Prêt à Voter: All-In-One

Z. Xia, S. Schneider, J. Heather, P. Ryan, D. Lundin, R. Peel and P. Howard

June 15, 2007

Abstract

A number of voter-verifiable electronic voting schemes have been introduced in the recent decades. These schemes not only provide each voter with a receipt without the threat of coercion and ballot selling, but also the ballot tallying phase can be publicly verified. Furthermore, these schemes are robust because the power of authorities can be threshold distributed. Generally speaking, the homomorphic encryption schemes are efficient but they are unable to handle some preferential elections, such as STV elections and Condorcet elections. The mix network schemes are versatile, but they are not as efficient as the homomorphic encryption schemes in approval elections.

In this paper, we will present a new electronic voting schemes which is secure, versatile and efficient. We call our proposal scheme the Prêt à Voter: All-In-One because it is based on the re-encryption version of the Prêt à Voter scheme and inherits most of its security properties. Our scheme not only handles both approval elections and preferential elections, but also the ballot tallying phase will always be the most efficient because according to different elections, different tally strategies can be applied.

1 Introduction

Electronic voting has attracted a lot of interests in the recent decades, and a number of e-voting schemes have been introduced. Compared to traditional election methods, electronic voting is not only more efficient but also more secure. Using cryptographic techniques, the security of e-voting schemes is mainly depending on hard cryptographic problems instead of election authorities or equipment used in the election. Generally speaking, e-voting schemes should achieve the following security properties:

- **Integrity:** only eligible voters are accepted by authorities and each authorised voter can cast one and only one ballot. Only valid ballots are included for tabulation.
- **Privacy:** voter's choice information will be kept secure if voter does not want to reveal it.
- **Coercion-free:** voters have no way to prove others how they have voted. This prevents voter coercion, intimidation and ballot selling.
- **Individual verifiability:** voters be able to verify that their votes have been correctly recorded. This is normally achieved by providing each voter with a receipt.
- **Public verifiability:** any interested party can verify that the final result is correctly tallied from the received votes.
- **Robustness:** the election system can tolerate some mistakes made by participant parties without affecting the correct performance.
- **Reliability:** the election system can recover from attempted cheating.

However, majority of the existing schemes are only designed for approval elections, e.g. First-Past-The-Post (FPTP) elections. Generally speaking, only mix network schemes can be implemented in some preferential elections, such as Single Transferable Voting (STV) elections and Condorcet elections. But these schemes are not as efficient as the homomorphic encryption schemes in FPTP elections.

In this paper, we will introduce a secure, versatile and efficient electronic voting scheme which can be considered as the combination of the homomorphic encryption scheme and the mix network scheme. Our proposal scheme achieves most of the security properties. Furthermore, it handles both approval elections and preferential elections, and the ballot tallying phase will always be the most efficient because according to different elections, different tally strategies can be applied.

We will first introduce how our proposal scheme can be implemented in STV elections, then how can it be implemented in FPTP elections, Borda Count elections and Condorcet elections will be briefly explained.

2 Issues about STV elections

STV elections have been used in many different local, regional, and national electoral systems around the world. For example, in 2006, they were used for elections in Australia, the Republic of Ireland, Northern Ireland (except elections to the British House of Commons) and Malta. They are also used for some local government elections in New Zealand. However, in the cryptographic literature, although a number of voter-verifiable voting schemes have been introduced over the last few years, very few of them can be implemented in STV elections.

In STV elections, voters need to indicate not just a single preferred candidate but a preference ranking of a partial or all of the candidates on the ballot. Therefore, in order to maintain the privacy and coercion-free properties, the order of the candidate list on the ballot has to be totally random instead of just cyclic shifted. Otherwise, if adversaries know which candidate is most preferred by a particular voter, they might find out the rest choices of this voter, and a similar problem will occur if the adversaries know which candidate is least preferred by this voter.

When tallying the ballots, in the first round, only the first preference of each vote is evaluated. If the candidate achieves the winning quota, the election ends and this candidate wins. Otherwise, the following processes will be repeated until some candidate achieves the winning quota: the candidate with the least votes will be eliminated, all votes for this candidate will be transferred to other candidates according to the next preference. Because of this, each ballot has to be tallied as a whole.

