Cryptographic Voting Protocols: A short Overwiev

Björn Breitmeyer
(bjbreitm@cs.tu-berlin.de)

Seminar "Internet Security" ,
Technische Universität Berlin

SS 2008 (Version from June 6, 2008)

Abstract

Cryptographic voting systems are trying to ensure privacy, security and verifiability to the electronic voting systems. They try to guarantee these traits independent of the used software. As those protocols rely on another set of hardware, the hole system needs to be evaluated. In this paper i will discuss the schemes of Andrew Neff and David Chaum based on the paper from Karlof, Sastry and Wagner [2]. It also contains a small motivation why the DRE systems are not sufficient and alternatives like cryptographic voting protocols are needed. Several weaknesses have been discovered in the analysis, most of them hard to solve. They are all induced by the underspecification of the protocols, as they have not described how the most vital parts of the protocols should be implemented, they are just described as black boxes. Karlof has hopes for the voting protocols if they specify the rest of the system, even though he does not propose the use already. In contrast to them, i highly doubt they will be capable reaching their goal of becoming a secure, software independent system. With new specifications, new problems will occur and all proposes of Karlof induced new weaknesses or a rollback to the paper based system. So in my opinion cryptographic voting protocols were aiming for a good but kind of unreachable target and there has to be found a way to trust at least one vital part of an election to ensure security for the hole election.

1 Introduction

Voting is a crucial element in every Democracy. An election has to fulfill a certain number of conditions. Those are the verifiability of an election, voting is done private and coercion resistance.

Paperless "Direct Recording Electronic" voting machines(DRE, see figure1 as a sample) do not fulfill all of these conditions. The Voter has no way to get a proof that his vote was cast as intended and that a corrupted DRE does not undermine the privacy of the election. The DRE's, working without voting protocols, rely on security through obfuscation, that this is not a good idea, has been proofed by[3] Khono and co. Also there are several other cases where the obfuscation could be broken and the system was not secure anymore, since the underlying crypto system had flaws, due to the fact it had not been evaluated by the majority of experts. Crypto systems normally have to be evaluated in public for a reason, the system has to be secure even though the principle of the system

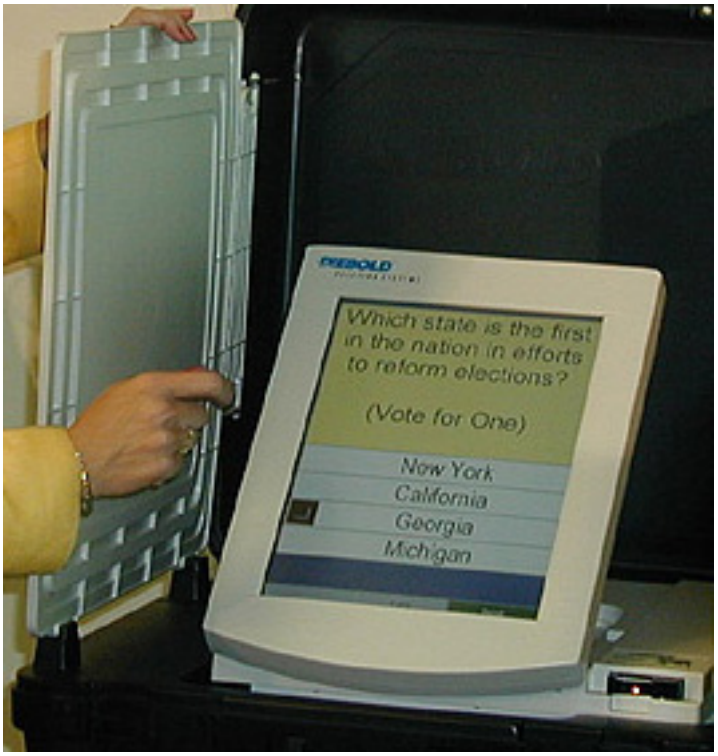

Figure 1: This is a sample DRE from accuVote, picture taken from [3]

is well known. Some of the several weaknesses in DRE voting systems are the smart cards voter's use to identify them self to the DRE, as the difference between a normal and and administration card is very low. Also the check if voting is still allowed with this smart card depends on data at the smart card. Its just enough if the smart card reject the reset of this data, so you can vote as much as you want[3]. This are just the major flaws of a very often used DRE system, but it already implies the great need for a system that does not rely at the underlying hardware.

