Chapter 12

Prêt à Voter — The Evolution of the Species

Peter Y. A. Ryan
Department of Computing Science and Communications
University of Luxembourg, Luxembourg
peter.ryan@uni.lu

Steve Schneider
Department of Computer Science
University of Surrey, UK
s.schneider@surrey.ac.uk

Vanessa Teague
Department of Computing and Information Systems
University of Melbourne, Australia
vjteague@unimelb.edu.au

CONTENTS

12.1 Introduction 311
12.1.1 End-to-End Verifiability 311
12.2 Outline of Prêt à Voter 313
12.2.1 The Voting Ceremony 313
12.2.2 Vote Counting 314
12.2.3 Advantages of Prêt à Voter 316
12.3 Auditing the Election 317
12.3.1 Auditing the Ballot Generation Authority 317

12.3.2 Auditing Mixing and Decryption 318
12.3.2.1 Auditing the Mixes 318
12.3.2.2 Auditing the Decryption Tellers 318
12.4 Cryptographic Components 319
12.4.1 Decryption Mixes 319
12.4.2 Re-encryption Mix-nets 320
12.4.2.1 Re-encryption Mixes with Cyclic Shifts 320
12.4.2.2 Re-encryption Mixes with Affine Transformations 321
12.4.2.3 Re-encryption Mixes with Full Permutations 321
12.4.3 Distributed Generation of Ballots 322
12.4.4 The Bulletin Board 322
12.5 Facilitating Verification and Privacy 323
12.5.1 Encouraging Cast-as-Intended Verification (Ballot Auditing) 323
12.6 Enhancing Robustness Using Parallel Verification Mechanisms 324
12.6.1 Verified Encrypted Paper Audit Trails 325
12.6.2 Human Readable Paper Audit Trails 325
12.6.3 Confirmation Codes and Signatures 326
12.7 Accountability, Dispute Resolution and Resilience 327
12.7.1 Cast-as-Intended Verification 327
12.7.2 Authenticity of Receipts (Included-as-Cast Verification) 328
12.7.3 Tally Verification 329
12.8 Vulnerabilities and Countermeasures 329
12.8.1 Ballot Stuffing 329
12.8.2 Information Leakage 330
12.8.3 Retention of the Candidate List 330
12.8.4 Forced/Coerced Randomization 331
12.8.5 Chain Voting 331
12.8.6 Trash Attacks 332
12.8.7 Clash Attacks 332
12.8.8 Psychological Attacks 332
12.9 Prêt à Voter Goes Down-Under 332
12.9.1 Significance of the VEC Election 333
12.9.2 Challenges of Combining End-to-End Verifiability with Traditional Victorian Paper Voting 334
12.9.3 Specific Design Choices 335
12.9.3.1 Computer-Assisted Voting 335
12.9.3.2 Unified Scanner and EBM 335
12.9.4 Handling Complex Ballots and Printing Them on Demand .. 336
12.9.5 The Web Bulletin Board 336
12.9.6 vVote-Specific Vulnerabilities and Countermeasures 338
12.9.7 Practical Experiences 339
12.10 Conclusions 340
Acknowledgments 341

12.1 Introduction

Prêt à Voter provides a practical and highly usable approach to end-to-end verifiable elections with a simple, familiar voter-experience. It assures a high degree of transparency while preserving secrecy of the ballot. Assurance arises from the auditability of an evidence trail created by the execution of the election, rather than the need to place trust in the system components. The original idea has undergone numerous enhancements since its inception in 2004, driven by the identification of threats, the availability of improved cryptographic primitives and the desire to make the scheme as flexible as possible. This evolution culminated in the development and deployment of the system for use in the State of Victoria in Australia for the state election in November 2014. This chapter presents the key elements of the approach and describes the evolution of the design up to that deployment. We also describe the voter experience, and the security properties that the schemes provide.

The main advantage of Prêt à Voter compared to other end-to-end verifiable voting systems is that the auditing necessary for cast-as-intended verification can be performed before the voter has expressed any preference. Hence voters can be helped to verify without any impact on the privacy of the vote and without the possibility of any dispute arising as to what selection the voter made.

In the next section we give an outline of the notion of *end-to-end verifiability*. This is followed by a high-level outline of the Prêt à Voter approach to E2E V. Section 12.3 describes the logic and structure of verification audits. Section 12.4 gives an overview of the underlying cryptographic protocols, with explanations of how they developed, a description of options for different voting schemes and pointers to more detailed descriptions in prior publications. This concludes the theoretical description of Prêt à Voter. In Section 12.5 we describe pollsite procedures for encouraging voters to act according to those assumptions, including making ballot audits an automatic side-effect of ballot casting.

In Section 12.6 we describe some complementary methods of verification that can be used to improve the overall robustness of the process, because they depend on different assumptions from those of the cryptographic protocol. Section 12.7 introduces the notions of accountability and dispute resolution, i.e., the importance of being able to distinguish genuine verification failures from false claims of failure. Section 12.8 is a general discussion of potential attacks on the basic version of Prêt à Voter and possible countermeasures. The final section is an overview of the deployment of Prêt à Voter in a state election in Victoria, Australia, the first time an end-to-end verifiable voting system has run in a state election anywhere in the world (Section 12.9). Open problems and research directions are included in the conclusion.

12.1.1 End-to-End Verifiability

The purpose of end-to-end verifiability (E2E V) is to provide a high level of assurance to all parties that the announced outcome is correct in terms of the legitimately cast votes. This assurance should not depend on the correct behavior of the various components, but rather should stem from immutable evidence generated from the execution of the election. In particular, E2E V seeks to provide each voter with the means to confirm that her vote is correctly included in the tally. However, this has to be done with great care to ensure that it does not provide any way for the voter to demonstrate to a third party how she voted. In practice, this goal is achieved by providing voters, when they cast their vote, with a receipt that holds their vote in encrypted form. They can later use this receipt to confirm that their vote is correctly included in the tally. Once we have the votes collected together in encrypted form we can use standard cryptographic techniques to anonymize and decrypt the votes in a way that can be checked independently, providing universal verifiability of the processing of the votes.

The primary innovation of Prêt à Voter is the way that this encrypted vote is created: voters receive a pre-printed ballot form listing the candidates in a randomized order along with an encryption of this order. This serves as a reference against which they express their choices or preferences. The plaintext version of the candidate list is then destroyed leaving the receipt which carries the voter's selection in encrypted form. A number of important benefits flow from this approach that we detail in Section 12.2.3.

Prêt à Voter is designed to provide the three key ingredients required for E2E verifiability:

Cast-as-intended verification Each voter gets evidence that their vote is cast as they intended;

Recorded-as-cast verification Each voter gets evidence that their vote is included unaltered in the tally;

Universally verifiable tallying Everyone can check that the list of (encrypted) recorded votes produces the announced election outcome.

Eligibility verifiability can also be incorporated by simply attaching a verifiable identity to the public encrypted vote. This is discussed more below.

This chapter outlines the evolution of Prêt à Voter, drawing on the earlier literature on Prêt à Voter and incorporates lessons from the recent deployment in a binding government election in the Australian state of Victoria [184].

Candidates	Your vote
Obelix	
Panoramix	
Asterix	
Idefix	
	7*rJ*94*K*
Destroy	Retain

Figure 12.1: Prêt à Voter ballot form.

12.2 Outline of Prêt à Voter

Here we outline the main ingredients of the Prêt à Voter scheme [494, 151, 504]. The key innovation of the Prêt à Voter approach is the way votes are encoded in a randomized frame of reference, i.e., a randomized candidate list. An important observation about this way of encoding the vote is that, in contrast to previous schemes, there is no need for the voter to communicate her vote to an encryption device. What is encrypted is the information that defines the frame of reference for any given ballot form, and this can be computed in advance. We will return to the significance of this observation later when we discuss the threat model. Incidentally, this encoding has another advantage: the randomization of the candidate list results in fairness as a fixed ordering tends to favor candidates near the top of the list. However, it precludes preprinted cards or fixed instructions telling voters how to vote, because the instructions must vary according to the permutation. Depending on the electoral practice in force, this may or may not be an issue.

In the next section we present the basic voter experience when using Prêt à Voter, leaving aside at this stage the various auditing options available to the voter.