Majority e-voting schemes can be classified into three categories: *blind signature schemes*, *homomorphic encryption schemes* and *mix network schemes*. However, we believe only mix network schemes can handle STV elections because:

- Blind signature schemes are not practical in real use because of two problems: one is that the number of received votes may be less than the number of listed eligible registered voters. But it would be difficult to decide whether this difference comes from voters who decide not to cast their votes, or because some valid votes have been removed by fraudulent authorities. The other problem is that faulty authorities can cast votes for abstaining voters. Although this cheating can be detected, the fraudulent votes cannot be removed because it is infeasible to distinguish such fraudulent votes from valid ones.

- Homomorphic encryption schemes are very efficient in the ballot tallying phase. However, all received votes are aggregated before decryption. If no candidate achieves the winning quota in the first round, it is infeasible to transfer the votes of the candidate with the least votes to other candidates. Therefore, we assume that in e-voting schemes, if the ballot tallying phase is only implemented by applying the homomorphic property, they are impractical for STV elections.

3 Introduction of the Prêt à Voter schemes

In [2, 10], Ryan et al. have introduced two versions of the Prêt à Voter schemes which can be implemented in STV elections. We now present a brief overview of these Prêt à Voter schemes. For full details, see [2, 10]. In this paper, we will denote the Prêt à Voter with the decryption mix network as PAV 2005, and the other one with re-encryption mix network as PAV 2006.

In both Prêt à Voter schemes, the whole protocol can be divided into three phases: *ballot construction phase*, *ballot casting phase*, and *ballot tallying phase*. Although the two schemes are different in the ballot construction phase and ballot tallying phase, their ballot casting phases are very similar from the voter's point of view.

In both Prêt à Voter schemes, an authenticated voter in the voting booth will be provided with a ballot form as shown in Figure 1.

Bob	
Crystal	
David	
Alice	
	7q3Kyr

Figure 1: A Prêt à Voter ballot form

A ballot form consists of two columns. The left hand column lists the candidate names, and the right hand column is blank for a voter to cast her vote. At the bottom of the right hand column, there is an encrypted value, called an *Onion*, which can be used to reconstruct the order of the candidate list in the left hand column if properly decrypted. The candidate list in the left hand column is in some canonical order but are randomly cyclicly shifted and varies between different ballot forms (Note that in PAV 2005 [2], the candidate list can be made totally random). Therefore, it is infeasible to predict the order of the candidate list.

When receiving such a blank ballot form, a voter makes a mark against her preferred candidate and then tears the ballot form apart along some perforation between the two columns. After that, the left hand column has to be destroyed, and the right hand column can be kept as the receipt after being scanned by the election authorities. All scanned information will be published onto the bulletin board. Therefore any voter can verify whether her vote has been correctly recorded by checking whether her receipt is correctly displayed on the bulletin board.

An attractive property of the Prêt à Voter schemes is that, even by providing the receipt, without properly decrypting the *Onion*, nobody except the voter herself can know the content of the vote, thereby preventing vote coercion, intimidation, and ballot selling.

3.1 PAV 2005

In [2], all ballot forms are generated by administrators in advance. In order to audit the ballot tallying phase with *Randomized Partial Checking* (RPC) [3], if there are k tellers (who implement the mix network), each *Onion* contains $2k$ layers. For each ballot, $2k$ germ values $(g_0, g_1, \ldots, g_{2k-1})$ are randomly selected, and then the *Onion* is encrypted using each teller's public key iteratively as $D_{i+1} = \{g_i \,||\, D_i\}_{PK_i}$, in which g_i contributes to d_i times cyclic shift to the candidate list as $d_i = hash(g_i) \pmod{v}$. Finally, D_{2k} will represent the *Onion*, and the total cyclic shift θ that will be applied to the candidate list on this ballot form will be computed as $\theta = \sum_{i=1}^{2k-1} d_i \pmod{v}$.

