Distributed creation of the ballot form in Prêt à Voter using an element of Visual Encryption

D. Lundin*, H. Treharne*, P. Y. A. Ryan†, S. Schneider* and J. Heather*

*University of Surrey †University of Newcastle

***Abstract*— In this paper we present an extension of the Prêt à Voter e-voting system that introduces visual encryption to solve the chain voting problem and ensures that no organisation sees the layout of the ballot form prior to its use. The chain voting problem in Prêt à Voter is that anyone who can see the ballot prior to its use can coerce a voter by noting down the details of the form and then requiring it to be used by the voter. This solution is based on requiring the tellers to be active in ballot creation.**

Further benefits are the use of symmetric key encryption to potentially provide higher cryptographic security, automatic and mandatory well-formedness checks on the ballot forms used can be performed and visual encryption also results in a solution to the problem of enforced destruction of the candidate list.

***Index Terms*— Visual encryption, election integrity, ballot secrecy, receipts and coercion resistance, verifiability**

I. Introduction

CHAUM [1] introduces an electronic voting scheme that fundamentally uses visual cryptography to provide voter verifiability. The vote is printed in plain text by the voting machine but in two layers so that when these are separated and one is discarded, what remains is a receipt that can be decrypted into a vote that in turn can be counted. The decryption is performed serially by a number of independent *trustees* leading to an output that is verifiable but makes it impossible to tie a vote to a voter.

Chaum *et al* [2] builds on this initial configuration by setting the system up with ballot forms that are printed on normal paper prior to the start of the election in what they call *Prêt à Voter*. One main advantage of this system is that the voting devices never learn the intention of the voter, leaving a compromised device unable to change the vote based on its contents. The system also adds further benefits of simplicity and user recognition.

A. Voting in Prêt à Voter

The Prêt à Voter system uses printed ballot forms (example in Figure 1) with a candidate list to the left, boxes in which to indicate a choice on the right and an *onion* printed below the boxes. The ordering of the candidate list is based on randomness unique for each ballot form and encapsulated cryptographically within the onion, so named because it has many layers of encryption underneath each of which is found a *germ*. The sum of the germs is the reordering of the candidate list, thus requiring all layers of the onion to be peeled off before the order can be reconstructed.

HOMER	
MARGE	
BART	
LISA	
MAGGIE	

a45K1&s

Fig. 1. Prêt à Voter ballot form

When casting a vote the candidate list is detached and discarded, leaving the *receipt*. This contains only the position of the X (indicating the choice of the voter) and the onion. The receipt is scanned in, stored and transmitted electronically. The voter is allowed to take the receipt away.

In Prêt à Voter the ballot receipts, intermediate decryption steps and the final plain-text votes are published on a secure, publicly available *web bulletin board*, thus providing voter verifiability. The onion can be used as search criterion to check that the vote has been included in the tally by comparing the receipt held with the one shown on the bulletin board.

B. Shortcomings of Prêt à Voter

Ryan and Peacock [3] analysed Prêt à Voter from a systems perspective and raised a number of concerns, the most important being that voter anonymity can potentially be compromised by anyone who can see the ballot forms before they are used. Anyone who can register the connection between a particular onion and the ordering of the candidate list knows the vote cast resulting in a receipt with that particular onion. This opens up the possibility not only for a corrupt government agency to register all votes (completely circumventing the cryptographic steps) but also for *chain voting* or simple coercion. For example, a coercer can note down the candidate list ordering and the associated onion before marking his own intention on the form and giving it to another person. If that ballot form has been used to cast a vote then the onion will appear on the web bulletin board and the coercer knows the content of that vote.

In this paper we present an approach which reduces the potential for identifying voting choices.

C. Three phases

An election has three phases: *ballot creation phase* (ballot forms are created and distributed), *election phase* (ballots are cast) and *tallying phase* (ballots are counted). The changes to Prêt à Voter that we propose in this paper are concerned only with the first two phases and the last phase remains unchanged.

