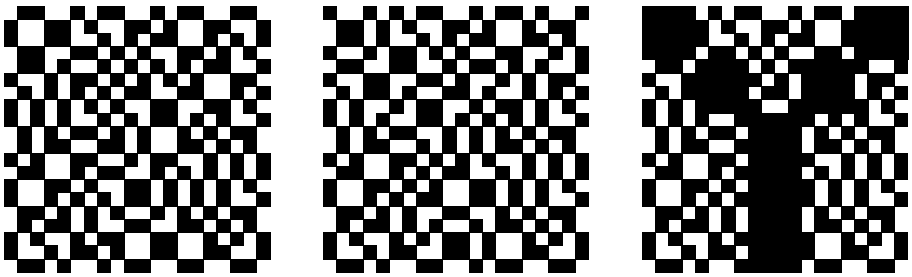
Fig. 1. Screen view

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 as in Figure 2, using the optical cryptography of [8]. This involves first creating a "One-Time Pad" (OTP), then creating a cipher text, as in for example Figure 2. Individually each of these is unreadable, but when the patterns are overlaid her selection is visible as a ballot image formed out of a pattern of opaque and semi-opaque cells, as in Figure 2.

Although the obvious way to print these images would appear to be to print the OTP on one layer and the cipher text on the other, this would in fact be vulnerable to a attack, detailed in Section 5. Instead, the OTP and the cipher text are interwoven, so that on one layer every even cell is taken from the same cell in the cipher text image, and every odd cell is taken from the same cell in the OTP, while on the other layer every even cell is taken from the OTP, and every odd cell is taken from the cipher text.

The creation of the OTP itself is described in Section 3.2.

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. Suppose, without loss of generality, Alice chooses the top layer. Once she has made this choice, some further information is printed onto both sheets alongside the pixel patterns printed earlier. Mathematically, the retained layer now comprises the



One-Time Pad Cipher Text Overlaid Image

Fig. 2. Construction of the Ballot Image

six-tuple

$$\langle {}^tL, q, {}^tD, {}^bD, \{q\}_{t_S}, \{{}^tL, q, {}^tD, {}^bD, \{q\}_{t_S}\}_{t_O} \rangle$$

where

- ${}^tL$ is the encrypted layer of the ballot image (unreadable of itself).
- q is the ballot receipt serial number.
- ${}^tD$ and ${}^bD$ contain information on how to reconstruct the noise used to encrypt the ballot image. This information is itself encrypted in such a way as to be readable only by a collaboration of all the trustees, and its construction is detailed shortly.
- $\{q\}_{t_S}$ is the serial number encrypted with the one of the booths secret keys, and
- the final item is a encryption of the first five items under another of the booths secret keys.

Alice now detaches the sheets from the printer and separates them before leaving the both. 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.

Before leaving the voting station she will be encouraged to run her receipt through a device that can read the information on the receipt and 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.

3.2 Creating and Transmitting the One-Time Pad

The OTP is actually created as two independent "half-size" OTPs, which we call ${}^tW$ and ${}^bW$. ${}^tW$ will end up wholly on the top layer (${}^tL$), and ${}^bW$ wholly on the bottom layer (${}^bL$). Because only the voter's choice of receipt layer will be decrypted, only one of these OTPs will need to be applied. The use of the redundant one is explained later.

To build the ${}^tW$ and ${}^bW$, the voting machine requires

- q: the sequence number (which can be consecutive or linked to the voter) There is a unique sequence number for each voter.
- h, h': publically known pseudo-random sequence functions whose composition ($h'(h(.))$) yields a 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 3.3).

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

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

Then, for both OTPs, for $1 \leq l \leq k$, the booth computes

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

The ${}^{t}W$ and ${}^{b}W$ matrices are formed from all these bitstrings by `XOR`ing them together and turning the bitstrings into matrices as follows:

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

However, rather than pass the components of the OTPs directly to the trustees, the booth passes the intermediate d' components. This is done by preparing two k-layered "dolls"[2] or "onions" ${}^{t}D$ and ${}^{b}D$. The two dolls are made up of k layers, where k is the number of partial decryptions that will take place. Each layer is encrypted with a different trustee encryption key e_l. This is defined as

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

and represented pictorially as Figure 3.

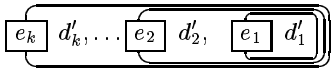


Fig. 3. The Layers of a Doll.

Each layer is "wrapped" with a different encryption function 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 OTP.

[2] The terminology of "dolls" comes from an analogy with Russian dolls in [2].

3.3 Decrypting a Vote

The booth now passes the encrypted votes to a web site to be publicly posted along with the other receipts for the election. Note here that although all the chosen layer (with every alternate cell derived from cipher text and the remainder purely white noise) gets passed to the trustees, only the component made up of cipher text can be decrypted. In this section, therefore, we suppress the superscripts, and let R be the cipher text component of the chosen layer and W the white noise component which will decrypt R.