In the ballot tallying phase, all received votes are decrypted in batches by these k tellers. Suppose the input list to the $Teller_i$ is $(r_{1,i+1}, D_{1,i+1}), \ldots, (r_{n,i+1}, D_{n,i+1})$, for each value in the input list, $Teller_i$ will behave as follows:

1. Let j denote the ballot serial number, for $j \in \{1, \ldots, n\}$, $Teller_i$ will first decrypt $D_{j,i+1}$ using her private key SK_i as $\{D_{j,i+1}\}_{SK_i} = g_{j,i} \,||\, D_{j,i}$.
2. Then $Teller_i$ applies the hash function to the germ value as $d_{j,i} = hash(g_{j,i}) \pmod v$, and calculates the cyclic shift as $r_{j,i} = r_{j,i+1} - d_{j,i+1} \pmod v$. Here v denotes the number of candidates.

In order to break the links between inputs and outputs, $Teller_i$ permutes the output list by applying a random permutation π, and then outputs the result $(r_{\pi(1),i}, D_{\pi(1),i}), \ldots, (r_{\pi(n),i}, D_{\pi(n),i})$ to the next teller through the bulletin board. The following tellers will perform the same processes. Finally, the result will be revealed by the last teller.

3.2 PAV 2006

In [10], Ryan et al. have introduced another version of the Prêt à Voter scheme, which has improved PAV 2005 in the following aspects:

- All ballot forms are generated in a distributed fashion. Therefore, these ballot forms can be kept secret before use unless all authorities in charge of the ballot construction phase collude.
- Ballots are printed on demand when voters cast their votes. Therefore, the chain voting attack (see [7]) will not be a problem.
- The size of the *Onion* is not proportional to the number of tellers.
- The re-encryption mix network in PAV 2006 is more robust and versatile than the decryption mix network in PAV 2005, because the shuffle phase and the decryption phase are separated.

In PAV 2006, the ballot form is slight different, each ballot form contains two *Onions*, one at the bottom of each column. The left *Onion*, which is encrypted under the public key of the voting machine, is decrypted when voters cast their votes, and the right *Onion*, which is encrypted under the public key of the decryption authorities, is only decrypted in the ballot tallying phase.

When constructing ballots, firstly, the voting machine (operated by threshold parties) and the threshold decryption authorities publish their public keys (α, β_T) and (α, β_R) respectively. Then the election authorities randomly select γ and publish it. As follows, a number of clerks will generate the ballots in a distributed fashion as:

1. Suppose p and q are large primes, where $p = 2q+1$. The first clerk C_1 selects $x_1, y_1, s_1 \in_R Z_p^*$ and calculates D_1 as
$$D_1 = (\alpha^{x_1}, \beta_R^{x_1} \cdot \gamma^{-s_1}), (\alpha^{y_1}, \beta_T^{y_1} \cdot \gamma^{-s_1})$$
2. For $i = 2, 3, 4, \ldots, n$, the clerk C_i first selects $\bar{x}_i, \bar{y}_i, \bar{s}_i \in_R Z_p^*$, and then calculates D_i as
$$\begin{aligned} \bar{D}_i &= (\alpha^{\bar{x}_i}, \beta_R^{\bar{x}_i} \cdot \gamma^{-\bar{s}_i}), (\alpha^{\bar{y}_i}, \beta_T^{\bar{y}_i} \cdot \gamma^{-\bar{s}_i}) \\ D_i &= \bar{D}_i \cdot D_{i-1} = (\alpha^{x_i}, \beta_R^{x_i} \cdot \gamma^{-s_i}), (\alpha^{y_i}, \beta_T^{y_i} \cdot \gamma^{-s_i}) \end{aligned}$$
where
$$\begin{aligned} x_i &= x_{i-1} + \bar{x}_i \\ y_i &= y_{i-1} + \bar{y}_i \\ s_i &= s_{i-1} + \bar{s}_i \end{aligned}$$
3. Finally, the last clerk C_n will output the result as:
$$\begin{aligned} Onion_L &= (\alpha^{x_n}, \beta_R^{x_n} \cdot \gamma^{-s_n}) \\ Onion_R &= (\alpha^{y_n}, \beta_T^{y_n} \cdot \gamma^{-s_n}) \end{aligned}$$

In the ballot tallying phase, the received votes are first shuffled by a re-encryption mix network, e.g. Neff's mix [5], in order to break the voter-vote links. Then the decryption authorities will decrypt each vote in a threshold fashion and count the final result.