D. Contributions of this scheme

We acknowledge that the original Prêt à Voter paper [2] puts forth that removing the visual encryption of the vote and using ballot forms that voters recognise will increase simplicity as well as user acceptance. However, in this paper we propose to augment Prêt à Voter with a visual encryption of the candidate list. This enables the creation of paper ballots where no one organisation can learn the ordering of the candidate list. In our proposed system the visual encryption of the candidate list is achieved by going through a number of teller permutations in the same way as the onion is composed in Prêt à Voter.

It is necessary to involve a number of tellers in this process so that no single teller can learn the ordering of the candidate list.

The main objective of introducing visual encryption is to encapsulate the ordering of the candidate list in such a way that no one can see it prior to the election but where it can easily be decrypted by the individual voter whilst in the booth. This removes the need for the destruction of the left-side of the ballot form, containing the candidate list. This mandatory destruction is one of the potential weaknesses of the Pret a Voter system. The decryption must not be a difficult task because every single voter must be able to perform the decryption discussed in Section III.

Using visual encryption also means that a stronger symmetric key encryption can be used instead of asymmetric key encryption, and that the tellers are employed in an *oracle mode* [2], performing an automatic well-formedness check on the onion during the casting of a vote.

E. Overview of this paper

The remainder of this paper is concerned with the changes introduced into the ballot creation and election phases. The visual encryption and transformations are described in detail and illustrated with examples before future work is identified.

II. Ballot creation phase

We propose that the ballot creation process is started by a central organisation that creates an image of the candidate list in the base ordering. This is then split into two layers, creating a visual encryption of the list where both layers are needed to render the list legible. This process can be done in the open and publicly scrutinised. The image and the two layers can even be published so that anyone wishing to verify them can do so. Details of the visual encryption, or division into two layers, is presented in Section V.

The creation of a ballot form is started by the submission of the top layer to the first teller and this randomly chooses a germ. This germ is used to reorder the encrypted candidate list in the same way as previously done in [2], but it is also used to perform a transformation of the candidate list image, as described in Section V. The germ is encrypted into the onion using a secret key held by the teller.

Performing these transformations to the top and bottom layers separately result in valid top and bottom layers that when brought together shows the candidate list in the new ordering. It should be noted that the tellers must ensure that each part of the visual encryption is supplied strictly only during each respective period, in order to guarantee that an organisation cannot create a ballot form with known ordering, nor that the ordering can be checked during the election.

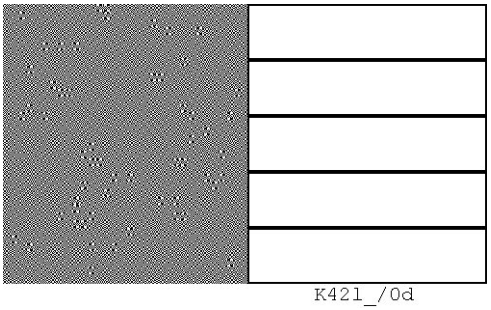

Fig. 2. Visually encrypted ballot form

III. Election phase

The voter selects a ballot form (shown in Figure 2) from the many supplied inside or outside the booth and places it on a horizontal touch screen on the voting device. This prompts the machine to read the barcode on the form and submit the onion to the tellers. These return a visual encryption of the candidate list but contrary to the process of creating the ballot forms, it is now the bottom layer of the encryption that is returned. Note that the top layer of the visual encryption, which is printed on the ballot form, is never read in by the machine and so will never co-exist on the same medium to compromise security.

When this layer of the visual candidate list encryption is displayed on the touch screen underneath the ballot form, the candidate list will appear in plain text for the voter to see because both layers of the visual encryption have now been brought together and are aligned. The details of this visual encryption are explained later in this paper.

Revealing the candidate list indicates to the voter that the onion is well formed, adding to the original Prêt à Voter scheme by making it possible to test the well-formedness of the actual form used to cast a vote. In the previous scheme, the user had to assert him- or herself of the validity of the scheme by testing the well-formedness of any number of preliminary forms but could not test the form used to cast the vote because this would enable the voting device (and the extended system) to learn the intention of the voter. [2]

Before viewing the plain text ballot form, the voter must perform some action[1] to commit to using this to cast the vote, ensuring that a voter cannot learn the ordering of the list without that ballot form being taken out of circulation. Failing to do this would enable an eligible voter to check the candidate lists on any number of forms and use these to coerce other voters.

