

Prêt à Voter with re-encryption mixes

P Y A Ryan¹ and S A Schneider²

¹University of Newcastle, ²University of Surrey

Abstract. We present a number of enhancements to the voter verifiable election scheme Prêt à Voter [CRS05]. Firstly, we propose a mechanism for the distributed construction by a set of independent clerks of the ballot forms. This construction leads to proto-ballot forms with the candidate list encrypted and ensures that only a collusion of all the clerks could determine the cryptographic seeds or the onion/candidate list association. This eliminates the need to trust a single authority to keep this information secret. Furthermore, it allows the on-demand decryption and printing of the ballot forms, so eliminating chain of custody issues and the chain voting style attacks against encrypted receipt schemes identified in [RP05].

The ballot forms proposed here use ElGamal randomised encryption so enabling the use of re-encryption mixes for the anonymising tabulation phase in place of the decryption mixes. This has a number of advantages over the RSA decryption mixes used previously: tolerance against failure of any of the mix tellers, full mixing of terms over the Z_p^* space and enabling the mixes and audits to be fully independently rerun if necessary.

1 Introduction

The Prêt à Voter scheme, presented in [CRS05], is a cryptographic voting scheme that enables voter-verifiability: at the time of casting their vote, voters are provided with an encrypted receipt. They can then check, via a secure Web Bulletin Board (WBB), that their receipt is accurately included in a robust anonymising mix process. Various checking mechanisms serve to detect any corruption in any phase of this process: encryption of the vote, recording and transmission of the encrypted ballot receipt and the decryptions of the votes. Full details can be found in [CRS05]. Henceforth we will refer to this version of the scheme as Prêt à Voter'05.

Prêt à Voter seeks to achieve the goals of accuracy and ballot secrecy with minimal trust in the system: software, hardware, officials. Assurance is achieved through a high degree of transparency and we thus verify the correctness of the election rather than attempting to verify the system.

This scheme has the benefit of providing a very simple and familiar voter experience, but certain vulnerabilities and trust assumptions have been identified, see [RP05]. In this paper we present a number of enhancements designed to counter these threats and eliminate the need for these trust assumptions.

The construction of the ballot forms presented here also enables the use of re-encryption mixes in the anonymising/tabulation phase. This also provides a number of advantages over the RSA/decryption mixes of Prêt à Voter'05.

The structure of the paper is as follows: in the next section we give the key elements of Prêt à Voter'05. Section 3 summarises some of the threats to and trust assumptions needed in Prêt à Voter'05. Section 4 presents the distributed construction of encrypted ballot forms. Sections 5 and 6 describe how these forms can be used in the vote casting process. Section 7 describes the use of this construction for re-encryption mixes during the anonymising and tabulation phase. Sections 8 and 9 describe the new auditing procedures required for the new ElGamal style ballot forms. Sections 10 and 11 discuss some further extensions to deal with more general voting methods and remote voting.

2 Outline of Prêt à Voter 2005

We now present an overview of the Prêt à Voter voter-verifiable scheme. Voters select at random a ballot form, an example of which is shown in Figure 1.

Obelix	
Asterix	
Panoramix	
Idefix	
	<i>7rJ94K</i>

Fig. 1. Prêt à Voter ballot form

In the booth, the voter makes her selection in the usual way by placing a cross in the right hand column against the candidate of choice, or, in the case of a Single Transferable Vote (STV) system for example, they mark their ranking against the candidates. Once the selection has been marked, the left hand strip is detached and discarded. The remaining right hand strip now constitutes the receipt, as shown in Figure 2.

X
<i>7rJ94K</i>

Fig. 2. Prêt à Voter ballot receipt

The voter now exits the booth and casts their vote in the presence of an official. The ballot receipt is placed under an optical reader or similar device that records the random value at the bottom of the strip and an index value indicating the cell into which the X was marked. The receipt is digitally signed and franked and the voter now retains this as their receipt.

Possession of a receipt might appear to open up the possibility of coercion or vote-buying. However, the candidate lists on the ballot forms are independently randomised for each ballot form. Thus, with the left hand strip removed, the right hand strip alone does not indicate which way the vote was cast.