3.3 Prêt à Voter schemes in STV elections

In STV elections, each voter has to make a full or partial ranking of the candidates instead of ticking a single preferred candidate. Therefore, the order of the candidate list has to be totally random instead of just cyclic shifted. And in the ballot tallying phase, the voter's choice indexes have to be changed through the mixes. Because of this requirement, the suggested method in [2, 10] of implementating the Prêt à Voter schemes in STV elections becomes quite complicated, compared to handle FPTP elections.

4 Prêt à Voter for STV Elections

In this section, we will introduce our proposal scheme to efficiently handle STV elections.

4.1 Election parameters

We suppose that in our case, there are $v = 6$ candidates, Alice, Bob, Crystal, David, Elaine and Frank with indexes 1, 2, 3, 4, 5, 6 respectively. We assign each candidate a unique value $M^1, M^2, \ldots, M^6$ according to their index number, where $M > v$. Here, we simply set $M = 7$. (α, β_T) and (α, β_R) are the public keys of the voting machine (operated by threshold parties) and decryption parties respectively.

4.2 Ballot construction

Denote $c = E_{pk}(m, r)$ represent the Paillier encryption [6], therefore $c = g^m r^n \pmod{n^2}$. Similar to PAV 2006, the ballot construction phase is implemented by a number of clerks. The first clerk randomly decides an order of the candidate list, e.g. "Bob, Frank, Crystal, Elaine, David, Alice", and by index as "2,6,3,5,4,1". She then generates a blank ballot as shown in Figure 2.

	☐ $E_{pk\_R}(7^2, r_1)$
	☐ $E_{pk\_R}(7^6, r_2)$
	☐ $E_{pk\_R}(7^3, r_3)$
	☐ $E_{pk\_R}(7^5, r_4)$
	☐ $E_{pk\_R}(7^4, r_5)$
	☐ $E_{pk\_R}(7^1, r_6)$
E_{pk_T} (263541, r)	

Figure 2: An unfinished STV ballot form

The following clerks just randomly re-encrypt each *Onion* in the ballot form as $c' = c \times t^n = E_{pk}(m, r')$, where $r' = r \cdot t$, so that the plaintext corresponding to the candidate does not change. After the ballot form is re-encrypted by the following clerks, the first clerk cannot trace the ballot any more. And the following clerks do not know the order of the candidate list. Therefore, the privacy of the ballot forms will be maintained unless all clerks collude.

Note that in this phase, the first clerk and the last clerk may cheat. The first clerk can deliberately set all candidate list just cyclic shifted or some certain candidate always at the bottom. One solution is that the first clerk commits to all the ballots she has generated, and then some trusted third parties randomly select half of these ballots and ask the first clerk to reveal how she has generated them. If no cheating is detected, the remaining ballots can be passed to the following

clerks. Also, if the last clerk collude with the voting machine, voter privacy will be violated. A possible solution to this problem might be that the voting machine is operated by a group of threshold parties, and each of these parties can communicate with the voter through untappable channel. When read the *Onion* on the left hand column, each party decrypts it using her share of the private key and sends the result to the voter. When receives enough shares of these results, the voter can calculate the order of the candidate list by herself. Therefore, no body else except the voter knows the order of the candidate list unless more than half of these parties collude.

4.3 Auditing the ballot construction

We advocate to use the strategy introduced by Ryan et al. [8] to audit the ballot construction phase, that each voter is provided with two ballot forms. Every voter can randomly choose one for auditing and the other one to cast her vote. Furthermore, some third parties are allowed to audit the original ballot forms as well.

4.4 Ballot casting

The process of casting a vote in our proposal scheme is similar to PAV 2006, when a voter receives a blank ballot form similar as in Figure 2, she first tears it apart, inserts the left hand column into a voting machine which reads the *Onion* at the bottom, and then the voting machine prints the candidate names in the left hand column. As follows, when this voter aligns the two columns, the ballot form will look as in Figure 3.