As the DRE's purpose was a paperless election without the necessity of manual counting, so a DRE which prints ballots as a proof and backup system is not the solution. Cryptographic Voting Protocols may be a solution to that problem.

Their goal is to ensure voter privacy, verifiability and protection against other cheating attempts independent from the underlying hardware/software. So they have to make sure the protocol reveals cheating attempts to the election officials and the voters at a very high probability. In this paper we will discuss the schemes of David Chaum and Andrew Neff, they are the first protocols which allow the voter to verify his vote was cast and count as intended. Due to the fact that both combined cryptographic subsystems that are secure, it's needed to focus on the interactions of these systems to ensure the security promises the protocols give.

Karlof Sastry and Wagner[2] have found several weaknesses in both protocols, which are only apparent due to the fact that both systems are underspecified. Due to those problems the Protocols are not yet to be considered for use in voting systems, even though some companies are implementing them already. The results of the analysis from Karlof, Sastry and Wagner are summarized in Table1. We will discuss the human factors, social engineering- and DOS attacks in detail.

Weakness	Protocols	Thread Model	Affects
Random subliminal channels	Neff	Malicious DRE colluding with outsider	Voter privacy, coercion resistance
Semantic subliminal channels	Chaum	Malicious DRE colluding with outsider	Voter privacy, coercion resistance
Message reordering attacks	Neff	Malicious DRE and human error	Election integrity, public verifiability
Social engineering attacks	Neff, Chaum	Malicious DRE and human error	Election integrity, public verifiability
Discarded receipts	Neff, Chaum	Malicious DRE or bulletin board	Election integrity
Other human factor attacks	Neff, Chaum	Malicious DRE	Ability of voter to prove DRE is cheating
Denial of service attacks	Neff, Chaum	Malicious DRE or tallying software	Voter confidence, election integrity

Table 1: Table 1: Summary of weaknesses Karlof, Sastry and Wagner found in Neff's and Chaum's voting schemes

2 Preliminaries

Since billions of dollars are at stake in national elections it has to be assumed that groups with an interest to manipulate the election have nearly no financial limit. So the protocols have to assure that there is no way to manipulate the election without being caught as long as not all elections officials cooperate. Attacks where the coercer has to be physically present are not treated, as they can not be countered with voting protocols. But malicious DRE's, cooperating elections officials and attacks on the pool of already casted ballots are treated. Especially manipulated DRE's and security through obfuscation are the Problem of active DRE's[3].So the voting Protocols have the following security goals.

2.1 Security Goals[2]

- **Cast-as-intended:** A voters ballot should represent his choices.
- **Counted-as-cast:** The final tally should be an accurate cast of the created ballots.
- **Verifiabilty:** The voter should be able to verify his vote was cast as intended and everyone should be able to verify the final tally is an accurate representation of the casted votes.
- **One voter/one vote:** It has to be ensured that everyone has just one vote and that nobody is capable of deleting, changing or adding votes.
- **Coercion resistance:** A voter should not be able to prove how he voted to others.
- **Privacy:** Ballots should be secret to anyone.

2.2 Common principles of the protocols

Both protocols use a four staged sequence to achieve these goals. *election initialization*, *ballot preparation*, *ballot tabulation*and *election verification*. Before the elections starts, a set of *election trustees* with competing interests is chosen. Ballot preparation starts with the voter casting his vote at a DRE, which generates an electronic ballot and posts it to a public bulletin board. At the same time the DRE provides a receipt to the voter, enabling

the voter to verify his vote was cast as intended, and he can verify his vote was published encrypted on the public bulletin board. Receipts are designed to resist vote buying and coercion. Also each voters ballot is assigned a unique ballot sequence number (BSN).