12.2.1 The Voting Ceremony

At the polling station, our voter Anne pre-registers and chooses at random a ballot form from a pile of forms individually sealed in envelopes. Example forms are shown in Figures 12.1 and 12.2. Note that the order of the candidates and the cryptographic values vary from form to form.

In the privacy of the booth, Anne removes the ballot from its envelope and 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, she marks her ranking against the candidates. Once the selection has been made, she detaches and discards the left-hand column that carries the candidate order. The remaining right-hand column now constitutes the (encrypted) receipt, as shown in Figure 12.3.

Candidates	Your vote
Asterix	
Idefix	
Panoramix	
Obelix	
	*N*5077*t*3
Destroy	Retain

Figure 12.2: Another Prêt à Voter ballot form.

Your Vote
X
*N*5077*t*3
Retain

Figure 12.3: Prêt à Voter ballot receipt encoding a vote for "Idefix."

Anne now exits the booth with this receipt, registers with an official and casts her receipt in the presence of the official: the ballot receipt is placed on an optical reader or similar device that records the cryptographic value at the bottom of the strip, that we will refer to henceforth as the ballot *cipher* and denote Θ, and an index value ι indicating the cell into which the X was marked.

In practice, the ballot *cipher* will actually be a unique serial number or pointer to a full ciphertext previously committed to the web bulletin board (WBB), an append-only, public broadcast channel with memory.

The digitized copies of the receipts are transmitted to a central tabulation server which posts them to the WBB. Only the tabulation server, and later the tabulation tellers, can write to this and, once written, anything posted will remain unchanged. Voters are encouraged to visit this WBB and confirm that their receipt appears correctly and, if their receipt does not appear, or appears incorrectly (i.e., with the X in the wrong position), they can appeal. Note that, as the voters hold physical, authenticated receipts, they have demonstrable grounds for complaint if their receipt fails to appear on the WBB.

It is also possible to have representatives of *helper organizations* [43] at hand at the polling stations. They could offer a service: helping audit ballots, checking digital signatures and possibly checking of posting of receipts to the WBB.

12.2.2 Vote Counting

The value printed on the bottom of the receipt, which we will refer to as the ballot *cipher*, is the key to extraction of the vote. Buried cryptographically in this value is the information needed to reconstruct the candidate order and so interpret the vote value encoded on the receipt. This information is encrypted under the secret keys shared using a threshold scheme among a number of tellers. Thus, only a threshold set of the tellers acting in concert are able to reconstruct the candidate order and so interpret the vote value encoded on the receipt. In practice, the value printed on the ballot form will be a pointer to the full ciphertext committed to the WBB during the setup phase.

The ciphertext associated with each ballot will have as its plaintext a value σ that encodes a permutation of the N candidates. Thus each ballot form may be thought of as a tuple:

$$(\pi, \Theta)$$

where π is the candidate order and the encrypted term $\Theta = \mathcal{E}(\sigma)$ is the ciphertext. The ballot is well-formed if and only if the plaintext candidate order printed on the ballot agrees with the order encoded in the σ value. A receipt has the form:

$$(\iota, \Theta)$$

where ι is an index value indicating where the voter placed her X or a vector recording her rankings, approvals, etc., depending on the voting method.

After a suitable period, once any disputes over recording and inclusion of receipts are suitably resolved, we can start the counting process. Here we will describe counting using *anonymizing mixes* to guarantee ballot secrecy, but other approaches are possible, for example, if Paillier encryption is used it would be feasible to use homomorphic tabulation [49].

We assume that the candidate list information has been encrypted in the ballot ciphertexts using a randomizing algorithm that supports re-encryption, e.g., ElGamal or Paillier. We also assume for the moment that the index value, indicating the position of the X for example, has been absorbed into the ciphertext to give a pure ElGamal or Paillier term. Details will be presented a little later. Tabulation proceeds in two phases: first a mixing phase to provide privacy (rather like shaking the ballot box) followed by a decryption phase (unsealing and unlocking the ballot box). The first phase will be performed by a set of *mix tellers*. The mix tellers do not need to know any secret keys in order to perform a re-encryption, only the public key under which the encryption was performed.

At the end of the mixing process, the batch of receipts will have undergone a number of re-encryptions and shuffles and are ready to be decrypted. Decryption will then be performed by a threshold set of decryption tellers who hold secret shares for the ballot decryption key.

Once all this is completed, the final, decrypted, anonymized votes appear in the final column of the WBB and these can be tallied in a conventional fashion and can be verified by anyone.

12.2.3 Advantages of Prêt à Voter

The main advantages of Prêt à Voter over earlier E2E schemes is that ballot auditing is independent of the vote and that it is readily adaptable to handling more complex voting methods such as STV.

Voters may challenge a preprinted ballot to check that its ciphertexts match the plaintext vote encoding (candidate order) shown on the ballot. When they have challenged enough ballots to feel confident that the ballots are all likely to be well-formed, they choose an unchallenged one to vote on. This means that ballot auditing is not privacy invasive and furthermore dispute resolution is clear cut: the ballot is either well-formed or it isn't and there is no question of having to rely on the voter's claim as to what vote they input. Wombat and STAR-Vote also print a plaintext representation of the vote beside the voter's ciphertext, which can be demonstrated not to match, however even if they do match the voter could still claim to have input something different.

Another advantage of the Prêt à Voter approach is that voters do not have to spend time inputting (dummy) selections before auditing. This is in contrast with schemes such as Helios and Benaloh's simple voter-verifiable elections, in which a voter has to go all the way through the process of inputting a vote selection before challenging. For simple ballots, it probably does not make much difference, but for complex voting schemes, the ease of verification before voting is a significant advantage.

Later systems, such as Punchscan and Scantegrity, that adopted the idea of pre-prepared ballot forms, inherit these features.

Another main difference between Prêt à Voter and many other schemes is that the secret information in the ballot is actually used for ballot printing, rather than for expressing the vote. This shifts the risk of privacy breach from a DRE (or equivalent, used for vote casting) to the processes that generate the ballot forms. Depending on the process details and the threat model, this may be an advantage or a disadvantage. Centralized printing might allow the machine to be better protected, but the paper itself then has to be carefully kept secret. Print-on-demand in a polling place means weaker assumptions on printout secrecy, but also many machines spread through polling places that must be protected. This is discussed further in Section 12.4.3.

Note also that clash attacks, in which the same receipt is provided against two votes that are the same, are harder in Prêt à Voter, because an attacker would have to guess how someone will vote before contriving a collision of receipts. In other E2E systems in which a device creates an encryption of the vote on demand, the device could simply record previous votes and when it encounters a repeat vote it simply copies the receipt it produced earlier. Such threats are typically countered by using

a trustworthy source of entropy for the encryption, but even here we have to ensure that the device actually employs the entropy provided.

Backup with a plaintext paper record is an emerging theme of end-to-end verifiable voting systems. Scantegrity II [133], STAR-Vote [77] and Wombat [80] all provide this feature as a natural artefact of the voting process. This allows auditing or manual counting of the paper trail to provide evidence of election integrity independent of the end-to-end verifiability. Prêt à Voter does not automatically produce such a centralized paper trail, and its only deployment in a real election was in a case in which avoiding paper records was specifically requested (see Section 12.9). However, in general it improves robustness to combine end-to-end verifiability with a more traditional paper trail. This is explored in Section 12.6.

12.3 Auditing the Election

So far we have described the process under the assumption that all the steps are executed correctly. However, we do not want the integrity of the election to rely on the entities involved behaving correctly and so we now introduce the mechanisms to detect any malfunction or corruption.

12.3.1 Auditing the Ballot Generation Authority

The first place that things could go wrong is in the creation of the ballot forms. If a ballot form is incorrectly constructed, in the sense that the candidate list shown on the form does not correspond to the order given by the σ value buried in the ballot ciphertext, then the voter's choice will not be accurately encoded. Note that throughout this chapter we are assuming that the voter makes her mark or marks in the right place relative to the printed candidate list. We therefore need a mechanism to detect incorrectly constructed forms, without revealing encryption keys.