Bob	☐ $E_{pk\_R}(7^2, r_1')$
Frank	☐ $E_{pk\_R}(7^6, r_2')$
Crystal	☐ $E_{pk\_R}(7^3, r_4')$
Elaine	☐ $E_{pk\_R}(7^5, r_5')$
David	☐ $E_{pk\_R}(7^4, r_6')$
Alice	☐ $E_{pk\_R}(7^1, r_7')$
$E_{pk\_T}(2673541, r')$	

Figure 3: A standard STV ballot form

Then this voter casts her vote by ranking in the right hand column. After that, the left hand column has to be destroyed and the right hand column can be kept as the receipt after being scanned by the election authorities.

4.5 Ballot tallying

The ballot tallying phase of our proposal scheme, as shown in Figure 4, is divided into three steps:

1. *Ballot Transferring:* Paillier encryption [6] enjoys the following homomorphic properties:

$$\begin{aligned} E_{pk}(m_i, r_i) \times E_{pk}(m_j, r_j) &= E_{pk}(m_i + m_j, r_i \cdot r_j) \\ (E_{pk}(m_i, r_i))^k &= E_{pk}(k \cdot m_i, {r_i}^k) \end{aligned}$$

A ranked ballot can be transferred into a single *Onion* as

$$E_{pk}(m_k, r_k) = \prod_{i=1}^{v} (E_{pk}(m_i, r_i))^{k_i}$$

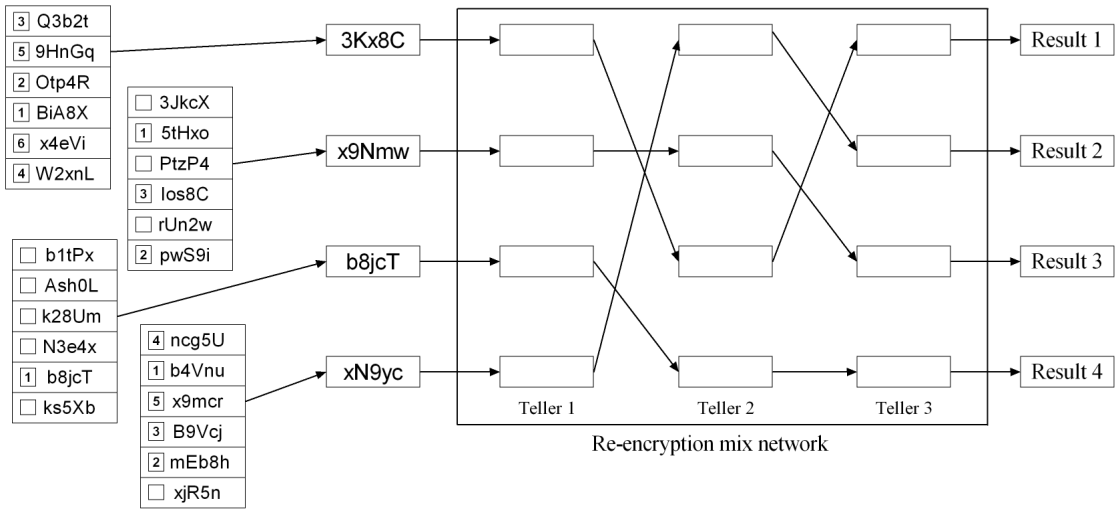

Figure 4: The ballot tallying phase

where k_i represents voter's choice index, $m_k = \sum_{i=1}^{v} m_i \cdot k_i$ and $r_k = \prod_{i=1}^{v} {r_i}^{k_i}$. Here, m_k has absorbed all choices for the corresponding candidates, and the value of r_k does not affect the result after decryption. For example, in this step, when receives a vote as shown in Figure 5, the election authorities will transfer it into an encrypted information $E_{pk}(m_k, r_k)$, where $m_k = 3 \times 7^2 + 1 \times 7^6 + \ldots + 4 \times 7^1$.

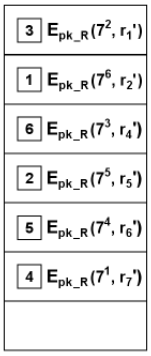

Figure 5: An example vote

2. *Ballot Shuffling:* the ciphertext in the previous step is then shuffled by a re-encryption mix network, in which each teller receives a batch of votes $(c_1, c_2, \ldots, c_n)$. For each vote, the teller re-encrypts it as ${c_i}' = c_i \times {t_i}^n$, and then outputs the result batch $({c_{\pi(1)}}', {c_{\pi(2)}}', \ldots, {c_{\pi(n)}}')$ to the next teller through the bulletin board. This phase can be audited using the *Randomized Partial Checking* (RPC) [3].

