

The Prêt à Voter Verifiable Election System

Peter Y. A. Ryan^{*†}, David Bismark[†], James Heather[†], Steve Schneider[†], Zhe Xia[†]

^{*}University of Luxembourg [†]University of Surrey

Abstract—The Prêt à Voter election system has undergone several revisions and enhancements since its inception in 2004, resulting in a family of election systems designed to provide end-to-end verifiability and a high degree of transparency while ensuring secrecy of the ballot. Assurance for these systems arises from the auditability of the election itself, rather than the need to place trust in the system components. This paper brings together the variations of Prêt à Voter, presents their design, describes the voter experience, and considers the security properties that they provide.

I. INTRODUCTION

A. Background and motivation

The use of technology in supporting elections offers many benefits, including more accurate and faster tallying, cost savings, and encouraging greater voter participation. However, it is important to retain confidence in the election processes, and elections should not only run correctly but be seen to run correctly. Recent developments in the US have focused attention on this issue, and have raised questions about the reliability of election equipment and the confidence that can be placed in its correct operation.

Proposals for electronic election systems have been emerging over the past 20 years [22], [40], [10], [51], [36], [30], [28], [14], [7], [6], [15], [16], [5], often with some use of cryptography providing the basis for security, both in terms of vote privacy, and in terms of defence against election fraud. Over the past five years the Prêt à Voter election system and developments of it have been proposed [47], [12], [50], [48]. This election system obtains its assurance from its *auditability*: it is designed to enable checking, by the voter and by audit teams, of the various phases of collecting and processing the votes, and provides mechanisms for challenging the election if fraud is identified. This is termed *end-to-end verifiability*. Individual voters obtain a receipt, containing their vote in encrypted form, that they can use for checking that their vote has indeed been included in the tally. Audit teams can check the decryption of the votes. The Prêt à Voter approach supports different kinds of election, from voting for individual candidates, to complete ranking of candidates. Votes remain private, even when the election is challenged. It is versatile enough to allow different encryption schemes (RSA, ElGamal, Paillier) and cryptographic mechanisms (threshold, zero-knowledge proofs, use of homomorphic properties) which can be used to meet different requirements as appropriate.

This paper describes the Prêt à Voter approach. It introduces the general approach to electronic voting systems and Prêt à Voter in particular, discusses the assumptions about the context of the system, and the kinds of security property that are applicable to electronic voting systems. The paper then gives

a system overview, describes the voter experience of using the system, presents the technical details of how the system works, and a threat analysis. While we present informal arguments to support claims that the schemes support the properties, we do not here attempt formal analysis. Formal analysis along with full system analysis is the subject of ongoing research.

B. Roles in electronic voting systems

Voters: The normal requirement on a voter is to cast a vote. In Prêt à Voter, voters also have the opportunity to verify that their vote has been recorded as cast, and the receipt provided by Prêt à Voter provides the mechanism to achieve that. In practice we cannot expect all voters to carry out this verification, but confidence in the election will increase with the number of voters carrying out this check.

Election authority: Many of the responsibilities of the election authority are practical: distribution of ballot forms, recruitment of local officials, aggregation of votes, publishing information, announcing the result and so forth. In a verifiable voting system, most of these duties remain but those leading to the announcement of the outcome of the election must be verifiable. Furthermore, the secrecy of the votes may, in some systems, be dependent on procedures carried out by the election authority and it seems likely that such procedures will always be a part of election systems [33]. Within Prêt à Voter key elements of the election authority include the mix servers and tellers, which are responsible for processing and decrypting the recorded votes so that they can be tallied.

Auditors: The role of the auditor is to provide an expert opinion on evidence of proper function published by the electronic voting system, by checking or auditing the published information. Auditors can be any interested party, or those appointed by different interest groups to ensure they are trusted by the electorate and to act on their behalf.

Help organisations [1]: These are the parties in the polling stations who are available to help voters correctly follow the procedures involved in voting and checking, or to act on voters' behalf in carrying out the non-private elements of the voting and checking procedures.

C. Assumptions

The Prêt à Voter system in this paper is presented in the context of various assumptions about the systems and processes that provide other aspects of an overall election. The system focuses on obtaining the votes from the voters, and processing those votes towards an election result. It is important to recognise the assumptions which underpin any claims of a trustworthy election system.

- *Electoral roll.* The electoral roll provides the list of eligible voters for an election. It is clearly important that this information is correct, and that only eligible voters are allowed to vote, and that all eligible voters are allowed to vote. Responsibility for the correctness of the electoral roll is outside the scope of Prêt à Voter, but it is clear that the reliability of the election is dependent on its veracity. Another aspect of the electoral roll is tracking who has voted in a particular election to prevent a voter casting multiple votes. Thus the whole question of who is entitled to cast a vote at a particular time is an important one to get right, but is outside the scope of Prêt à Voter. The assumption of Prêt à Voter is that any votes cast into the system are indeed valid, and it is the responsibility of the election authorities to ensure this.
- *Chain of custody.* We assume that ballot forms cannot be forged, and that they are kept confidential between their creation and use, so that unauthorised parties do not have access to them. The voter privacy issues rest on this assumption, since it is necessary that entire ballot papers should not be available to anyone other than the voter.
- *Privacy of polling booth.* We assume that the voters are able to cast their vote in private, without the possibility of being observed. Hence we assume for example that there are no hidden cameras in the polling booth where the voter marks her vote, and that the casting of the vote takes place in a controlled environment.
- *Bulletin board.* The system requires information about the election to be posted publicly, so that voters and public auditors can have the necessary access to the information to carry out their verification checks. The bulletin board provides a way of publishing the various stages of the election so that individual and public verification can take place. It is important that once the information is published it cannot be adjusted. We assume that there is a secure mechanism for doing this. In other words, we assume that we are able to publish information reliably and in a tamper-proof way [27].

D. Tallying methods

As our work aims at providing a secure and verifiable electronic voting system usable in all kinds of election, we believe it important to ensure that the system can be used to run not only First-Past-The-Post races but also Single Transferable Voting (STV) and similar ranking schemes, in those places where they are used, such as the Republic of Ireland, Australia, and Malta. To ensure that Prêt à Voter is able to do this, we here describe a number of different versions of the system. It does not seem feasible to come up with a single, practical scheme that will cater for all voting/tallying methods. Consequently it seems more sensible to propose variations of the theme tailored to the particular methods.

E. Key Security Properties

Integrity: Integrity in the context of an election system is the property that the result of the election is not manipulated or altered in any way. This means that all the steps involved

in processing the votes preserve the information that they are processing. In particular, steps that transform the representation of the vote do not alter the vote itself.

The election process may be considered in three stages: casting the vote, recording the vote, and then tallying the votes. Integrity will require that each of these stages is honest. These requirements are respectively termed:

- *cast as intended*, meaning that the vote captured by the system (on a ballot form, touch screen, optical scan, lever system, or any other method provided to the voter to cast her vote) should correspond to the vote that the voter intended to cast. This is not a security property as such, but it is a usability requirement and one that is necessary to justify the claim that the outcome of the election reflects the will of the voters;
- *recorded as cast*, meaning that the vote data to be processed by the system corresponds to the votes that were cast by the voters. We take this encompass the requirement that the voter's choice be correctly encoded in any receipt;
- *counted as recorded*, meaning that the process of tallying the votes gives the result corresponding to the votes that were recorded.

Privacy: One of the principles of modern elections is the 'secret ballot', whereby it should not be visible externally how any particular voter voted. This property may be considered in terms of anonymity or secrecy. Roughly:

- *anonymity* requires that for any particular vote, it is not known which voter cast that vote.
- *vote secrecy* requires that for any particular voter, the vote that they have cast is not known.