After all ballots have been posted to the public bulletin board, the ballot tabulation begins. Now the trustees execute a multistage mixnet[1] in which the ballots are anonymously decrypted. Before ballots enter the mixnet the BSN is stripped from it. The final result are the plaintext ballots, ready for counting. In electronic voting protocols the mixnet is universally verifiable, which means the trustee's provide a proof that they correctly executed their part of the mixnet. At various points during that process, voters and other election participants can verify that their votes are cast as intended that the mixnet is executed correctly. Both schemes rely on printers to create the receipts for the voters. These printers should not be able to make changes to the receipts without the voter being capable of detecting it.

2.3 Short introduction to mixnets[1]

This section can be savely skipped by readers that have already a good guess what mixnets are.

A mixnets goal is to decrypt cyphertexts in some steps and provide anonymity. A further goal for use in voting protocols is the verifiability that the mixnet was executed correctly without loosing the anonymity. As already introduced each trustee generates a key, then those keys are combined and used for encryption of the ballots. So in each step a trustee decrypts the ballots with his key and hands the results to the next trustee in a permuted order. So the principle can be described as opening envelopes shuffling them and then give them to the next one who can open the next set of envelopes until you finally get the letter. At the end all ballots are plaintext and there is no verifiable relation between input order and output order. The issues with proofing the correctness of the mixnet were solved the following way. Each trustee server will reveal a random input output pair to proof he has executed the net correctly. Due to the randomness which input output pairs are revealed, there is a very low probability that you can reconstruct the initial order for even a single ballot.

3 The two Protocols

Here we will discuss both schemes in detail.

3.1 Neff's Scheme

The election starts with the trustees perform a distributed key generation protocol to get a public master key. The decryption is only possible if all trustees cooperate. There is a security parameter l, it describes the probability of a Cheating DRE not to get caught with the probability of 2^{-l}. Neff suggests $10 \leq l \leq 15$ even though it may be obvious to choose a very large l,this inflicts the time needed for decryption and the memory needed for storage to grow exponential. When the voter cast his vote, an encrypted electronic ballot, with a unique BSN, is generated and an according receipt is given to the voter. This receipt enables the user to make sure his vote was represented correctly with a high probability. The Scheme works for elections with multiple races, but simplicity only a one race election is used for explanation.

The Candidates are represented by $C_1, C_2, ..., C_n$. After the voter made his choice the DRE generates an verifiable Choice(VC). A VC is an encrypted ballot representing the voters choice(see table2). A VC is a $n \times l$ matrix of ballot mark pairs(BMP's), one row per candidate(recall that l determines the cheating detection probability). Each BMP

	1	2	3	$\cdots$	l
C_1	01	10	01	$\cdots$	01
C_2	10	01	10	$\cdots$	10
C_3	11	00	11	$\cdots$	11
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
C_n	10	01	10	$\cdots$	10

Table 2: A VC in Neff's Scheme, for Candidate C_3

	1	2	3	$\cdots$	l
C_1	01	10	01	$\cdots$	01
C_2	10	01	10	$\cdots$	10
C_3	$\underline{1}\mathbf{1}$	$\mathbf{0}\underline{0}$	$\mathbf{1}\underline{1}$	$\cdots$	$\underline{1}\mathbf{1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
C_n	10	01	10	$\cdots$	10

Table 3: A VC in Neff's Scheme, for Candidate C_3, with pledgebit **p** and the users choice $\underline{p}$

is a pair of El Gamal ciphertexts. Each ciphertext is an encryption of 0 or 1 with the public master key. In the plaintext the chosen row has columns with a 11 or 00 in it, the unchosen rows carry either 01 or 10 entries.