The voter now indicates his or her choice by marking an X within the box immediately to the right of the preferred candidate's name. This makes a mark on the paper of course, producing the receipt in the original scheme [2], but also replicates this mark on the screen below the paper. The voter

[1]This is an area of future study, see Section VI.

Fig. 3. The two pixel symbols

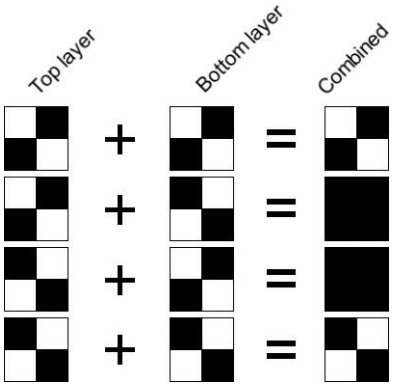

Fig. 4. The three resulting pixel symbols

can check that the mark appears in the same box on the screen as it does on the paper.

The vote is now submitted by the machine by storing or transmitting the same data as in the original scheme, namely the onion and the position of the X. The bottom layer of the candidate list is thus discarded, rendering it impossible for a third party to reassemble the parts.

A. A solution to the problem of forced destruction of the candidate list

We have already noted that in the original Prêt à Voter scheme it is absolutely essential that the voter destroys the left hand side of the form, the candidate list, or at least detaches it from the form and is given "dummy" candidate lists [3], so that it is impossible to prove the contents of the vote. In this scheme, because the candidate list is only visible in plain text during the actual voting taking place in the voting booth, the destruction of the candidate list does not have to be enforced.

IV. Visual encryption

The visual encryption, or division of the image into two layers, is the same as presented in [1], in turn derived from [4]. It is based on the division of each plain image pixel into four sub-pixels, each of which is one of two different pixel symbols as shown in Figure 3.

If two of these pixel symbols are placed in different layers they will yield one of three possible results, shown in Figure 4, when placed immediately above and below each other. The resulting pixel symbol that is completely black is perceived by the human eye as black and the two resulting symbols that have white sub-pixels are perceived as white. Thus, the plain image is in fact simply represented by white pixels on a black background.

It is easy to see from Figure 4 that a white pixel in the plain image is represented by the same pixel symbol in both layers; it can be either symbol as long as they are the same in both layers. The black pixel is similarly represented by different symbols in the two layers, but which symbol is in which layer is not dependent on the result (nor vice versa).

A. Mathematical notation of the visual encryption

A simple mathematical model for the visual encryption is presented here to provide completeness.

The pixel symbols are represented by the integers 0 (white), 1 (white) and 2 (black) and thus in this representation the following is true:

$$0 + 0 = 0 \tag{1}$$
$$1 + 1 = 1 \tag{2}$$
$$0 + 1 = 2 \tag{3}$$
$$1 + 0 = 2 \tag{4}$$

It is easy to see that we can use this system to calculate the contents of the plaintext image when the two layers have been overlaid. The first constituent is the top layer, the second is the bottom and the resultant is the plain-text image.

B. Example of visual encryption

By the following example it should be more clear how the visual encryption of the image of the candidate list is done. An image of the list is created and shown in Figure 5. In simple terms we start by creating a bottom, random layer where both dimensions are twice those of the original image and the area has been randomly filled with the pixel symbols in Figure 3, resulting in the layer shown in Figure 10.

Because we randomise the visual contents of the bottom layer, this means that the top layer will not be random but dependent on the bottom. From a cryptoanalytic perspective one might put forth that the fact that all the information is in the top layer that is printed onto paper and thus no information about the contents on a ballot paper can be derived from the pixel symbols used in the bottom layer that is handled electronically. This could be argued to enhance the security of the system because the layer handled electronically stands a slightly higher chance of being stolen.