These are useful informal definitions, though they will not cover corner cases such as a unanimous election, in which it is trivially known how everyone voted. A more precise definition covering both properties is:

- Let A_1 and A_2 be two voters, and v_1 and v_2 be two votes. No-one should be able to distinguish between a case where A_1 casts v_1 and A_2 casts v_2 , and a case where A_1 casts v_2 and A_2 casts v_1 .

Anonymity and vote secrecy are two sides of the same coin: in both cases they require that there is no externally observable link between a voter and the vote that they have cast. Anonymity comes from the point of view of the vote, and secrecy from the point of view of the voter.

Secure voting systems such as Prêt à Voter can introduce features into the voting system not seen in traditional ballot-box elections. One common feature is the use of a receipt to provide a voter with some evidence of their vote, and to enable her to verify later that it has been correctly included. The introduction of such a receipt introduces new vulnerabilities and hence new security requirements on voting systems:

- *receipt-freeness* is the requirement that voters are not able to prove to a third party how they voted. In other words, voters should not have, or be able to generate, evidence of how they voted. This is important to avoid vote selling, or demonstrating to a coercer after the election that the voter has voted in a particular way. A receipt can provide

evidence that some vote was cast, but not which vote was cast.

- *coercion-resistance* means that the system provides mechanisms that would foil any potential coercer, who is in a position to require a voter to vote in a particular way. Even if the voter is interacting with the coercer during the voting process, the coercer should not be able to establish whether the vote was cast in the way demanded.

Verifiability: One of the key aspects of Prêt à Voter and other secure voting systems is the notion of *verifiability* of the election. This is the property that the result of the election, and the processing of the votes, can be publicly verified or audited after the election has taken place.

- *individual verifiability* We take this to refer to the ability of individual voters to confirm that their choice has been correctly encoded in their receipt.
- *public verifiability* means that anyone can verify that the receipts posted to the Bulletin Board have been correctly decrypted and tallied.
- *end-to-end verifiability* means that all the stages of the election, from the casting of the vote, through to the tallying of all the votes, can be verified: that the declared election result really is the correct tally of all the votes that were cast. End-to-end verifiability can be public or individual, or a combination of the two (where individuals verify some aspects, and public auditors verify other aspects).

This form of verifiability is concerned with auditing the election *data*. Verifiability therefore requires that this data is published during or after the election, to enable the checks to take place. It is not concerned with the reliability or verification of the election machinery itself, since its correct operation is checked through verifying the published election data. Thus concerns about tampering with or replacing equipment are addressed. The claim that integrity of the election was upheld becomes a mathematical theorem concerning the publicly available data, and this theorem can be mechanically checked.

Robustness: This is concerned with resilience in the face of random faults as well as deliberate attempts to disrupt the election, such as denial of service attacks. One aspect of this is an ability to recover from cheating when it is detected. Another aspect is the ability to run the election even in the face of a minority of dishonest election authorities, for example tellers refusing to decrypt ciphertexts, or mix servers failing to operate. Techniques such as fault tolerance, threshold cryptography and voter-verifiable paper audit trails [35] can be used to provide robustness.

F. Non-functional properties

In addition to the security properties described above, any voting system should be capable of supporting real elections with the voting public. Aspects of the properties necessary to achieve that are as follows:

- *Timely interaction* In the vote casting phase, it should operate at ‘human speed’: the speed a voter would be able

Donald	
Barack	
Alice	
Crystal	
Edward	
	a6Gq21p

Fig. 1. A Prêt à Voter ballot form

to use it, and its response times for interacting with voters should be of the order of seconds or tens of seconds.

- *Timely tallying* It should process and count the votes at least as quickly as the system it is designed to replace.
- *Usability* The system should have an intuitive way of voting—ideally, the procedure should be very similar to the one voters are already familiar with. It should also be easy for the election officials to run an election—as easy or easier than with conventional elections.
- *Election Versatility* A versatile election system should be able to handle a variety of tallying methods, and provide support for different ways of voters casting their votes (in particular, for voters who are unable to cast a vote in the usual way).

II. SYSTEM OVERVIEW

The Prêt à Voter system operates in four quite distinct parts: ballot generation, vote capture, vote processing, and auditing. In this section, we shall give an overview of the operation of, and design philosophy behind, each part. The precise details vary according to which flavour of Prêt à Voter is being used, but the basic idea remains the same.

A. Ballot generation

A Prêt à Voter ballot paper, as shown in Figure 1, contains a detachable list (usually the left-hand half of the paper) of candidate names, given in a random order, and corresponding boxes into which the voter’s preference should be recorded (the right-hand half). This right-hand half also contains encrypted information that enables the system to reconstruct the candidate order, but encrypted in such a way that no single party is able to perform the decryption alone. For historical reasons this encrypted information is called an ‘onion’. The early proposals for Prêt à Voter [47], [12] built up the encrypted information in a series of layers, so the layers could be ‘peeled’ off one at a time by the decryption stages, hence the use of the term ‘onion’. However, the terminology now commonly applies to any encrypted information regarding the associated with a candidate ordering used in this way.

The random candidate ordering is what provides voter privacy. The left-hand side will be detached and destroyed before the vote is scanned, and the voter will retain the right-hand side (either the original or a copy) as a receipt. Provided that no-one except the voter knows the ordering, and the link between the voter and this particular ballot paper is lost before

the vote is decrypted, no-one else (including the scanner in the booth) will ever know the voter's preference.

B. Vote capture

Vote capture is simply a matter of reading in the right-hand side of the ballot paper and sending it to the vote database. No cryptographic operations need to be performed, other than perhaps applying a digital signature to the receipt. The voter retains the right-hand side as a receipt; the booth machine marks the receipt as authentic. The encrypted vote (that is, the right-hand side) is published so that anyone in possession of the receipt can check that it appears on the bulletin board unaltered.

C. Vote processing

The essential idea behind the vote processing part of Prêt à Voter is to transform the set of encrypted votes into a set of unencrypted votes, but without allowing anyone (including those involved in the decryption) to perform end-to-end matching. Three tasks need to be performed: mixing, decrypting, and tallying; some Prêt à Voter variants combine the mixing and decrypting phases, and some combine the decrypting and tallying phases.

A tool introduced by David Chaum [10], the mix net, is used in electronic voting system to anonymise the source of an encrypted vote while guaranteeing that the source is valid and that the vote has not been changed. In general a whole set of encrypted votes are passed between a set of mix servers and shuffled in secret one or more times by each party. At each stage the set of encrypted votes are made to look different (see Sections IV-A and IV-B, including a diagram in Figure 7) to hide the secret shuffle. Both the shuffle and this hiding are subsequently verified using one of several methods now available.

The arguably simplest method used to verify a mix network is the Randomised Partial Checking [30]. Here each layer in the mixnet performs its secret shuffle and transformation of the votes, and passes the result to the next layer. The set of votes at all stages is published. To audit this process, a random set of the encrypted votes are selected in a public process, ensuring that the choices could not have been known in advance. The mixnet is then obliged to publish proof of the source of these votes. The votes to audit must be chosen so that no complete path from vote as recorded to vote as decrypted is exposed, to preserve voter anonymity. The chances of undetected dishonesty in the mixnet decreases exponentially with the number of votes tampered with.

We note that it may also be possible to use homomorphic tabulation techniques in place of mixes. This was done for example in *Scratch and Vote* [2], which uses Prêt à Voter ballots but with homomorphic tabulation.

D. Auditing

In order for the voters, officials, candidates etc. to be convinced that the published tally corresponds to the votes cast, they need to be able to check that the mixing, decrypting

and tallying phases have all been performed correctly. Every cryptographically protected operation here publishes enough information for voters (and others) to be able to verify correctness. The details vary: the information may consist of a zero-knowledge proof; it may consist of information enabling anyone to reverse the operation that has been performed and confirm that the output can be transformed back into the input. In many forms of Prêt à Voter, the decryption phase results in publication of all of the (anonymized) unencrypted votes; in such cases, the tallying operation can be publicly verified without any further information.