Any other configuration is an indication for a cheating DRE. Now the pair (BSN, hash(VC)) is printed on the receipt, allowing to the voter to verify his vote was correctly published on the public bulletin board. After voting the voter shall verify the DRE hast constructed a VC representing his choice. To do this the DRE selects a pledge bit p for each BMP and then ask's the voter to choose between the left and right bit, which will then be decrypted(see table3). As there are l entries for each row(The voters choice is called challenge) and the chance of being detected in each column is $\frac{1}{2}$, so the chance for the DRE to get away with a undetected cheating attempt is indeed $\left(\frac{1}{2}\right)^l$. In fact an average voter won't be able to take this process unassisted, so the DRE let's the user do the process and then print's the result(VC + challenge) on the receipt, so the voter can later verify that his vote was cast as intended, with a trusted software. The DRE can not cheat by revealing the entries, as the VC is part of the OVC which is published at the bulletin board. That way the voter can ensure the same entry is on the bulletin board and theoretically the plaintext ballot can be checked against the OVC to ensure the DRE did not cheated. Even though this check is not explicitly mentioned at the scheme. After that the DRE constructs an open verifiable choice(OVC), using the voters opened values and opening random other bits in the other rows (can also be chosen by the voter), so that the choice of the voter is not revealed in public. This OVC is then printed on the public bulletin board, along with the hash of the VC, so the voter can verify, that the hash on his receipt is present on the bulletin board.

3.2 Chaum's Scheme

Chaum relies on visual cryptography, his receipt is a two layer transparency. With both transparencies together the choice of the voter is revealed(see figure2). In the transparency the keys of the trustees are used to be capable to encrypt the ballot with just one layer. So from one transparency alone, no information can be taken, unless the keys of all trustees

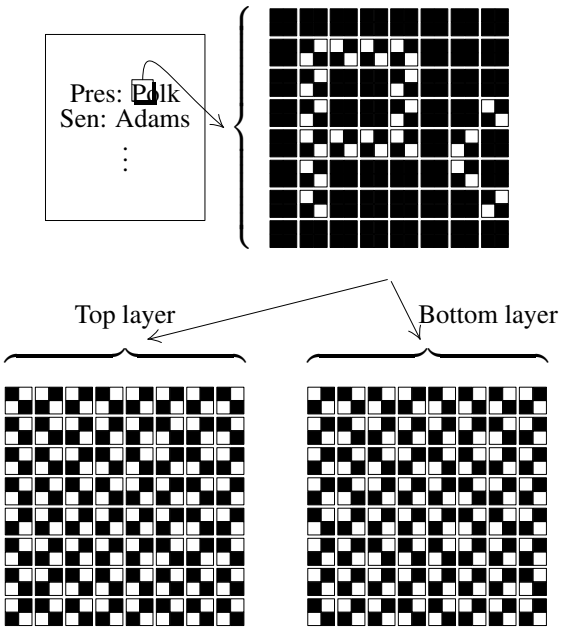

Figure 2: Representation of the printed ballot and transparencies. The top two images show the ballot as well as a zoomed in portion of the two overlayed transparencies portrayed below.[2]

are known.

The voter chooses one layer as a receipt, the other layer is destroyed, the retained layer is posted on the public bulletin board, so the voter can verify his vote was cast correctly.

Chaum's protocol satisfies three properties.

1. **Visual Verifiability:** The DRE hast to create two transparencies T and U, which fulfill the condition $T \oplus_v U = B$ with B representing the voters choice as an ballot image.($\oplus_v$, see figure3)
2. **Recovery:** given a single layer and the trustees keys it is possible to decrypt the ballot for evaluation.
3. **Integrity:** there is now way given B and one of the layers to create a witness transparency that does not decrypt(with the layer) to B.