If, as we have done above, we assume that the ballot forms are created in advance, we can perform a random audit on a proportion of the forms. So, we require the ballot creation authority (or authorities) to create an excess number of forms, perhaps four or five times as many as actually required, and allow independent organizations to make random selections of an appropriate proportion. For these selected forms, the randomization factors are revealed, so allowing the auditors to recompute the Θ and π and confirm that they agree with those printed on the forms. Alternatively, zero-knowledge proofs of correct decryption could be provided, which avoids having to reveal the randomization.

A rather elegant way of revealing the audit information for selected forms while ensuring that it is kept secret for ballot forms that are used to cast votes, is the "Scratch and Vote" mechanism of Adida and Rivest [43]. Here, the audit information is printed on the ballot forms but concealed by a scratch strip. Revealing the infor-

mation by removing the strip automatically invalidates the form for voting. In the '06 version of Prêt à Voter [504], this information was revealed by having the decryption tellers online to reveal the audit information. The Adida/Rivest approach avoids the need to have the tellers online and provides a procedural mechanism to enforce the mutual exclusion of casting and auditing.

The process of auditing a ballot form is accomplished by recomputing the values on the form from the cryptographic values. Thus the ciphertext is recomputed from the representation of the candidate order σ, the randomization and the teller public keys.

$$\Theta = \{\sigma\}_{PK_T}$$

If these agree with the values printed on the form, we may conclude that the form was correctly formed.

Voters should always have the opportunity to audit on demand. In addition, audits may be performed by appropriate authorities before, during and after the election.

12.3.2 Auditing Mixing and Decryption

Next, we need to confirm that the mix tellers perform all their actions correctly. In the case of decryption mixes the mixing and (partial) decryptions occur in parallel. We need to show that each teller correctly decrypts its layer of encryption for each ballot. The point is to ensure that the set of votes in the input is the same as the set of votes output.

The technique for auditing the mix tellers in the early versions of Prêt à Voter that used decryption mixes was based on *randomized partial checking* [321]. Half the links are randomly chosen to be revealed and verified. The choice of links, while essentially random, is carefully constrained in such a way as to ensure that no decrypted vote can be traced back to the original ballot receipt. We do not go into the details here as the approach has been superseded by the zero-knowledge proofs of shuffles for re-encryption mixes described below.

When we use re-encryption mixes, the mixing and decryption phases are separated out and we deal with these in the next two sections.

12.3.2.1 Auditing the Mixes

Numerous techniques have been proposed for proving correct mixing of encrypted values quite efficiently, for example the approach proposed by Neff [406]. For a comprehensive survey see [43] or [189].

12.3.2.2 Auditing the Decryption Tellers

Finally, once the ballots have been mixed, we also need to confirm that they are all correctly decrypted. Here we can be more direct and can separately audit every decryption as we do not need to worry about anonymity at this stage. Given that we are using randomizing encryptions here, the process of checking the correctness of the decryptions is not quite trivial: we cannot simply perform the encryption of the claimed plaintext and check the result agrees with the ciphertext, as would be possible for a deterministic algorithm. And of course we don't want to reveal the secret keys. There are efficient ways to prove correct distributed decryption in zero knowledge, for example using [178].

12.4 Cryptographic Components

Most of the published versions of Prêt à Voter use anonymizing mixes to eliminate any link between the receipts and their eventual decryption. In theory, homomorphic tabulation could also be used, given suitable encodings of the votes and encryption algorithms. In fact, the vVote implementation, described in detail in Section 12.9, uses a hybrid of techniques to compress and uncompress votes (a vector of preferences) in order to make their pass through the mix more efficient.

In this section we briefly describe the evolution of the mixed-based approaches.

12.4.1 Decryption Mixes

The original version of Prêt à Voter [151], employed Chaumian decryption mixes and RSA encryption. The ciphertexts defining the candidate permutations were constructed as layers of RSA encryption. The ith layer is encrypted under the public key of the ith mix server. Suppose that we have m layers. To mix a batch of such terms, the batch is passed to the mth server who strips off the outer layer of encryption, shuffles the resulting terms and outputs them to the $(m-1)$th server. The $(m-1)$th server then strips off the next layer, shuffles them and passes them on to the $(m-2)$th server and so on.

Decryption mixes had one very pleasing feature when used in Prêt à Voter: permutations can be encoded in the randomization at each layer of the encryption. The final permutation of the candidates printed on the ballot can thus be constructed as a product of the permutations defined at each layer, moving outwards from the inner layer. Consequently, during the tabulation mixes, the representation of the vote can be transformed as the ballots passed through the mixes by the inverse permutations, moving in reverse order from the outer layer inwards. Suppose that the vote is represented as a vector. For a simple choice of one candidate the vote will have a 1 on the position corresponding to the voter choice in the basis given by the original candidate

ordering on the ballot, with 0s elsewhere. As the ballot moves through the tabulation mix, the vector representation is transformed according to the inverse of the permutations that was used to form the final permutation shown on the ballot form. In effect the transformations of the vectors that occur during the tabulations mixes undo the permutations applied in the construction of the ballot.

The result of this is that it is possible to arrange for all the ballots to emerge from the tabulation mixes with the voter's selection presented in the standard candidate ordering. Consequently, all information about the original ordering on the ballot is washed out, avoiding any danger of an attacker being able to partition the mix according to such information. Full details of these constructions can be found in [151].

However, decryption mixes have several downsides. Mix servers need decryption keys and hence have to be pre-determined. If auditing reveals that one of them has cheated there is no privacy-preserving way to redo the mixing and audit. A further drawback is that the size of the ciphertexts grows with the number of mixes used.

In re-encryption mixes, the servers do not need any secret keys, just knowledge of the public key. Consequently, any failing server is easily swapped out. Note also that re-encryption mixes can be independently repeated with different re-encryption factors, or even run in parallel. A number of techniques to prove in zero-knowledge the correctness of a shuffle are available for re-encryption mixes, whereas decryption mixes require partial random checking (RPC) techniques, [321, 342] that necessarily leak some information about the shuffles.

Such considerations prompted the investigation of the use of re-encryption mixes in place of decryption mixes. This has many advantages but also entails some drawbacks when applied to Prêt à Voter, as described below.

12.4.2 Re-encryption Mix-nets

12.4.2.1 Re-encryption Mixes with Cyclic Shifts

Re-encryption mixes for Prêt à Voter were first investigated in [504]. Here, the ciphertexts on the ballots defining the candidate order comprise a single layer encryption with a randomizing algorithm such as ElGamal. Now mixing involves each mix server taking the input batch of ciphertexts, re-encrypting each term, i.e., re-randomizing and outputting the resulting set in secret, shuffled order to the next server.

As mentioned earlier, there are a number of clear advantages in moving to re-encryption mixes, but there is also one major downside, at least when done in the most obvious way. With decryption mixes we have a natural way to transform the vector of voter choices as the ballot moves through the mixes in such a way as to ensure that the final vector aligns with the canonical order of the candidates. With re-encryption mixes there is no obvious analogue of this. We could simply send the

receipts, $(\iota.\Theta)$ through the mix leaving the ι vector unchanged and finally decrypt the Θ to reveal the original ballot permutation. This would indeed allow correct tabulation, but it reveals the original candidate permutation and allows an attacker to partition the mix according to the various ι values.

A possible fix, which works at least for simple voting methods in which the voter just selects a single candidate, is to restrict the candidate permutations to cyclic shifts of the canonical order. We can then exploit the homomorphic nature of the encryption algorithm to absorb the index value into the Θ term. Suitably constructed, this now becomes an encryption of the index value translated to the canonical order. Thus we can now treat the Θ' terms as regular ElGamal ciphertexts and send them through the re-encryption mixes in the usual way. We omit the details here; full details can be found in [503].

This eliminates the partitioning problem, but still has a couple of problems: it will only work for simple "*X* marks the candidate" style voting, and it is arguably rather fragile. Suppose that an attacker is able somehow to undetectably shift the position of the *X*s on some ballots. Suppose further that he knows that the majority of these votes will be for some candidate *B* while he favors candidate *F* five down in the list. Then he simply shifts a suitable number of the *X*s to $X+5 \pmod N$, where N is the number of candidates.

Of course, the auditing mechanisms should make this impossible, but nonetheless this possibility is a bit troubling, hence the next enhancement described in the next section.