End-to-end verifiability results from voters being able to verify that their intent has been properly recorded, and public auditors being able to verify that the votes as a whole have been counted as recorded.

III. VOTER EXPERIENCE

In this section we describe what a voter would experience when using a Prêt à Voter voting system. There are several developments of the Prêt à Voter voting system, which give rise to variations in the details of the voting experience, but they have common features.

The heart of the voting process provides a way of obtaining an encrypted vote from a voter. This requires the information required to vote (normally the list of candidates) to be presented to the voter, so that the vote can be captured in an encrypted form. The information is destroyed once the voter has made use of it to cast the vote, so the vote can then only be understood via decryption.

A. Ballot Form Layout

The standard Prêt à Voter ballot form has been pictured in Figure 1. It is designed so that the two halves of the form can be separated, for example by means of a perforation running down the middle of the ballot form. This enables the plaintext information concerning the candidate list to be removed once the vote has been cast, leaving the vote in purely encrypted form. A vote on the remaining half of the form, together with the encrypted information, comprises an encrypted vote.

All elections that place votes against candidates can be run using this ballot form: whether votes involve selection of a single candidate, a number of candidates, or a ranked list of candidates, and however these will be tallied. Different underlying decryption mechanisms are required in different cases, but from the voters' perspective all such elections can be supported with this ballot form, and votes cast in the traditional way.

The original Prêt à Voter scheme [47], [12] proposed the use of a particular form of permutation: cyclic shifts of the candidate list, whereby the list is always in the same order, but can start at any arbitrary point, and wrap around. This is appropriate for capturing a vote against a single candidate, since the position of the vote gives no information about the candidate voted for, but it is not appropriate where votes are cast against more than one candidate, since the relative positions of candidates, and likely combinations of votes, will leak some information about the vote cast from the

Donald	
Barack	X
Alice	
Crystal	
Edward	
	a6Gq21p

Donald	4
Barack	1
Alice	5
Crystal	2
Edward	3
	a6Gq21p

Fig. 2. Completing the ballot form: single vote (left) or preference list (right)

X
a6Gq21p

Fig. 4. The scanned vote and receipt

Donald	
Barack	X
Alice	
Crystal	
Edward	
	a6Gq21p

Fig. 3. Detaching the candidate list

pattern of votes recorded on the right-hand side. In fact, the scheme generalises to allow arbitrary permutations of the candidate list, and can then be used to record votes for multiple candidates as well.

The re-encryption schemes of [50], [48] are only able to handle cyclic shifts of candidate lists, because of the properties of their respective cryptographic mechanisms used in processing the votes. Hence they are only appropriate for supporting votes for a single candidate. However, later developments [55], [54], [26] do support arbitrary permutations of candidate lists. This requires an onion for each candidate; these might appear on the ballot form on the right hand side against the candidate names, but the information could equally well be at the bottom of the form and associated with the votes after they have been cast. From the voter’s point of view this can be set up to look exactly like the earlier schemes.

B. Vote Casting

The voter casts a vote by filling in the boxes on the right hand side of the ballot form, against the chosen names. If a single name is to be chosen, an ‘x’, tick or other mark is placed against the name. If a preference list of names is to be chosen, then the appropriate preferences are placed against the chosen names, in the conventional way. This is pictured in Figure 2.

The two halves of the ballot form are then separated, as pictured in Figure 3. The left-hand half, consisting of the list of candidates, is destroyed. Its destruction is necessary to ensure that the voter cannot later prove how she voted, and hence provides resistance to coercion and vote-selling. Thus the voter must be required to destroy the left-hand half as part of the voting process.

The right-hand half of the ballot form, pictured in Figure 4, consists of the vote to be cast. It will be first scanned into the

Prêt à Voter voting system, and then signed. After that, the voter can retain it as the receipt of the vote cast. Note that the help organisations in the polling station are supposed to help the voter to check that the signature in her receipt is valid. This provides protection against election challenges with fake receipts—that is, receipts for votes which were never cast. The scanning system will need to extract the position of the vote or preferences on the sheet, and the encryption of the list ordering, in order to process the vote.

A possible variation on the receipt is for the Prêt à Voter system to print its own (signed) record of its scanned information, and provide this as a receipt. By matching this with the original vote, this will allow the voter to confirm that the information on the form has been correctly understood by the system.

C. Audit of ballot forms

Voters may wish to check that the order of candidates claimed to be encrypted on the right-hand side does indeed correspond to the list printed on the left-hand side. If this were not the case then a vote cast for one candidate may be considered after decryption as a vote for a different candidate. To provide such reassurance, voters may elect to ‘audit’ a ballot form. This involves removing the left-hand side of the ballot form, and asking the system to decrypt the candidate list from the onion on the right-hand side. The voter can then check that the decrypted list matches the list of candidates printed on the left-hand side. In principle, this audit can be carried out as often as the voter wishes. This gives the voter confidence that the ballot forms have been correctly constructed.

However, the voter is not allowed to cast a vote on a decrypted ballot form. Once the candidate list associated with a onion is known, vote privacy, and hence resistance to coercion and vote-selling, is lost. The audit process gives an individual voter confidence that the ballot forms are correctly constructed, but does not allow her to check the ballot form that she is using to cast the vote.

An alternative approach, proposed in [50], is to print a ballot form on each side of the ballot paper, in such a way that the removal of the candidate list on one side does not affect the information on the other side. The two ballot forms must be independent. The voter chooses arbitrarily one side of the ballot form on which to cast the vote, marks her vote in the usual way, and detaches the list of candidates so the vote can

be cast. The other side still constitutes a complete ballot form, and can be audited, checking that the decrypted candidate list corresponds to the printed candidate list.

D. Verifying the vote

Votes that have been cast are published on a bulletin board. Voters can check that the information on their receipt (the vote and the onion) appears on the board, and this gives them the assurance that their vote was indeed accepted into the system as a cast vote. If their vote does not appear on the board, or appears incorrectly, then they can use their receipt to challenge the election, since it provides evidence of a vote that has not been included in the tally. Voters are thus expected to retain their receipts as a protection against their vote being changed. A verifiable encrypted paper audit trail [34] or voter-verifiable paper audit trail [35] can also be used to provide additional assurance.

The process of decrypting the votes is also subject to audit processes, but the voter will not be directly involved in this. However, it will be known that agents from all parties, and from neutral organizations, participate in the auditing process, and this should be sufficient to provide confidence in the decryption process. This reflects current practice, whereby voters are aware that there are audit or checking processes that they have confidence in, but do not participate in directly. Once the votes are decrypted they can be made public, enabling the vote tally and election result to be checked by any interested voter.

E. Pre-print/on-demand printing of ballot forms

Secrecy of the ballot relies on the fact that the list of candidates cannot be deduced from the onion without the ability to decrypt. However, the ballot form itself, when entire, provides an association between the onion and the candidate list. This means that ballot forms need to be managed carefully, and the chain of custody between the creation of a ballot form and its use in a polling station needs to be trusted.

An alternative approach is to print ballot forms at the point they are needed. This can be achieved by providing a ballot form with the candidate list encrypted in two different ways: one which can be decrypted in the polling station, and one which can be decrypted by the Prêt à Voter tellers as previously. The list of candidates is then printed on demand, when the ballot form needs to be used at the polling station. This addresses the chain of custody issues, and provides additional assurance to the voter that external parties cannot have seen the candidate list associated with their particular ballot form or receipt. Ballot forms can be audited in the same way as previously, by asking for the candidate list associated with the right-hand side to be revealed, and checking that it matches the left-hand side.

With the exception of having the ballot form printed on demand as a voter wishes to vote, the voting experience—casting a vote, auditing a ballot form, separating a ballot form into two halves, checking the vote—is identical with the experience of using a pre-printed ballot form.