The white noise is not passed directly to the trustees, rather the corresponding doll D (Again, the superscript is suppressed.) In the language of [2] the (R, D) pair is a *duo*.

The task of the trustees is therefore to progressively subtract W from the R component of the duo, by opening up the doll and making use of the information contained.

This will leave an image 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.

Duos are batched together and each trustee must perform a number of tasks on the batch of duos they receive: they strip off a layer of encryption from each of the duos and then perform a secret shuffle on the batch. They then post this shuffled, partially decrypted set of duos to the web site, strip off a second layer of encryption from each of the duos and perform a second secret shuffle on the batch.

Finally they post this second shuffled batch on to the web site and pass it to the next trustee. A record of the permutation used is retained by the trustee, but not published. 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.

More precisely, for an indivual duo, if the trustees begin at n and work through to 1, trustee n uses its first private decryption key e_k^{-1} (where $k = 2n$) to decrypt D and produce the (partial seed, doll) pair:

$$\{{}^{y}D_k\}_{e_k^{-1}} := d'_k, D_{k-1}$$

then uses h' to calculate

$$h'(d'_k) := d_k$$

Trustee n then performs the first part of the decoding of the ballot image by `XOR`ing d_k and R

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

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

$$trustee\, n \rightarrow website : R_{k-1}, D_{k-1}$$

Trustee n then performs the above sequence again, this time using its second secret decryption key e_{k-1}^{-1}

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

The same function h' is used to generate the next layer of W:

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

and the result used to perform the second decryption

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

R_{k-2} and D_{k-2} are then posted on the website.

$$trustee\, n \rightarrow website : R_{k-2}, D_{k-2}$$

This time, trustee n passes R_{k-2} and 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

$$trustee\, n-1 \rightarrow website : R_{k-3}, D_{k-3}$$

and

$$trustee\, n-1 \rightarrow website : R_{k-4}, D_{k-4}$$

Each trustee repeats this pattern, and when trustee 1 has finished, the result posted on the website is

$$\begin{aligned} & R \oplus d_k \oplus d_{k-1} \oplus \cdots \oplus d_0 \\ = {} & R \oplus W \end{aligned}$$

which is although not the full image seen on Alice's receipt earlier, is human readable because of the pixel redundancy at the font level.

The overall effect then is to have posted on the web site, in the left hand column say, the batch of initial receipts as 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 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 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 anonymity.

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 should 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 several such processes could be written by interested parties, allowing cross-checking).

4 Checking on the Trustees

Some checks must be performed to ensure that the trustees themselves do not alter, remove, inject or corrupt votes.

Recall that each trustee performs two decryptions on each duo, and therefore each trustee performs two permutations on each batch. They post each of these mixes.

However, if no further information about the process was revealed, trustees could alter votes without fear of detection. For example, if the last trustee falsified 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. The solution to this problem is derived from [6]. Trustees are required to provide *some* linkage information between the votes: enough to make the possibility of successful corruption vanishingly small, but not enough to allow any of the final decrypted votes to be traced back to the original votes cast.

For each posting half of the input links and half of the output links are revealed. 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 next posting all the duos not pointed to by links opened in the first batch have their outgoing links opened. No full link from a single final vote to a single original vote can therefore be drawn.

For each of the revealed links, if the duo has been transformed correctly we should have:

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

where

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

Note that, along with the links, 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 e_l key is public so the auditor can also calculate

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

and check that this equals D_l.

The crypto hash h' and the fact that the d' is revealed along with the link is significant in that it foils a potential attack on the secrecy: knowledge of the d'_l

value is essential in performing the checks. As this is the pre-image of a crypto-hash 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: 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.

5 Checking on the booth

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. But this is a big "if". Should 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. We really want ways to check on the behaviour of the booth to catch any attempts to cheat (or indeed to detect innocent malfunctions).

Alice should perform two checks that serve to detect attempts by the booth to cheat. 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, for example, The Electoral Reform Society or similar. These devices check, given the information on the receipt, that the *unused* doll is properly formed. The details of this check can be found in [2, 1]. This check 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.

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 in the absence of such measures.

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 optical cryptography. 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 to the first trustee 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 optical 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 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.

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 $\mathcal{Z}_2$:

$$W \oplus C = B$$
$$W' \oplus C = B'$$

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.

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 in [2, 1].

6 The trust model

From the point of view of the accuracy requirement, the trust that we need to place in the booths and trustees is minimal since the scheme has been designed to detect any significant attempt at fraud.

Some degree of trust is required in the auditors who perform the partial random checks on the trustee transformations. This can be minimised as necessary by using a number of independent auditors. One place where a higher level of trust is required is in the process for the selection of the links to be revealed. This is one step in the process that cannot be duplicated.