The cryptographic value printed on the bottom of the receipt, the ‘onion’, is the key to extraction of the vote. Buried cryptographically in this value is the information needed to reconstruct the candidate list shown on the left hand strip. This information is encrypted under the secret keys shared by a number of tellers. Thus, only the tellers acting in concert are able to reconstruct the candidate order and so interpret the vote value encoded on the receipt.

Once the election has closed, all the receipts are transmitted to a central tabulation server which posts them to a secure WBB. This is an append-only, publicly visible facility. Only the tabulation server, and later the tellers, can write to this and, once written, anything posted to it will remain unchanged. Voters can visit this WBB and confirm that their receipt appears correctly.

After a suitable period, the tellers take over and perform a robust, anonymising, decryption mix on the batch of posted receipts. Various approaches can be used to ensure that the tellers perform the decryptions correctly. Details of this can be found in [CRS05].

Prêt à Voter’05 proposes an Authority responsible for the generation of the entropy for the crypto seeds and prior printing of the ballot forms. Random auditing, by independent organisations, of the forms before, during and after the election serve to detect any attempt by the the Authority to pass off incorrectly formed ballot forms. Later in this paper we propose an alternative approach using on-demand creation and printing of forms and post-auditing.

The Prêt à Voter’05 approach has the advantage of simplicity and results in a very simple and familiar experience for the voters: they simply register, collect a form, mark their selection in the booth and then cast the form.

For full details of the mechanisms used in the 2005 version of the scheme to detect any malfunction or misbehaviour by the devices or processes that comprise the scheme, see [CRS05]. The construction of the ballot forms used here calls for rather different monitoring and auditing mechanisms that we detail later.

3 Threats and trust models

The simplicity of the original scheme, in particular the use of a single authority and the pre-printing and pre-auditing of the ballot forms, comes at a certain

cost: various trust assumptions need to be made. In this section we briefly recall the threats and assumptions of Prêt à Voter'05 identified in [RP05].

3.1 The need to trust the Authority for confidentiality

In Prêt à Voter 2005, a single entity creates the ballot forms. Whilst it is not necessary to trust this entity from the point of view of accuracy, it is necessary to trust it not to leak the ballot form information. Clearly, if the Authority were to leak this information, the scheme would become susceptible to coercion or vote buying.

3.2 Chain of custody

Just as we need to trust the Authority not to leak ballot form information, we also need to assume that mechanisms are in place to ensure that none of this information is leaked during storage and distribution. Various counter-measures are possible: for example, ballot forms could be kept in sealed envelopes to be revealed only by the voters in the booth. Alternatively, a scratch card style mechanism along the lines suggested in [RP05] could be used to conceal the onion value until the voter reveals it at the time of vote casting. The ballot forms would also need to be stored and distributed in locked, sealed boxes. All of these counter-measures are rather procedural in nature and so require various trust assumptions.

3.3 Chain voting

Conventional, pen and paper elections may be vulnerable to a style form of vote buying known as chain voting. The UK system in particular is vulnerable. Here, the ballot forms are a controlled resource: on entering the polling station, the voter is registered and marked off on the electoral roll. They are given a ballot form which they take to the booth, mark and then cast in the ballot box. In principle, officials should observe the voters casting their form.

The attack works as follows: the coercer smuggles a blank ballot form out of the polling station. The controls on the distribution of the forms should make this a little tricky, but in practise there are many ways it could be achieved. Having marked the form for the candidate of their choice, the coercer intercepts a voter as they enter the polling station. The voter is told that if, when they exit the polling station, they hand a fresh, blank form back to the coercer they will receive an reward. The attack can now proceed inductively until a voter decides to cry foul. Note that, once initialised, the controls on the ballot forms works in the coercer's favour: if the voter emerges from the polling station with a blank form, it is a strong indication that they did indeed cast the marked form they were given by the coercer.