F. Summary

From a voter’s perspective, the voting experience consists of several parts, though in fact only the first part, casting the vote, is absolutely required for participation. The remaining parts are optional for the voter to obtain additional assurances in the integrity of the election, though it is to be hoped that sufficient users would exercise these options to deter attempts to subvert the election.

- casting the vote: filling out a ballot form (either pre-printed, or printed on demand), removing the left-hand side, having the right-hand side scanned into the voting system;
- optionally auditing ballot forms, checking that the decrypted candidate lists correspond to the candidate lists already printed;
- optionally retaining the receipt of the vote. This might be the original scanned right-hand half, or it might be generated by the voting system and checked in the polling station against the original vote;
- optionally checking (some time later) that the vote has been included in the list of cast votes, by checking the receipt against the web bulletin board it should be published on;
- optionally checking the tallying, and the election result, from the published decrypted votes.

The deliberate aim is to keep as much of the complexity of the system, required to ensure the security of the election and described in the rest of this paper, away from the voter. The voter’s experience needs to be kept as simple as possible, and the demands on the voter should be kept to a minimum.

IV. TECHNICAL DETAILS

In this section, we explain how to generate the ballot forms and, after voters have cast their votes, how these votes can be tallied. The procedures in this section are only implemented by the election authorities and they are transparent to ordinary voters, and so we call them the back-ends of the election systems. We first discuss how the back-ends can be designed using decryption mixnets, and then we show how they can be designed using re-encryption mixnets without affecting the voter experience.

A. Back-ends by decryption mixnets

The decryption mixnet was introduced by David Chaum in [10]. We explain how this mixnet can be used to design the back-ends of Prêt à Voter. The involved parties are an *election authority* who generates the ballot forms and a number of *mix servers* who shuffle and decrypt the received votes. We discuss later how the election authority may be distributed.

1) *Ballot generation*: Suppose there are k mix servers in the election system. Before generating the ballot forms, each mix server generates two RSA key pairs. They publish their public keys and distribute their private keys among all other mix servers in a threshold fashion [52]. All ballot forms are generated by an election authority. For each ballot, $2k$ germ values $(g_0, g_1, \dots, g_{2k-1})$ are randomly selected by

this authority. Then the onion is encrypted using each mix server's public keys iteratively as $D_{i+1} = \{g_i, D_i\}_{PK_i}$, for $i = 0, 1, \dots, 2k - 1$. The final output D_{2k} is the onion

$$D_{2k} = \{g_{2k-1}, \{\dots, \{g_1, \{g_0, D_0\}_{PK_0}\}_{PK_1} \dots\}_{PK_{2k-2}}\}_{2k-1}$$

At the same time, the candidate list in alphabetical order is cyclically shifted by $\theta = \sum_{i=0}^{2k-1} d_i \pmod{v}$, where $d_i = \text{hash}(g_i) \pmod{v}$, and v denotes the number of candidates. For each ballot form, the candidate list and its corresponding onion will be printed onto the ballot form by the election authority. This is illustrated in Figure 5.

Donald	
Edward	
Alice	
Barack	
Crystal	
	a6Gq21p

Fig. 5. A ballot form example

2) *Voter experience*: A voter first marks the choice against her preferred candidate, as shown in Figure 6. Then the left hand column of the ballot will be detached and destroyed. This voter's vote is only the right column of the ballot in Figure 6, which contains her mark $r_{j,2k}$ and the onion $D_{j,2k}$.

Donald	
Edward	
Alice	
Barack	X
Crystal	
	a6Gq21p

Fig. 6. A marked ballot form

3) *Ballot tallying*: In the tallying stage, all received ballots are shuffled and decrypted by a sequence of mix servers. Suppose the inputs to the i th mix server S_i are a list $L_{2i} = \{(r_{1,2i}, D_{1,2i}), \dots, (r_{n,2i}, D_{n,2i})\}$. For each value pair in the list, S_i will behave as follows:

- 1) Let j denote the ballot serial number, for $j \in \{1, \dots, n\}$, S_i first decrypts $D_{j,2i}$ using her private key SK_{2i} as $\{D_{j,2i}\}_{SK_{2i}} = (g_{j,2i-1}, D_{j,2i-1})$.
- 2) Then S_i applies the hash function to the germ value as $d_{j,2i-1} = \text{hash}(g_{j,2i-1}) \pmod{v}$.
- 3) S_i calculates the cyclical shift as $r_{j,2i-1} = r_{j,2i} - d_{j,2i-1} \pmod{v}$.
- 4) Now S_i obtains a list $\{(r_{1,2i-1}, D_{1,2i-1}), \dots, (r_{n,2i-1}, D_{n,2i-1})\}$. S_i then sorts the list in the lexical order, resulting a list $L_{2i-1} = \{(r_{\pi(1),2i-1}, D_{\pi(1),2i-1}), \dots, (r_{\pi(n),2i-1}, D_{\pi(n),2i-1})\}$.

S_i repeats the above procedures again, outputting the list L_{2i-2} to the next mix server. The following mix servers execute similarly. The last mix server's output is the decrypted votes.

4) *Audit the ballot tallying phase*: The above ballot tallying phase can be publicly verified using Randomised Partial Checking (RPC) [30]. The basic idea of RPC is to provide strong evidence that the mixnet is correctly implemented.

By using RPC, each mix server needs to implement two shuffles and every two adjacent mix servers are paired, as show in Figure 7. To audit a mixnet, each mix server pair should be verified separately as:

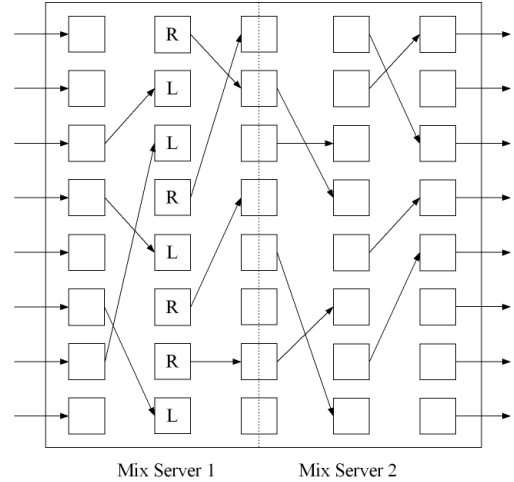


Fig. 7. Randomised Partial Checking

- 1) For the left mix server, the auditor will go down the middle column and randomly assign half the units L and the other half R .
- 2) For units assigned L , the auditor requires the left mix server to reveal the corresponding links in its first shuffle (incoming links).
- 3) For units assigned R , the auditor requires the left mix server to reveal the corresponding links in its second shuffle (outgoing links).
- 4) For the right mix server, for exactly half of its received inputs, their incoming links have already been revealed. We denote that these units are in the group G_1 while other units are in the group G_2 . Then the auditor randomly selects half of the units in G_1 and half of the units in G_2 and requires the right mix server to reveal its outgoing links.
- 5) In the last shuffle, for the units whose incoming links have not been revealed, the right mix server is required to reveal its outgoing links.

When considering the two mix servers as a pair, the above auditing process will not reveal any information about how these two mix servers have shuffled the mixes. A proof of security of this process appears in [25]. A single round of verification will provide strong evidence that both mix servers have behaved correctly. Note that the random challenges can be generated by a number of sources, e.g., lottery style choosing

of elements or using the Fiat-Shamir heuristics [20]. In some cases it may be necessary to include shuffle commitments to deny the server any freedom in how it reveals links under audit. Most variants of Prêt à Voter ensure that all receipts have distinct plaintexts, which removes the need for such shuffle commitments. The distinct plaintext requirement needs to be enforced by running *Plaintext Equivalence Tests* [9], [29] across the posted receipts. This needs to be done in any case to counter ballot doubling attacks against privacy.