12.4.2.2 Re-encryption Mixes with Affine Transformations

In order to counter the fragility issue described above, and to slightly broaden the scope of voting methods, the use of affine transformations of the candidate list instead of simple cyclic shifts is proposed in [505]. Now we have two ciphertexts, one concealing a shift factor and the other a scaling factor. The approach works best with a prime number of candidates. The result is less fragile: an attacker does not know how to manipulate the position of the *X* to produce a predictable switch of candidates.

The approach also accommodates voting where up to two candidates can be selected, but of course this is a rather modest improvement in flexibility. The real challenge is to handle full permutations, which we address in the next section.

12.4.2.3 Re-encryption Mixes with Full Permutations

For preferential voting the vote is a full vector of rankings. The cipher schemes used in re-encryption mixes (ElGamal and Paillier) do not provide a homomorphic way of composing permutations, and so it is not possible to combine the list of preferences with the encrypted candidate list to obtain the vote in encrypted form as a single ciphertext. It is an open problem as to whether this can be done in some other way.

Therefore the way we handle full permutations coupled with re-encryption mixes is simply to provide N ciphertexts, one for each candidate, in the permuted order. A vote is the rank ordered list of these ciphertexts, which can then be processed by the mix-net in the usual way, simply handling tuples (lists) rather than single ciphers. This approach first appears in [581] and is used in the vVote system that we describe in detail later.

12.4.3 Distributed Generation of Ballots

As already mentioned, one key feature of Prêt à Voter is the absence of any single machine that knows both the ciphertext and the corresponding plaintext vote. However, on closer examination this is not quite so clear cut: anyone who observes the printed ballot form with its attached ciphertexts and candidate list can easily infer the contents of the vote when it appears on the bulletin board. This also applies to a single machine that generates the preprinted ballot form. Such a machine could also use deliberately badly-chosen randomness to leak this information to a third party without explicitly communicating it (this is sometimes called a "subliminal attack" [530][1]). This has motivated many schemes for distributing the generation of ballot forms [477], for printing them on demand so that only the intended voter sees them [182], or altering the form that will appear on the bulletin board so that it is no longer recognizable by the entity that originally created it.

In summary, the overall aims of such constructions are:

- To make sure the randomness is good,
- To make sure no single (electronic) entity knows the ciphertext-plaintext link.

Overall, however, this remains an incompletely solved problem. The ballot forms must eventually be printed, and although there are some preliminary investigations into distributing the printing process itself [217], these do not yet seem practical.

12.4.4 The Bulletin Board

Prêt à Voter shares with most end-to-end verifiable voting schemes the assumption of a bulletin board, which is an authenticated broadcast channel with append-only memory. Although in theory a fairly simple object, the bulletin board turns out to be difficult to implement in practice. One important insight of the vVote project in Victoria was the advantage of splitting the two functions of the bulletin board. One part is a robust and secure database with redundancy, tolerance of failures, and a method of acknowledging receipts. This is the real-time repository of election data. The second part, which corresponds to the theoretical "bulletin board," is a static transcript of the day's transactions, including vote generation, auditing, casting and,

[1] In earlier versions of this work we referred to this as a kleptographic attack, which is not quite the right term.

on the final day, mixing and decryption. It contains all of the information that the system commits to before, during and after the election.

12.5 Facilitating Verification and Privacy

E2E V ensures that errors or corruption are detectable, but this is not enough: it is essential that audit steps are actually performed to ensure that problems are actually detected. Thus, electoral integrity depends on people actually performing the verification tasks, so it's important to make it easy to do so. In the case of voters, it is important that they understand the purpose and importance of the checks available to them and are thus motivated. Some procedures are optional for the voters, but some really need to be enforced, for example, receipt freeness follows from the *compulsory* shredding of the candidate list in the polling place.

Auditing the proper construction of the ballot is arguably the most important and least intuitive of Prêt à Voter audits. We describe below some possibilities for making it easier.

12.5.1 Encouraging Cast-as-Intended Verification (Ballot Auditing)

The separation of ballot auditing from actual voting in Prêt à Voter permits many methods of encouraging cast-as-intended verification without any appearance of authorities influencing people's votes. Anyone can help anyone else to audit a preprinted ballot. Electoral authorities or other parties could offer explicit encouragement or incentives to audit, or appoint independent ballot auditors who would challenge ballots in public in addition to voter-initiated auditing.

It is important to make auditing as effortless and automatic as possible. There are various ways this could be achieved. One proposal is the double-sided forms of [497], which has the advantage of automatically assigning two ballots to each voter and enforcing the mutual exclusion of voting and auditing of any ballot. Here, each side of a form carries an independent Prêt à Voter ballot form. The voter arbitrarily selects one side to vote and the other for audit. The forms actually have a third, blank column opposite the candidate list on the other side, as shown in Figures 12.4 and 12.5. These images of the two sides should be thought of as being related by a rotation about the vertical axis. Thus, detaching the candidate list on the voted side detaches the blank column of the flip side, so leaving an intact Prêt à Voter ballot for audit. The two sides of the resulting receipt where the voter has cast a vote for Idefix on the second side is shown in Figures 12.6 and 12.7.

Now the side selected for casting is scanned and posted to the WBB as usual, but also the flip side, with a complete Prêt à Voter ballot is scanned and posted for

Obelix		
Asterix		
Idefix		
Panoramix		
	$3Wa3Kc$	

Figure 12.4: Dual Prêt à Voter ballot form; side 1.

Asterix		
Idefix		
Obelix		
Panoramix		
	$Yu78gf$	

Figure 12.5: Dual Prêt à Voter ballot form; side 2.

public auditing. Note that this mechanism enforces the mutual separation of casting and auditing.

12.6 Enhancing Robustness Using Parallel Verification Mechanisms

This section describes several methods of providing evidence of a correct election outcome that are different from the main logic of end-to-end verification. The idea is to provide some redundant means of auditing or verifying the election outcome, relying on different assumptions that may remain true even if the assumptions of the end-to-end verification protocol are not. Examples include confirmation codes (Section 12.6.3), or retaining at the polling place either a plaintext human-readable vote (Section 12.6.2) or a copy of the encrypted receipt (Section 12.6.1).

Obelix	
Asterix	
Idefix	
Panoramix	
	$3Wa3Kc$

Figure 12.6: Dual Prêt à Voter ballot receipt; auditable side.

X	
*Yu*78*gf*	

Figure 12.7: Dual Prêt à Voter receipt; vote carrying side.

Combining such methods requires careful thought because the assumptions are different, and may result in an inconsistent state. For example, the plain human-readable paper record may be corrupted even when the end-to-end verifiable voting data are correct. Resolving discrepancies between different evidence trails could be problematic.

12.6.1 Verified Encrypted Paper Audit Trails

A VEPAT (verified encrypted paper audit trail) mechanism was proposed in [495] as a way to counter the possible lack of diligence of voters in checking receipts against the WBB. The idea is to keep one or more complete copies of the cast votes locally at the polling site. These copies could then be made available to independent observers to perform checks against the WBB. Of course, great care has to be taken to ensure the integrity of such paper audit trails, as corruption of the trail could end up undermining the credibility of a valid electronic count. A possibility is to record the ballots on a till roll which is very hard to tamper with. Usually for VVPAT, i.e., with plaintext ballots, this is not possible because of the privacy concerns arising from preserving a record of the order in which votes are cast.

Note that this also serves to counter the so-called *trash* attacks in which voters throw away their receipts which normally would indicate to an attacker that these receipts will not be checked, and hence allow an insider attacker to manipulate or suppress publication of such votes.

12.6.2 Human Readable Paper Audit Trails

The idea of introducing a human readable paper audit trail (HRPAT) was introduced in [496] and later elaborated in [374]. This was prompted by the observation that many people seem uncomfortable with relying on cryptography and so having a more conventional plaintext record of cast votes as a fall-back could assuage such concerns and foster a higher level of trust in the system. Again there may be dangers in this in that manipulation of the audit trail could end up casting doubt on a valid electronic count.

