Authentication Codes

Chris Culnane
University of Surrey

David Bismark
University of Surrey

James Heather
University of Surrey

Steve Schneider
University of Surrey

Sriramkrishnan Srinivasan
University of Surrey

Zhe Xia
University of Surrey

Abstract

The Prêt à Voter end-to-end verifiable voting system makes use of receipts, retained by voters, to provide individual verifiability that their vote has been recorded as cast. The paper discusses issues around the production and acceptance of receipts, and presents an alternative approach to individual verifiability based on Authentication Codes. These codes are constructed, in the encrypted domain, by the peered Web Bulletin Board when the vote is cast, and provide the voter with an assurance that their vote has been properly received. The approach is designed to work in a uniform way with ranked elections and single preference elections.

1 Introduction

A number of different Prêt à Voter schemes have been proposed over the years [2][11][10], however, they all share a common receipt scheme. Primarily, the receipt contains a digital signature, provided either by the booth scanner or more commonly by the Web Bulletin Board (WBB). The receipt, which the voter gets, has a barcode of some sort (more recently 2D) that contains the digital signature. The purpose of this digital signature is twofold: first it should protect the WBB from malicious voters wrongly accusing the WBB of cheating and second it should provide assurance to the voter that the receipt she has received is genuine and can be used to challenge the election if she feels this is necessary. However, in the particular scenario of electronic voting the receiver of the signed message, the voter, has no trusted equipment available to verify the signature.

In [6] the issue of malformed digital signatures going unchecked was raised in relation to both [3] and [7], both of which are DRE (Direct Recording Electronic) based voting schemes. The issue was discussed in [10] in relation to Prêt à Voter. The solution proposed in [10] was to make use of helper organisations [1] to verify the digital signature to prevent the system from discrediting voter challenges by providing false signatures. However, this approach may prove to be too expensive and infeasible on a large scale.

Below we provide a brief overview of the fundamentals of Prêt à Voter. In Section 2 we will give further details of how not verifying the digital signature could lead to an attack against the system and provide a further discussion on why the digital signature alone does not provide the properties required. In Section 3 we discuss why the acknowledgement code scheme used in Pretty Good Democracy (PGD) [1][13][5] is not applicable to Prêt à Voter in its current form and hence justify why an alternative approach is required. We discuss our new approach in Section 4, which allows a voter to verify by herself, in the polling station, that her vote has been recorded as cast, without the need for digital signatures or checking on a WBB. In Section 5 we discuss some of the wider implications of using such a scheme and additional compromises that have to be made.

[1] We compare to PGD because of the similarity between the two schemes — primarily that PGD uses a randomised candidate ordering in certain settings.

1.1 Overview of Prêt à Voter

Prêt à Voter is an end-to-end verifiable voting system. There have been various different varieties proposed over the years, with various different enhancements and capabilities for handling different types of election methods. Prêt à Voter can handle both plurality and ranked elections, using a unified interface. One of the key features of Prêt à Voter is that the ballot form is perforated down the middle, with a left hand side containing the candidate list and the right hand side containing the voter selection and a barcode. When a voter comes to vote with Prêt à Voter she fills in her preferences on the right hand side, whether it be an 'X' or a series of rankings. She then tears down the perforation and destroys the left hand side. The right hand side is then scanned by the system and submitted to the Web Bulletin Board, which produces a digitally signed receipt that is returned to the voter. In addition, the candidate list is in a randomised order for each ballot. It is this randomised ordering that is stored, in encrypted form, inside the barcode on the right; this encrypted value is known as an onion due to the multiple layers of encryption. These multiple layers of encryption are performed by different trusted parties, which means a voter only needs to trust a single one of the trusted parties to act honestly, for the secrecy of their vote to be protected. In order to count the votes each layer of the onion is peeled off (decrypted) by the relevant trusted party and the votes mixed, in a mix net, to ensure secrecy. Once all the layers have been decrypted the original candidate ordering can be reconstructed and the vote read and added to the count. At this point the vote has been through the mix net and thus cannot be connected with the original vote cast, ensuring vote secrecy. There are various audit measures in place to provide integrity assurances and allow the voter to verify that the mixing and decryption has been performed honestly. A voter is able to check that her right hand side is present on the Web Bulletin Board by entering her serial number into a website, which will return the relevant right hand side. As part of the auditing process a voter or any other interested party can then check that all the encrypted votes that were on the Web Bulleting Board were submitted to the mix network. A list of the key properties of Prêt à Voter and a sample ballot form can be found in Appendix A.

2 Weaknesses of the Digital Signature and potential Attacks

In the current Prêt à Voter system [10] the digital signature aims to protect two different parties, the WBB and the voter. The WBB is protected from a voter creating a fake receipt and then claiming that the WBB has cheated and thus wrongfully undermining the election. The voter is protected from the booth or WBB recording her vote incorrectly. Whilst the digital signature does not prevent the incorrect recording in itself, it provides the voter with the required evidence to successfully challenge the voting system should something have gone awry[2]. However, this is dependent on the voter receiving a valid digital signature.