We now create a representation of the original image, expanding each pixel into one of the pixel result symbols in Figure 4. For each pixel, if the current pixel is white then the pixel symbol used in this representation must be the same as the symbol in that particular place in the random bottom layer. Otherwise the symbol is simply the completely black. This complete representation can be found in the final image in Figure 10.

From the complete representation of the image and the random bottom layer we can create the top layer simply by going through each pixel and checking which symbol is in place in the complete representation. If that pixel is black then the pixel used in this layer must be the opposite to the one used in the bottom layer. Otherwise the pixel in the top layer must be the same to that of the bottom layer. The resulting top layer is shown in the middle image in Figure 10.

Thus, the superimposing of the top layer upon the bottom layer is shown in the final image in Figure 10.

HOMER
MARGE
BART
LISA
MAGGIE

Fig. 5. The image of the candidate list

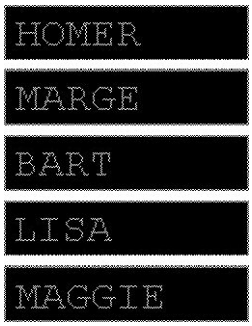

Fig. 6. Illustration of how the ballot form is split into smaller images

V. The visual transformations

The transformations applied to the layers by a teller do not have to be reversible. In order to mitigate the buffering problem described above, they do however have to be possible to apply in any order and still yield the same result. One can say that the scrambling of the image is performed in order to mask the reordering at each stage.

A. Reordering of the list

The teller treats the image of the candidate list as a set of vertically stacked smaller images (as shown in Figure 6), each of which contains the name of one candidate. By reordering these smaller images the teller also reorders the candidates, though without knowing which image has within it the name of which candidate. The basis for the reordering is of course the germ selected by that same teller in the ballot creation phase.

In this first instance the reordering of the candidate list is based on cyclic shifts.

B. Scrambling of the image

Also based on the germ created by that teller, the teller performs a scrambling of the image so that the reordering is not apparent to a spectator. If the scrambling is not performed, it is trivial to simply reorder the image of the base order list until a match is found. The theory is also that if the same transformations are applied to both layers, the final output will be a different but still legible candidate list.

One requirement on the scrambling of the list is that the transformations must be possible to apply in any order and still yield the same result, that is to say that the top layer should be possible to create in a forward teller order and the bottom layer in a reverse teller order and still yield a legible list.

To accomplish this the image is divided into a number of smaller images along the vertical axis, one for each candidate. The same scrambling is then applied to the same pixels of all these smaller images. If the scrambling described in Section V is applied to all these smaller images, it is evident that their ordering in the larger image does not matter but the result is the same.

C. Scrambling of the smaller images

The scrambling of each of the smaller vertical images is simple. The teller uses its germ to create a map of the image with a *true* or *false* value for each pixel. The pixel symbols in the positions with a *true* value are switched to the respective other symbol and those in positions with a *false* value are simply left as they are.

If such scrambling is performed in the same manner to both layers that make up the plain image, this yields the same plain image as if no such scrambling had been performed. This is because two of the same pixel symbol result in a white pixel and two different pixel symbols result in a black. So if the pixel symbols in both layers are switched to the corresponding other symbol, the result will be the same.

D. Mathematical expression of the scrambling

The upper layer L_2 and the bottom layer L_1 are represented by two two-dimensional arrays. From Section IV-A we know that the two pixel symbols in these layers are represented by the integers 0 and 1. Thus two examples of these layers are

$$L_1 = \begin{matrix} 1 & 0 & 0 & 1 & \dots \\ 1 & \ddots & & & \\ 0 & & \ddots & & \\ 0 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \tag{5}$$

$$L_2 = \begin{matrix} 0 & 1 & 1 & 1 & \dots \\ 1 & \ddots & & & \\ 1 & & \ddots & & \\ 1 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \tag{6}$$

The sum of these layers is thus

$$L_0 = \begin{matrix} 2 & 2 & 2 & 1 & \dots \\ 1 & \ddots & & & \\ 2 & & \ddots & & \\ 2 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \tag{7}$$