5) *Handling ranked elections*: The above protocol also can be implemented in ranked elections. However, the candidate list in the ballot form needs to be randomly permuted instead of just cyclically shifted; otherwise, a receipt might leak information about the vote. Therefore, when generating a ballot form, the election authority should use the germ values to determine a random permutation as $\phi_i = f(\text{hash}(g_i))$, where hash is a collision-resistant hash function and f is a suitable, unbiased mapping from the domain of the hash function to the set of permutation on the set of candidates. The total permutation can be represented as $\phi = \phi_0 \circ \phi_1 \circ \dots \circ \phi_{2k-1}$. Note that although this avoids information leakage from the receipt, it does not avoid the so-called *Italian attack*, in which lower-ranked preferences are used as a covert channel to provide a mechanism for coercion.

B. Back-ends by re-encryption mixnets

We now discuss how the ballots can be generated using re-encryption mixnets without affecting the voter experience. The involved parties are a set of election authorities, called *clerks*, who generate the ballot forms, a sequence of mix servers who shuffle and re-encrypt the received votes, and a number of *tellers* who decrypt the election result in a threshold fashion.

1) *Key generation*: The back-ends by re-encryption mixnets can be implemented using either exponential ElGamal [19] or Paillier [41]. Here, we only introduce the ones based on ElGamal, but it is a trivial modification to replace the ElGamal cipher with Paillier.

Suppose the ElGamal public parameters (α, γ, p, q) are made public in advance, where p and q are large primes such that $q|p-1$, and α, γ are generators of G_q which is a subgroup of Z_p^* with order q . A set of tellers first generate a secret key $x_T \in Z_q$ in a threshold fashion [42], [24]. (Note that if Paillier is applied in the mixnets, the techniques in [52] enable the tellers to generate the key pair in a threshold fashion, and the techniques in [21] can be used to decrypt the Paillier ciphertexts in a threshold fashion.) They publish the corresponding public key $\beta_T = g^{x_T}$. Furthermore, the voting machine randomly selects a private key $x_R \in Z_q$ and reveals its public key $\beta_R = g^{x_R}$. To ensure robustness, its private key needs to be distributed among the threshold tellers using verifiable secret sharing [18].

2) *Ballot generation*: An initial clerk C_0 randomly selects a batch of initial seeds s_i^0 from a binomial distribution centred around 0 and standard deviation σ , where σ can be chosen to be order of v , the number of candidates. Because the exponential ElGamal cipher is not a trapdoor function, this requirement ensures that m can be retrieved from h^m efficiently; if Paillier

is used, this requirement is not necessary. C_0 then generates a batch of onion pairs

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

where the blinding factors x_i^0, y_i^0 are randomly drawn from Z_q .

After that, the remaining l clerks perform as follows: each clerk takes the batch of onion pairs output by the previous clerk and injects fresh entropy into the seed values. For each onion pair, the same entropy is injected into the seed value of both onions. We suppose the batch of onions pairs received by the clerk C_j is

$$(\alpha^{x_i^{j-1}}, \beta_R^{x_i^{j-1}} \cdot \gamma^{-s_i^{j-1}}), (\alpha^{y_i^{j-1}}, \beta_T^{y_i^{j-1}} \cdot \gamma^{-s_i^{j-1}})$$

C_j randomly selects a batch of seeds \bar{s}_i^j from the same binomial distribution centred around 0 and σ . Then C_j calculates her output batch of the onion pairs as

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

where the blinding values \bar{x}_i^j, \bar{y}_i^j are randomly drawn from Z_q , and

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

Finally, the last clerk C_l will output a batch of onion pairs

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

where

$$\begin{aligned} x_i &= x_i^l = x_i^0 + \sum_{j=1}^l \bar{x}_i^j \pmod{q} \\ y_i &= y_i^l = y_i^0 + \sum_{j=1}^l \bar{y}_i^j \pmod{q} \\ s_i &= s_i^l = s_i^0 + \sum_{j=1}^l \bar{s}_i^j \pmod{q} \end{aligned}$$

For each ballot form, an onion pair is printed at the bottom, as shown in Figure 8. If all clerks are honest, the two onions in a ballot form will contain the same seed value. The onion in the left hand column is encrypted using public key β_R , thus the voting machine can decrypt it and retrieve the seed value. The onion in the right hand column can be decoded only by a quorum of tellers.

3) *Voter experience*: The voter experience is similar to the description of the previous section. An additional task for the voter is to insert the left hand column into the voting machine, which will read the onion, decode it and print the corresponding candidate list on the ballot form. The candidate

4m9xe	7q3Kyc

Fig. 8. A blank ballot form example

list is determined by $s_i \pmod{v}$, where v is the number of candidates. After that, the voter marks her choice and cast her vote as normal. A vote example can be illustrated as in Figure 9.

X
7q3Kyc

Fig. 9. A vote example

4) *Ballot tallying*: After the election, all received votes are collected from the bulletin board. For each vote, the election officials will first perform some calculation to absorb the voter's choice index value ι into the onion as

$$(\alpha^{y_i}, \beta_T^{y_i} \cdot \gamma^\iota \cdot \gamma^{-s_i}) = (\alpha^{y_i}, \beta_T^{y_i} \cdot \gamma^{\iota - s_i})$$

The above calculation is done publicly. These encrypted values will then be inserted into a sequence of mixnets, which will shuffle and re-encrypt these terms by changing the randomisations while leaving the seed values untouched. Following this stage, the outputs of the mix network will be decoded by a quorum of tellers in a threshold fashion. Finally, the decrypted votes will be tallied and the election result will be announced.

5) *Auditing the ballot tallying phase*: The above ballot tallying phase also can be audited using RPC. However, in this case RPC needs to be run several times. The purpose is to provide a proof of correct operation. Otherwise, if the RPC only provides strong evidence, voter privacy might be violated without being detected if adversaries apply the attacks in [44]. Instead, if exponential ElGamal cipher is used in the re-encryption mixes, we can use either the Furukawa-Sako mix [23] or Neff's mix [36], [38] to challenge the mix. For Paillier re-encryption mixes, the techniques in [39], [43] can be applied.

6) *Handling ranked elections*: The above protocol does not directly handle ranked elections because the candidate ordering is restricted to cyclic shifts rather than permutations of the canonical ordering. However, some of its later improvements can be implemented in ranked elections. For example, the techniques in [54] give a general method by which all election methods can be handled using re-encryption mixnets.

Election systems with information-rich votes, such as STV and Condorcet, however, will still be vulnerable to the Italian attack. More recently, Heather (in [26]) and Teague *et al.* (in [53]) have shown how to implement STV elections in such a way as to counter the Italian attack.

V. THREAT ANALYSIS OF PRÊT À VOTER

We now briefly analyze the two Prêt à Voter protocols with respect to the requirements proposed in Section I. For simplicity, Prêt à Voter with decryption mixes is denoted as PAV 2005 and the one based on re-encryption mixes is denoted as PAV 2006.

- *Integrity*: We first argue that both Prêt à Voter protocols achieve the integrity property if all involved parties are honest. For every ballot form that has been properly generated, the onion in the right hand column can be used to reconstruct the candidate ordering in the left hand column. Thus any vote which contains the onion and the voter's choice index can be used to derive this voter's intent. Hence both schemes are able to ensure *cast as intended*. Also, if a voter's vote has been correctly displayed on the bulletin board, she can be ensured that her vote is *recorded as cast*. Furthermore, both schemes achieve *counted as recorded* because if all received votes are properly tallied (including shuffle and decryption), the counting of the tally outputs will reveal the election result.
- *Verifiability*: The verifiability property ensures that all involved parties have to behave honestly. Otherwise, their cheating behaviour will be detected with overwhelming probability. In any case if cheating behaviour is detected, dishonest parties will be removed and their role will be implemented by other parties. To see why both Prêt à Voter schemes have achieved *individual verifiability*, recall that each authenticated voter can be provided with a number of ballot forms, and she can randomly choose one ballot to cast her vote and challenge the other ones. Even in the case where a voter is provided with only two ballots, if one of her ballots is not properly constructed, she has a 50% chance of detecting cheating. Hence any attempt to cheat in more than a very small number of ballots would surely be detected. Furthermore, each voter will be provided with a receipt which contains her vote. She can check whether her receipt has been correctly displayed on the bulletin board. Otherwise, her receipt can be used as a proof to challenge the election. Therefore, both Prêt à Voter protocols enable voters to verify by themselves that their votes are correctly recorded in the election systems. In both Prêt à Voter protocols, the receipts are tallied using mixnets. *Public verifiability* can be achieved because it can be publicly verified that the mixnets act honestly, and therefore that the encrypted votes are correctly transformed into the decrypted votes.
- *Anonymity*: The anonymity property contains three levels of assurance. *Coercion-resistance* implies *receipt-freeness*, which implies *voter privacy*. Generally speaking, all Prêt à Voter systems only achieve voter privacy and receipt-freeness. (Coercion-resistance can be

achieved in the JCJ-style remote voting [31].) In Prêt à Voter schemes, voters are required to cast their votes in a secure voting booth. When a voter is casting her votes, the entire ballot form is available for her. Thus only the voter herself knows how she has voted. Afterwards, although each voter will be provided with a receipt, it cannot be used to prove others how she has voted. Therefore, both schemes provide some assurance of voter privacy and receipt-freeness.

- *Robustness*: There are various aspects of robustness. Both Prêt à Voter schemes achieve a high level of robustness against faulty election authorities. In PAV 2005, if any mix server is found cheating, it will be removed and its role will be simulated by a quorum Q of other mix servers. Therefore, as long as there are at least Q honest mix servers, the correct result will always be output. In PAV 2006, if any mix server is found cheating, she can be simply ignored or she can be replaced by another party. Moreover, if there are at least Q honest tellers, the outputs of the mixnet will always be correctly decrypted. However, there are other issues related to the property of robustness in the practical aspects of implementation, and they are future work.
- *Usability*: The usability of both schemes is very good. Voters do not need any special knowledge or ability to cast their votes, and their mandatory tasks have been reduced to the minimum: just vote-and-go. Also, the ballot form layout is familiar to voters' previous experience. Furthermore, it is easy for the election authorities to set up and control the election system.
- *Versatility*: PAV 2005 is very versatile. It can implement not only FPTP elections, but also approval voting, Borda Count voting, STV and Condorcet voting as well. Although PAV 2006 in its original form can be used only for FPTP elections, some of its adaptations have shown how the scheme can be extended to handle other election methods.

A. Vulnerabilities

Although we have illustrated that both Prêt à Voter protocols have achieved most of the desired properties, they still suffer several vulnerabilities.

- *Authority knowledge attack*: Only PAV 2005 suffers this attack, because all ballot forms are generated by a single election authority. If this party is dishonest, she can not only learn how a voter has voted just from the receipt, but also can apply the *subliminal & Kleptographic channel attack* [32] to enable her colluding parties to have such power. The 2006 version and later versions counter this by introducing distributed constructions for the ballot forms.
- *Discarded receipt attack*: This attack is suffered in both Prêt à Voter protocols. Voters need to use their receipts to check whether their votes are correctly recorded by the election system. Thus if some voters have discarded their receipts, adversaries may safely display their votes incorrectly on the bulletin board without being accused.

To resolve this problem, we could use a *Voter-Verifiable Paper Audit Trail* (VVPAT) [35], or techniques specifically designed for Prêt à Voter in [34].

- *Chain voting attack*: This attack is applicable only if all ballot forms are printed before the election. If adversaries successfully smuggle a blank ballot form out of the voting booth, they can use this ballot to coerce a large number of voters. The adversaries force or bribe a voter to vote in a particular way using this ballot and bring out another blank ballot form. Thus after this voter has cast her vote, the adversaries still have a blank ballot form. They can repeat this attack to coerce other voters. But if all ballot forms are printed on-demand, it will be much more difficult for adversaries to launch this attack.
- *Italian attack*: PAV 2005 for ranked elections is vulnerable to the Italian attack simply because it involves publication of all decrypted votes. (PAV 2006 cannot support ranked elections in its original form, and so the issue does not arise.)
- *Randomisation attack*: Both Prêt à Voter protocols suffer this attack. Adversaries can coerce voters to bring out their receipts with the choice marks always at the top. Although they do not know how these voters have cast their votes, they make these voters vote in a random manner.

B. Comparison of the two approaches

Now, we give a brief comparison of the above two approaches. Compared with PAV 2005, PAV 2006 has the following advantages:

- In PAV 2006, the shuffle phase and the decryption phase are separated, and the parties who execute the shuffle phase do not need to know the private key. Thus if some of them are absent in the mix, we can simply ignore them and replace them by some other parties. In contrast, the absence of any mix server in PAV 2005 will require expensive strategies to recover the private key share of the absent mix server.
- In PAV 2006, the size of the onion is constant. In contrast, it is proportional to the number of mix servers in PAV 2005.
- In PAV 2006, all ballot forms are generated by a number of clerks in a distributed fashion. If there exists at least one honest clerk, the privacy of ballot forms can be properly preserved. In contrast, all ballot forms in PAV 2005 are constructed by a single party. Thus this party can break the privacy of all ballot forms. Furthermore, she can apply subliminal & Kleptographic channel attacks to enable colluding parties to break voter privacy.
- In PAV 2005, if adversaries can smuggle a blank ballot form (as shown in Figure 5) out of the polling station, they can coerce a lot of voters using the chain voting attack. PAV 2005 also has chain of custody issues. But in PAV 2006, all ballot forms can be printed on-demand. Thus this scheme provides better assurance against chain voting attacks and chain of custody issues.

VI. DISCUSSION

We have discussed the challenges in designing a verifiable, secret ballot voting scheme and presented the design philosophy of Prêt à Voter. We have sought to illustrate the key steps in the evolution of Prêt à Voter and their motivation, without seeking to give an exhaustive history. The design has evolved significantly from the original version presented in [47]. This evolution has been driven in part by a desire to improve the design, in part in response to the identification of vulnerabilities. Some design decisions lead to clear-cut improvements, some give rise to rather subtle trade-offs. A prime example is the trade-off between pre-printed and on-demand ballots. The former allows for auditing of ballot forms prior to the election, and so to a generally easier process, but requires careful chain of custody mechanisms to ensure accuracy and privacy. The latter avoids the chain of custody problems but requires mechanisms, typically cut-and-choose style protocols, to detect malfunctions or malfeasance by the booth printing device.

The design of Prêt à Voter has been driven by the aim to make the voter experience as simple and familiar as possible while providing high levels of transparency and auditability. The Prêt à Voter approach has spawned a suite of trustworthy schemes while also providing voters with a voting experience almost identical with currently existing manual systems. It is also very flexible and adaptable, being capable of supporting a number of different tallying methods. As such, it would appear to be one of the strongest contenders for a deployable scheme.

A number of challenges remain. One technical point stems from the observation that with the original, RSA based, decryption mix version it is straightforward to deal with full permutations of the candidates on the ballot forms, at least if we set aside issues arising from Italian-style attacks. Whether it is possible to handle full permutations with the re-encryption mix versions without multiplying the number of onions remains an open question. We have shown how cyclic shifts can be handled with re-encryption mixes. While this appears to be satisfactory for simple, single choice of candidate elections, it is clearly not suitable for ranked or approval elections.