2.1 Verifying the Digital Signature

When the voter receives the digital signature on her receipt she has no way of verifying that it is authentic or that it contains the correct information. The signature scheme itself is not human verifiable and the embedding of the signature in a 2D barcode prevents the voter from even reading its contents. Only by using some computing device is the voter able to verify the signature. However, she cannot use the voting machine (or scanner device) because it cannot be trusted and could be the party that is cheating. It therefore needs to be some form of trusted hardware, or third party, that provides the verification of the signature. As was mentioned in [10] voters would need additional assistance in polling stations to check the signature. The use of helper organisations

[2] It has been proposed in [11] that the right hand side is franked to provide proof it was submitted in addition to a digital signature. The details of how this might be achieved would require further investigation. Its security properties would rely on the physical nature of franking, in contrast to the approach proposed in this paper.

was the method proposed in [10][3], an alternative approach would be the use of a trusted application on a voters' mobile phone. In the case of using mobile phones in the polling stations this is not compliant with election law, or good practice, to encourage the use of mobile phones and photographic equipment in this setting. Therefore it is not practical to use a mobile phone to verify the signature. Trusted third parties attending all polling stations to provide verification services to voters is also not practical: the cost for third party organisations to provide such services would be very high when coverage of all polling stations is required.

2.2 Attacks against the system when the signature is not verified

The obvious question is "what are the consequences of not being able to verify the digital signature?" The answer is that it provides three possible attacks, one from each party involved: the voter, the booth and the WBB. An attack mounted by the WBB is virtually identical to that mounted by the booth. However, the WBB attack can be mitigated by requiring the booth to verify the digital signature received from the WBB. Thus, to successfully mount the WBB attack would require both the WBB and the booth to be compromised. Since the same effect can be achieved by just attacking the booth there seems little mileage in exploring a more complicated attack that does not provide a greater pay-off. We have therefore omitted details of the WBB attack and only discuss the voter and booth attacks.

2.2.1 Booth Attack

One of the key properties of the Prêt à Voter scheme is that the voting booth[4] need not be trusted. As such, any misbehaviour by the voting booth should be detectable and provable. However, if the voter cannot verify the digital signature on the receipt then it is possible for the booth to cheat and in turn accuse the voter of being malicious. The booth can change the vote cast and if it is not able to do this in an informed way[5] then it can still certainly randomise the votes. The booth then has the choice of putting the valid signature, returned from the WBB, which is for the changed vote that the booth submitted on behalf of the voter, onto the receipt or replacing the signature altogether with invalid data. In both cases it is not possible to detect whether the booth, the WBB or the voter is the source of the invalid signature. Thus it is not possible to determine whether the vote is genuine or not and the digital signature scheme becomes irrelevant.

2.2.2 Voter Attack

The voter is able to create a new receipt and transfer a valid digital signature from a valid vote to this invalid receipt. Because she is not able to verify the digital signature at the time of voting (before leaving the polling station) she can claim that the receipt was created by the electronic voting system[6]. The WBB and booth have no evidence to indicate that they acted honestly and therefore cannot contest the accusation. Whilst the voter does not have sufficiently strong evidence to prove cheating (in that she cannot even identify who it is that has cheated) she can undermine the system.

The general problem is that the digital signature cannot be verified at the time of voting and as a result it cannot be challenged at a later stage. This leads to ambiguity that results in an inability to prove who is cheating and this opens the system up to the po-

[3] Note that in this case it may be necessary not only to have one third party organisation in each poll station but several independent organisations so that the voter can select one or more that she trusts.

[4] This is the term used for the Prêt à Voter scanning equipment used to scan and submit encrypted votes and to receive and print receipts.

[5] This may seem like an inconsequential attack because the booth does not know the contents of the vote and can only stage a randomisation attack. However, in real world elections the votes are not evenly distributed within constituencies and are typically clustered in regions within the polling district around particular polling stations. Therefore randomisation attacks are particularly effective in certain polling stations that are in areas that favour a specific party, thus influencing the final outcome. Furthermore, we here assume that full permutations are used and not merely cyclic shifts as this leaks information useful in this attack [12].

[6] It is this same weakness that allows the booth and WBB to conduct their attacks.

tential attacks. What is required is an alternative authentication scheme that is verifiable by the voter at the point of receipt, without the need for trusted third parties or hardware. In Section 4 we propose such a scheme.

3 Human Verifiable Authentication Scheme

The principles behind a human verifiable authentication scheme are well established, in particular when used as acknowledgement codes in schemes such as Pretty Good Democracy (PGD) [13][5]. However, it is not possible to merely implement the same acknowledgement code schemes in Prêt à Voter. The acknowledgement code scheme used in PGD is connected to the use of code voting and the construction of the code voting ballot. Whilst some of the methods are transferable to Prêt à Voter, it is not possible to make a wholesale transfer of the PGD acknowledgement code scheme to Prêt à Voter. In Section 3.1 we shall discuss why we cannot just utilise the proposed methods in PGD in Prêt à Voter. Instead, our goal is to produce an equivalent acknowledgement code that provides guarantees both that the vote has been received by the WBB and it has been recorded as intended. Another approach to introducing a PGD style confirmation code into Prêt à Voter is given in [9].

3.1 Applying PGD ACK codes to Prêt à Voter

We wish to maintain the existing Prêt à Voter front-end. Whilst we acknowledge that we will need to provide additional features on the ballot form, we do not wish to change the fundamentals of how the voter interacts with the system, therefore ruling out the use of a fully code voting approach. That is not a criticism of code voting or PGD, just a desire to maintain the fundamental Prêt à Voter approach. In this section we shall describe the problems that arise if we were to make a wholesale transfer of the PGD acknowledgement scheme to Prêt à Voter. The attacks and weaknesses described here are only relevant when the PGD acknowledgement scheme is used with Prêt à Voter, they are not applicable to PGD.