We do not go into the details here, as they can be found in [374], but we just give an indication of the mechanism. Now the ballots have two layers: the lower layer is a regular Prêt à Voter ballot form, but without the usual cipher term. The upper part comprises another copy of just the receipt column, without the candidate list but with the usual cipher term, overlaid over the lower layer. The voter marks her selection on the upper layer in the usual fashion and a carbon paper or similar mechanism transfers the marks to the lower layer. The voter then detaches the upper layer that now forms the receipt in the usual way. The lower part still carries the candidate list and so the vote is represented in the clear. The lower layer is folded and dropped into a ballot box observed by the voting officials. This creates a plaintext paper audit trail and, inter alia, provides a mechanism to help ensure the voter relinquishes the candidate order.

Whether such a mechanism actually increases the assurance in a technical sense is debatable, if there is any possibility of manipulation of the paper audit trail this may actually weaken the overall assurance by undermining the credibility of the electronic count. Another concern is that voters might deliberately manipulate the ballot marking in such a way as to produce differently marked upper and lower ballots, leading to a discrepancy later. Possible countermeasures are proposed in [374] but as always, they are not infallible.

This relies on different assumptions, in particular that the voters verify their plaintext paper vote, and that the paper records are properly audited (or manually counted) afterwards.

12.6.3 Confirmation Codes and Signatures

There are various methods for confirming that a (threshold) number of authorities have recorded each ballot. Although this doesn't contribute to end-to-end verifiability (because all the trustees may be corrupt, and may subsequently drop or modify the ballot), it can give voters an immediate and easy way to check, in the polling place, that their vote has been received and recorded.

Several standard methods of acknowledging receipt by a threshold set of trustees are applicable to Prêt à Voter, including

- acknowledgment codes that need a threshold to decrypt/derive and return them to the voter,
- (threshold) digital signatures.

The first approach was proposed in [498] and introduced ideas and construction from Pretty Good Democracy, [506], into Prêt à Voter. A threshold set of trustees cooperate to register the vote and reveal a secret shared confirmation code.

The second was proposed for vVote in [181] and instead of revealing a confirmation code it provides a threshold signature. In both cases, the voter needs a way to

check, in the polling place, that the returned code or signature is valid. Procedures need to be established to handle situations in which codes do not match or signatures prove not valid.

12.7 Accountability, Dispute Resolution and Resilience

The possibility for voters to verify parts of an election process introduces new attack vectors: to falsely claim that an attempted verification has failed in order to cast doubt on the accuracy of the election outcome. Of course, traditional voting is vulnerable to the same problem, although they typically provide little or no opportunity for voters to uncover errors or fraud. There are numerous examples of disappointed candidates who contest results alleging electoral fraud. Distinguishing truthful from false claims is essential for any election system.

In general, every possible defaming attack is matched to some genuine attack on electoral integrity—the point is that the claim of a failed verification is plausible because it corresponds to some possible attack that could alter the votes. The aim of design for accountability and dispute resolution is to distinguish between a true detection of the attack, and a false claim of detection.

Prêt à Voter provides evidence of most kinds of attack. However, depending on the exact polling place procedures and mechanisms of authenticating printouts and receipts, there are some ways of fabricating printouts or receipts that make it look as if they were received from the legitimate authority but are invalid.[2] Some ideas for attempting to defend against such attacks are described in Section 12.7.2.

12.7.1 Cast-as-Intended Verification

As discussed already, in Prêt à Voter cast-as-intended verification is equivalent to establishing the well-formedness of the ballots. This is achieved by challenging preprinted ballots, which means that this verification step automatically produces evidence when there is a failure. This is in contrast to most other E2E V systems that depend on the voter's declaration of how she asked the machine to vote. In Prêt à Voter, in order to fabricate a failed verification, an attacker would have to generate a ballot form that looked genuine but encoded inconsistent values and smuggle it into the system. Countermeasures to this of course exist: the usual anti-counterfeiting mechanisms such as special paper stock, digital signatures, chain of custody, etc. In the vVote setting for Victoria's state election (Section 12.9), printed ballots were dig-

[2]For example, a voter could make a good receipt/vote but not cast it, then complain later that it doesn't appear on the bulletin board. Depending on exact procedures, and how much work is put into authenticating receipts and ensuring that only authenticated receipts are left with voters, this sort of attack might succeed.

itally signed to prevent this attack. The techniques described below for guaranteeing the authenticity of receipts could be applied to preprinted ballots too.

12.7.2 Authenticity of Receipts (Included-as-Cast Verification)

When a voter complains that her receipt has been omitted from the bulletin board, there should be some process for establishing whether the problem lies in a misbehaving electoral system or a misbehaving accuser. A verified encrypted paper audit trail (Section 12.6.1) helps, but assumes that the paper audit trail is hard to manipulate. Such assumptions are routinely made for conventional voting systems, and indeed for a VEPAT mechanism with a till roll style the assumption is probably more justifiable. Nonetheless, it is important that the voter's receipt itself needs to include evidence that it is genuine, which can be tested by a third party.

There are several methods of providing evidence of the validity of a receipt, listed here in increasing order of difficulty of forgery. In each case, the crucial questions are

- how difficult it is for an accuser to forge a valid-looking receipt, and
- how difficult it is for a misbehaving poll-site to trick a voter into leaving with an invalid receipt.

Possible technologies include:

Anti-counterfeit paper: Receipts could be printed on paper that is distinctive and otherwise unavailable. Producing a fake one would be as hard as stealing some of the paper. Voters need to be able to recognize the paper in order to identify false receipts from a corrupted polling place.

Franked or stamped receipts: Receipts could be stamped, die-cut or signed (by hand) at the polling place. The security properties would be similar to traditional commercial receipts, and voters would need to know what sort of mark, stamp or die-cut to expect.

Digitally signed receipts: Voters receive a digital signature on all the data on their receipt. The voter must be able to check the signature in the polling station, before she leaves. Since obtaining the receipt happens in controlled conditions and the voter is under supervision through to doing the signature check, a claim of an incorrect signature at that point demonstrates that there has been an attack.

This is based on the assumption that a voter behaves honestly while in the polling station. (For example, we have to guard against smuggling in and switching in a fake receipt.) Once a voter leaves the polling station then she cannot later come back in with an incorrect signature.

This solution was used in the vVote project—see Section 12.9.

Confirmation codes This approach was originally proposed by Chaum for Scantegrity II. The idea is that when the voter casts her vote she gets a code specific to that ballot and vote. Such codes are fairly sparse and hence hard to guess. If we can ensure that the voter only sees the code for her vote, then knowledge of this code can provide evidence in the event of a challenge. Suppose that the voter checks the code posted to the WBB against her ballot and finds that it does not agree with the code revealed to her. When she challenges there is a procedure to verifiably open all the codes for that ballot. If it turns out that the code claimed by the voter is indeed a valid code and different to the one posted then this provides good support from the voter's challenge. If however, the code she claims does not correspond to any valid code for this ballot then this is a strong indication that the challenge is false. Of course, great care has to be taken to ensure the integrity and secrecy of these codes. Furthermore, the voter needs a way to confirm that she has been given the correct code at the time of casting.

12.7.3 Tally Verification

Disputes about the proper mixing and decryption of receipts on the bulletin board are the easiest to resolve, because multiple verifiers can be applied and examined until they agree. Thus, if a problem is detected with say the posted output of one of the mix servers, then it is clear which entity is responsible. Furthermore, for many such errors in the tabulation process we can recover quite easily, for example by swapping out a defective/corrupt mix server and re-running the mix.

12.8 Vulnerabilities and Countermeasures

This section describes known vulnerabilities in the basic version of Prêt à Voter. Some depend on the exact version of Prêt à Voter being used. Most have workarounds at the cost of some extra complexity in the procedures or protocol.

12.8.1 Ballot Stuffing

Prêt à Voter, like most E2E systems, does not by itself address ballot stuffing. The fact that the link from a voter's receipt to the voter's identity does not need to be kept secret permits some simple means of guaranteeing that only eligible voters can vote. One possibility is simply to list the voter's name against their receipt on the bulletin board. This allows for public eligibility verifiability, because everyone can check the entire list of included voters. It also protects privacy, though not everlastingly—in time the cryptographic algorithms may be broken and votes may be revealed [397].

To counter concerns about the long-term secrecy of votes, we could simply list the names of voters who cast votes in random order with respect to the list of receipts.

Another possibility, since Prêt à Voter is an attendance voting scheme, is to use existing procedures for roll markoff (however secure or insecure they are) and ensure that the number of votes legitimately cast at each location is correctly recorded and reconciled with the numbers on the bulletin board. This means that prevention of ballot stuffing depends on the security of polling-place procedures.

12.8.2 Information Leakage

The device that prints the ballot forms learns their candidate ordering. Depending on the exact version of Prêt à Voter being used, it can probably also recognize the corresponding receipt afterwards. If it leaks this information it violates vote privacy.

Although an honestly constructed receipt does not leak information about the vote except given the decryption key, a dishonestly constructed ballot form might produce a receipt that does leak the vote, in two conceptually different ways:

print channels: This involves embedding information in subtle features of the printed ballot form, and hence into the receipt. Possibilities include slightly modified layout, fonts or colors. Depending on the method of tabulation, such information may or may not be carried onto the bulletin board, but could certainly be used by someone with direct contact with the voter's printed receipt.

subliminal channels: This involves embedding information into the randomness used to generate the encrypted ballot form itself. (Public key encryption generally requires additional randomness to produce each ciphertext.) The randomness could be chosen in a way that was deliberately weak, or deliberately known in advance to a colluding attacker. This information would be immediately replicated on the bulletin board.

Methods for secure distributed generation of randomness exist and can be employed here, but they solve only the subliminal attacks, not the plain information leakage or the leakage via print channels. There is some work on distributed printing of secret information [217], but at the moment the production of ballot forms still has important trust assumptions for privacy.

12.8.3 Retention of the Candidate List

A voter who retains their randomized candidate list can use it to prove how they voted. It is challenging to devise a polling place procedure that forces the destruction of the list without revealing the list to observers. The HRPAT mechanism mentioned earlier could provide such a mechanism. An alternative, rather pleasing approach is to have a supply of alternative candidate strips in the voting booths. Now a coercer

does not know if the voter retained the real strip or just picked up an alternative. Of course care needs to be taken to avoid the coercer being able to distinguish real from dummy by, for example, aligning perforations.

12.8.4 Forced/Coerced Randomization

Although a voter cannot produce evidence of having voted a particular way, and hence cannot be coerced to do so, a coercer can still insist on a receipt of a particular form, for example, one with a checkmark in the first location, or one in which preferences are in ascending order. This makes the voter cast a random vote. The effect is ameliorated if the voter can select from a variety of ballots, including for audit, without observation by the coercer. She may then be able to choose one that simultaneously keeps the coercer happy and expresses the vote she wants. This may not necessarily work for complex ballots such as preferential ones, however, because it may take a very large number of ballot forms to make the coercer's demand consistent with the voter's intention.

12.8.5 Chain Voting

In the chain voting attack, as it applies to conventional voting, a coercer smuggles a ballot out of a polling place, fills it out and gives it to a voter approaching the polling station with instructions to cast it and bring an unmarked ballot back out. The coercer uses the new ballot to repeat the attack with a new voter.

In Prêt à Voter, the chain voting attack is to smuggle out a printed ballot form, record the candidate order, then send a voter back into the polling place with instructions to vote in a particular way and return with both a receipt derived from that form and a new, unmarked, ballot form. Since the coercer has already recorded the voter's candidate order, the receipt reveals the vote. The new ballot form is used to repeat the attack with a new voter.

As in conventional paper-based voting, [296], a simple counter is to give the receipt a unique, detachable serial number and record this at the time the ballot is passed to the voter. When the voter returns to cast her vote, the official checks that it matches the one recorded and then detaches it from the ballot before it is cast. This ensures that the voter casts the same ballot she was provided when she registered. Another possibility is setting a limit for the time elapsed between receiving a ballot form and casting the vote—this doesn't completely solve the problem, but it reduces the window of opportunity for the attack.

A further idea [501], is to have a scratch strip over the ballot serial number, possibly overprinted with an ephemeral serial number. This is required to be intact until the point at which the receipt is scanned. Election officials check the ephemeral serial number is intact and corresponds to the number handed out to the voter. It is then removed to reveal the true serial number before scanning. This ensures that

everyone votes on the form they received at registration time, without giving officials a way to link the final receipt with its candidate list.

12.8.6 Trash Attacks

The "trash attack" [88] is an attack on counted-as-cast verifiability. If the attacker controls the bulletin board or uploads his own data to the bulletin board, and happens to know that a particular voter will not check the inclusion of their receipt, then the encrypted vote can be safely substituted. The name "trash attack" refers to such an attacker noticing a certain receipt in the trash, and hence inferring that the voter won't be able to verify it later. There are simple mitigations for the classic form of this attack, for example a VEPAT (Section 12.6.1), or a photocopier in the polling place allowing voters to distribute their receipt to others.

In general without an independent copy, the more general class of attack, in which a cheating authority somehow learns that a certain voter will not verify the inclusion of their receipt, remains an important problem.

12.8.7 Clash Attacks

The "clash attack" [357] is a vote dropping technique that applies to many cryptographic voting schemes. An attacker (as a server or ballot generator) arranges to give several different voters identical receipts. All affected voters see their receipt appear on the public WBB, and yet only one vote has been counted.

The attack works only if the voters subsequently cast identical votes. The attack is in general harder for Prêt à Voter than for direct-encrypting schemes such as Helios and Wombat, because the attacker must commit to the identical ballot before learning the person's vote.

Overall this attack is no more effective, requires more conspirators and has a higher probability of detection than the simple misalignment of the candidate names on the ballot by a corrupt printer.

12.8.8 Psychological Attacks

Particularly for privacy, there is always the possibility for a coercer to claim that they can infer a person's vote when they actually can't. Successful coercion depends on the voter believing this claim, not on the claim being true. Since Prêt à Voter's public-key-based privacy mechanisms are probably too complicated for many voters to understand, and since there are indeed genuine opportunities for some information leakage by particular components, it is hard to defend against this attack. This problem is not specific to Prêt à Voter.

12.9 Prêt à Voter Goes Down-Under

This section describes a design based on Prêt à Voter for the Australian state of Victoria. More details are given in [184], of which this section is a summary. The system ran successfully in the state election in Victoria (Australia) in November 2014, taking a total of 1121 votes from supervised polling places inside Victoria and at the Australian High Commission in London.

The protocol itself is end-to-end verifiable, meaning that there are no human or electronic components which must be trusted for guaranteeing the integrity of the votes (although vision impaired voters must assume that at least one device reads accurately to them). There are probabilistic assumptions about the number of voters who audit Prêt à Voter ballots, the number of voters who check that their receipt printout matches their intended vote, and the number who check that their receipt appears on the Web Bulletin Board (WBB). It also provides voters with evidence of malfeasance, assuming that they check the signature on their receipt before they leave the polling station. Since this is a polling-station scheme, we do not address eligibility verifiability. Prevention of ballot stuffing is by existing procedural mechanisms.

12.9.1 Significance of the VEC Election

End-to-end verifiable election protocols are well studied in the academic literature, but (with the notable exception of the Scantegrity II project in Takoma Park, MD [133]) had not previously been deployed in binding government elections. This project demanded new protocols for addressing issues that arise in practice but had not been adequately considered in the literature, and provided new insights into the important difference between practical requirements and academic security goals. Our main contributions are:

1. A version of Prêt à Voter usable enough for real people, even for the very complex ballots used in Victoria, with some practical evidence about its use in a real election.

2. Scalable cryptographic protocols that are fast enough for long preferential ballots. Details are in [182].

3. A clear account of what is achieved by running an end-to-end verifiable system as part of an electoral process that also includes a traditional paper-based system for other votes. The paper elements mean that the whole electoral process is not end-to-end verifiable, but end-to-end verifiability of the subset improves the weakest links in the paper system and hence the security of the overall system. This is substantially better than substituting an unverifiable electronic system in the same place.

4. An informative account of the challenges of implementing and deploying a

verifiable system and lessons about the distinction between theory and practice.

This project does not achieve verifiability all the way to the announcement of the election result, because it runs alongside an existing paper-based system that relies on scrutineers to check that the cast votes are included unaltered in the final count. In summary, the vVote system provides:

- cast-as-intended verification,
- recorded-as-cast verification and
- an output list of decrypted recorded votes, with a universally verifiable proof of proper mixing and decryption.