The voter can visually control his vote was cast as intended, exploiting the transparencies properties. So the human eye does the xor and the voter can immediately see if the DRE tried to cheat him. The pixels have a type $\in \mathbf{P}, \mathbf{E}$ in addition to their values. On a transparency no two pixels of the same type shall be adjacent. Additionally and **E** pixel should only stack a **P** pixel and vice versa. The **P** pixel is generated from a pseudorandom stream, which consists of trustee streams, each trustees stream contains the trustee's number and the voter's BSN, which are then encrypted with the trustees public key. The **E** pixel is then set accordingly, so hat **E** and **P** pixel decrypt to the ballot pixel. That way it is guaranteed that one layer alone reveals no information.

To evaluate the election the trustees perform a multistage mixnet, where each trustee decrypts his part of the layer and permutes the ballots, after the last trustee decrypted the ballots are plaintext again and can be counted.

Encoding for Transparency	1:  0: 
Encoding for Overlay	$\hat{1}$:  $\hat{0}$:  or 

$\oplus_v$ Truth Table	$0 \oplus_v 1 = \hat{1}$  $\oplus_v$  = 
	$0 \oplus_v 0 = \hat{0}$  $\oplus_v$  = 
	$1 \oplus_v 1 = \hat{0}$  $\oplus_v$  = 
	$1 \oplus_v 0 = \hat{1}$  $\oplus_v$  = 

Table 4: table3[2]: Visual Cryptography. A printed pixel on a single transparency has a value in 0, 1, encoded as shown in the first row. We apply the visual xor operator v by stacking two transparencies so that light can shine through areas where the subpixels are clear. The pixels in the overlay take values from $\hat{0}, \hat{1}$. The bottom table shows the truth table for the visual xor operator and its parallels to the binary xor operator.

4 Security Discussion

There are several security issues in the protocols. Since we cant cover them all, we will just list them here and discuss some in detail. We have subliminal channels, meaning either manipulation of the voting process through trusted hardware(semantic) and those who rely on insecure random number generators. Since the voters participate in a complex cryptographic voting protocol, the probability that the voters won't notice slight changes in the procedure is very large, this may change the voting experience significantly but the security impact can be huge. Also with social engineering the DRE can try to force the voter, helping him to cheat. And finally its possible to stop the election process with a Denial of Service(DoS) attack.

4.1 Message Reordering

Message reordering changes the order of the in which the voter follows the voting protocols. In Chaum's scheme the question which transparency is kept as a receipt can be asked before the printing of them. This is a change hard to notice for the voter, but it has a great effect on security, since the DRE could then create an layer that decrypts to the ballot the voter expects to see, but the layer which is posted on the bulletin board decrypts to an other ballot. This is caused through the xor function that allows the DRE an easy construction knowing how the correct layer for the cheating vote has to look like, and then xor with the image that the voter expects, will give the layer that makes the user think his vote was cast correctly. Since the produced layer is destroyed than, it can not be detected that the layer was malformed and the DRE cheated.

A similar attack can be done on Neff's scheme, if you look at the implementation of voteHere.com[2] the voter can choose if he wants a basic or a detailed receipt. The basic receipt does not allow the verification that the vote was cast as intended. This ok due to the fact the DRE does not know if the user will choose a detailed or a basic receipt, by creating the ballot. But if the voter is asked if he wants a basic or a detailed receipt before the ballot is created, the DRE can cheat as it knows it wont be detected.

Even further cheating is possible if the voters choice is known to the DRE before Constructing the VC. This allows the DRE to prepare the row i(choice of the voter) so that it will pass the tests in the OVC and the voter see's his vote on bulletin board, but his vote was cast for another candidate.

4.2 Social Engineering attacks

Social engineering attacks use the same principle as message reordering attacks, but the order of the voting process is not permuted, but new steps are created. An example in Chaum's scheme is the following: The DRE ask the user which transparency he will choose later on, or even more direct. It could try to lead the voter taking a certain layer. That way the DRE has a much lower chance of being detect at a cheating attempt. The voter may not be aware of the fact that each voting process should be different(challenges in Neff's scheme for example). Due to this fact the DRE could reboot after gaining the voters challenge, feigning an error, if the user choose the same challenge again, the DRE could forge a ballot for a different candidate. If the user does not chose the same challenge, the DRE can just reboot to avoid detection.