In the original Pretty Good Democracy proposal [13] only single preference based voting schemes could be used, for example First Past The Post (FPTP). PGD is an enhanced form of Code Voting that provides additional verifiability. A voter receives a ballot form containing a list of candidates and a corresponding vote code for each candidate. To cast a vote the voter submits the relevant vote code for her chosen candidate, or a series of vote codes for ranked elections. In [13] there is a single acknowledgement code also printed on the ballot form, this acknowledgement code is returned by the WBB to provide assurance the vote has been recorded. In [5] the approach is further expanded with a number of different protocols that involve multiple acknowledgement codes and randomised candidate orderings. We discuss why these approaches are not transferrable to Prêt à Voter in more detail below.

In [13] the single acknowledgement code is generated from the vote code. Since we are not using the code voting front end the single acknowledgement code could not be calculated. Even if we found an alternative way to calculate the acknowledgement code it would not provide the guarantees required since the single acknowledgement code cannot encode the position of the submitted vote, thus allowing the booth or WBB to switch vote position.

An approach to using PGD for more expressive voting schemes was proposed in [5], in the form of three different protocols.

Protocol A proposes using a unique acknowledgement code for each candidate. However, it relies on the voter submitting each rank separately and waiting for the appropriate code to be sent back. This could be applied in single preference votes in Prêt à Voter, since the act of submitting the right hand side is equivalent to submitting just a first preference. However, for ranked elections it is not applicable, since the entire vote is submitted as one whole. If the vote is submitted as a single entity the booth or web bulletin board (WBB) can switch the preferences and still return a seemingly valid set of acknowledgement codes. This is due to the booth/WBB learning all of the relevant ACK codes. Since Prêt à Voter is designed to work for both single preference and ranked elections, we need an approach that is applicable in both. Having alternative schemes for different elections is possible, but the feature of a unified front-end is a nice property of Prêt à Voter that we would like to preserve.

Protocol B requires the voter to submit her vote codes in preference order. The voter receives the candidates in a random order and also receives a list of preference acknowledgement codes. This would initially appear to be applicable to Prêt à Voter, since the vote is submitted as a single whole and the construction of the ballot papers seems similar, with the randomised candidate order. However, this approach is also open to manipulation by the booth and WBB. The reason this works in the PGD scheme is that the mapping between candidate and preference order is kept secret. In Prêt à Voter the candidate to preference order is revealed when the voter submits the vote — due to the vote being submitted in the random order they appear on the ballot form, as opposed to preference order. As such, the booth and WBB can learn all the information they need (order and acknowledgement code) to switch preferences around, whilst maintaining a seemingly valid acknowledgement code.

Protocol C proposes the use of a matrix of vote codes. This is obviously not applicable for Prêt à Voter, however, an approach using a matrix of acknowledgement codes could be. This would provide unique acknowledgement codes for each candidate in each possible preference, thus preventing the switching around of acknowledgement codes. However, such an approach is likely to be too cumbersome and difficult for the average voter to use.

The acknowledgement schemes described in [5] are not fully transferable to Prêt à Voter, in particular when handling ranked elections, and therefore a new approach must be devised. Since it is not known how to provide a human verifiable digital signature, we will have to rely on some form of acknowledgement code in order to provide the same properties we want from the digital signature. As we have seen above we cannot simply apply an existing acknowledgement code scheme, since it is either too complex (in the case of matrix style PGD) or not secure. We therefore propose a new alternative acknowledgement code called Authentication Codes for Prêt à Voter.

4 A New Approach

It is intended that the scheme proposed here is standalone and independent of the digital signatures and WBB check. In Section 5.3 we shall discuss some of the potential problems in keeping a WBB check in the light of unverified digital signatures.

The Authentication Code consists of a string of digits, some of which are the rankings that the voter has given to the candidates and the rest being random digits. Looking at the Authentication Code without an indication about which digits are those rankings, it should be hard to accurately guess which the ranking digits are. The voter can create the Authentication Code by simply filling out a few empty boxes on an Authentication Strip given to her together with the ballot form and the voting system then essentially proves to her the correct recording of her vote by giving the code back to her separately — the Authentication Code becomes a secret shared between the voter and a set of peer WBBs. This is achieved through distributed homomorphic cryptography, for example the Paillier cryptosystem [8][4], and redundancy in the Authentication Code. In the following sections we shall discuss the Authentication Code scheme from the various different perspectives, giving an overall picture of how it works.

4.1 Overview of Approach

In this section we will provide a brief description of the desired properties of each component and any assumptions we make. We require an Authentication Code that the voter can compute by hand without needing to trust hardware or individuals. It should not be possible for either the WBB or booth equipment to provide a forged Authentication Code that the voter will accept, even if they know the submitted encrypted vote and computed Authentication Code. The Authentication Code will be constructed by the Election Manager, the same authority who constructs the ballot forms in Prêt à Voter. It is assumed that the Election Manager is honest and is destroyed following the construction of the ballot forms and Authentication Codes. The Authentication Strip will be sealed inside a tamper evident envelope with the corresponding ballot form. A voter should reject any envelope that appears to have been tampered with. This does lead to a greater chain of custody requirement, which we discuss further in Section 5.2. The WBB peers construct shares of the complete Authentication Code, but it is only after the public combining step that the final Authentication Code is learned. The Authentication Code shares and Authentication Values are stored in encrypted form by the WBB peers. As such no single peer should be able to learn a partial Authentication Code or Authentication Value. The Authentication Code should be checked and either accepted or challenged whilst still within the booth, once a voter leaves the polling station no further challenges can be made. If the voter is not happy about the received Authentication Code she should be able to withdraw her vote and cast another.