Each teller that performs a transformation of the image creates a two-dimensional array which is a map of the cells that will be changed. The contents of this array is dependent on the teller's germ and some secret function. In the array, the integer 0 indicates that the pixel symbol will not be changed and the

Layer contents	Change	Result
0	0 (no)	0
0	1 (yes)	1
1	0 (no)	1
1	1 (yes)	0

TABLE I

PIXEL SYMBOLS THAT ARE CHANGED AND THE RESULTS

integer 1 indicates that the pixel symbol will be changed to the other. The following is an example of such an array:

$$F = \begin{matrix} 1 & 0 & 0 & 1 & \dots \\ 1 & \ddots & & & \\ 0 & & \ddots & & \\ 1 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \quad (8)$$

The changes that are performed are shown in Table I and we can see from it that the resulting pixel symbol in the layer is determined by the XOR function. We can annotate this in the following way where $G(x, y)$ is the layer being modified, $F(x, y)$ is the modifier and $R(x, y)$ is the resulting layer:

$$R(x, y) = G(x, y) \oplus F(x, y) \quad (9)$$

The following two arrays are (5) and (6) with (8) applied to them:

$$L'_1 = \begin{matrix} 0 & 0 & 0 & 0 & \dots \\ 0 & \ddots & & & \\ 0 & & \ddots & & \\ 1 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \quad (10)$$

$$L'_2 = \begin{matrix} 1 & 1 & 1 & 0 & \dots \\ 0 & \ddots & & & \\ 1 & & \ddots & & \\ 0 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \quad (11)$$

We now add together layers L'_1 and L'_2 and form the following resulting image:

$$L'_0 = \begin{matrix} 2 & 2 & 2 & 0 & \dots \\ 0 & \ddots & & & \\ 2 & & \ddots & & \\ 2 & & & \ddots & \\ \vdots & & & & \ddots \end{matrix} \quad (12)$$

Because the integers 0 and 1 are used to represent the "white" pixels, we find that those pixels are found in the same positions in (7) and (12). We can thus deduce that we have altered the contents of the layers L_1 and L_2 but the resulting layers L'_1 and L'_2 still yield the same visual contents to the human eye.

Fig. 7. The result of the scrambling

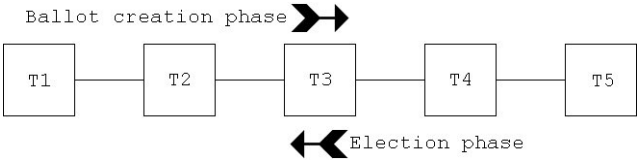

Fig. 8. Teller communication order

E. Example of scrambling of the image

A Java application has been written to perform the visual encryption of the image described earlier and the manipulations used in this section as examples of how the scrambling might work. The application takes an image such as Figure 5, encrypts it by splitting it into two layers and then allows the user to perform any number of manipulations as described in this section, saving the results to files. Figure 9 shows the top and the bottom layer going through reordering and scrambling with the same seeds but in different order. Figure 7 shows the final top layer superimposed upon the final bottom layer — displaying the candidate list in the legible form only ever occurring within the voting booth.

F. Election phase visual well-formedness check

When the voter places the ballot form on the voting machine the onion is electronically read and sent to the tellers, in reverse order to the ballot form creation phase, as illustrated in Figure 8. The first teller to receive the onion removes its layer of encryption from the onion and extracts its germ. It then takes the bottom layer of the original visual encryption of the candidate list and using the germ it then reorders the candidate list and performs the transformations described in Section V.

When all tellers have performed this decryption, reordering and transformation in order the result is passed to the voting machine where it is displayed on a screen underneath the printed copy of the top layer, yielding a legible candidate list.