3. *Ballot Decryption:* all shuffled votes are decrypted by the threshold decryption authorities, then the voters' choices are separated and the final result is counted. Because $M > v$, the choice for each candidate can easily be separated. For example, in our case in Figure 5, the plaintext m_k will be retrieved after the corresponding vote has been decrypted by the threshold decryption authorities, where $m_k = 3 \times 7^2 + 1 \times 7^6 + \ldots + 4 \times 7^1$. Then m_k div 7^6

can retrieve the choice for Frank. The remainder div 7^5 can retrieve the choice for Elaine, and so on.

4.6 How to allow partial ranking and spoiling ballots

However, our proposal scheme introduced above, as well as PAV 2005 and PAV 2006 are all suffering two problems:

- If some voters do not rank all the way down the ballot form, these voter's privacy will be violated in some extent. In our proposal scheme, although adversaries cannot trace such ballots through the re-encryption mix network, they might link the final result directly with the receipts.
- In some situations, some voters want to spoil their ballots if they do not prefer any of the candidate. A satisfactory e-voting system should allow voters to express their right in this way. However, in majority of existing e-voting systems, voters can only spoil their ballots without casting their votes. But this will give some information to adversaries that who has cast her vote and who has not cast her vote.

In order to solve the above two problems, we have made some additional improvements to our proposal scheme. A modified ballot form is shown as in Figure 6

Bob	☐ $E_{pk\_R}(8^2, r_1')$
Frank	☐ $E_{pk\_R}(8^6, r_2')$
Spoil	☐ $E_{pk\_R}(8^7, r_3')$
Crystal	☐ $E_{pk\_R}(8^3, r_4')$
Elaine	☐ $E_{pk\_R}(8^5, r_5')$
David	☐ $E_{pk\_R}(8^4, r_6')$
Alice	☐ $E_{pk\_R}(8^1, r_7')$
$E_{pk\_T}(2673541, r')$	

Figure 6: A modified STV ballot form

The base order M has been increased to 8 and an additional choice *Spoil*, with the index 7 and unique value M^7 is randomly inserted into the candidate list. In the voting booth, an authenticated voter will be provided with a ballot as shown in Figure 6. Suppose this voter wants to vote David as her first choice, Elaine as her second choice and she does not want to vote for other candidates. She can then mark the first choice against David and mark the second choice against Elaine. Then she has to randomly rank all other candidates after marking the third choice against *Spoil*. This helps the receipt to record which are the voter's genuine choices and which choices are randomly appended. As a result, the adversaries have no way to know how many candidates are selected by this voter.

In the ballot tallying phase, each ballot is transferred into a single *Onion* at first. Then followed by shuffling and threshold decryption, the result is separated. As follows, the randomly appended candidates are removed and only the genuine candidates selected by this voter are remained in the final result. Such a result is shown in Figure 7.

If a voter wants to spoil her ballot, she can mark the first choice against *spoil* and then randomly ranks all other candidates. When this vote is decrypted, it is publicly verifiable that it is a spoilt vote. In our proposal scheme, the voter spoiling her ballot is provided with a receipt as well. This gives several advantages:

Alice	
Bob	
Crystal	
David	1
Elaine	2
Frank	

Figure 7: A result of ballot form

- The receipt does not give adversaries any clue whether this voter has voted for some candidates or has spoilt her ballot.
- Although the spoilt ballots do not affect the final result, they play a similar role as dummy votes.
- Although a certain voter has not voted for any candidate, she can use the receipt to audit the election process, and accuse if any cheating is detected. The receipt can make voters more confident about the system.
- If some voters vote for spoilt ballot, they are not allowed to cast another vote. This prevent fraudulent election authorities cast votes for voters who want to spoil their ballots.