4.2 The Voter Perspective

It is imperative that the voter experience is simple and easy to follow and this is why the voter's task is limited to merely writing down her vote in the appropriate boxes and comparing two lists of numbers. There is an implicit chain of custody related to the ballot form and more importantly the Authentication Strip. In Prêt à Voter [2][11][10] the chain of custody of the ballot form protects the secrecy of the vote. With the addition of an Authentication Strip the chain of custody also protects the integrity. The implications of this additional burden on the chain of custody is discussed in Section 5.2. One way of helping to verify and maintain the chain of custody is to enclose the ballot form and Authentication Strip in a tamper evident envelope. The voter collects her ballot form and also receives the Authentication Strip associated with that ballot form (most likely contained in a single tamper evident envelope). This Authentication Strip contains a list of numbers with n blank boxes, where n is the number of candidates. Once the voter has completed her ballot form she writes down the preferences into the blank boxes on the Authentication Strip. Where there is no preference (no rank assigned to a candidate) a zero is entered in the box on the Authentication Strip. Figure 1 shows how an Authentication Strip would be filled in, with the voter simply transcribing her vote into the empty boxes in the Authentication Code.

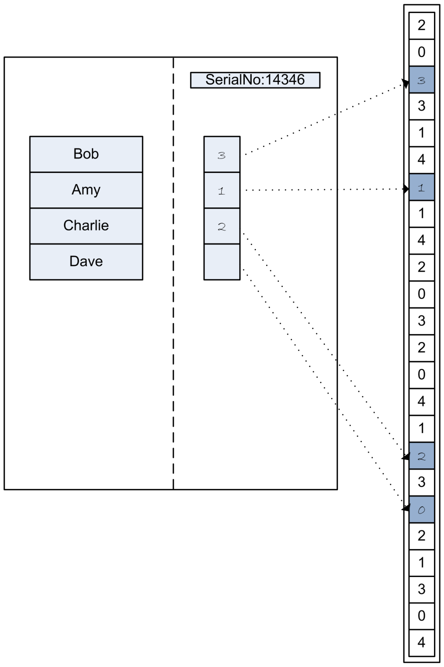

Figure 1: Example Authentication Strip

Having completed this stage the standard Prêt à Voter scheme continues with the voter submitting her right hand side. It is important that the Authentication Strip is kept secret[7] prior to the scanning and

[7] A particular danger would be if the Authentication Strip became known to an attacker who had subverted scanning equipment or the WBB or anyone conspiring with such an attacker.

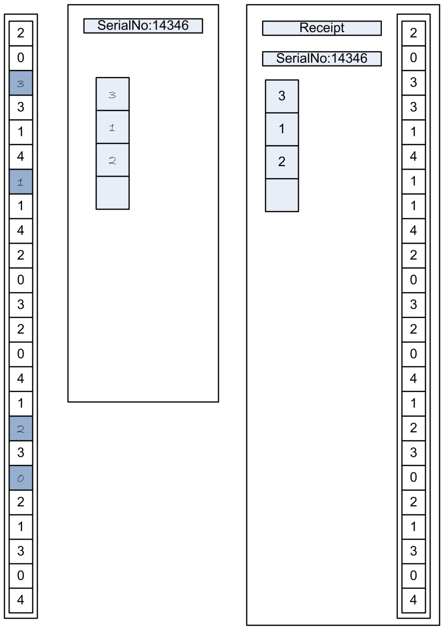

Figure 2: Example Receipt

submission of the vote. The receipt received back from the WBB, and printed by the booth, contains the candidate ranking as normal and also a copy of the completed Authentication Strip, as shown in Figure 2. The voter can then check that the two strips match exactly: the Authentication Code that she created on her Authentication Strip by transcribing her vote matches the Authentication Code printed on her encrypted receipt. If the codes match then the voter can be sure that the vote has been recorded as it was cast. If they do not, the voter can immediately challenge the election and she should not accept that her vote has been included in the tally unless the Authentication Codes match exactly[8].

4.3 The Election Manager Perspective

The Authentication Code is a series of randomly permuted values with n (where n is the number of candidates) blank spaces added. To create the Authentication Code the Election Manager takes the list of values between 0 and m (where $m = n$ in ranked voting and $m = 1$ in single choice voting) plus the blank space (denoted ϕ) and randomly permutes and concatenates these values n times.

For example, with $n = 4$, the values 0,1,2,3,4,ϕ are randomly permuted 4 times[9]

$$20\phi314, \phi14203, 2041\phi3, \phi21304$$

and concatenated into a list of 24 digits[10]:

$$20\phi314\phi142032041\phi3\phi21304$$

Having generated this random list of digits, the positions of the blank spaces are extracted:

$$3, 7, 17, 19$$

At each location 0 is inserted.

$$200314014203204103021304$$

The next stage is to create the individual Authentication Values for each candidate. The first candidate Authentication Value corresponds to the first blank location, the second candidate Authentication Value to the second blank location and so on. To create the Authentication Value a string of zeros is created that is the same length as the Authentication Code, for example:

$$000000000000000000000000$$

The zero in the location of the first blank location generated above is replaced with a 1.

$$001000000000000000000000$$

[8] If she leaves the polling station with non-matching Authentication Codes she has forfeited her right to challenge the election at a later time.

[9] Here $m = n$ because ranked voting is used. We have chosen $n = 4$ as an arbitrary example.

[10] There may be a more optimal way of generating this Authentication Code that would allow it to be shorter, but initially we shall use this approach since the blinding effect will become obvious. We use the value 4 as an example of an election with 4 candidates.

