UNIVERSITY OF
NEWCASTLE

University of Newcastle upon Tyne

COMPUTING SCIENCE

Pret a Voter with Paillier Encryption

P. Y. A. Ryan.

TECHNICAL REPORT SERIES

No. CS-TR-965 June, 2006

TECHNICAL REPORT SERIES

No. CS-TR-965 June, 2006

Pret a Voter with Paillier Encryption

P. Y. A. Ryan

Abstract

In a previous paper, a version of the Pret a Voter verifiable election scheme using ElGamal encryption and enabling the use of re-encryption mixes was presented. In order to ensure that the construction of the ballot forms mesh with the re-encryption mixes, it was necessary to draw the seed values from a statistical distribution, e.g., a binomial. In this paper we present a similar construction of the ballot forms but using Paillier encryption in place of ElGamal. The advantage of this is that the homomorphic properties of Paillier are ideally suited to our construction and removes the need to constrain the distribution of seed values.

As with the scheme using ElGamal, we have a distributed construction of encrypted ballot forms. This enables on-demand decryption and printing of the ballot forms and so eliminates the need to trust a single authority to keep this information secret. It also avoids chain of custody issues as well as chain voting style attacks.

Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

RYAN, P. Y. A..

Pret a Voter with Paillier Encryption
[By] P. Y. A. Ryan.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2006.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-965)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-965

Abstract

In a previous paper, a version of the Pret a Voter verifiable election scheme using ElGamal encryption and enabling the use of re-encryption mixes was presented. In order to ensure that the construction of the ballot forms mesh with the re-encryption mixes, it was necessary to draw the seed values from a statistical distribution, e.g., a binomial. In this paper we present a similar construction of the ballot forms but using Paillier encryption in place of ElGamal. The advantage of this is that the homomorphic properties of Paillier are ideally suited to our construction and removes the need to constrain the distribution of seed values.

As with the scheme using ElGamal, we have a distributed construction of encrypted ballot forms. This enables on-demand decryption and printing of the ballot forms and so eliminates the need to trust a single authority to keep this information secret. It also avoids chain of custody issues as well as chain voting style attacks.

About the author

Peter Ryan is a Professor of CSR. He is responsible for the security and privacy aspects of the DIRC program and is involved in the European MAFTIA project. Prior to joining the CSR, he conducted research in formal methods and information assurance at GCHQ, CESG, DERA, SRI Cambridge, the Norwegian Computing Centre Oslo and the Software Engineering Institute, Carnegie Mellon University. Before migrating into information assurance he was a theoretical physicist and holds a BSc in Theoretical Physics and a PhD in Mathematical Physics from the University of London for research in quantum gravity. He has published numerous articles; the most recent being "Mathematical Models of Computer Security," a chapter in LNCS 2171, is based on lectures given at the FOSAD 2000 Summer School. He is co-author of the book "Modelling and Analysis of Security Protocols," Pearson 2001.

Suggested keywords

VERIFIABLE ELECTIONS. VOTING,
CRYPTOGRAPHY,
RE-ENCRYPTION MIXES

Prêt à Voter with Paillier Encryption

P Y A Ryan[*]

Abstract

In [9], a version of the Prêt à Voter verifiable election scheme [2] using ElGamal encryption and enabling the use of re-encryption mixes was presented. In order to ensure that the construction of the ballot forms mesh with the re-encryption mixes, it was necessary to draw the seed values from a statistical distribution, e.g., a binomial. In this paper we present a similar construction of the ballot forms but using Paillier encryption in place of ElGamal. The advantage of this is that the homomorphic properties of Paillier are ideally suited to our construction and removes the need to constrain the distribution of seed values.

As with the scheme of [9], we have a distributed construction of encrypted ballot forms. This enables on-demand decryption and printing of the ballot forms and so eliminates the need to trust a single authority to keep this information secret. It also avoids chain of custody issues as well as chain voting style attacks identified in [8].

1 Introduction

The Prêt à Voter scheme, first presented in [5] and [6] and elaborated in [2], is a cryptographic voting scheme that enables voter-verifiability: at the time of casting their vote, voters are provided with an encrypted receipt which enables them to check, via a secure Web Bulletin Board (WBB), that their receipt is accurately included in a verifiable anonymising tabulation process. Various checking mechanisms serve to detect any corruption in any phase of this process: encryption of the vote (more precisely, in the case of Prêt à Voter, in the construction of the ballot forms), recording and transmission of the encrypted ballot receipt and the decryption and tabulation of the votes.

[*] University of Newcastle

Full details can be found in [2]. Henceforth we will refer to this version of the scheme as Prêt à Voter'05.

Prêt à Voter'05 [8] uses RSA encryption and a layered construction for the ballot form onions and decryption mixes at the tabulation stage. In [9] this was adapted to use ElGamal encryption in place of RSA enabling the use of re-encryption mixes in place of decryption mixes. This provides a number of advantages, such as separation of mix and decryption phases, easier recovery in the event of faulty mix tellers or corruption detection during the audit phase and so on.

The special representation of the receipts in Prêt à Voter as a pair of an index value and an encrypted term, means that they cannot be put directly through a re-encryption mix, unless we are prepared to leave the index values unchanged through the mix. Invariant index values would of course allow an observer to partition the mix. In the case of Prêt à Voter'05, there was a natural way to transform the index values as the terms moved through the mix as a function of the germ values revealed at each stage.

The resolution of this problem suggested in [9] is to encrypt the seed values as exponents rather than as pure terms in order to allow the index value to be absorbed into the onion term. This results in a pure ElGamal term that can now be put through a conventional (robust) re-encryption mix. The drawback of this approach is that the raw votes appear in the final decryption as exponents and so if unconstrained seed values are used, we would have to solve the discrete log problem in order to extract the votes. This was avoided by constraining the seed values to a suitable statistical distribution, i.e., a binomial. This renders the decryption tractable whilst avoiding edge-effects that might compromise secrecy in some instances if the seed values where simply bound to an interval.

In this paper we explore the use of Paillier encryption in place of ElGamal. This has the advantage that the homomorphic structure of Paillier is ideally suited to our purposes and we are able to sidestep the obstruction described above and so relax all constraints on the choice of the seed values.

We note also that in a recent paper [1], Adida and Rivest propose a "Scratch and Vote" scheme that uses Paillier encryption, but for a different purpose, i.e., to allow homomorphic tabulation.

Obelix	
Idefix	
Asterix	
Panoramix	
	$7rJ94K$

Figure 1: Prêt à Voter ballot form

X
$7rJ94K$

Figure 2: Prêt à Voter ballot receipt

2 Outline of Prêt à Voter

We recall the key ingredients of the Prêt à Voter scheme. The key innovation of the Prêt à Voter approach is, to encode the vote in a randomised frame of reference, i.e., a randomised candidate list. Other schemes require that the vote value be communicated to a device which then encrypts the value. Prêt à Voter thus has the advantage that the device does not learn the voter's selection and so the threat of various side channels and subliminal channels is neatly sidestepped.

At the polling station, voters are assigned (or choose) at random a ballot form, and example of which is shown in Figure 1. Note that the order of the candidates shown on any ballot form is random and unpredictable.

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

The voter now exits the booth with this receipt, registers with an official and casts their encrypted receipt in the presence of the official. The ballot receipt is placed against an optical reader or similar device that records the random value at the bottom of the strip and an index value indicating the cell into which the X was marked. The physical receipt is digitally signed and franked and this is retained by the voter retains as their receipt. The independent randomisation of the candidate list for each ballot form ensures the secrecy of the vote.

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

Once the election has closed, the digitized copies of the receipts are transmitted to a central tabulation server which posts them to a secure WBB. This is an append-only, publicly visible facility. Only the tabulation server, and later the tellers, can write to this and, once written, anything posted to it will remain unchanged. Voters can visit this WBB and confirm that their receipt appears correctly. If their receipt does not appear, or appears incorrectly (i.e., with the X in the wrong position) they can appeal. Note that, since they hold receipts, they have sound grounds for complaint if their receipts fails to appear.

After a suitable period, the tellers take over and perform an anonymising mix on the batch of posted receipts. Various mechanisms can be used to ensure that the tellers perform the decryptions correctly. These are described in section 9.

3 Paillier encryption

Paillier encryption is a randomising algorithm that is ideally suited to adapting Prêt à Voter to re-encryption mixes. Key generation proceeds as follows: firstly generate an RSA integer $n = p.q$ and compute the Carmichael function of n: $\lambda := lcm(p-1, q-1)$. Define $L(x) := (x-1)/n$. Find a generator g of $Z^*_{n^2}$ such that $g = 1 \pmod{n}$. (n, g) is published as the public key whilst λ forms the secret key.

The encryption of a message $m \in Z_n$ is computed as:

$$c = g^m r^n \pmod{n^2}$$

Where r is a freshly generated random value drawn from Z_n.

Decryption is given by:

$$m = \frac{L(c^\lambda \pmod{n^2})}{L(g^\lambda \pmod{n^2})} \pmod{n}$$

Due to the way that the payload is carried in the exponent, Paillier enjoys the homomorphic property:

$$\mathcal{E}_P(a) \otimes \mathcal{E}_P(b) = \mathcal{E}_P(a \oplus b)$$

This makes Paillier ideally suited to our construction as it allows us to absorb the index value of the receipt into the onion value.

4 Distributed generation of Paillier encrypted ballot forms

We now present a mechanism for the distributed generation of the seed values and ballot forms. Throughout, we use Paillier encryption.

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

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

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

An initial clerk C_0 generates a batch of initial seeds $\bar{s}_i^0$ drawn at random from Z_n. From these, C_0 generates a batch of pairs of "entangled" onions by encrypting each $\bar{s}_i^0$ under the Registrar key and the Teller key:

$(\{\bar{s}_i^0\}_{PK_R}, \{\bar{s}_i^0\}_{PK_T})$.

Expressed in full as Paillier encryptions these have the form:

$$\{(\alpha^{\bar{s}_i^0}.(x_i^0)^n), (\beta^{\bar{s}_i^0}.(y_i^0)^n)\}$$

for fresh random values $\bar{x}_i^0$, $\bar{y}_i^0$ drawn from Z_n.

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

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

$$\begin{array}{c}
\{(\alpha^{s_i^{j-1}}.(x_i)^n), (\beta^{s_i^{j-1}}.(y_i)^n)\} \\
\downarrow \\
\{(\alpha^{s_i^{j-1}}.\alpha^{\bar{s}_i^j}.(\bar{x}_i^j)^n.(x_i^{j-1})^n), (\beta^{s_i^{j-1}}.\beta^{\bar{s}_i^j}.(\bar{y}_i^j)^n.(y_i^{j-1})^n)\} \\
\downarrow \\
\{(\alpha^{(s_i^{j-1}+\bar{s}_i^j)}.(\bar{x}_i^j \times x_i^{j-1})^n), (\beta^{(s_i^{j-1}+\bar{s}_i^j)}.(\bar{y}_i^j \times y_i^{j-1})^n)\} \\
\downarrow \\
\{(\alpha^{s_i^j}.(x_i^j)^n), (\beta^{s_i^j}.(y_i^j)^n)\}
\end{array}$$

where

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

Having transformed each onion pair in this way, the Clerk C_j then performs a secret shuffle on the batch and outputs the result to the next Clerk, C_{j+1}.

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

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

or:

$$s_i = \sum_{j=0}^{l} \bar{s}_i^j \pmod{n}$$

$$x_i = \prod_{j=0}^{l} \bar{x}_i^j \pmod{n}$$

$$y_i = \prod_{j=0}^{l} \bar{y}_i^j \pmod{n}$$

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

For each pair, assuming correct behaviour of the clerks, the s values in the two onions will match. We'll discuss mechanisms to detect corruption of the forms in section 8. As the seed values, and hence the candidate orders, remain encrypted, none of clerks knows the final seed values and they would all have to act in collusion to determine the final seed values. These encrypted forms can now be stored and distributed in encrypted form, thus avoiding the chain of custody problems mentioned above.

5 On-demand creation of ballot forms

The above construction of the "proto-ballot" forms means that the ballot form material can be stored and distributed with the candidate orders in encrypted form. For example, forms might be pre-printed with the onion pairs printed at the bottom, one at the bottom of each column, see Figure 3.

The booth device needs to be able to decrypt the left hand onion, or to obtain a decryption by invoking a threshold set of the registrars. Thus, we could arrange for the LH onion to be encrypted under a key held by

$z6Y33p3$	$7rJ94K$

Figure 3: Prêt à Voter ballot form

the booth device. Alternatively, if vesting such keys in the booth device is considered too fragile, the booth device could communicate the onion value to an appropriate set of registrars who hold threshold keys who then return the seed value to the device. All of this would need to be done over suitably protected channels. From the seed value the device can derive the candidate order using the publicly agreed function from the seed space to the permutation space and print this onto the ballot form to give a conventional Prêt à Voter ballot form.

Note that we should prevent the booth device from learning the RH onion value, that will constitute the receipt onion. This is to avoid the device learning the association of the receipt onion and the candidate order. It may be possible to use a scratch strip mechanism to conceal the RH onion, along the lines of suggested in [8]. This would only be removed at the time of the supervised vote casting.

6 Supervised casting of a ballot

The voter now has a "conventional" Prêt à Voter style ballot form with the candidate list and the associated right hand (Teller) onion. In the booth, the voter marks her choice with an X against her candidate. The left hand strip, that carries the candidate order and the LH onion, is detached and discarded and the voter leaves the booth with the RH strip that constitutes the receipt. Note that the RH onion is automatically discarded in this process. The voter now registers with an official and casts their vote in the presence of the official exactly as described previously. Their receipt is recorded digitally as $(r, onion_R)$, where r is the index value indicating the position of the X. The receipt can be digitally signed and franked at this point to counter receipt faking attacks.

Once the election has closed, copies of the digitised receipts will be posted to the WBB exactly as before and the voters can visit this and assure themselves that their receipt has been correctly registered. In addition to this, a Verified Encrypted Paper Audit Trail mechanism could be deployed: at the time of casting, an extra paper copy of the receipt is made and retained by the returning officer for example. This can be used to independently check the correspondence with the receipts posted to the WBB, see [7].

7 Re-encryption/tabulation mixes

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

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

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

Suppose that s_i is the seed for the ith ballot and let $s_i \pmod{m}$, where m is the number of candidates, be the shift of the candidate list. We can absorb the index value r on the ballot receipt into the onion:

$\{\text{r}, (\alpha^{-s}.y^n)\} \rightarrow (\alpha^r.\alpha^{-s}.y^n) = (\alpha^{r-s}.y^n)$

Note that, for convenience, we encrypt $-s$ rather than s. This gives a pure Paillier term and the plaintext value $r-s$ which, taken modulo m, gives the voter's the original candidate choice in the base ordering. These Paillier terms can now be sent through a conventional re-encryption mix by a set of mix tellers, see for example [3]. These mix tellers do not hold any secret keys but read in a batch of Paillier terms from the WBB, re-encrypt each of them and then post the resulting terms in random order to the WBB. After an appropriate number of such anonymising re-encryption mixes, (a threshold set of) the decryption tellers take over to extract the plaintext values.

8 Auditing the Ballot Forms

The mechanisms described above allow for the distributed generation of ballot forms, just-in-time decryption of the candidate list and printing of the ballot forms. This has clear advantages in terms of removing the need to trust a single entity to keep the ballot form information secret and avoiding chain of custody issues. On the other hand, it means that we can no longer use the random pre-auditing of pre-printed ballot forms as suggested in [2]. Consequently, we must introduce alternative techniques to detect and deter any corruption or malfunction in the creation of the ballot forms in the booth.

The voter is furnished with two or perhaps more encrypted ballot forms. All of these are decrypted by the device and the candidate orders printed on them. The voter selects just one at random to cast her vote, the other will be audited and discarded. Care has to be taken to avoid introducing dangers of double voting or chain voting etc. The double sided forms of [4] provide a possible mechanism to keep a clear account of the distribution of ballot forms. Here, each side of a form carries an encrypted "proto-ballot". The LH onions on both sides are decrypted by the device in the booth and the candidate orders printed on the appropriate sides. This results in two independent ballot forms being generated, one on each side of the form. The voter selects one side to vote and the other for audit. The forms actually have a third, blank column opposing the candidate list on the other side. Thus, detaching the candidate list on one side, the side chosen for voting, detaches the blank column of the flip side, so leaving an intact form for audit.

Auditing of the ballot forms could take place at several natural points in the process: immediately at the time of casting, or just after using devices provided by independent organisations. Additionally, auditing could take place on material posted to the WBB. All the information on both sides of the receipts would be posted to the WBB. The the voted and audit sides should be posted to separate regions of the WBB in such a way as to loose any association between the two sides. The voted sides would be processed via the tabulation mixes whilst the audit sides could be verified by anyone.

A variant of the Adida/Rivest off-line auditing mechanism, [1], could be employed here: at the time of printing, audit information is also printed on the LH strip of the ballots. In the case of ElGamal or Paillier encryption this would be the randomising factors used in the encryption of the RH/receipt onion. This information would be discarded for voted sides of the receipts but preserved on the audit sides, so allowing auditing without the need to invoke the tellers. Details of this approach will be written up separately.

9 Auditing the Anonymising Mixes

In order to detect any malfunction or corruption by the mix tellers, we can again use the Partial Random Checking approach of [3]. Here the checks on audited links will be slightly different to those of Prêt à Voter 2005: rather than revealing the seed information for the layer in question, the teller is required to reveal the re-randomisation factor used to re-encrypt the link selected for audit.

Auditing of the decryption tellers is quite straightforward as we don't need any further mixing at this stage (the anonymising mixes will be enough to ensure ballot secrecy). The correctness of the decryptions can thus be directly checked by simply encrypting the final values with the public keys and checking that these agree with the initial terms.

10 Handling full permutations and STV style elections

In order to deal with full permutations of the candidate list it is not immediately clear how to generalise the approach of section 7.

One solution is simply to have one onion for each candidate position. For a single candidate selection the ballot receipt would in effect simply be the onion value against the chosen candidate. This feels rather inelegant and inefficient in that it multiplies the number of onions required.

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

11 Conclusions and Future Work

We have described the adaptation of Prêt à Voter 2006 to use Paillier encrytption in place of ElGamal. This appears to retain all the advantages of Prêt à Voter 2006 whilst leading to a more straightforward construction of the receipts compatible with the re-encryption mixes at the tabulation stage.

It remains to find a completely satisfactory way to handle full permutations of the candidate list in the re-encryption mix. In addition, we intend to explore alternatives to Partial Random Checking [3].

We also intend to explore ways to counter the possibility of colluding entities establishing a link between the receipt onion and the decrypted candidate list. in the current scheme, it is possible for the final ballot creation clerk and the booth device to collude to establish this link. Performing a final re-encryption of the receipt onion at the time of casting is a possible avenue, but introduces an extra complication to the protocol that would require careful evaluation.

12 Acknowledgements

The author would like to thank Ran Cannetti, Michael Clarkson, Bob Cunningham, Joshua Guttman, James Heather, Markus Jakobsson, Thea Peacock, Jean-Jacques Quiquater, Ron Rivest, Steve Schneider and Paul Syverson for helpful comments and discussions.

References

[1] B. Adida and R. L. Rivest. Scratch vote: Self-contained paper-based cryptographic voting. In *Workshop on Privacy in the Electronic Society, to appear*, 2006.

[2] D. Chaum, P.Y.A. Ryan, and S. Schneider. A practical, voter-verifiable election scheme. In *European Symposium on Research in Computer Security*, number 3679 in Lecture Notes in Computer Science. Springer-Verlag, 2005.

[3] M. Jakobsson, A. Juels, and Ronald Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *USENIX Security Symposium*, pages 339–353, 2002.

[4] B. Randell and P.Y.A. Ryan. Voting technologies and trust. *IEEE Security & Privacy*, 2005. To appear.

[5] P.Y.A. Ryan. A variant of the chaum voting scheme. Technical Report CS-TR-864, University of Newcastle upon Tyne, 2004.

[6] P.Y.A. Ryan. A variant of the chaum voting scheme. In *Proceedings of the Workshop on Issues in the Theory of Security*, pages 81–88. ACM, 2005.

[7] P.Y.A. Ryan. Verified encrypted paper audit trails. Technical Report (to appear), University of Newcastle upon Tyne, 2006.

[8] P.Y.A. Ryan and T. Peacock. Prêt à voter: a systems perspective. Technical Report CS-TR-929, University of Newcastle upon Tyne, 2005.

[9] P.Y.A. Ryan and S. Schneider. A practical, voter-verifiable election scheme. In *European Symposium on Research in Computer Security*, number 4189 in Lecture Notes in Computer Science. Springer-Verlag, 2006.