4.7 Discussion of our proposal scheme

In our proposal scheme, all steps in the ballot tallying phase are recorded on the bulletin board. The ballot transferring step is publicly verifiable because any party with the mathematical knowledge can redo this step and compare the result. The shuffle step can be audited by public because we apply it using some existing publicly verifiable mix networks. The decryption step is done in a threshold fashion.

Our proposal scheme introduced above maintains all properties of PAV 2006. Tallying STV ballots in our scheme nearly has the same computational complexity as tallying FPTP in PAV 2006. Besides, our scheme solves the problems of partial ranking and spoiling ballots. It is infeasible for adversaries to trace the voter-vote links even if some voters spoil their votes or do not give their preference to all the candidates. Furthermore, we feel our scheme is very elegant, therefore easy to be understand and decrease the error possibility in real implementation.

5 Prêt à Voter: All-In-One

The requirement of condorcet elections is similar as STV elections, that each ballot need to be tallied as a whole. The proposed scheme introduced above can directly handle condorcet elections. Furthermore, the proposed scheme can easily be improved to handle other election methods, such as FPTP elections and Borda Count elections.

Denote N to represent the maximum number of voters. To handle FPTP elections, we just need to raise the base order M such that $M > N$, then the ballot tallying phase can only be implemented applying the homomorphic property, no mix network is needed. This is exactly the same as the Scratch & Vote [1], which could be considered as a special case in our proposal scheme.

Similarly, to handle Borda Count elections, the base order M just need to achieve the requirement that $M > N \cdot v$, where v is the number of candidates. In the ballot tallying phase, voter's

ranking preference will be firstly transferred to points. Then all points are aggregated by applying the homomorphic property, and finally, the result can be decrypted in a threshold fashion.

The original Prêt à Voter schemes can handle all these election methods as well, but Prêt à Voter: All-In-One is much more efficient and versatile than the original schemes because according to different election methods, different tallying strategies can be applied.

6 Conclusion and discussion

We have introduced an improvement of the Prêt à Voter schemes which is secure, versatile and efficient. We have done some system perspectives to the Prêt à Voter: All-In-One similar as in [4, 7, 9], but because of space limitations, we omit the details here. Our proposal scheme needs three compulsory assumptions: first, the bulletin board has to be secure and reliable; second, the voters cast their votes in a secure place; third, the election authorities who sign voter's receipt have to be honest or voters are able to verify the signatures by themselves. Generally speaking, the Prêt à Voter: All-In-One scheme has achieves most of the e-voting desirable security properties and it is immune to most of the known attacks except the Italian attacks.

We would like to appreciate the anonymous reviewers who have given their comments and criticism on this work.

References

[1] B. Adida and R. Rivest. Scratch & Vote: self-contained paper-based cryptographic voting. *Proceedings of the 5th ACM workshop on Privacy in Electronic Society*, pages 29–40, 2006.

[2] D. Chaum, P. Ryan, and S. Schneider. A practical voter-verifiable election scheme. *Technical Report of University of Newcastle*, CS-TR:880, 2005.

[3] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, 2002.

[4] C. Karlof, N. Sastry, and D. Wagner. Cryptographic voting protocols: a systems perspective. *Proceeding of USENIX Security Symposium*, pages 186–200, 2005. LNCS 3444.

[5] C. A. Neff. A verifiable secret shuffle and its application to e-voting. *Proceedings of the 8th ACM conference on Computer and Communications Security (CSS'01)*, pages 116–125, 2001.

[6] P. Paillier. Public-key cryptosystems based on discrete logarithms residues. *Advances of Eurocrypt'99*, pages 223–238, 1999. LNCS 1592.

[7] P. Ryan and T. Peacock. Prêt à Voter: a system perspective. *Technical Report of University of Newcastle*, CS-TR:929, 2005.

[8] P. Ryan and T. Peacock. Putting the human back in voting protocols. *Technical Report of University of Newcastle*, CS-TR:972, 2006.

[9] P. Ryan and T. Peacock. Threat analysis of cryptographic election schemes. *Technical Report of University of Newcastle*, CS-TR:971, 2006.

[10] P. Ryan and S. Schneider. Prêt à Voter with re-encryption mixes. *Technical Report of University of Newcastle*, CS-TR:956, 2006.