This forms the first Authentication Value. This is repeated for each candidate:

$$000000100000000000000000$$

and

$$000000000000000010000000$$

and

$$000000000000000000100000$$

These Values, along with the Authentication Code itself, are then encrypted under the WBB public key

$$E_{PK_{wbb}}(200314014203204103021304)$$

$$E_{PK_{wbb}}(001000000000000000000000)$$

$$E_{PK_{wbb}}(000000100000000000000000)$$

$$E_{PK_{wbb}}(000000000000000010000000)$$

$$E_{PK_{wbb}}(000000000000000000100000)$$

and copies of all encrypted Authentication Codes and Values, with references to ballot form serial numbers, are copied to all WBB peers.

4.4 The Web Bulletin Board Perspective

Unlike the existing digital signature scheme, a single WBB does not produce the Authentication Code by itself. We envisage the use of a distributed WBB, either with a central website that handles contact with all the peers or an entirely peered WBB network. WBB peers (who hold the distributed private key shares) jointly construct the Authentication Code. In short, as there are several parties involved no single party ever knows what the Authentication Code is (before it has been issued) and therefore cannot cheat. The construction of the peered WBB is outside the scope of this paper, but we include the details on the construction of the Authentication Code.

The WBB, as it has been extensively described in literature, has a table that links the serial number of the ballot form to the relevant ciphers[10]. We expand this table to also include an encrypted Authentication Code and individual Authentication Values for each of the candidates. Both the Code and the Values are encrypted under the WBB public key[11] and are created by the Election Manager during the setting up procedure for the election. This means that after completion of the setting up procedure a threshold set of the WBB peers must work together if they wish to decrypt the Authentication Code or Values.

4.4.1 Receiving a Vote

In the structure with a central distributor communicating with WBB peers, the central distributor records the vote and then notifies each of the peers that an Authentication Code needs to be constructed. Each peer than takes a copy of the submitted vote from the central distributor and commences the Authentication Code construction. In the scenario where there is no central distributor and it is entirely peered, each of the WBB peers will receive the vote separately and will be able to commence construction of the Authentication Code themselves.

To construct the Authentication Code each WBB peer performs the following steps:

1. The list of rankings made by the voter, with zeros inserted wherever no ranking or selection is made, is received. Continuing the example from above the vote is

$$3, 1, 2, 0$$

2. The list of encrypted Authentication Values is retrieved from the peer's own database and each Value is homomorphically scaled by the corresponding ranking.

$$E_{PK_{wbb}}(003000000000000000000000)$$

$$E_{PK_{wbb}}(000000100000000000000000)$$

$$E_{PK_{wbb}}(000000000000000020000000)$$

$$E_{PK_{wbb}}(000000000000000000000000)$$

[11] The corresponding private key is shared in a threshold fashion, each share held by a WBB peer.

3. Each of the Authentication Values are homomorphically added

   $$E_{PK_{wbb}}(0030001000000000020000000)$$

   with the Authentication Code

   $$E_{PK_{wbb}}(2003140142032041030213 04)$$

   resulting in a single, encrypted Authentication Code:

   $$E_{PK_{wbb}}(2033141142032041230213 04)$$

4. The peer performs a partial decryption (with corresponding proof of decryption) of the Authentication Code that it has constructed using its private key share.

5. The partial decryption and the proof of decryption are published.

Once a threshold set of peers have provided their partial decryptions the public combining step is performed and the decrypted Authentication Code is recorded and returned to the booth and voter.

4.4.2 Election Process

As with most schemes the technology alone does not provide enough to handle the inherently unpredictable nature of people, we also need processes and standard operating procedures to cover a wider variety of situations. In such a process we set out what each party's responsibilities are and how disputes are resolved. The voter has the responsibility of checking that the Authentication Code matches the one on the Authentication Strip. If it does not she must challenge the election at that point, before leaving the polling station. The leaving of the polling station is an implicit acceptance of the vote shown by the Authentication Code. This is necessary to prevent a malicious voter leaving the polling station and fabricating a fake receipt and then attempting to undermine the election.

4.4.3 Security Discussion

When looking at the security of the Authentication Code we must look at what has to be undertaken in order to attack it. The primary goal of any attacker, be it a booth or WBB, would be to return a valid Authentication Code, whilst recording a different result. If this was achieved it would allow the cheating component to return a valid Authentication Code to the voter, who would accept it and leave the polling station. Once the voter has left she cannot challenge the result based on her Authentication Code alone — she is dependent on the write once aspect of the WBB. This is due to the malicious voter problem discussed above. As such the cheating component would get away with the cheating by diverting blame onto the voter.

Given the Authentication Code and a single voter preference, changing the code to another valid code to match a different preference cannot be achieved with better than 50% probability (as there are, for example, two 1's to choose from). In ranked voting, correctly guessing each preference would have an associated 50% probability and thus the chance of correctly guessing and changing all preferences would decrease exponentially with the number of preferences changed.[12] Furthermore, the chance of successfully changing many votes, without being caught, decreases exponentially with the increasing number of votes changed. It is clear that the WBB itself and the peers cannot cheat individually, since they are operating in the encrypted domain and do not see the Authentication Code until it is publicly combined. They could make random changes but given the level of redundancy in the Authentication Code they are far more likely to make a detectable change than get away with it.

If an individual WBB tries to record a different vote preference the proof of partial decryption will not hold. If the central WBB distributes an incorrect preference the Authentication Code will not match

[12] These are worst case probabilities, in most cases the probability of correctly guessing the locations is far smaller, for example 25%. An example of where the worse case scenario is applicable would be where a voters preference is removed or an additional preference is added.

the one the voter has.