An important practical advantage of an end-to-end verifiable election scheme, compared to simpler methods of electronically assisted voting, is that it provides for electronic transfer of ballot information from distant supervised locations, supported by verifiable evidence of correctness. This is particularly important for distant polling places (e.g., overseas) and for allowing any voter to vote at any polling place. Since this project commenced, a problem in the transport of West Australian Senate ballots in the 2013 federal election has focused national attention on the security of processes for transporting paper ballots. A security problem and verification flaw in an Internet voting system in neighboring New South Wales [282] has emphasized the importance of genuinely verifiable electronic election outcomes.

12.9.2 Challenges of Combining End-to-End Verifiability with Traditional Victorian Paper Voting

A large part of the challenge arises from the special requirements of Victorian parliamentary elections. Voting is complex: for some ballots, voters typically choose from among about 30 candidates—they rank at least 5, and up to all candidates in their order of preference. Each polling place must accept votes for any race, thus serving residents of any district in the state.

The system provides privacy and receipt-freeness under reasonable assumptions about the correct randomized generation and careful deletion of secret data, and of course assuming a secure mix-net and that a threshold of decryption key sharers do not collude. It depends on both the electronic ballot marker and the printer protecting their secret data. It does not defend against ballot signature attacks [203] (often called "Italian Attacks") or other subtle coercion issues, but neither does the current paper-based system. It also reveals how many preferences a person cast. A precise statement about privacy, its assumptions and limitations, is in [184].

Another challenge is producing an accessible solution for voters who cannot fill

out a paper ballot unassisted. This is a primary justification for the project, but producing a truly verifiable solution for such voters is extremely difficult, because many of them cannot perform the crucial check that the printout matches their intention (though see [148] for a verifiable and accessible protocol). We provide a way for them to use any other machine in the polling place to do the check, in which case the cast-as-intended property depends upon at least one of the machines in the polling station not colluding with the others to manipulate the vote.

12.9.3 Specific Design Choices

12.9.3.1 Computer-Assisted Voting

The main departure from standard Prêt à Voter is the use of a computer to assist the user in completing the ballot. This is referred to as an "electronic ballot marker" (EBM). This modification is necessary for usability for all voters, and especially for particular voter groups: as described above, one of the drivers for the vVote project was accessibility, with the requirement to provide the secret ballot to blind, partially sighted and motor-impaired voters, for whom electronic assistance is necessary.

A ballot form can consist of a permuted list of about 30 candidates. It seemed infeasible for a voter to fill in a Prêt à Voter ballot form correctly without assistance. Indeed, simply filling in an ordinary paper ballot with about 30 preferences is a difficult task.[3] Computerized assistance is an important benefit of the project, and trusting the device for privacy seemed an almost unavoidable result of that usability advantage. Hence vVote depends on stronger privacy assumptions than standard Prêt à Voter, because the machine used for voting learns the vote. The vVote implementation took steps to minimize this risk, including the deletion from the EBM of all information about the vote cast, and deleting all information about the ballot form from the printer device after it has been printed.

12.9.3.2 Unified Scanner and EBM

We have already described why completing the ballot needs to be assisted by a computer. Our original design [124] included separate steps for filling in the ballot and then scanning the printed receipt. This was designed to separate the information of how the person voted from the knowledge of what their receipt looked like: the EBM learned how the person voted, but could not subsequently recognize their ballot (and hence link it to the individual voter), while the scanner knew the receipt but did not know the corresponding plaintext. However, user studies determined that a three-step voting process was too cumbersome for use. Also the necessity of print-on-demand

[3] Since some people deliberately vote informally, it is difficult to say exactly what percentage of people accidentally disenfranchise themselves by incorrectly filling in their vote. About 2% of votes in the 2006 state election were ruled informal because of "numbering errors" [567] , but the overall informality rate is closer to 10%, especially when there are many candidates on the ballot. See, e.g., `https://www.vec.vic.gov.au/Results/stateby2012distributionMelbourneDistrict.html`

meant that there was already an Internet-connected machine in the polling place that was trusted for maintaining privacy of the information on the printed ballot, including which candidate ordering corresponded to which receipt. For these reasons, the Victoria Electoral Commission insisted that the protocol unify the job of the scanner and the EBM, though it retains a separate print-on-demand step. The voter first collects their ballot form, and has an opportunity to audit it, then goes to an EBM to fill in the ballot, then the EBM sends the receipt electronically and also prints a paper record of the receipt column for the voter to check. This now means there are two online machines in the polling place (the EBMs and the ballot printers) that are trusted for vote privacy.

Receipts are digitally signed by the WBB when printed by the EBM. We provided a signature checking mobile app on Google Play that voters could download. In principle this provided accountability and defense against defaming. In practice the process for allowing voters to check, before they leave the polling place, that they had received a valid signature, was too complex for most voters to bother with.

Other significant departures are print on demand (rather than ahead of time) and printing the two halves separately (rather than overprinting a ballot), and hence the need to commit to the ciphertexts on the bulletin board before they are printed.

12.9.4 Handling Complex Ballots and Printing Them on Demand

A key requirement of the vVote project was that every polling place should allow voting in any of the 88 races being conducted across the State of Victoria. For the numbers of ballots required, and the number of candidates on each ballot, it would be infeasible for ballot forms to be centrally generated and distributed, due to the computational load required for all of the cryptographic operations necessary. This led to the design of new protocols for the distributed generation and on-demand printing of complex ballots by the print servers in the polling place [182]. Although prior designs had existed in the literature, they were not computationally feasible for the number and complexity of Victorian ballots.

The novel idea is that the printer generates a permuted list of candidate ciphers using randomness values generated by a distributed set of peers. (A similar construction is also used in Wombat.) The printer undertakes the expensive crypto operations, but does not have any influence over the values used in those operations. This prevents the printer from mounting subliminal attacks which use the randomness to encode information, or otherwise having any influence over the ciphertexts.

12.9.5 The Web Bulletin Board

The design of Prêt à Voter, in common with other end-to-end verifiable voting systems, assumes the availability of a web bulletin board (WBB): a way of posting

information publicly which cannot be removed or changed once it has been posted. Although conceptually straightforward, no previous implementation of a WBB was available with the key properties listed below required to support Prêt à Voter, and so a WBB design was developed to provide the required functionality. Details of the protocols and properties can be found in [185].

The key properties are as follows:

1. the WBB displays only items that have been posted to it;
2. the WBB needs to provide a receipt in real-time (in the order of seconds) for any post that it accepts, so that voters obtain the evidence of their vote being accepted naturally as part of the voting process;
3. the WBB needs to be able to decide in real time whether it will accept or reject a post (for example, it will not accept two votes associated with the same ballot form, and it will not accept both a vote and an audit request for the same ballot form);
4. any item for which a receipt is provided must appear on the WBB;
5. it must not be possible to undetectably alter previous posts.

Properties 2, 3 and 4 together mean that a mechanism such as a blockchain for receiving posts is not suitable here: blockchains are not designed to handle posts where immediate posting is critical; it typically takes tens of minutes for a new block of items to appear.

Receipts are signed by the WBB to counter the possibility of dishonest voters faking receipts to falsely claim that posted items have been removed (known as a *defamation attack*). Hence alteration of posted items could be detected and challenged by means of the signed receipts. A smart-phone app was developed to allow voters to check the signatures on the spot.

Furthermore, the WBB is required to be robust in order to support a live election, and in particular there should be no single point of failure. This necessitates a peered implementation, with several servers collecting the information and agreeing on which posts to accept and reject. A peered implementation requires the use of a threshold signature scheme so that the peers can collectively contribute to signing printed ballots and receipts, and where agreement of a sufficient number is required for agreement of the WBB as a whole.

These requirements were achieved by separating the function of the WBB into two elements: receiving items, and posting information publicly. Receipt of items needs to be done in real time because voters are waiting; and the system also needs to be able to handle many simultaneous posts. Hence a distributed protocol for the WBB peers was designed for which each peer had its own view of the state of the WBB, and could take local decisions on whether or not to accept a post and provide its signature share on a receipt. The global WBB is an aggregation of the local peers,

and behaves in the required way: if a threshold of peers provides their share of a signature then the voter obtains a receipt, and the post is held by the majority of the peers.