G. The buffering problem

This problem applies to a trivial implementation of visual encryption through the tellers. Although none of the tellers are able to make out the plaintext candidate list either during the ballot creation phase OR the election phase, they would be able to buffer their images of the top and the bottom layers to apply them to the ballot form during the counting phase succeeding the election phase. This problem would not occur

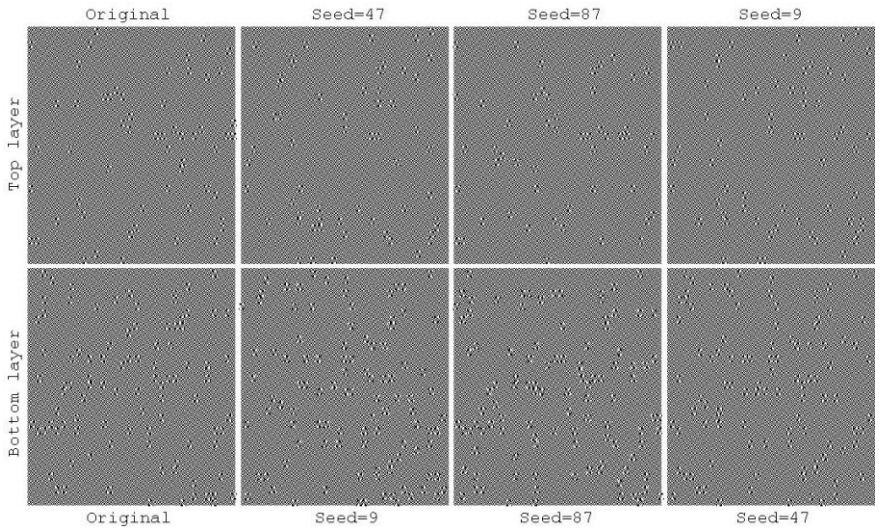

Fig. 9. The scrambling of the visual encryption

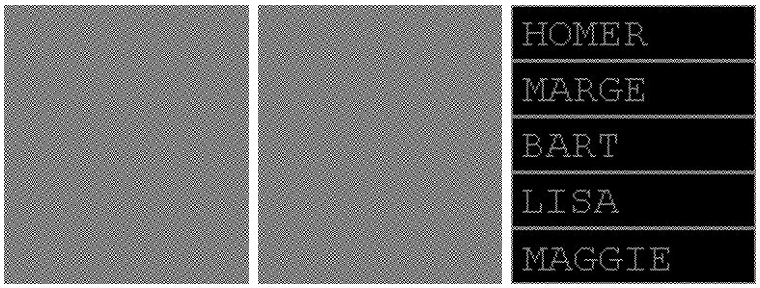

Fig. 10. Top layer, bottom layer and both overlain

in the original Prêt à Voter scheme [2] because all the tellers see is the receipt - they never get any sight of the candidate list so they cannot learn the contents of the vote.

It is thus important to ensure that the visual encryption does not lead to the tellers being able to learn the contents of the vote.

H. Natural mitigation of the buffering problem by reverse teller order

The proposed transformations in Section V are associative and commutative. In other words, they can be applied in any order to the image and as long as all transformations are applied, the result is the same. This means that when the first teller removes the top layer of encryption and extracts its germ it can immediately apply the re-ordering of the list and the transformation of the image. In other words, the processing of the bottom layer does not have to be done in the same order as the top. This results in none of the tellers seeing the same transformations applied to both layers, seemingly solving the buffering problem.

However, the weakness of the current scheme presented here is that the first and the last teller can collude. The weakness is that the last teller saves a copy of the top layer when it has gone through the transformations of all tellers. During the election phase when the bottom layer is created, the first teller will see the bottom layer after it has gone through the transformations of all tellers. If the two tellers work together they can create an image of the candidate list as it is shown to the voter in the booth. In the example shown in Table II a, b, c, d and e are transformations performed by tellers 1, 2, 3, 4 and 5 respectively.

Table II shows how the tellers start by processing the top layer of the encrypted candidate list in steps 1 to 5. They then perform the same transformations on the bottom layer in steps 6-10. As the transformations can be applied in any order, it is easily deduced that the knowledge of T_5 after step 5 and that of T_1 after step 10 can be combined to create a plain-text version of the ballot form, the exact image shown to the voter in the voting booth.

Overcoming this weakness is one focus of future work, see Section VI.