4.5 Short Code Alternative

The method proposed above results in a code length that grows quadratically with the number of candidates. This is obviously undesirable, particularly in elections with large numbers of candidates. In smaller elections this is not so much of an issue, since it is still quite manageable with 3 to 4 candidates.

Ideally we would like to have a code that grows linearly with the number of candidates. In this section we will propose such an approach. The back-end processing is identical and the principle stays the same. However, the front-end involves a further level of indirection. The change to the front-end involves permuting the position of each candidate's "blank location". Recalling the original scheme, the blanks were in the same order as the ballot form (top most candidate related to top most blank). This extra level of indirection means we can reduce the number of elements we need to blind the voter's preference. However, it does require the voter to do some additional work in locating where she needs to write in her preference values on the Authentication Strip.

The code length is thus given by the equation $2n + 1$, where n is the number of candidates. The reason for requiring the additional 1 is to include a blinding 0 value to handle partial rankings and FPTP. This results in a code that at worst has a 1/2 chance of an adversary guessing the location of any particular value. This chance worsens when trying to guess multiple partial ranking locations or in FPTP. However, the code length can be further generalised to adjust the chance of guessing. The code length can then be given by $n + (p-1)(n+1)$ where p is the $1/p$ probability of guessing any particular value.

In order to make it easy for the voter to make the link between the candidate preference box and the location of the corresponding blank in the Authentication Strip the relevant boxes are labelled. In this example we have used letters to label the boxes, but any form of image or text could be used, depending on cultural sensitivities. Figure 3 shows a ballot and Authentication Strip for the Short Code variant.

The comparison step for evaluating the Short Code

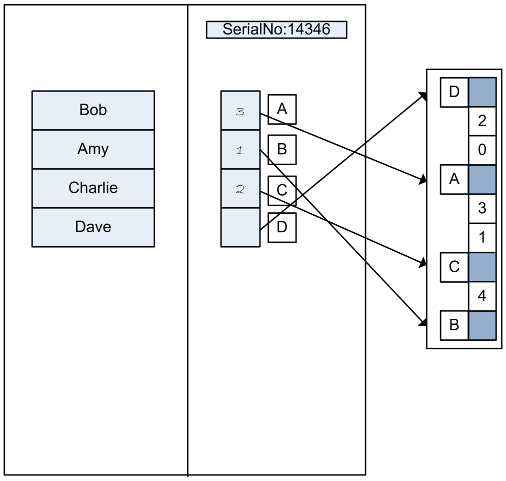

Figure 3: Example Short Code Authentication Strip

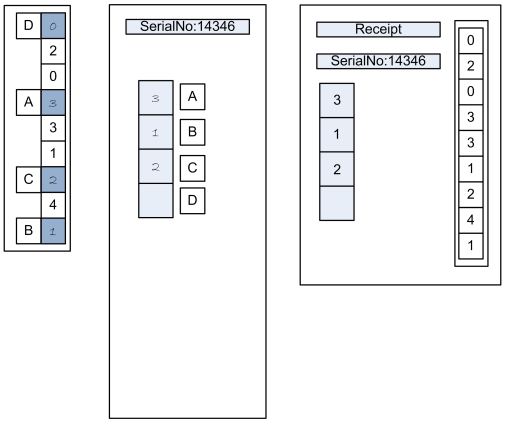

Figure 4: Example Short Code Receipt

variant is the same as before, however, the code is now considerably shorter.

4.5.1 Short Code Construction

The construction for the Short Code shares a number of the same steps as before, but due to the shorter length and additional permutation we explain it separately below.

To create the Short Code variant the Election Manager takes the list of values between 0 and n (where n is the number of candidates) plus n blank spaces (denoted ϕ) and randomly permutes this list. For example, with $n = 4$, the values 0,1,2,3,4,ϕ,ϕ,ϕ,ϕ are randomly permuted

$$\phi 20 \phi 31 \phi 4 \phi$$

Having generated this random list of digits, the positions of the blank spaces are extracted:

$$1, 4, 7, 9$$

At each location 0 is inserted.

$$020031040$$

The next stage is to create the individual Authentication Values for each candidate. The first candidate Authentication Value corresponds to the first blank location, the second candidate Authentication Value to the second blank location and so on. To create the Authentication Value a string of zeros is created that is the same length as the Authentication Code, for example:

$$000000000$$

The zero in the location of the first blank location generated above is replaced with a 1.

$$100000000$$

This forms the first Authentication Value. This is repeated for each candidate:

$$000100000$$

and

$$000000100$$

and

$$000000001$$

These values, along with the Authentication Code itself, are then encrypted under the WBB public key. The Authentication Values at this point refer to n labels in canonical form (A,B,C,D).

$$E_{PK_{wbb}}(020031040)$$

$$E_{PK_{wbb}}(100000000)(A)$$

$$E_{PK_{wbb}}(000100000)(B)$$

$$E_{PK_{wbb}}(000000100)(C)$$

$$E_{PK_{wbb}}(000000001)(D)$$

The next step requires us to create a random indirection mapping. We randomly permute the labels associated with each Authentication Value:

$$E_{PK_{wbb}}(020031040)$$

$$E_{PK_{wbb}}(100000000)(D)$$

$$E_{PK_{wbb}}(000100000)(A)$$

$$E_{PK_{wbb}}(000000100)(C)$$

$$E_{PK_{wbb}}(000000001)(B)$$

This permuted list of letters is printed to the left of each empty position on the Authentication Slip, as shown on Figure 3. The final step is to then reorder the Authentication Values to the canonical order of the letters that correspond to the labels.

$$E_{PK_{wbb}}(020031040)$$

$$E_{PK_{wbb}}(000100000)(A)$$

$$E_{PK_{wbb}}(000000001)(B)$$

$$E_{PK_{wbb}}(000000100)(C)$$

$$E_{PK_{wbb}}(100000000)(D)$$

Each candidate position on the ballot form[13] is labeled alphabetically starting with A.