4.3 Mitigation Strategies

Message reordering and social engineering have a huge impact at the security of an election. The only way to prevent them properly is voter education. But it's not realistic to assume the average voter will have confidence in a voting system where he has to get trained for usage. Some will just reject to learn anything about the system and others will be awed by the system. An other method would be auditing the DRE's, but this has a lot of flaws. For instance to audit a DRE, the DRE should not be able to tell it is audited, so the vote will be committed to the bulletin board. And its not possible to remove it from there, as this would undermine the goal that no one can delete votes on the bulletin board. Should the DRE be audited with special smart cards the DRE could notice it is audited, the same goes for an entry on a ballot(not existing candidate or party) thats just for the audit, would make it possible to detect that the DRE is audited(normally candidates and parties that participate in the election are known before the election day, so an unexpected entry would be a hard evidence. If you would say those votes are insignificant it would allow the testers to audit the DRE very often, manipulating the election that way. And even then there is no way to be sure that a DRE was not detected during the audits.

4.4 Denial of Service attack

To ensure people does not forge any faked recipes to invalidate a voters vote, receipts are signed. But this holds a problem that can invalidate the hole election and is kind of irrecoverable. If a cheating DRE wants to invalidate an election, tries to avoid detection or wants to make sure that even if it is detected cheating, the election as a hole is in doubt, it prints invalid signatures on the receipts. If this happens the voter cannot proof he was cheated by the DRE, as he has no valid receipt. On top of that even if he could proof it, he had to reveal his voting intentions. This would violate the privacy of the election. If it wants to avoid detection it can also do not print his own machine id on the ballots, but those of other DRE's. If the election is not going as the cheaters wanted it, the DRE could just print always invalid signatures. As soon as the first voter discovers that, others will follow and the hole election is in doubt.

A far more Dangerous thing is that the DRE can ignore the voters input and forge a valid ballot, challenge and receipt as the DRE wants. The user will have a hard time proofing he was cheated that way, as everything is correct but the voters choice was not represented. Only auditing the same DRE and experiencing the same thing again would be a valid proof. But the DRE does not have to cheat the hole time, so this is not a guarantee for detection. And even if it is detected, it is not known if other voters experienced the same thing and did not notice, so the hole election is in doubt again. This attack is also to recoverable.

This problem can be pursued further, if the DRE is capable of assigning BSN's or is

able to guess or know the valid BSN's, it can forge valid vote's for each BSN, even though this might lead to collisions and becomes detectable, there is no way of recovery as there are multiple ballots with the same BSN. IF the DRE just wants to let the election fail, it just hast to create votes with already cast BSN's so its again not possible to check which vote is the correct one.

A possible solution to that would be a time stamp at the receipt and the ballot, but even that can be manipulated and on top of that erode voter privacy, as it becomes verifiable who voted at that time.

The bulletin board is another vulnerable spot for a DoS attack. If the bulletin board misbehaves there is no way of recovery, it could delete the trustees keys and insert and delete ballots. This especially irrecoverable if the tallying has begun.

Since it is unclear if the bulletin board is a global system or there are as many bulletin boards as polling stations we do not know how the results are computed. As there are several ways of putting the results together. Either it is transported manually on a electronic device or per network connection, both can be attacked with either an Dos attack or even more worse the transported data could get corrupted.

4.5 Mitigation Strategies

There are two obvious countermeasures are revoting and a voter verified paper audit trail.

Revoting has some troubles as the first question is who will be allowed to revote. A complete revote costs a lot of money and has the problem that in the next complete revote another DoS attack could happen. Letting only the ones revote that have proven to be cheated or letting the voting districts in which cheating was detected revote also has a great flaw. As it is not ensured that other voters who were less attentive could have been cheated too.