3.4 Kleptographic channels

A further, rather subtle vulnerability can occur where a single entity is responsible for creating cryptographic variables: kleptographic attacks as described in [YY96]. The possible relevance of such attacks to cryptographic voting schemes is described in [M. 06]. The idea is that the entity may carefully choose the values of the crypto variables in order to leak information to a colluding party.

In the case of Prêt à Voter, the Authority might choose the seed values in such a way that an agreed, keyed cryptographic hash of the onion value indicates the candidate order. Clearly this may require quite a bit of searching and computation to find suitable values. Note however that such an attack could pass unnoticed: the distribution of seed values would look perfectly random to anyone ignorant of the cryptographic hash function.

4 Distributed generation of encrypted ballot forms

Many of the above attacks stem from the fact that a single entity is able to determine, in the sense of being able both to know and to control, the seed values. We now present a mechanism for the distributed generation of the seed values and ballot forms. Throughout, we will use ElGamal encryption rather than RSA as used in Prêt à Voter'05 and we will work in Z_p^* , p a (large) prime.

An analogous construction is possible for the distributed creation of the RSA, layered onions of Prêt à Voter'05. However, as we want to introduce re-encryption mixes at the tabulation stage, we present the construction for ElGamal encryption here. We note also that the term *onion* is a slight misnomer where ElGamal terms are used but we will retain it here for historical reasons.

The ballot forms will be generated by a set of l clerks in such a way that each contributes to the entropy of the crypto seed and this remains encrypted throughout. Consequently the candidate list, which is derived from the seed, remains concealed and all the clerks would have to collude to determine the seeds values.

We assume a set of decryption tellers who hold the key shares for a threshold ElGamal primitive with public key: (p, α, β_T) . These will act much as the tellers of the original scheme and will be responsible for the final decryption stage after the anonymising, re-encryption mix phase. Details of the anonymising and decryption/tabulation phases will be given in section 7.

We also assume a set of Registrars with threshold secret key shares corresponding to the public key: (p, α, β_R) . These public keys are known to the Clerks and are used in the construction of the ballot forms.

An initial clerk C_0 generates a batch of initial seeds s_i^0 . These seeds are drawn randomly from a binomial distribution centred around 0 with standard deviation σ . σ would probably be chosen to be of order n , the number of candidates.

From these, C_0 generates a batch of pairs of "entangled" onions by encrypting each s_i^0 , actually in the form $\gamma^{-s_i^0}$, under the Registrar key and the Teller key:

$$(\{\gamma^{-s_i^0}\}_{PK_R}, \{\gamma^{-s_i^0}\}_{PK_T}).$$

Expressed as ElGamal encryptions these have the form:

$$(\alpha^{x_i^0}, \beta_R^{x_i^0} \cdot \gamma^{-s_i^0}), (\alpha^{y_i^0}, \beta_T^{y_i^0} \cdot \gamma^{-s_i^0})$$

for fresh random values x_i^0, y_i^0 drawn from Z_p^* .

Notice that, for convenience later, we have encrypted the value $\gamma^{-s_i^0}$ for some generator γ of Z_p^* rather than encrypting s_i^0 directly. The reason for this will become apparent shortly.

The remaining $l - 1$ Clerks now perform re-encryption mixes and transformations on this batch of onion pairs. Each Clerk takes the batch of pairs output by the previous Clerk and performs a combined re-encryption along with an injection of fresh entropy into the seed values. For each pair of onions, the same entropy is injected into the seed value of both onions to ensure that these values continue to match for each pair.

More precisely, for each pair of the batch, the j th Clerk C_j generates a new, random values \bar{x}, \bar{y} and \bar{s} and performs the following mix/transformation on each onion pair of the batch:

$$\begin{aligned} & \{(\alpha^{x_i^{j-1}}, \beta_R^{x_i^{j-1}} \cdot \gamma^{-s_i^{j-1}}), (\alpha^{y_i^{j-1}}, \beta_T^{y_i^{j-1}} \cdot \gamma^{-s_i^{j-1}})\} \\ & \quad \downarrow \\ & \{(\alpha^{x_i^{j-1}} \cdot \alpha^{\bar{x}_i^j}, \beta_R^{x_i^{j-1}} \cdot \beta_R^{\bar{x}_i^j} \cdot \gamma^{-s_i^{j-1}} \cdot \gamma^{-\bar{s}_i^j}), (\alpha^{y_i^{j-1}} \cdot \alpha^{\bar{y}_i^j}, \beta_R^{y_i^{j-1}} \cdot \beta_R^{\bar{y}_i^j} \cdot \gamma^{-s_i^{j-1}} \cdot \gamma^{-\bar{s}_i^j})\} \\ & \quad \downarrow \\ & \{(\alpha^{(x_i^{j-1} + \bar{x}_i^j)}, \beta_R^{(x_i^{j-1} + \bar{x}_i^j)} \cdot \gamma^{-(s_i^{j-1} + \bar{s}_i^j)}), (\alpha^{(y_i^{j-1} + \bar{y}_i^j)}, \beta_R^{(y_i^{j-1} + \bar{y}_i^j)} \cdot \gamma^{-(s_i^{j-1} + \bar{s}_i^j)})\} \\ & \quad \downarrow \\ & \{(\alpha^{x_i^j}, \beta_R^{x_i^j} \cdot \gamma^{-s_i^j}), (\alpha^{y_i^j}, \beta_T^{y_i^j} \cdot \gamma^{-s_i^j})\} \end{aligned}$$

where

$$\begin{aligned} x_i^j &= x_i^{j-1} + \bar{x}_i^j \\ y_i^j &= y_i^{j-1} + \bar{y}_i^j \\ s_i^j &= s_i^{j-1} + \bar{s}_i^j \end{aligned}$$

The \bar{x}, \bar{y} denote fresh random values drawn from from Z_p^* generated by the Clerk during the mix. Similarly the \bar{s} values are freshly created random values except that these are again chosen randomly and independently with a binomial distribution mean 0 and standard deviation σ . Having transformed each onion pair in this way, the Clerk C_j then performs a secret shuffle on the batch and outputs the result to the next Clerk, C_{j+1} .

Thus, each Clerk performs a re-encryption mix along with the injection of further entropy into the seed values \bar{s} .

So the final output after $l - 1$ mixes is a batch of pairs of onions of the form: $\{(\alpha^{x_i}, \beta_R^{x_i} \cdot \gamma^{-s_i}), (\alpha^{y_i}, \beta_T^{y_i} \cdot \gamma^{-s_i})\}$ where:

$$x_i = x_i^l, y_i = y_i^l, s_i = s_i^l$$

thus:

$$x_i = \sum_{i=1}^l \bar{x}^i$$

etc.

The final s_i values will have binomial distribution mean 0 and standard deviation $\sigma\sqrt{l}$.

We will refer to the first onion as the “Registrar onion” or “booth onion” and the second onion as the “Teller onion”.

For each pair, assuming correct behaviour of the clerks, the s values in the two onions should match. We’ll discuss mechanisms to detect corruption of the forms later. As the seed values, and hence the candidate orders, remain encrypted, none of clerks knows the seed values and only if they all acted in collusion could they determine the seed values. These “proto-ballot form” can now be stored and distributed in encrypted form, thus avoiding the chain of custody problems mentioned above. The seed values can now be revealed on demand by a threshold set of the Registrars.

5 On-demand creation of ballot forms

The above construction of the proto-ballot forms means that the ballot form material can be stored and distributed in encrypted form. Once registered at the polling station, voters are assigned at random one of these forms:

<i>onion_L</i>	<i>onion_R</i>

The voter proceeds to the booth in which they find a device that reads the left-hand onion. In the simplest case, the secret key to decrypt the left-hand onions could be held in the devices in the booths. Thus, the left hand onion

could be decrypted in the booth, the seed value s revealed and the candidate order π derived as some agreed function of s . If lodging the keys in a single device is considered rather fragile, the left-hand onion could be encrypted under a threshold key held by a number of registrars. The onions could be transmitted to these registrars and a threshold set of these would then decrypt the onions and return the seed to the booth device.