On the other hand, publication of the posts does not need to be done immediately. If peers have different views of the collection of posts, then they need to run a conciliation protocol which brings them back into agreement, and they then jointly sign and publish the collection of posts. In the Victorian election this was carried out once per day: the record of that day's voting was published at the end of each day.

The threshold required is strictly greater than $2/3$ of the number of peers. For example, if there are 7 peers then any collective signature requires a share from at least 5 of them. As long as a threshold of peers follows the protocol correctly for any particular post, a receipt will be given, and the post will appear on the collectively agreed bulletin board at the end of the day. Hence the WBB is robust against fewer than $1/3$ of peers going down, requiring rebooting, being dishonest or being the subject of an attack, and will deliver on the key properties listed above as long as a threshold behaves properly. It is also important to note that in fact property 5 will hold no matter how many peers are dishonest, but that there will be no attempt to alter previous votes (even if detectable) as long as a threshold of peers is honest.

12.9.6 vVote-Specific Vulnerabilities and Countermeasures

vVote's departures from standard Prêt à Voter, described in Section 12.9.3, have security implications that are described there. vVote also requires specific, and in some cases novel, countermeasures against particular attacks as described here:

Chain voting: vVote includes some technical measures to defend against chain voting. Printed ballot forms expire after 5 minutes if they have not been used to start a session, and the private WBB refuses to allow the same ballot form to be used to start another voting session once it has been used to start one. This means someone who sneaks an unused printed ballot form out of the polling place has 5 minutes to send it in with another voter. If someone sneaks one out having used it to start a session (and the tablet sits there with session active), then attempting to sneak this back in will not work as the ballot cannot be used to start a fresh session and the abandoned session "locks."

Clash attacks: Serial numbers are jointly generated to guarantee their uniqueness, but this doesn't prevent a corrupt printer from printing off exactly the same ballot, with the same serial number, for many different voters. The attack is detectable by ballot printing audit: such an audit would identify that the ballot has already been voted on and thus expose the corrupt printer.

The printer would have to collude with a corrupt EBM that merely reused the WBB signature, without resubmitting multiple instances of the same vote to the WBB. If the second voter voted differently from the first, the EBM would

be unable to produce a valid signature on the receipt, and unable to post it to the WBB.

subliminal attacks: The output of the printers is entirely determined by the randomness that is sent to them, and other publicly committed information. Hence they have no opportunity to provide any information which may be skewed in a particular way. Correct information posted therefore cannot leak information from the printer. Incorrect information will be detected with some non-negligible probability by the ballot-generation audit processes.

Although the whole group of randomness generation authorities can collude to mount a subliminal attack, a smaller collusion has insufficient information.

12.9.7 Practical Experiences

A University of Surrey survey of voters leaving the Australia Centre in London having cast their votes is most indicative of the system with the entire voter cohort. The VEC (Victoria Electoral Commission) ran an anonymous online questionnaire of the poll workers asked questions about equipment setup and voter support. Both surveys asked about verifiability, trust and security.

The overall results were that voters were generally satisfied with the usability of the system, but there was a wide variation in understanding the security assurances provided. For example, some voters answered that they thought that the receipt revealed their vote. Although most voters trusted the system implicitly they nonetheless took part in the verifiability steps and many said they would check receipts at home. No voters reported that the process took "too long" despite the added task of aligning and checking printed preferences in the receipt column against the candidate list.

The electoral commission also ran their own survey of poll workers. Their main findings were:

- System features for accessibility were well used.
- The system did not require much intervention in the voting session.
- The verifiability measures were well used. A quarter of respondents saw electors check their printed preferences against their candidate list. Only two respondents handled electors who believed that the result did not match their vote.
- Staff may have not fully understood verifiability.
- Although more than half of respondents stated the system was Too Difficult to Operate or Not Very Reliable, two thirds stated they would be happy to support it if more voters came to use it.

Unfortunately the importance of ballot printing audits was not well understood by the VEC, who decided not to advertise the option to voters. They did implement the necessary code, and train the poll workers to respond properly to a voter who requested an audit, but the only mention to voters was in an article on a University of Melbourne website, written by the authors. In principle the deterrent against ballot manipulation was still present, because a cheating printer that rearranged the candidate list would have had to consider the possibility of a ballot audit. However, the fact is that there were no such audits, and hence no quantifiable evidence to support the accuracy of the election result. If vVote is deployed in future elections, the ballot auditing step must be advertised to all voters, and they must be encouraged to perform it.

Another significant deviation from Prêt à Voter processes was a decision not to provide for voters to shred the candidate lists, but rather to keep them securely for shredding later. This decision followed from the security requirements of a traditional polling place, which of course vVote had to run alongside. Traditionally, paper ballots are carefully controlled and kept well away from shredders, for good reason. Future deployments of vVote will have to consider how to provide the crucial privacy process for Prêt à Voter, without compromising the integrity of the parallel paper-based process.

More detail about the practicalities of the deployment is given in [126].

12.10 Conclusions

The Prêt à Voter approach to verifiable voting has proven to be highly fruitful and adaptable, leading to many variants and enhancements and ultimately to a real deployment in a public, binding election. The Prêt à Voter approach has one great advantage:

- a simple ballot auditing procedure that depends only on the well-formedness of the ballot and is thus wholly independent of the vote or voter. This ensures full accountability, avoids vote privacy issues and means that, in addition to voter initiated audits, audits can be performed by independent scrutineers.

The original concept has been refined in many ways over the years to make it more usable, more secure and more flexible, but much more remains to be done. For example, there is doubtless still scope to make the auditing steps simpler and more natural, ideally as a side effect of the main task: to cast the vote. The procedures need to be refined and elaborated to address resilience and recovery.

Further research is required into the socio-technical aspects of verifiable voting systems in general. Such systems are large complex, security critical systems comprising, aside from the cryptographic protocols at the core, physical components, humans, procedures, etc. To date most analysis has been of the cryptographic prim-

itives and protocols, with little work understanding the environment in which these reside.

It is also important to better understand the behavior and attitude of the users to the security procedures they are asked to perform. It is essential that voters, and election officials, understand the principles of the system well enough to have confidence in the security guarantees it provides and be sufficiently motived to perform the checks. Verification protocols have to be simple enough for people to understand and use. It is desirable for voters and officials to understand the security measures well enough not to be deflected from the correct procedures by social engineering style attacks.

It is one thing to design and even implement a cryptographic protocol, and quite another to organize the polling place verification procedures that underpin the protocol's fundamental assumptions. In a real deployment the designers can recommend, but not dictate, the procedures deployed in the polling place. This increases the necessity of making those procedures simple and intuitive. End-to-end verifiability is a property of a system, but verification is hard work that someone actually has to do. If the system is verifiable but not verified then it may not produce the evidence trail that it was designed to build.

The practical advantage of the vVote deployment was to improve the flexibility and verifiability of distant pollsite voting, obviating the need for the VEC to return paper ballots by courier, while still providing evidence that the ballots were correct. Prêt à Voter is very adaptable, but it is well suited only to an environment where a reasonable number of voters actually perform the verification steps. The researchers' challenge is to streamline the verification steps so that they produce genuine evidence of a correct election outcome, while the whole system remains appealing enough to be selected by election officials for deployment.

Acknowledgments

Many people have contributed to Prêt à Voter, its design and implementation, over many years. These include Craig Burton, who was the overall vVote project lead for the Victorian Electoral Commission; Chris Culnane, who was the lead architect of the Surrey contribution to vVote; David Chaum, James Heather, Rui Joaquim, Thea Peacock, Sriramkrishnan Srinivasan, Zhe Xia, and many others. We should also like to thank Olivier Pereira for a careful reading of the chapter and many helpful comments and suggestions and the many other people with whom we have had fruitful discussions over the years, notably Ron Rivest. Ryan would like to thank the EPSRC for funding under the DIRC project and subsequently the FNR Luxembourg for support under the SeRTVS and STAST projects. Schneider is grateful to EPSRC for funding under grant EP/G025797/1.