The Authentication Code and the Authentication Values, in their permuted order, are then stored onto the WBB peers.

4.5.2 Short Code Receiving a Vote

The process undertaken on receipt of a vote is identical to that in Section 4.4.1. This is because the permuted Authentication Values will modify the relevant part of the completed Authentication Code. As such, the WBB peers do not need to know what the permutation is, because it is dealt with during the homomorphic addition.

5 Discussion

In this section we will further discuss some of the issues raised during the description of construction and use of the Authentication Code. This section aims to pose some open questions whose answers are still open to debate and discussion.

5.1 Usability

We believe the short code variant to be more usable, even though there is an additional level of indirection. However, the advantage of the shorter code we believe outweighs this. Both schemes have been implemented using the Paillier cryptosystem [4] and are at proof of concept stage. We intend on conducting a wider usability analysis in future Prêt à Voter trials. There are a number of usability properties to evaluate. For example, whether numerical representations are the most desirable as opposed to letters or symbols. Additionally, whether in plurality schemes we should use blanks to represent '0' and 'X' to represent '1'. We aim to explore these further in the future. It should be noted that such modifications do not impact on the security of the scheme, they are purely related to usability and accessibility.

5.2 Chain of Custody

In Section 4.2 we briefly mentioned the greater burden on the chain of custody when including an Authentication Code. In the existing Prêt à Voter schemes, which involve pre-constructed ballot forms, the chain of custody is required to protect the secrecy of the vote. If an adversary is able to gain access to the ballot forms he will be able to record the candidate ordering and uncover how a voter has voted from the receipt. With the addition of an Authentication Code the chain of custody is also required to protect the integrity of the election. If an adversary is able to gain access to the Authentication Strip he can use knowledge of the locations of the blanks to then corrupt a scanning machine to be able to produce forged Authentication Codes. Various techniques can be used to try and maintain the chain of custody and allow the voter to verify that it has been maintained. For example, tamper evident envelopes would help in this regard. Whilst it is clearly undesirable to have a greater burden on the chain of custody, it would appear to be unavoidable. The chain of custody problem is evident in all schemes using precomputed acknowledgement or confirmation codes.

It should be noted that the removal of trust in opaque devices is a key component of Prêt à Voter and since the secrecy of Prêt à Voter is based on the chain of custody of the secret ballot forms, providing a higher level of assurance by keeping another piece of paper in the same sealed envelope seems only a marginal extra risk. Merely breaking the chain of custody of the Authentication Code does not immediately break the integrity of the voting system, only secrecy is broken and that has been compromised already by the attacker seeing the ballot form. The attacker must also subvert the booth device to perform an attack on a single vote. Without our contribution, subverting the device would potentially allow the attacker to perform the attack on all votes cast through that device; with our contribution the attacker must also break the chain of custody of the Authentication Code for each vote.

A possible way to improve the situation would be to allow the voter to construct her own Authentication Codes in an out-of-band fashion. Thus allowing

[13] Note that this is the positions from top to bottom on the form: in Prêt à Voter the position on the form does not correspond to a particular candidate as the candidate list is in a random order on each form.

a voter to have verifiable communication with the peered WBB without having to trust the chain of custody of the Authentication Strip. This forms part of our future work, but would result in a significant departure from how Prêt à Voter currently works.

5.3 Issuing Receipt and the WBB Check

The provision of an Authentication Code provides the voter with a guarantee that her vote has been recorded by a threshold set of WBB peers, assuming that the chain of custody of the Authentication Strip has been maintained. It is natural to then question whether a receipt, with or without a digital signature, should be still be issued and thus whether the checking of the WBB by the voter should still be required.

It is often advocated that maintaining the issuing of the receipt and the WBB allows the voter to verify that her vote is correctly recorded. However, as was discussed in Section 2.2 this is only true if the signature is being verified. Since we are unaware of any practical way of allowing the voter to verify the digital signature, what are the implications of still issuing the receipts? It would appear that continuing to issue the receipts allows a malicious voter to continue to maliciously undermine the voting system. The data being provided by the WBB cannot be demonstrated to be genuine, even if it is. Therefore the system as a whole does not have a defense against malicious voters. Since a voting system is only of use if it is trusted, providing an avenue for malicious voters to potentially undermine the trust in the voting system seems unwise. Whilst removing the receipt and WBB check is a significant departure from the Prêt à Voter scheme, it may well provide better protection for the system as a whole, without any loss in terms of security.

6 Conclusion

In this paper we have identified the need for an additional, or possibly alternative, method for a voter to verify that her vote has been recorded as cast. We have proposed a new acknowledgement scheme entitled Authentication Codes that provides a human verifiable technique for verifying the accurate recording of the vote. We have identified possible weaknesses related to the chain of custody of acknowledge code based schemes in general and discussed possible techniques for mitigating the problem.

7 Future Work

A combinatorial analysis of the length and construction of the Authentication Code may allow the length of the code to be shortened, particularly in the case of single preference voting.

The proposed approach will reveal the number of candidates that the voter has included in her ranking. This is an underlying problem in Prêt à Voter as it currently stands, since that information is posted to the WBB and printed on the receipt. It could be solved by including an additional STOP candidate and then requiring full ranking. However, the usability of such an approach would need to be carefully evaluated. The Authentication Code itself does not reveal any additional information than the underlying Prêt à Voter system.