Even for a choice of a single candidate, cyclic shifts feel rather fragile in that, if an adversary does have a way to alter receipts in an undetectable way, he has a simple way to effect a predictable shift in the semantics of receipts. Suppose that the adversary knows that a majority of votes will be cast for candidate A and he wishes to swing the election in favour of candidate C; he simply applies a cyclic shift of 2 to the position of the \times on a suitable proportion of the receipts. Of course, the audit mechanisms should catch such manipulation, but in the spirit of deploying a strategy of layered defence, it is desirable to avoid this possibility.

It turns out that we can do better than just cyclic shifts with re-encryption mixes and prevent this bias attack, as shown in [53]. Here, the set of permutations allowable on the ballots is enriched using permutations derived from *Florentine squares*. The result is that for any receipt, any shift in the position of the \times will be as likely to change the vote to one

candidate as another. Consequently, by trying to shift the \times , the adversary can at best randomize a proportion of the votes and so shift the outcome toward a draw.

In this paper we have sought to provide informal arguments that the scheme provides the claimed properties, primarily of accuracy and ballot secrecy. Such arguments are typically based on known results about cryptographic algorithms and protocols (in particular, mixes). All this forms part of the broader issue of developing more formal and systematic ways to analyse voting schemes. This is still a rather new area of research and what work has been published is typically focused on certain formal properties of the cryptographic core of the schemes. To date there is no consensus as to the precise definitions of the properties that a voting system should provide. Furthermore, there is to date very little work addressing voting systems as socio-technical systems, taking account not just of the core algorithms and protocols but also of the roles of the various, procedures, and so on.

We have sought to demonstrate that the Prêt à Voter approach is a fruitful, flexible and promising one in the search to develop trustworthy, practical and, we hope, trusted voting schemes.

A. Future Directions

A number of issues call for further research. Some of these are quite general issues, facing the verifiable voting community in general, and some are specific to Prêt à Voter-based systems. First, more systematic and formal analysis techniques and tools need to be developed to deal with high-assurance voting systems.

More work needs to be done in exploring efficient and usable versions of Prêt à Voter to deal with a larger class of voting systems—STV, approval, Borda Count, Condorcet—in a way that avoids Italian-style attacks.

A number of vulnerabilities have been identified. For these, counter-measures have been proposed, but in many cases these either introduce extra complexity into the voting process or leave further vulnerabilities. A prime example of this is the requirement to ensure that it is not possible for the voter to retain any proof of the candidate order associated with their receipt. Various counter-measures have been proposed. Arguably the most pleasing is to ensure a supply of decoy left hand strips so that if the voter does attempt to smuggle the strip out of the booth this will not constitute a proof to a coercer.

The MarkPledge approach is interesting in this context in that it avoids the need to destroy information in the process of forming the receipt. Rather, the information is masked by additional faked information. It would be very satisfying to find a way of incorporating such an approach in Prêt à Voter, but it is far from clear how this can be done without excessive complexity.

To date, the Prêt à Voter program has concentrated on supervised voting, where voters cast their vote in the enforced privacy of a booth in a polling station. There is considerable interest in allowing remote voting via various channels such as the internet or telephone. While there are obvious attractions

in terms of convenience in such remote voting, there are serious concerns about how one counters threats of coercion and vote buying in such a context. To date, no satisfactory solution to such threats appears to exist. Some proposals have been made—notably that of Juels, Catano and Jakobson [31], and Clarkson *et al.* [13] and Traore *et al.* [4]—that appear technically sound, but from a usability and user perception point of view they are likely to be problematic.

Nonetheless, while most experts remain wary of proposing the use of remote voting for politically binding elections [17], it is clearly interesting to explore remote schemes and there are likely to be contexts in which the coercion threats can be disregarded (for example, the election of officials to professional organisations). Remote variants of Prêt à Voter are being investigated including the *Pretty Good Democracy* scheme [49], that combines ideas from *code voting* and Prêt à Voter.

Another issue that has been rather neglected by the verifiable voting community is that of how one actually implements the concept of the secure bulletin board. Most of the literature simply assumes that such a primitive exists or at least can be implemented.

Even if one succeeds in designing a scheme that is trustworthy, the challenge remains to persuade the stakeholders that they should trust and use the system. This is true of any new technology but is especially challenging for verifiable schemes such as Prêt à Voter. First, it is essential that the system achieves close to universal acceptance by the electorate as well as election officials and politicians. Secondly, such systems make heavy use of cryptography which, by its very nature, is rather mysterious and daunting. How to convey sufficient level of understanding to engender trust remains a major challenge.

B. Related Work

There has been an explosion of activity in recent years in the area of verifiable voting schemes. The Prêt à Voter approach was inspired by a study of Chaum’s visual cryptographic scheme [11] and the desire to develop a conceptually and technologically simpler scheme achieving the same goals. Around the same time as Chaum’s scheme, Neff proposed the VoteHere scheme [36], and later MarkPledge [37]. These are ingenious schemes that provide high degrees of assurance, especially of the correctness of the encoding of the vote in the receipts. The drawback is that the voter has to participate in quite a complicated challenge-response-style protocol with the booth device to generate the receipt.

Adida and Rivest proposed *Scratch and Vote* [2], based on Prêt à Voter but using homomorphic tabulation and proposing the use of scratch strips to allow off-line auditing of ballots.

Bingo Voting [8] is another recent and interesting approach. The idea here is that on the receipt there is a random-looking string against each candidate. For the chosen candidate the string is drawn from a pool of pre-assigned codes. The strings against the non-chosen candidates are chosen at random using a trusted random number generator. The approach is ingenious but does rely on trust in the random number generation, and it is difficult for voters to confirm that their vote is correctly encoded.

One response to the observation that the use of cryptography is likely to be an inhibiting factor in the uptake of verifiable schemes is to explore schemes that achieve similar levels of verifiability but without the use of cryptography. An early effort in this direction is Randell and Ryan’s scheme [45], which is based on Prêt à Voter but uses scratch strips to mimic the effect of cryptography. Another approach is Rivest’s *ThreeBallot* scheme [46], in which the vote is encoded across three ballots, only one of which is kept as the receipt. Yet another approach is the *Farnel* based schemes [3], which rest on the observation that verifiability does not require the voter to retain a copy of the her own receipt. Accordingly, the Farnel schemes propose mechanisms that allow voters to be given a copy of one or more previously generated receipts. Thus the anonymization occurs up front, rather than later in the mix/tabulation phase. In practice, implementing the shuffling of receipts before they are passed out to the voters is difficult without a significant level of trust in procedures and mechanical devices.

These non-cryptographic schemes are interesting in that they do not require an understanding of cryptographic mechanisms. Nonetheless, the assurance arguments are still quite subtle, more subtle than those associated with conventional voting systems. Vulnerabilities in all three of these schemes have been identified and they do not achieve the same levels of assurance of the more advanced cryptographic schemes.

REFERENCES

- [1] Ben Adida. *Advances in cryptographic voting systems*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, August 2006.
- [2] Ben Adida and Ronald L. Rivest. Scratch & Vote: self-contained paper-based cryptographic voting. *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, pages 29–40, 2006.
- [3] Roberto Araújo, Ricardo F. Custódio, and Jeroen van de Graaf. A verifiable voting protocol based on farnel. *Proceedings of IAVoSS Workshop on Trustworthy Elections (WOTE’2007)*, pages 57–64, 2007. Ottawa, Canada.
- [4] Roberto Araújo, Sebastien Foulle, and Jacques Traoré. A practical and secure coercion-resistant scheme for remote elections. *Proceedings of IAVoSS Workshop on Frontiers of Electronic Voting (FEV’07)*, 2007.
- [5] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC’01)*, pages 274–283, 2001. New York, NY, USA.
- [6] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC’94)*, pages 544–553, 1994. New York, NY, USA.
- [7] Josh Benaloh and Moti Yung. Distributing the power of a government to enhance the privacy of voters. *Proceedings of the 5th ACM Symposium on Principles of Distributed Computing (PODC’86)*, pages 52–62, 1986. New York, NY, USA.
- [8] Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich. Bingo Voting: secure and coercion-free voting using a trusted random number generator. *Proceedings of the 1st International Conference of E-Voting and Identity (VOTE-ID 2007)*, pages 111–124, 2007. LNCS 4896.
- [9] Fabrice Boudot, Berry Schoenmakers, and Jacques Traoré. A fair and efficient solution to the socialist millionaires’ problem. *Journal of Discrete Applied Mathematics*, 111(1-2):23–36, 2001.
- [10] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
- [11] David Chaum. Secret ballot receipts: true voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.
- [12] David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical voter-verifiable election scheme. *Proceedings of the 10th European Symposium on Research in Computer Science (ESORICS’05)*, pages 118–139, 2005. LNCS 3679.