Assurances of anonymity are provided by the secret, random permutations on the ballot duos applied by the trustees. If there are n trustees we can tolerate the subversion of $n-1$ of these and still provide anonymity.

We are assuming that the cryptographic primitives have the appropriate properties and are of sufficient strength. In particular the booths' signature

algorithms will need to be highly resistant to plaintext/ciphertext pairs attack as the scheme generates a vast number of such pairs for the well-formedness checks on the ballot receipts.

We further assume that the checks are performed sufficiently assiduously, i.e. a sufficient proportion of the voters do choice to run checks on their receipts and check that their receipt is correctly posted on the web site. We also assume, for this paper at least, that the auditors perform their checks correctly and that there is no collusion between trustees and auditors. Note that if a trustee knew in advance which links would be selected for audit then they could corrupt votes on the other links.

7 Public trust

The Chaum scheme has been carefully devised, and the checks carefully constructed, so that minimal trust need be placed in the components of the system. Thus, with respect to the accuracy requirement, it would be extremely difficult for either a booth or a trustee to corrupt of falsify votes undetected. The scheme thus offers a high degree of transparency: it is not necessary to place a high degree of trust in any of the components. Significant malfunctions or compromises of booths or trustees will be detected, as long as the checking porcedures are applied reasonably assiduously.

The Chaum scheme is one of great beauty and intricacy, a veritable Harrison Martime Chronometer of voting schemes. Our initial investigations indicate that it does provide high levels of assurance with respect to the requirements of voting systems. From a purely technical point of view it would appear that, subject to appropriate assumptions, a scheme like the Chaum scheme is at least as trustworthy as existing pen and paper systems.

Being trustworthy, however, is not the same as being trusted. The subtlety and complexity of the scheme could prove problematic from the point of view of public uptake. It is extremely difficult to provide a simple explanation that the average voter would find both understandable and convincing as to why the scheme should be regarded as trustworthy.

The optical cryptography and provision of a ballot receipt and the voter's ability to check that their receipt has been accurately entered into the tallying process should serve to provide some confidence. However, the arguments that support the claim that there is a binding between the noise used to encrypt the ballot receipt and the noise used by the trustees to reveal to original ballot iamge are very subtle. Equally, the arguments supporting the assertion that the trustees will produce the correct decryptions of the ballot receipts are rather subtle.

The majority of the public therefore would have to take on trust the claims for such a scheme. Presumably having the system reviewed by a number of authorities and experts would help engender public confidence, but would this suffice?

Curiously enough, the transparency of the scheme may prove to be a handicap from the point of view of public confidence. Attempted frauds would, with high probability, be detected. The voting process itself could be recovered, but recovering public trust may be much more difficult. For a less transparent system, fraud could go largely undetected, and public trust could be unaffected.

8 Conclusions

We have discussed the dependability requirements of digital voting systems and have presented the elements of the Chaum scheme. We have argued that the scheme appears to meet these requirements to a high level of assurance and requires minimal trust to be placed in the components. It offers a high level of transparency and allows all the steps to be independently audited.

We have briefly discussed the obstacles to engendering and maintaining public trust in such a system. In future work we will investigate further the socio-technical aspects of this and similar digital voting schemes.

9 Acknowledgements

The authors would like to thank David Chaum for many helpful clarification regarding details of the scheme. We would also like to thank the members of the DIRC project for useful discussion and input.

References

1. Jeremy Bryans and Peter Ryan. A Dependability Analysis of the Chaum Voting Scheme. Technical Report CS-TR-809, Newcastle University School of Computing Science, 2003.
2. David Chaum. Secret-Ballot Receipts and Transparent Integrity: Better and less-costly electronic voting at polling places. http://www.vreceipt.com/article.pdf.
3. The Electoral Commission. Modernising Elections: A Strategic Evaluation of the 2002 Pilot Schemes. http://www.electoralcommission.gov.uk/about-us/modernisingelections.cfm, Oct 2002.
4. Marcin Gomulkiewicz, Marek Klonowski, and Miroslaw Kutylowski. Rapid mixing and security of Chaum's visual electronic voting. In *ESORICS*, 2003. To appear.
5. Interdisciplinary Research Collaboration in Dependability. http://www.dirc.org.uk.
6. M. Jakobsson, M. Juels, and R. Rivest. Making Mix Nets Robust for Electronic Voting by Randomised Partial Checking. In *USENIX'02*, 2002.
7. Rebecca Mercuri. Questions for voting systesm vendors. http://www.notablesoftware.com/checklists.html.
8. 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.

9. Office of the E-envoy. http://www.edemocracy.gov.uk, July 2002.
10. 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.
11. Steve Schneider and Abraham Sidiropoulos. CSP and Anonymity. In *ESORICS*, volume 1146 of *LNCS*, 1996.
12. Geoffrey Willans and Ronald Searle. *Compleet Molesworth*. Pavilion Books., 1984.