The candidate list can now be printed by the device in the booth to give a standard Prêt à Voter ballot form:

Obelix	
Asterix	
Panoramix	
Idefix	
$onion_L$	$onion_R$

As an additional precaution, the left-hand onion might be separately destroyed.

The point of the paired onions is now clear: we arrange for the booth device to see only the left hand onion and so it will not know the association of the candidate list with the right hand, teller onion that will appear on the receipt. Various mechanisms are possible to ensure that the booth device does not see the right-hand onion. The scratch strip mechanism could be invoked here again for example: the right-hand onion would be covered by a scratch strip that would only be removed at the time of casting, or even at some time after casting. The voter only really needs to reveal the teller onion when they come to check their receipt on the WBB.

Strictly speaking, the l th clerk in collusion with the booth device could form the candidate list/onion association. Elaborations of the scheme to counter the threat of such collusion attacks are the subject of ongoing research.

6 Supervised casting of a ballot

The voter in the booth now has a “conventional” Prêt à Voter style ballot form with the candidate list and the associated right hand (teller) onion. His vote can now be cast in the usual way by marking an X against the candidate of their choice. The left hand strip is detached and discarded and the voter leaves the booth and casts their vote in the presence of an official exactly as described previously. Their receipt is recorded digitally as $(r, onion)$, where r is the index value indicating the position of the X .

The receipt can be digitally signed and franked at this point to counter any receipt faking attacks.

Once the election has closed, copies of the digitised receipts will be posted to the WBB exactly as before and the voters can visit this and assure themselves that their receipt has been correctly registered. In addition to this, a Verified Encrypted Paper Audit Trail mechanism could be deployed: at the time of casting, an extra paper copy of the receipt is made and retained in a sealed audit box. This can be used to independently check the correspondence with the receipts posted to the WBB.

7 Re-encryption/tabulation mixes

Our construction leads to ElGamal onions which appear to be well suited to being put through re-encryption mixes. However, the form of the ballot receipts means that this is not quite straightforward: in addition to the onion term we have the index value, in the clear as it were. An obvious approach would be to send the receipt terms through the mix re-encrypting the onions whilst leaving the index values unchanged. The problem with this is that an adversary is able to partition the mix according to the index values. There may be situations in which this is acceptable, for example large elections in which the number of voters vastly exceeds the number of voting options. In general it seems rather unsatisfactory.

A more satisfactory solution, at least for the case of a simple selection of one candidate from the list, is described in this section. We will discuss how to achieve full mixing in the more general case in section 10.

In this case we restrict ourselves to just cyclic shifts from the base ordering of the candidate list from a base ordering. For single candidate choice elections, this is sufficient to ensure that the receipts do not reveal the voter's selection. For more general styles of election, in which for example voters are required to indicate a ranking of the candidates, we of course need to allow full permutations of the candidate list. Indeed, even in the case of single selection elections, it is preferable to allow full permutations in order to eliminate any possibility of a systematic corruption of votes. For this moment we discuss the approach of simple cyclic shifts.

Let s_i be the shift of the candidate list for the i th ballot form. We can absorb the index value r into the onion:

$$(\alpha^y, \beta_T^y \cdot \gamma^{r-s_i})$$

This gives a pure ElGamal term and the value $r-s_i$ taken modulo n indicates the voter's the original candidate choice in the base ordering. These ElGamal terms can now be sent through a conventional re-encryption mix by a set of mix tellers, see for example [JJR02]. These mix tellers do not hold any secret keys but read in a batch of ElGamal terms from the WBB, re-encrypt each of them and then post the resulting terms in random order to the WBB. After an

appropriate number of such anonymising re-encryption mixes, (a threshold set of) the decryption tellers take over to extract the plaintext values.

Thus, in contrast to the decryption mixes uses previously, the anonymising and decrypting phases are separated out in re-encryption mixes.