- [13] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: toward a secure voting system. *2008 IEEE Symposium on Security and Privacy*, 2008.
- [14] Josh Cohen and Michael Fisher. A robust and verifiable cryptographically secure election scheme. *Proceedings of the 26th IEEE symposium on the Foundations of Computer Science (FOCS'85)*, pages 372–382, 1985.
- [15] Ronald Cramer, Matthew Franklin, Berry Schoenmakers, and Moti Yung. Multi-authority secret-ballot elections with linear work. *Advances in EUROCRYPT'96*, pages 72–82, 1996. LNCS 1070.
- [16] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. *Advances in EUROCRYPT'97*, pages 103–118, 1997. LNCS 1233.
- [17] Dagstuhl. Dagstuhl accord on electronic voting. <http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=07311,2007>.
- [18] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. *Advances in CRYPTO'89*, pages 307–315, 1989. LNCS 435.
- [19] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on IT*, 31(4):467–472, 1985.
- [20] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. *Advances in CRYPTO'86*, pages 186–199, 1986. LNCS 263.
- [21] Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. *Proceedings of Financial Cryptography (FC'00)*, 2000. LNCS 1962.
- [22] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. *Advances in Auscrypt'92*, pages 244–251, 1992. LNCS 718.
- [23] Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. *Advances in CRYPTO'01*, pages 368–387, 2001. LNCS 2139.
- [24] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Advances in EUROCRYPT'99*, pages 295–310, 1999. LNCS 1592.
- [25] Marcin Gomułkiewicz, Marek Klonowski, and Mirosław Kutylowski. Rapid mixing and security of Chaum's visual electronic voting. *Proceedings of the 8th European Symposium on Research in Computer Science (ESORICS'03)*, pages 132–145, 2003. LNCS 2008.
- [26] James Heather. Implementing STV securely in Prêt à Voter. *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF'07)*, pages 157–169, 2007. Venice, Italy.
- [27] James Heather and David Lundin. The append-only web bulletin board. *Proceedings of the 13th European Symposium on Research in Computer Security (FAST 2008)*, 2008.
- [28] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. *Advances in EUROCRYPT'00*, pages 539–556, 2000. LNCS 1807.
- [29] Markus Jakobsson and Ari Juels. Mix and match: secure function evaluation via ciphertexts. *Advances in ASIACRYPT'00*, pages 162–177, 2000. LNCS 1976.
- [30] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, 2002.
- [31] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society (WPES'05)*, pages 61–70, 2005.
- [32] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: a systems perspective. *Proceeding of the 14th USENIX Security Symposium*, pages 186–200, 2005. LNCS 3444.
- [33] David Lundin. Component based electronic voting systems. *Proceedings of IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, pages 11–16, 2007. Ottawa, Canada.
- [34] David Lundin and Peter Y. A. Ryan. Human readable paper verification of Prêt à Voter. *Proceedings of the 13th European Symposium on Research in Computer Science (ESORICS'08)*, pages 379–395, 2008. LNCS 5283.
- [35] Rebecca Mercuri. A better ballot box? *IEEE Spectrum*, 39:46–50, 2002.
- [36] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. *Proceedings of the 8th ACM Conference on Computer and Communications Security (CSS'01)*, pages 116–125, 2001.
- [37] C. Andrew Neff. Practical high certainty intent verification for encrypted votes. *VoteHere document*, 2004.
- [38] C. Andrew Neff. Verifiable mixing (shuffling) of ElGamal pairs. *VoteHere document*, 2004.
- [39] Lan Nguyen, Rei Safavi-Naini, and Kaoru Kurosawa. Verifiable shuffles: a formal model and a Paillier-based efficient construction with provable security. *Proceedings of the 2nd International Conference on Applied Cryptography and Network Security (ACNS'04)*, pages 61–75, 2004. LNCS 3089.
- [40] Tatsuaki Okamoto. An electronic voting scheme. *Proceedings of IFIP'96*, pages 21–30, 1996.
- [41] Pascal Paillier. Public-key cryptosystems based on discrete logarithms residues. *Advances in EUROCRYPT'99*, pages 223–238, 1999. LNCS 1592.
- [42] Torben P. Pedersen. A threshold cryptosystem without a trusted party. *Advances in EUROCRYPT'91*, pages 522–526, 1991. LNCS 547.
- [43] Kun Peng, Colin Boyd, and Ed Dawson. Simple and efficient shuffling with provable correctness and ZK privacy. *Advances in CRYPTO'05*, pages 188–204, 2005. LNCS 3621.
- [44] Birgit Pfitzmann. Breaking an efficient anonymous channel. *Advances in EUROCRYPT'94*, pages 339–348, 1994. LNCS 950.
- [45] Brian Randell and Peter Y. A. Ryan. Voting technologies and trust. *IEEE Security & Privacy*, 4(5):50–56, 2006.
- [46] Ronald L. Rivest. The ThreeBallot voting system. <http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf>, 2006.
- [47] Peter Y. A. Ryan. A variant of the Chaum voter-verifiable scheme. *Technical Report of University of Newcastle*, CS-TR:864, 2004.
- [48] Peter Y. A. Ryan. Prêt à Voter with Paillier encryption – extended journal version. *Journal of Mathematical Modelling of Voting Systems and Elections: Theory and Applications, special issue of Mathematics and Computer Modelling*, Ed Alexander S. Belenky, 2008.
- [49] Peter Y. A. Ryan. Pretty good democracy. *Unpublished menudraft*, 2009.
- [50] Peter Y. A. Ryan and Steve A. Schneider. Prêt à Voter with re-encryption mixes. *Proceedings of the 11th European Symposium on Research in Computer Science (ESORICS'06)*, pages 313–326, 2006. LNCS 4189.
- [51] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme. *Advances in EUROCRYPT'95*, pages 393–403, 1995. LNCS 921.
- [52] Victor Shoup. Practical threshold signature. *Advances in EUROCRYPT'00*, pages 207–220, 2000. LNCS 1807.
- [53] Vanessa Teague, Kim Ramchen, and Lee Naish. Coercion-resistant tallying for STV voting. *Proceedings of the 3rd USENIX/ACCURATE Electronic Voting Technology Workshop (EVT'08)*, 2008. San Jose, CA.
- [54] Zhe Xia, Steve A. Schneider, James Heather, Peter Y. A. Ryan, David Lundin, Roger Peel, and Phil Howard. Prêt à Voter: All-In-One. *Proceedings of IAVoSS Workshop on Trustworthy Elections (WOTE 2007)*, pages 47–56, 2007. Ottawa, Canada.
- [55] Zhe Xia, Steve A. Schneider, James Heather, and Jacques Traoré. Analysis, improvement and simplification of the Prêt à Voter with Paillier encryption. *Proceedings of the 3rd USENIX/ACCURATE Electronic Voting Technology Workshop (EVT'08)*, 2008. San Jose, CA.