The voter verified paper audit trail would solve all of those problems as plaintext ballot is printed from the DRE and then verified from the voter and put into an urn. This way the voter can be sure his vote is cast as intended. It would be easy to make existing DRE's printing these ballots. Karlof[2] suggests that it should be used as a backup system, this has a great flaw as cheating can still be undetected. But since the ballots would be printed then, it counting them could be done with scanners with detection software(This can leave security issues again but counting the paper audit should be done too.) , so the counting is faster and that way there can be checked if there is a difference between the paper audit trail and the voting protocol. However the greatest trouble with the paper audit trail is the fact that DRE originally should replace the paper based system.

5 Conclusion

Both schemes are underspecificated. This begins with the assignment of BSN's. This is a huge problem, as BSN's are vital to distinguish legitimate inserted votes from illegal inserted ones. The best way to stop attacks on BSN's would be getting them from an external source, if the voter enters a smart card. However that way the DRE is still capable of using the BSN just delivered to her. But then again it would be possible to get a relation between the a certain voter and his BSN. To avoid this a random BSN from a given set of BSN's is chosen. To ensure the DRE can not get a BSN from that set on her own, its necessary to split the voter authentication and the DRE, however that would create a new vulnerable spot(another untrustworthy hardware device). And in the end the DRE could try to guess BSN's, or even know the BSN's from a cheating trustee. Till now there is no good way to handle BSN's.

The bulletin board is also not specified, but its security has to be impenetrable as it stores the casted ballots, so the capability of modifying the contents of the board is giving

the cheater all options. He could manipulate every single vote and still show the voters choice to every observer. Both scheme did not mentioned a single security mechanism to ensure the bulletin board is not corrupted. On top of that it is not even specified how the DRE's communicates and authenticates with the board. so anyone with access to the bulletin board is potentially able to add votes to it.

Also the tallying process after the mixnet was executed is in no way specified. This is very problematic as the software could do an invalid count. A solution to that would be to use several different counting software, but that costs money and it is no guarantee that no all of them or the majority is corrupted as it would be very hard to distinguish the correct working from the malformed tallying software.

This implies another flaw, the summation of all partial results(if they are collected globally the same thing applies to the way to the global bulletin board) is a problem. This needs a secure transport mechanism. This is a problem as digital data can be corrupted on the way, even if it is transported physically, as the transporter could exchange the data during the delivery.

Due to the fact there are so many very vital flaws it is in no way considerable to safely use those systems in a real election. Cryptographic voting protocols have an interesting approach, as they try to ensure the correctness of an election independent from the underlying hardware. This however seems to be an unreachable goal. With every verification they proposed they introduced another weakness. In the end something or somebody has to be trusted without a doubt, as that cannot be guaranteed for hardware(or software), only the trustees and the voters are left. But since human can make mistakes thats also not a security guarantee.

This leads to the voter verified paper audit trail, however that is just what we had before the electronic voting systems raised. As assumed in the security discussion, an election with the DRE printing the ballots verifiable to the voter and secured in an urn can improve numeration time. This can be done, due to the fact that printed ballots can be evaluated from multiple scanners and then compared to the electronic result. If a mismatch is detected the paper ballots would have to be counted manually. This is a relative secure way of dealing with an election, however its very costly.

This introduces the question, are electronic voting system worth the effort, risks and money put in them, if they still need the traditional way of voting with a paper system? In my humble opinion they do not, as even the voting protocols have flaws, the risk of an corrupted election increases with an electronic voting and due to the preparation and asset costs, i highly doubt it will be any cheaper than a traditional paper based election.

References

[1] Markus Jakobsson, Ari Juels, and Ronald L-Rivest. Making mix nets robust for electronic voting by randomized partial checking. 2002.

[2] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A systems perspective. *14th USENIX Security Symposium*, pages 33–49, August 2005.

[3] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an electronic voting system. *Proceedings of the 2004 IEEE Symposium on Security and Privacy(S&P'*04, pages 27–40, May 2004.