This will yield decrypted terms of the form:

$$\gamma^{r-s_i} \pmod{p}.$$

Now we have to extract the values $r - s_i \pmod{n}$ to recover the original votes. The difficulty is that $r - s_i$ is the discrete log of γ^{r-s_i} in Z_p^* so in general, if the seed values had been drawn randomly from Z_p^* , computing this would be intractable. However, we have set things up so that the s values are drawn from a binomial distribution so we can search the space very efficiently. We could, for example, generate a look-up table for the logs out to some multiple of $\sigma\sqrt{l}$. Occasionally we will have an outlier that will require some search beyond the range of the look-up table.

7.1 Coercion resistance and plausible deniability

The point of using a binomial distribution for the seed value is to ensure plausible deniability or coercion resistance whilst at the same time avoiding the discrete log problem. An alternative approach would be to bound the possible seed values generated by the clerks to lie in some fixed range, between $-M$ and $+M$ say. This would have the problem that occasionally we would hit situations in which final decrypted $r - s$ values would take on extreme values, e.g., $r - s = -M$. In this case, an adversary could deduce that r must have equalled 0 and so be able to link this vote value back to a subset of the receipts, i.e., receipts with the index value 0.

Using a distribution avoids such “edge effects” whilst avoiding our having to compute arbitrary discrete logs in Z_p^* . Arguably, the adversary would be able to assign a non-flat probability distribution to the possible r values, but as long as no values of r can ever be eliminated, plausible deniability will be maintained.

We should also observe that even if it were possible to link a vote back to a particular index value, this would not typically violate ballot secrecy unless this it so happened that this identified a unique receipt, i.e., there happened to be only one receipt with this r value.

8 Auditing the Ballot Forms

The mechanisms described above allow for the distributed generation of ballot forms and just-in-time decryption of the candidate list and printing of the ballot forms. This has clear advantages in terms of removing the need to trust a single

entity to keep the ballot form information secret and avoiding chain of custody issues. On the other hand, it means that we can no longer use the random pre-auditing of pre-printed ballot forms as suggested in [CRS05]. Consequently, we must introduce alternative techniques to detect and deter any corruption or malfunction in the creation of the ballot forms.

A possible approach, in the supervised context at least, is to incorporate the two sided ballot form mechanism suggested in [Rya06] and re-introduce a cut-and-choose mechanism into the voter protocol. Here, a ballot form would be assigned two independent, entangled pairs of onions. One printed on one side of the form, the other on the flip side. In the booth, on each side, the left hand onion would be decrypted and the corresponding candidate list printed in the left hand column. The result is two independent ballot forms, one printed on each side, as illustrated in Figure 3.

Obelix		-----
Asterix		-----
Panoramix		-----
Idefix		-----
	<i>7rJ94K</i>	-----

Side 1

Panoramix		-----
Idefix		-----
Obelix		-----
Asterix		-----
	<i>Yu78gf</i>	-----

Side 2

Fig. 3. Prêt à Voter ballot form

Figure 3 shows the two sides of such a dual ballot form. These two sides should be thought of as rotated around a vertical axis. Note that each side has an independent randomization of the candidate order along with the corresponding cryptographic values. Thus each side carries an independent Prêt à Voter ballot form.

The voter uses only one side to encode their vote and makes an arbitrary choice between the sides. Suppose that the voter in this case chooses what we are referring to as side 2 and wants to cast a vote for Idefix. They place an X against Idefix on side 2 and then destroy the left hand strip that shows the candidate order for side 2. This results in a ballot receipt of the form shown in Figure 4.

These two sides should be thought of as being rotated around a vertical axis with respect to each other. Thus the shaded, third column of side 1 would oppose the candidate list of side 2.

The voter makes a random choice of which side to use to cast their vote and made their mark on the middle column against their candidate of choice and leave the flip, unselected side blank. The left hand column of the selected side is destroyed, and so the blank column of the flip side is destroyed. This results in a receipt on which the candidate list for the chosen side has been destroyed, whilst the ballot form on the slip, unselected side is intact, i.e., still has the onion value

Obelix	
Asterix	
Panoramix	
Idefix	
	<i>7rJ9AK</i>

auditable side
(remaining part of Side 1)

X	-----

<i>Yu78gf</i>	-----

vote encoding side
(remaining part of Side 2)

Fig. 4. Both sides of a Prêt à Voter ballot receipt

and candidate list. The information on both sides would now be recorded when the ballot is cast and posted to the WBB.

This flip side can now be audited and checked to ensure that the candidate list printed by the booth correctly corresponds to the onion value. Such checks could be performed immediately at the time of casting to detect any problems as soon as possible. Additionally, checks could be performed on the posted values.

In addition to such post-auditing of the dual ballot forms, we can do some pre-auditing of the committed onions pairs. This would help pick up any malfunctions or corruption in the preparation of the proto-forms at an early stage.

9 Auditing the anonymising mixes

In order to detect any malfunction or corruption by the mix tellers, we can again use the Partial Random Checking approach of [JJR02]. Here the checks on audited links will be slightly different: rather than revealing the seed information for the layer in question, the teller is required to reveal the re-randomisation value used to e-encrypt the select link. Auditing of the decryption tellers is quite straightforward as we don't need any further mixing at this stage (the anonymising mixes will be enough to ensure ballot secrecy). The correctness of the decryptions can thus be directly checked by simply encrypting the final values with the public keys and checking that these agree with the initial terms.

10 Handling full permutations and STV style elections

In order to deal with full permutations of the candidate list it is not immediately clear how to generalise the approach of section 7. As mentioned, one possibility is to leave the index values unchanged through the mixes. This might be acceptable in some situations but is clearly not satisfactory in general.

One solution is simply to have one onion for each candidate position. For a single candidate selection the ballot receipt would in effect simply be the onion

value against the chosen candidate. This feels rather inelegant and inefficient in terms of multiplying up the number of onions required.

For a ranked voting method, in which the voters are required to place a rank against each candidate, a ballot receipt would now comprise n pairs of rank value and onion. Each of these pairs could be put through the mix separately with the rank value unchanged (allowing the adversary to partition the mix according to the rank values seems not to matter). This approach works fine as long as the voting method does not require a voters rankings to be kept grouped for tabulation, as with STV for example.

11 Remote voting with Prêt à Voter

The encrypted ballot forms proposed here would appear to be adaptable to remote voting. We could for example, use a protocol like that described in [ZMSR04], to transform left-hand onions encrypted under the registrars' public key to terms encrypted under an individual voter's public key. The protocol of [ZMSR04] achieves this without having to reveal the underlying plaintext (seed) in the process. A pair of such ballot forms could be supplied to each voter in order to mimic the cut-and-choose mechanism described above. Details of such protocols are the subject of ongoing research.

Any remote voting scheme must face problems of coercion. A possible approach to counter such threats is the use *capabilities* as proposed in [JCJ02]. The possibility of using such a mechanism in conjunction with Prêt à Voter 2005 was explored in [CM05]. Voters are supplied with capabilities that are essentially encryptions of a nonce and a *valid* string. Votes are cast along with a capability and these go through the mix alongside the ballot terms. They emerge from the mix decrypted. A valid capability will decrypt to a valid plaintext. The validity or otherwise of the capability is not apparent until it is decrypted. As a consequence, a voter who is being observed whilst casting their vote has the possibility of deliberately and surreptitiously corrupting their capability. As long as the voter has some window of unobserved access to system he can cast his vote with his valid capability.

12 Conclusions

We have proposed some extensions to Prêt à Voter 2005 to counter vulnerabilities identified previously:

- Authority knowledge of ballot form crypto variables.
- Chain of custody threats.
- Chain voting attacks.

– Kleptographic channels.

The new version of the scheme counters these threats by enabling the distributed construction of encrypted ballot forms by a set of clerks. As a result, only a collusion of all the clerks could determine the cryptographic seed values. This eliminates the need to trust a single entity to keep this material secret and prevents Kleptographic attacks.

Our construction results in ballot forms in which the cryptographic seed values remain encrypted and can be decrypted on demand. Thus, the ballot forms with the candidate ordering can be created and printed in the booth, so eliminating chain of custody and chain voting threats.

The new construction uses ElGamal encryption and so is better suited to using re-encryption mixes for the anonymising/tabulation phase. Earlier work on robust ElGamal mixes may be found in [JJ01,Nef01,GJJS04]. The rather special representation of the ballot receipt in Prêt à Voter, index value plus cryptographic onion, means that it is not entirely straightforward to send such terms through a re-encryption mix. We have shown how, for single candidate selection and cyclic shifts of the candidate list at least, the ballot receipts can be transformed into pure ElGamal terms and so are adapted to re-encryption mixes. We have indicated how the approach may be generalised to deal with alternative electoral methods.

This version of the scheme is, we believe, technically superior to the 2005 version in that it requires less trust assumptions and is more robust against a number of threats. On the other hand, from a socio-technical point of view, it may have certain disadvantages. The voter experience is a little more complex, in particular the need for the cut-and-choose element on the voter protocol, which could have usability implications as well as opening up possibilities of “social engineering” style attacks, [KSW05]. Thus, it is possible that, for some situations like general elections perhaps, in evaluating the trade-off between the trust assumptions of Prêt à Voter 2005 and the usability issues of this scheme, the former might be deemed more acceptable.

13 Acknowledgements

The authors would firstly like to thank Paul Syverson and Ron Rivest for suggesting adapting Prêt à Voter to work with re-encryption mixes. We would also like to thank James Heather for many helpful discussions, as well as Ran Cannetti, Michael Clarkson, Rob Delicata, Joshua Guttman, Markus Jakobsson, and Thea Peacock. Finally, we are grateful to Paul Syverson and the anonymous referees for their comments on this paper.

References

- [CM05] M. Clarkson and A. Myers. Coercion-resistant remote voting using decryption mixes. In *Workshop on Frontiers of Electronic Elections*, 2005.
- [CRS05] D. Chaum, P.Y.A. Ryan, and S. Schneider. A practical, voter-verifiable election scheme. In *European Symposium on Research in Computer Security*, number 3679 in Lecture Notes in Computer Science. Springer-Verlag, 2005.
- [GJJS04] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. In *Proceedings of the 2004 RSA Conference, Cryptographers' track*, San Francisco, USA, February 2004.
- [JCJ02] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. To appear, 2002.
- [JJ01] M. Jakobsson and A. Juels. An optimally robust hybrid mix network (extended abstract). In *Proceedings of Principles of Distributed Computing — PODC'01*. ACM Press, 2001.
- [JJR02] A. Jakobsson, A. Juels, and R. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *USENIX Security '02*, 2002.
- [KSW05] C. Karlof, N. Sastry, and D. Wagner. Cryptographic voting protocols: A systems perspective. In *USENIX Security Symposium*, number 3444 in Lecture Notes in Computer Science, pages 186–200. Springer-Verlag, 2005.
- [M. 06] M. Gogolewski et al. Kleptographic attacks on e-election schemes. In *International Conference on Emerging trends in Information and Communication Security*, 2006. <http://www.nesc.ac.uk/talks/639/Day2/workshop-slides2.pdf>.
- [Nef01] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In *Proceedings of the 8th ACM Conference on Computer and Communications Security — CCS 2001*, November 2001.
- [RP05] P.Y.A. Ryan and T. Peacock. Prêt à voter: a systems perspective. Technical Report CS-TR-929, University of Newcastle upon Tyne, 2005.
- [Rya06] P.Y.A. Ryan. Putting the human back in voting protocols. In *Fourteenth International Workshop on Security Protocols*, Lecture Notes in Computer Science. Springer-Verlag, 2006. To appear.
- [YY96] A. Young and M. Yung. The dark side of black-box cryptography, or: Should we trust capstone? In *Crypto'96*, Lecture Notes in Computer Science, pages 89–103. Springer-Verlag, 1996.
- [ZMSR04] L. Zhou, M. Marsh, F. Schneider, and A. Redz. Distributed blinding for elgamal re-encryption. Technical Report TR 2004-1920, Cornell University, January 2004.