To limit the reliance on the chain of custody of the ballot forms with their corresponding Authentication Strips, it may be possible to devise a scheme whereby Authentication Codes are created out-of-band, initiated by the voters themselves. A voter, in the privacy of her own home, visits the election website and selects the option to create an Authentication Code. Running software on the election authority's server, on a trusted third party's server or on the voter's own computer, an Authentication Code with corresponding Authentication Values are created as described above, all encrypted under the public key of the WBB peers. The encrypted Authentication Code and Values are copied to the WBB peers and the voter prints the Authentication Strip on her own printer. The printed sheet contains a barcode with the serial number of the Authentication Code. When the voter allows her Prêt à Voter vote to be scanned in the polling station, she also allows the barcode of the Authentication Strip to be scanned — only the barcode should be scanned, the Authentication Strip itself must be kept secret. On the receipt printed by the voting system is the Authentication Code that the voter created herself. This does create a possible coercion issue, in that a coercer could request to see a voters Authentication Strip and then require a particular completed Authentication Code to appear on the WBB. This would allow a randomisation attack to be undertaken. However, the same can already occur in Prêt à Voter if the coercer requests to see a receipt with particular preference ordering on it.

Acknowledgements

We are grateful to the anonymous referees, and to Jeremy Clark, for their detailed and insightful comments which have improved this paper. We would like to thank Peter Ryan for the helpful and enjoyable discussions on both this work and his alternative approach [9], developed independently and concurrently to solve the same problem. This work was funded by the EPSRC 'Trustworthy Voting Systems' project under grant EP/G025797/1 and was conducted while James Heather was a Royal Society/Leverhulme Trust Senior Research Fellow.

References

[1] B. Adida. *Advances in Cryptographic Voting Systems*. PhD thesis, M.I.T., 2006.

[2] D. Chaum, P. Y. A. Ryan, and S. Schneider. A practical voter-verifiable election scheme. In *ESORICS*, pages 118–139, 2005.

[3] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.

[4] Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer, 2001.

[5] James Heather, Peter Y. A. Ryan, and Vanessa Teague. Pretty Good Democracy for more expressive voting schemes. *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS 2010)*, pages 405–423, 2010. LNCS 6345.

[6] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A systems perspective. In *In USENIX Security Symposium, number 3444 in Lecture Notes in Computer Science*, pages 33–50. Springer-Verlag, 2005.

[7] C. Andrew Neff. Practical high certainty intent verification for encrypted votes, 2004.

[8] Pascal Paillier. Paillier encryption and signature schemes. In Henk C. A. van Tilborg, editor, *Encyclopedia of Cryptography and Security*. Springer, 2005.

[9] Peter Y. A. Ryan. Prêt à voter with confirmation codes. In *Proceedings of the USENIX Electronic Voting Technology Workshop*, 2011.

[10] Peter Y. A. Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4):662–673, 2009.

[11] Peter Y. A. Ryan and Steve A. Schneider. Prêt à voter with re-encryption mixes. In Dieter Gollmann, Jan Meier, and Andrei Sabelfeld, editors, *ESORICS*, volume 4189 of *Lecture Notes in Computer Science*, pages 313–326. Springer, 2006.

[12] Peter Y. A. Ryan and Vanessa Teague. Ballot permutations in Prêt à Voter. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop*, 2009.

[13] Peter Y. A. Ryan and Vanessa Teague. Pretty Good Democracy. *Proceedings of the 17th International Workshop on Security Protocols*, 2009. LNCS.

A Prêt à Voter Overview

Figure 5 contains a sample ballot form for the Prêt à Voter system. Whilst the exact rendering of the ballot form may vary between versions, the fundamentals remain the same. Those fundamentals are as follows:

- Ballot form consists of a left and a right hand side perforated from top to bottom down the middle
- Left hand side contains the candidate list in a random order
- Right hand side contains the voting boxes, serial number and a barcode
- The barcode contains an "Onion" that is an encrypted representation of the randomised candidate list
- The right hand side is scanned by a machine in the polling station and submitted to a Web Bulletin Board (WBB)
- The WBB returns a receipt with a copy of the right hand side on it and a digitally signed copy of the submitted values
- The right hand side is published to a publically accessible WBB
- The voter can check their right hand side is present on the WBB by entering the serial number on their receipt/right hand side
- At the close of the election all the right hand sides on the WBB are submitted to the mix net and then finally decrypted and counted. (The exact methodology used depends on the election system and the version of Prêt à Voter being used.)

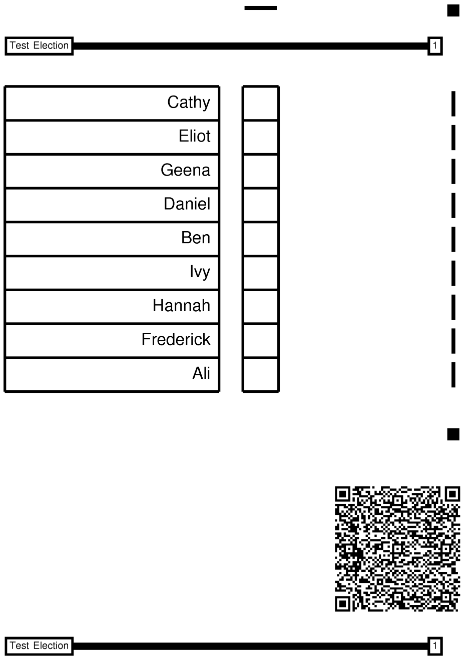

Figure 5: Example Prêt à Voter Ballot Form