VI. FUTURE WORK

Outstanding issues related to addressing whether the visual encryption approach is implementable in a real setting are currently under investigation, and we discuss them briefly here.

A. Prototype

As part of current research at the University of Surrey, a prototype of the contribution made by this paper to the Prêt à Voter scheme is currently being created. It will demonstrate the three entities involved: the candidate list creation agency, the teller(s) and the voting booth.

Step:	1	2	3	4	5	6	7	8	9	10
Teller:	T_1	T_2	T_3	T_4	T_5	T_5	T_4	T_3	T_2	T_1
Top layer	a	a b	a b c	a b c d	a b c d e					
Bottom layer						e	e d	e d c	e d c b	e d c b a

TABLE II
EXAMPLE OF SUM OF TELLER TRANSFORMATIONS

B. Teller tasks

The security of this scheme is based on the contents of the candidate list being hidden from the tellers even though these have to perform actions on a visual encryption of it. One potential source of weakness is the fact that the tellers have to perform all of three things: the creation of the ballot form, the creation of the bottom layer to be displayed in voting booths and the final tally. Because the ballot creation is done in a forward teller order and the creation of the bottom layer in a backward teller order, none of the tellers can reconstruct the candidate list alone. However, if the first and the last teller work together they can construct an image of the candidate list.

C. The correct layer at the correct time

To ensure the secrecy of the election it is important that the three phases of the election are recognised and enforced by all parties. It is essential that the correct layer is fed into the tellers at the right time, i.e. that the top layer that is to be printed onto the ballot forms is processed only during the ballot creation phase and that the bottom layer that is displayed in the voting booth only is processed during the actual voting phase — to ensure that no-one is able to check the ordering of the candidate list in advance. This must somehow be enforced. It might already be implicitly enforced by the mechanisms of this scheme, but this must be proven.

D. Committing to using a ballot form

It is essential that a ballot form that is displayed on a voting device must be immediately used or discarded: if this does not happen then it is possible for a voter to check the contents of a ballot form and then use this information to coerce another voter. A question that remains to be answered is this: how can this be enforced? If, for example, a further government agency is introduced into the system, in effect to "tick" the ballot forms used, then this agency could very easily remove unwanted votes from the final tally. Furthermore, the tellers cannot themselves identify a receipt that they are currently decrypting and thus cannot enforce this. Perhaps introducing databases in the tellers and embedding identification tags within the onion will enable these checks to be carried out — but this in turn will also lead to potential security problems.

This might be one of the most challenging issues.

E. Onion uniqueness

In Prêt à Voter a central agency creates the onion and can thus ensure that no two ballot forms are identified by the same onion. In this distributed example, some method must be put in place to impose similar restrictions so that a voter can find his or her receipt on the web bulletin board.

F. Aligning ballot form on machine

There are a host of physical considerations to be addressed, for example to make the aligning of the two layers easy and accurate for the voter in the booth. A series of perforations of the ballot form that correspond to protruding elements on the voting machine might aid in this.

VII. SUMMARY

By using visual encryption on the candidate list it is shown here how it is possible to create ballot forms that no organisation, in creation or transit, has seen, solving the chain voting problem discussed in [3].

In this scheme the tellers are involved in the creation of the onion and can therefore use stronger symmetric key encryption instead of asymmetric key encryption.

Furthermore, it follows of this configuration that each ballot form used to cast a vote is checked for well-formedness. In the case where it is not well formed, the voter will simply not see the candidate list and the ballot form is useless.

The system also solves the problem of forcing the destruction of the candidate list because the list is simply not legible outside the booth.

REFERENCES

[1] Chaum, D. (2004) *Secret-Ballot Receipts: True Voter-Verifiable Elections*, IEEE Security & Privacy
[2] Chaum, D., Ryan, P. Y. A., Schneider, S. (2005) *A Practical, Voter-Verifiable Election Scheme*, European Symposium on Research in Computer Security, LNCS 3679, Springer Verlag
[3] Ryan, P., Peacock, T. (2005) *Prêt à Voter: a Systems Perspective*
[4] Naor, M., Shamir, A. (1994) *Visual Cryptography*, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel