A Review Study on Electronic Election

Ahmad Baraani-Dastjerdi
Josef Pieprzyk
Reihaneh Safavi-Naini
The Centre for computer Security Research
Department of Computer Science
University of Wollongong
Wollogong, NSW, 2522, AUSTRALIA

June 2, 1994

Abstract

In a democratic society, there is a need for a system that citizen practise their democratic rights and vote with accuracy, privacy, and security. Any such system must provide citizen's rights including equal access to the voting process by all individuals, the ability to verify the registration, and the ability to vote in privacy without intimidation. Therefore, the problem of designing such a system has been the subject of interest among researchers for a number of years. This paper reviews known solutions on electronic election, describes what are currently thought, and points out the main features and characteristics of secure voting systems.

1 Introduction.

Modern societies more and more rely on computer systems and networks. The development of cryptographic techniques allows us to "computerize" some of the new untouched area of human activity. Task such as voting is a simple human activity in democratic communities. In this review we are going to study electronic implementation of the election process. In order to use computer effectively in voting processes a lot of research have been done and many protocols have been proposed.

The main requirements are: voters are able to check if their votes have been counted correctly, the voters should not be able to see votes of other people, and the system should be secure against illegal behavior of potential voters and outsiders.

An efficient and secure electronic election protocol requires a good design and an appropriate technology to support its implementation. Research in this area has been focused on understanding of cryptographic operations, their position in the protocol and the efficient implementation of the protocol using the VLSI technology.

For the first time, in 1958, Walter Cronkite and the CBS Election Headquarters team began using a computer for election forecasting. Now computers have become key strategic tools in large or small elections. During the election, computers can be used to:

- help candidate's campaign,
- count votes,
- conduct the whole election processes.

In Section 2, we present how campaigns of candidates can be supported electronically by computers. Next we show how to use computerized vote counting in the election process (Section 3). In Section 4, we review four protocols for electronic elections. Conclusions are given in Section 5.

2 Using Computers to Help Candidate's Campaign.

Campaigning has evolved from speeches on cabooses to televised debates. There is already well established industry which supports candidates' campaigns using the latest technology. Today's politicians rely increasingly on computerized devices to keep their campaigns running smoothly. Technology is ranging from touch-screen voting to various forms of electronic debating. Here, there are two following cases in which computers are used to help candidates in their campaigns during the 1992 US election season.

At the Democratic National Convention, delegates used a touch-screen-based electronic voting system to cast their votes. The system, made by NCR Corp, uses the Unix-based NCR 3B series that runs the back-office functions [3, 10].

The Republican National Convention used 27 different databases from WordPerfect Corp to access information on the convention's housing options, on the volunteers and on transportation alternatives [3].

Now, the question is that how the candidates can benefit from computer technology in their campaigns.

2.1 How Can Computers Be Used in Candidate's Campaign?

The computer systems can help candidate's campaign in the following ways:

1. Preparing a database of names and information on the voting tendencies of delegates.
2. Scheduling a candidate's appearances.
3. Precinct targeting that alerts campaign managers to areas needing a candidate's attention.
4. Reporting financial statements according to Federal Election Commission (FEC) regulations.
5. Transmitting candidate's message by using computerized media and side range of computer technologies (computer graphics, computer billboard, etc).

There are many security and privacy problems associated with the above computers during the campaigns. The campaigns should be carried out in such a way that voters' rights of privacy should be protected. Here we have some examples of possible privacy abuses.

The government can target groups of voters using information stored in government databases. For example, a group of fathers who were not paying child support can be identified from the Department of Social Security (DSS) databases. A group of people who are against a strict gun control can be extracted from a file of gun owners registered by the police.

In the next section, we will see how computers can be used in the vote counting and the security requirements that must be imposed in order to have a reliable voting result.

3 Using Computers in the Vote Counting Stage.

If the voter is the essential component of any election the next most essential component is the record of the voter's vote. Counting votes has to be error free. An electronic election should be implemented in such a way that it: (i) allows the voters to cast their votes in privacy, (ii) prevents partisan voters from casting their votes if they deviate from the correct voting procedure, (iii) forbids persons who are not eligible to vote, and (iv) disallows eligible voters to cast their votes more than once.[15]

In the USA, there is a vote-tallying system in use. Roughly half of the voters use election machines in the voting process.

Later in our paper, we shall introduce different types of vote-tallying equipments and we will examine protocols to ensure the correct count process. We will also discuss advantage and drawbacks of vote-count machines and give some examples.

3.1 Types of Tallying Equipment.

First tallying equipments (machines) were based on punched card readers. There are inherent problems with accuracy of these types machine, both in the count of ballots, and in the count of votes on the ballots [20].

Direct Recording machines (DRE) do not use ballots. Voters enter their choices directly into a storage unit of the machine with the use of push-buttons, touch screens or similar devices.

There are problems with DRE machines. The first is that there is no difference between a voter failing to vote and a machine failing to record a vote. The second problem is that the correctness of the machine data entry process cannot be checked. It is possible to make verification of the DRE machine summation of the voters' choices. Therefore, the DRE data entry function must be correct [20].

To have the correct tallying result while using tallying machines, it is essential to follow the policy that ensures the correctness of vote-tallying results.

3.2 Protocols for Computerized Vote-tallying System.

Saltman [20] gives some recommendations to ensure accuracy, integrity, and security of the tallying results. The recommendations can be divided into two classes. The first class relates to hardware recommendations, and the second class specifies the correct procedure which has to be followed while using vote to systems.

3.2.1 Hardware Recommendations.

1. Elimination of pre-scored cards. The use of pre-scored punch cards contributes to the inaccuracy both in the count of ballots, and in the count of votes on ballots. Therefore it is recommended to stop using pre-scored punch card ballots.

2. Elimination of setup-condition errors. Setup-condition errors involve specific election matters, such as wrong rotations, incorrect assignments of candidates to districts, incorrect summarizing of each candidate's votes, and causing wrong result against large number of ballots. It is necessary to eliminate setup-condition errors in pre-election system checkout.

3.2.2 Recommendations on Operational Procedures.

1. Audit Trails. Audit trails provide documentations through which the correctness of the results may be verified. Two types of audit trails are necessary to document operations and provide confidence in the results reported. One type records steps in the operation of the equipment, while the other records steps in the voting and vote-tallying processes.

2. Access Controls. Security controls must be in place during preparations for voting, voting itself, and vote-tallying. These controls concern access to sites, areas, facilities, equipment, documents, files, and data. The control covers transportation of ballots and telecommunication of results.

The controls on ballots also cover printing and distribution, accounting for use, validity, and prevention of error software that is intended for use in more than one election (it is usually written in a generalized form, and then specialized for specific election).

3. Assuring System Integrity. To protect voting-tallying system from unauthorized changes and other software vulnerabilities such as logical errors, the possibility of hidden codes, and undocumented changes strict administrative control must be imposed.

An important administrative control for protection against hidden code is the requirement that software be supplied only from the original manufacturer who is known, reliable and can be held accountable.

The vote-tallying software is provided by its developer with all necessary support software including the operating system, as a complete package. The provision of a complete software package enables an additional protection using data authentication code (DAC). The code is a binary string that is computed by randomly selecting a certain key value and then using the key to apply a specific mathematical function (for example hashing function) to the software package. The key value is kept secret. The computation of the DAC protects the software against both accidental and intentional unauthorized data modification. In application, the DAC would be generated for the original software deposited with state, and would be generated also for the supposedly identical software used in an election. The two values of the DAC will be identical if the software packages are identical.

Modifications to software must be strictly controlled to prevent unauthorized changes. Authorized changes should be documented. Previous versions of revised software should be retained according to the established procedures.

The system must be verified zero counts prior to opening of the polls.

4. Assuring System Security. Provision of all software, as a complete package, implies that the computer on which the software is to be run must be dedicated to the election operations. With the wide availability and downward cost trend of minicomputers and microcomputers, this possibility is reasonable, even for smaller election administrations.

3.3 Voting-Machine problems.

However the automated electronic voting systems have significant benefits in saving manpower costs involved in counting paper ballots, and process voting information in minutes rather than hours. Since the data provided by the voting machines are insufficient, and the programs are tamperable and vulnerable, it is difficult to demonstrate the correctness of the results. Therefore, voting machines have problems in their verifiability, completeness, and soundness [12, 20, 19].

3.4 Future Direction.

While vote-tallying using telephone [16], Integrated Touch Screen (ITS) [21] or Signature-Verification Imaging System [22] is technologically feasible, the decision to implement such a system must be based on more fundamental requirements. Any installed system must meet political and economic requirements as well as technical requirements of accuracy and privacy.

As we saw in the two previous computerized voting schemes, only part of the voting process was computerized. In most cases it is hard to support the privacy of voters and accuracy of voting. Therefore we need mechanisms for the entire voting process: registration, voting, counting, and administrating. In the following section, we discuss some protocols that achieve this goal.

4 Using Computers to Conduct the Whole Election Processes.

There has been extensive research into the design of cryptographic protocols which can be used to conduct the whole election process. The main problems related to electronic election are:

1. transmission of untraceable yet authentic messages. Therefore, we need a protocol to ensure secrecy and authenticity. Every message in the system must have come from some legitimate user, but no user can be associated with particular message transmitted.

2. every voter must be able to check that his vote is counted, without compromising the privacy of other voters.

3. prevention of the election protocol against both illegal activity of legitimate voters and dishonesty of the outsiders.

Next we give a list of properties which have to be satisfied to ensure that the electronic election protocol works correctly and possesses the requested security features. They are used to evaluate the electronic election protocols investigated in Section 4.3 through 4.6.

4.1 The Security Requirements for an Electronic Election Scheme.

A secure electronic election protocol must have the following properties:

Completeness : All valid votes are counted correctly.

Soundness : A dishonest voter cannot disrupt voting.

Privacy : All votes must be secret.

Un-reusability: No voter can vote twice.

Eligibility : Only those who are allowed to vote, can vote.

Verifiability : No-one can falsify the result of the voting.

Fairness : Nothing must affect the voting.

4.2 Reviews of Some Secure Voting Schemes.

Chaum [7] was the first who proposed an election protocol. Several other voting schemes have been proposed by other researchers emphasizing theoretical and practical aspects of voting [18, 1, 9, 14, 7, 2, 13].

In some proposals, the whole voting procedure is controlled by voters themselves. However many interactions among voters are necessary to prove that the voting procedure has been done correctly. As in real voting systems there is no interaction among voters, these protocols are interesting theoretically but not practical.

In order to design a secure election protocol that provides secrecy and authenticity, we can use the two following methods of transmission of untraceable yet authentic ballots:

1. the ballot is sent in an encrypted form;
2. the ballot is sent through an anonymous communication channel.

The first approach is proposed by Benaloh and Yung [1], Cohen and Fischer [9], and Iversen [14]. Their schemes utilize encryption techniques.

The second approach has been proposed by Chaum [7], Boyd [2], and Fujioka, Okamoto and Ohta [13]. Their schemes used an anonymous communication channel, and one administrator. They provide unconditional security against tracing the voting.

Next we will present secure protocols which are interesting theoretically and practically.

4.3 Pfleeger's Proposed Voting Protocol.

This model is managed by just the voters. Every voter has to do two public key encryptions and two private decryptions. It is a boardroom voting in which participants pass encrypted messages from one to another while performing encryption and decryption operations until a certain point is reached at which all are confident of the outcome of the vote. The problem is that the active participants must be known in advance and if one participant stops following its protocol during the election, the voting process cannot be continued .

4.3.1 Assumption.

Since it is possible that two people may have identical votes, the protocol must allow people to recognize their own vote, without being able to recognize other people votes. For this reason we have two public key encryptions. We use the following notations:

u : identify voter u.

E_u: public key encryption function for voter u. (The secret key function is known to u.)

R_u: second public key encryption function for voter u. For providing the desired secrecy, verifiability, and identifiability in case two voters have the same votes, votes are embedded in a random string, and then encrypted

D_u: private decryption function for voter u for public key function E_u.

Q_u: private decryption function for voter u for public key function R_u.

v_u: the vote of voter u.

4.3.2 Structure of Scheme.

To show how the protocol works we shall consider a voting example. Suppose the three voters are Joan (j), Keith (k), and Lee (l). Their public encryption functions, private decryption functions, and their votes are denoted by $(E_j, D_j, R_j, Q_j, v_j), (E_k, D_k, R_k, Q_k, v_k)$, and $(E_l, D_l, R_l, Q_l, v_l)$, respectively.

Voting Stage. A voter u chooses a vote v and computes:

$R_j(R_k(R_l(E_j(E_k(E_l(v_u))))))$

i.e. each voter takes his vote and encrypts it with his own public encryptions, then encrypts with every other voters' public encryptions (E_j, E_l, E_k). Then he/she embeds the massage in a random string and encrypts the result by R_j, R_l, R_K. All votes are then sent to one of the voters, in this case Joan (see Figure 1).

$R_j(R_k(R_l(E_j(E_k(E_l(v_l))))))$

$R_l(R_k(R_j(E_l(E_k(E_j(v_j))))))$

$R_j(R_l(R_k(E_j(E_l(E_k(v_k))))))$

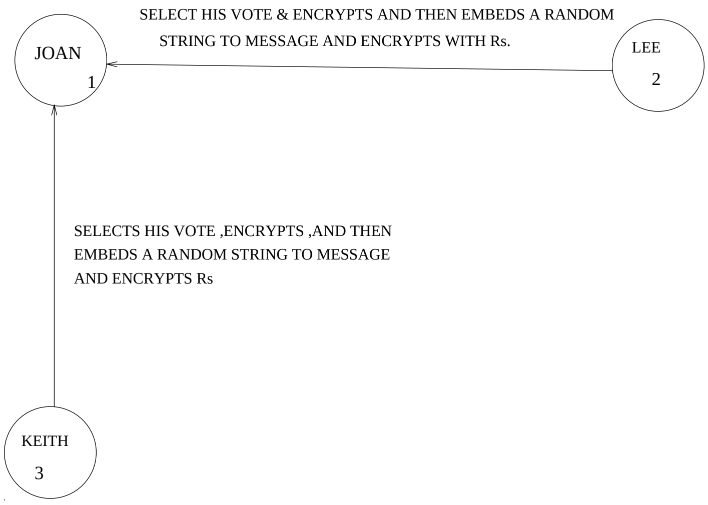

Figure 1: First stage voting

Verification Stage. Joan removes one level of encryption from all votes,

$Q_j[R_j(R_k(R_l(E_j(E_k(E_l(v_u))))))] = R_k(R_l(E_j(E_k(E_l(v_u)))))$

and forwards them in scrambled order to the next voter, say Keith. Keith verifies that his vote is in the set and is correct, then removes one level of encryption:

$Q_k[R_k(R_l(E_j(E_k(E_l(v_u)))))] = R_l(E_j(E_k(E_l(v_u))))$

and passes them to Lee who carries out the same process (see Figure 2).

$Q_l[R_l(E_j(E_k(E_l(v_u))))] = E_j(E_k(E_l(v_u)))$

At last, after finishing this stage votes will be as follows:

$E_j(E_k(E_l(v_j)))$

$E_j(E_k(E_l(v_l)))$

$E_j(E_k(E_l(v_k)))$

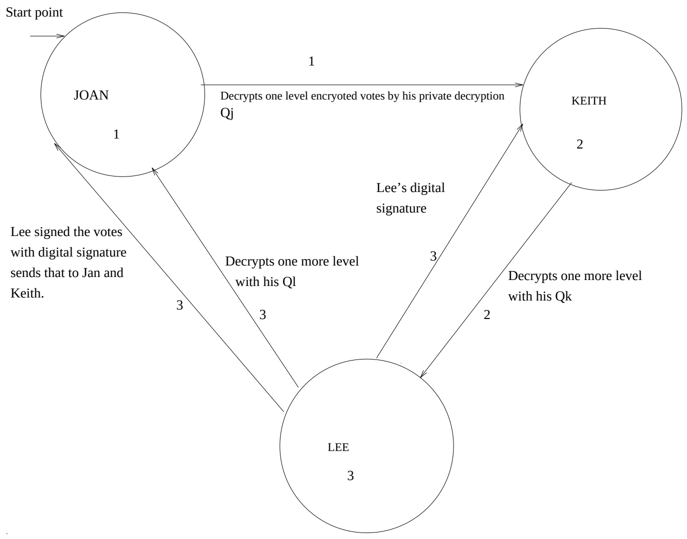

Figure 2: Second stage verification

Opening Stage. Lee passes all encrypted votes $[E_j(E_k(E_l(v_j)))$, and $E_j(E_k(E_l(v_j)))$, and $E_j(E_k(E_l(v_j)))]$ to Joan, and Lee also digitally signs the votes and passes the signature to both Joan and Keith (No. 3 in Figure 2).

Joan removes one more level of encryption, signs the ballots, and passes the votes to Keith, and the signatures to both Keith and Lee (No. 1 in Figure 3).

Keith receives $E_k(E_l(v_u))$ from Joan and does the same, passing the signed message to all other voters, leaving $E_l(v_u)$ for Lee (No. 2 in Figure 3).

Lee removes the final level of encryption and publishes the results and the signatures for Keith and Joan for verification (see Figure 3).

4.3.3 Analysis of Protocol.

The protocol works because of the chain of decryption used:

$J \longrightarrow K \longrightarrow L \longrightarrow J \longrightarrow K \longrightarrow L$

The results can be made public at each level without anonymity being lost.

Completeness: If all voters are honest the result can be trusted.

Soundness: Assume the public key system is secure.

The only way to disrupt the voting is to send a bogus vote. This means that the voter needs to obtain an extra set of public/private keys, this set would necessitate the independent production of the key set, put in the listing of the public key for all other voters. This contradicts the assumption.

Any invalid ballot will be detected in the first round and identified in the second round.

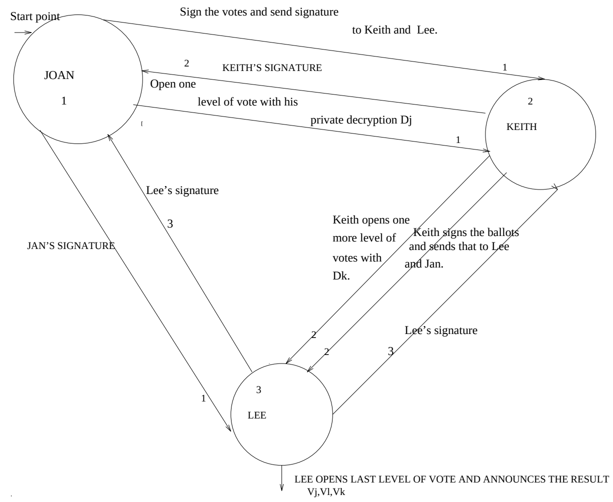

Figure 3: Third stage opening votes

Privacy: This is achieved through the two-pass system using a blinding technique of embedding votes in a random string before the R encryption to ensure secrecy.

Un-reuseability: Assume key distribution is secure.
In order to vote twice the voter must have two sets of keys as well as have the public keys listed. We already know the voter cannot produce an independent set, so this means the voter must obtain a second key set invalidly, which contradicts the assumption.

Eligibility: Assume a dishonest person can vote.
Then his/her (public) key must be listed for all users to encrypt with. This means the voter must have obtained a key set fraudulently, this contradicts our previous assumption, therefore the assumption that the dishonest person can vote is contradicted.

Verifiability: In the second round, anyone may move backward (for verification), but only one person may move forward, and they may move forward once. After publication, the result may be verified to ensure that the final voting set is the same as the initial voting set. Because each of the voters in turn signs the set during the second round, the guilty party can be found.

Fairness: The vote cannot be affected since everybody must vote and encrypt the messages before the vote-opening begins, and no votes may be added after beginning of the first round.

However it ensures verifiability, privacy, fairness, and un-reuseability but there are problems with this protocol:

1) It takes a lot of communication and computation overhead when the number of voters is large since every voter must check every other voter's vote. Therefore, this scheme is impractical for large scale elections, but is interesting theoretically.

2) In a practical secret voting scheme, we require additional participants, e.g., a trusted center or administrator to approve legitimacy of a voter, but in this scheme, all procedures are managed by just the voters.

3) It is required that all voters are present at the same time and they must be known in advance.

4) If one of participants does not follow up the protocol during, the election cannot be continued.

4.4 Chaum's Proposed Schemes.

The first practical secret election protocol was proposed by Chaum [4] in 1981, using a trusted "mix" to scramble pair of votes and digital pseudonyms in order to protect voter's privacy.

4.4.1 Background

Notations. We use the following notations:

S : sender.

B : receiver.

M : plaintext.

E_K : receiver's public key encryption function with public key K.

$D_{K^{-1}}$: private key decryption function (K^{-1} is the inverse of public key K).

E'_{K_1} : the public key encryption function of mix with public key K_1 .

R_0 and R_1 : two big random strings.

A : the address of receiver.

C : a big constant random number.

v : voter's vote.

What is Anonymous or Untraceable Communication Channel? In order that S sends message M to B anonymously, first, S seals message M with his public key K, Y=$E_K(R_0, M)$, where R_0 is a big random number in order to eliminate the possibility of checking Y=M (to increase security).

Next, S concatenates address of B to the encrypted message and then encrypts with public key of mix K_1, and then sends via mix to B (Figure 4):

$E'_{K_1}(R_1, E_K(R_0, M), A)$,

where R_1 is a random string number to increase security by eliminating the possibility of checking equality of plaintext and ciphertext, and A is address of B.

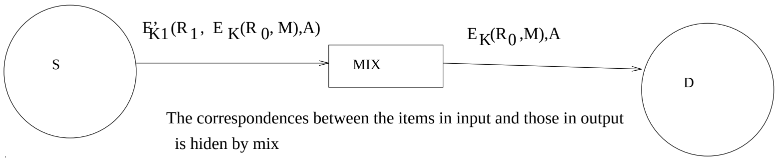

Figure 4: Mix

Definition of Mix. Someone becomes a user of public key cryptosystem who creates a pair of keys, and publishes one of them as a public key, say K, and keeps the other one secret, say K^{-1} . A message X is sealed with a public key K so that only the holder of private key K^{-1} can discover its content. If X is simply encrypted with E_K, then anyone could verify a guess that Y=X by checking whether $E_K(Y) = E_K(X)$. This threat can be eliminated by attaching a large string of random bits C to X before encrypting. The sealing of X with public key K is then denoted by $E_K(C, X)$, and the signing of X with the private key K^{-1} is denoted by $D_{K^{-1}}(C, X)$. Anyone can verify that $Y = D_{K^{-1}}(X, C)$ is signed by the holder of K^{-1} by finding $E_K(Y)$ and checking the value of C.

In a cryptosystem besides the correspondents, computer called 'mix' is that processes each item of mail before it is delivered.

Thus, if a participant wants to send a message M to someone at address A and public key K, first he/she seals M using E_K to obtain $E_K(R_0, M)$, then appends the address A and finally encrypts it with mix's public key K_1 to obtain $E'_{K_1}(R_1, E_K(R_0, M), A)$. The result is transferred to the computer, 'mix'. The mix decrypts its input with its private key, throws away the random string R_1, and outputs the remainder (Figure 4).

$E'_{K_1}(R_1, E_K(R_0, M), A) \longrightarrow E_K(R_0, M), A$

The purposes of the mix are:

1. To hide the correspondences between the items in its input and those in its output.

2. To ensure that no item is processed more than once by making part of each random string R_1 to contain something, such as a time-stamp, that is only valid for a particular batch.

Digital Pseudonyms. A digital pseudonym is a public key used to verify the signatures made by the anonymous holder of the corresponding private key. A roster (or list of pseudonyms) is created by an authority that decides which applications for pseudonyms to accept, but is unable to trace the pseudonyms in the completed roster. The applications may be sent to the authority anonymously.

4.4.2 Structure of Protocol.

Assumption: 1. There is a faithful and trustable administrator.

2. Voters and administrator communicate via an anonymous or untraceable channel.

3. Each voter has a digital pseudonym.

Preparation Stage.

Authority sends to each voter all the information required for the acceptance decision and a special unaddressed digital letter which includes the public key K and the voter's proposed pseudonym P. This letter is of the form $E'_{K_1}(R_1, K, P)$. (Figure 5).

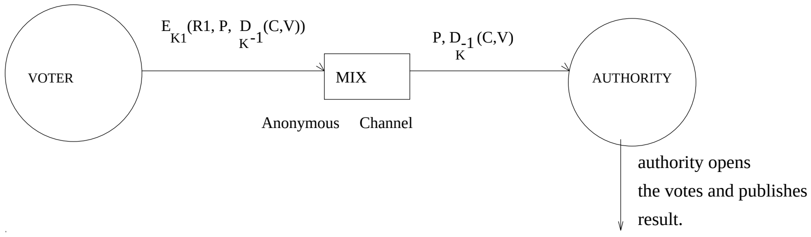

Figure 5: First stage preparation

Voting and Counting Stage. Figure 6 illustrates this stage.

1. Each registered voter submits a ballot of the form $E'_{K_1}(R_1, P, D_{K^{-1}}(C, V))$ where P is the voter's pseudonym and V is the actual vote.

2. The ballots are processed as single batch, as were the letters used to form rosters so the order of output batch is of the form $P, D_{K^{-1}}(C, V)$.

3. Checking pseudonyms: IF pseudonym P is in the list THEN decrypt the signed vote V, and count that ELSE reject that.

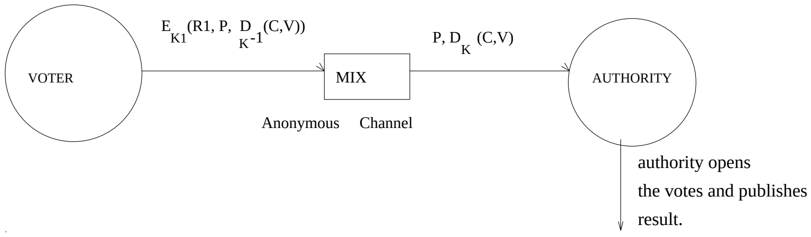

Figure 6: Second stage: voting and counting

4.4.3 Analysis of Protocol.

As we can see, votes are publicly revealed, but the identity of voters is protected by the mix.

Later in 1988, at EUROCRYPT'88, Chaum [7], proposed a new protocol which is drawn on two of his previous basic result ("Sender Untraceability" [5, 6] and "Blind Signature" [8]).

The voting scheme uses an anonymous communication channel and provides unconditional security against tracing the votes but the failure of a single voter can disrupt the election. However his method ensures that a failure can be traced, and the election can be restarted without the faulty voter. Therefore, this approach is not practical for large scale elections.

Another voting scheme which is based on the use of " multiple key ciphers" is proposed by Boyd [2] at EUROCRYPT'89. This voting scheme ensures that the votes can not be forged, and privacy is preserved, if the voters can deliver their votes anonymously. But the major problem of this scheme is that the administrator can see the votes delivered, and even worse, produce a false result by adding votes of its own choice. Next we look at this proposal .

4.5 Boyd's Proposed Protocol.

Boyd's scheme [2] is based on multiple key ciphers which is simply defined by the group of exponentiation transformations in a prime field with exponents prime to $p-1$. The important difference between multiple key and RSA is that this function does not possess the trapdoor property. It ensures that votes cannot be forged, and privacy is preserved, provided the voters can deliver their votes anonymously.

The major problem of this scheme is that the government can see the votes delivered, and even worse produce a false tally by adding votes of its own choice, hence there is no verifiability.

4.5.1 Structure of Protocol.

Participants. This model consists of voters and an administrator who will faithfully carry out elections and issue valid voting slips to every potential voter exactly once.

Other assumptions are the existence of some universally publicized large (enough) prime p that it is universally accepted, and that the discrete logarithm problem in the field of integers modulo p is a "hard" problem.

Preparation Stage.

1) Administrator selects three complementary keys a, b, and c. Two of them, for example a and b are selected randomly and coprime to $p-1$, and the third one c is selected such that $abc = 1 \bmod p-1$. One of these keys, say a, is published as public key for the purpose of election.

2) An element e that is prime to $p-1$ is also published.

Registration Stage. The process of registration is as follows (see Figure7):

1) Each voter registers for the election by forming a block M consisting of :

- a string "C" as a redundancy; it should be the same for all voters;

- a random string "R"; it should be chosen uniquely by each voter for the particular vote;

- the voter's voting intention "V", specified in a selected number of bits depending on the number of possible outcomes of the election.

2) Before the block M is sent to the authority for registration , the following two encryptions are performed on it by the voter:

a) First the voter forms

$$B = e^M \bmod p.$$

b) Second the voter blinds B by splitting the public key a in two number $a1$, and $a2$ such that $a1$ is chosen randomly and prime to $p-1$, and $a2$ is calculated by $a1a2 = a \bmod p-1$. Then B is encrypted with $a1$ by the voter and sent to the authority

$$B^{a1} = e^{(Ma1)} \bmod p.$$

3) The authority will check the voter's identity.

If the check is successful then the authority records that the voter has registered in order that no voter may vote twice, and the authority encrypts the voting block with secret key b and returns

$$B^{(a1b)} \bmod p$$

to the voter which is administrator's certificate for a potential voter.

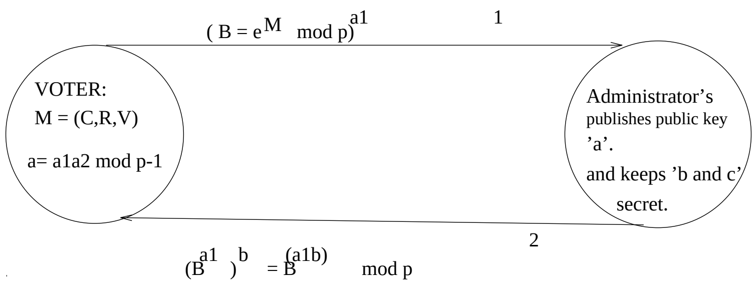

Figure 7: Second stage registration

Voting and Counting Stage. Voters complete their votes by encrypting the returned block with $a2$ to form

$$((B^{a1})^b)^{a2} \bmod p = B^{ab} \bmod p,$$

and forward them anonymously to the authority together with the original blocks M. Then the authority will do as follows:

1)The authority recovers the plaintext B by encrypting received anonymous block with c,

$$(B^{ab})^c = B^{abc} \bmod p = B.$$

Then he calculates $B = e^M \bmod p$ and checks it against the B computed above. If the redundancy condition is satisfied then the authority (Administrator) will accept the vote.

2) The authority will publish all plain voting blocks with the result of the election. Then each voter may check that his random number is presented to verify that his vote is counted. Figure 8 illustrates that.

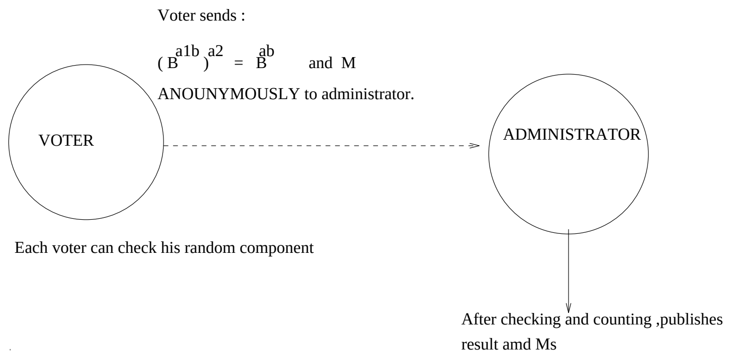

Figure 8: Third stage voting and counting

4.5.2 Analysis of the Protocol.

The proposed scheme ensures:

Un-reusability: The forger must be able to construct a pair (M, $e^{(Mab)}$) such that M satisfies the redundancy condition. Finding a (M, e^{Mab}) pair is equivalent to solving the discrete logarithm problem or breaking the Pohling-Hellman cryptosystem without finding the secret key.

Eligibility and Soundness: The dishonest person should find the secret authority key if he wants to send a pair (X, X^b) in registration phase or (M, $e^{(Mab)}$) in the voting phase. It means that he must solve the discrete logarithm problem.

Privacy: The privacy is preserved if the voters can deliver their votes anonymously.

Although the protocol ensures privacy, and authenticity of voters, but the essential drawback of this scheme is that the administrator or vote-issuing authority can see the votes delivered and even worse, produce a false tally by adding votes of its own choice. Thus the result of casting is not verifiable.

Iversen [14] in 1991 at CRYPTO'91, proposed a cryptographic voting scheme based on residue encryption technique. The participants of the scheme are the voters, the candidates and the government. It is assumed that the voter and the candidates communicate over a broadcast channel. The scheme ensures independence from the voters so they do not have to be present at the same time or go through several phases together and no global computation is needed. The scheme preserves the privacy of the votes against any subset of dishonest voters, and against any proper subset of dishonest candidates, including government. Also it ensures that no subset of voters can corrupt or disrupt the election. But the essential drawback of this approach is that if all centers conspire the privacy of the voters is violated. Moreover this scheme is less practical for large scale elections, since it takes a lot of communication and computation overhead when the number of voters is large.

The last scheme which we consider is the one presented by Fujioka et. al.'s [13]. This scheme is the most suitable scheme for large scale elections, since the communication and computation overhead is fairly small even if the number of voters is large. But the essential drawback of this scheme is that the assumption "no voter abstains from voting" is less practical then a failure of single voter can disrupt the election. Detailed discussion of their work is presented in the next section.

4.6 Fujioka et. al.'s Proposed Protocol.

The following protocol is proposed by Fujioka, Okamoto and Ohta in AUSCRYPT'92 [13]. This scheme is more suitable for large scale elections, since the communication and computation overhead is fairly small even if the number of voters is large. It ensures the privacy of the voter, even if both the administrator and the counter conspire, and voting fairness, i.e., no one can know even intermediate result of the voting, and finally it prohibits fraud by either the voter or the administrator.

The major problem of this scheme is that a failure of a single voter can disrupt the whole election process.

4.6.1 Notations.

V_i : Voter i.
v_i : The i-th voter's vote.
ID_i : The i-th voter's identification.
A : Administrator.
e_i= B(v_i, k_i) : Bit-commitment scheme for vote v_i using key k_i.
d_i= $S_A(e_i)$: Administrator's signature.
C : Counter.
$S_i(e_i)$: The i-th voter's signature scheme
$X(x_i, r_i)$: Blinding technique for ballot x_i and random r_i.
$R_A(d_i, r_i)$: Retrieval technique of blind signature.

4.6.2 Structure of Protocol.

Participants and Assumptions. The participants of scheme consist of voters, an administrator, and a counter. It is assumed that there exist : (i) an anonymous communication channel that the counter and voter communicate through [4], and (ii) a bit-commitment scheme for computing ballot[17]. Further Fujioka assumes that every voter has his own ordinary digital signature scheme [11], the administrator has a blind signature scheme [9], and the counter only creates a list of ballots, and publishes it.

Election processes are separated into three stages, registration, voting, and opening. The illustration of each stage is as follow:

First Stage : Registration. In order to be certified for the eligibility, each voter sends a request to the administration for registration. Registration requires co-operation between the voter and the administrator.

Voter. First voter V_i selects a vote v_i and computes the ballot using a key k_i randomly giving x_i=B(v_i, k_i) and also computes the message e_i using the blinding technique $e_i = X(x_i, r_i)$.

Second Voter V_i sings e_i to obtain $s_i = S_i(e_i)$ and sends the triple $< ID_i, e_i, s_i >$ to the administrator.

Administrator. The administrator checks that the voter V_i has the right to vote. If the check is satisfied then the administrator's certificate is sent to the eligible voter. Administrator, for every voter, checks that:

IF V_i has the right to vote, V_i has not already applied for signature, and signature s_i of message e_i is valid

THEN A signs e_i to obtain $d_i = S_A(e_i)$ and sends A's certificate to V_i

ELSE the administrator rejects the transaction.

Figure 9 illustrates this stage.

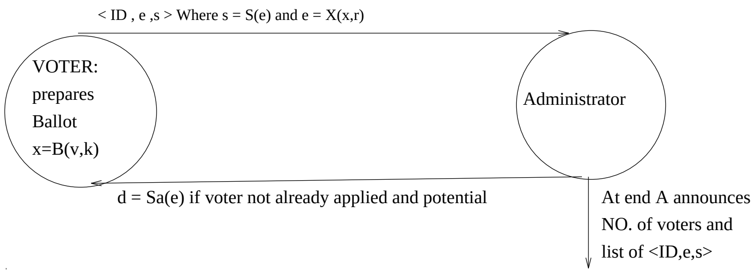

Figure 9: First stage registration

Voting Stage. Voters, after checking the validity of the administrator's signature, send their ballots and the ballot signature through an anonymous communication channel to the counter. Then the counter after performing essential checks makes a roster of valid voters' ballots and their signature (for more detail see following and Figure 10).

Voter: Voter V_i retrieves the desired signature y_i of the ballot x_i by retrieving $y_i = R_A(d_i, r_i)$ and then checks:

IF A's signature is valid THEN V_i sends (x_i, y_i) to the counter through an anonymous communication channel

ELSE V_i claims the failure by publishing the set (x_i, y_i).

Counter: The counter C checks the signature y_i of the ballot x_i using A's verification key. If the check succeeds then C enters $< l, x_i, y_i >$ into a list.

After all voters vote, C publishes the list which contains the following information:

ENTRY	BALLOT+ INFORMATION
1	x_j, y_j
$\vdots$	$\vdots$
l	x_i, y_i
$\vdots$	$\vdots$

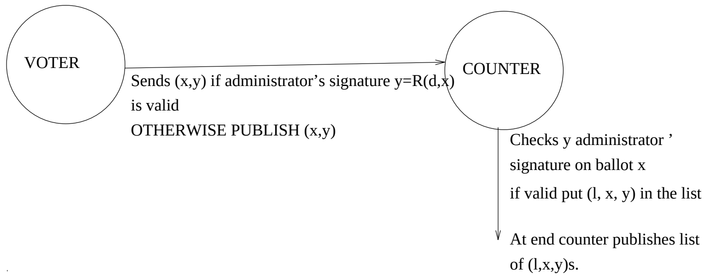

Figure 10: Second stage voting

Opening Stage.

Voter: Every Voter V_i checks :

IF the number of voters is not equal to the number of ballots in list THEN failure is claimed by opening r_i used in encryption

ELSE IF vote v_i is in the list THEN voter V_i sends key k_i and l to the counter through an anonymous communication channel

ELSE voter V_i claims failure by opening $< x_i, y_i >$ and the valid ballot plus the signature.

Counter: The counter C opens the ballot x_i, retrieves the vote v_i (or adds k_i, v_i to the list), and also the counter C counts the vote and announces the result.

Figure 11 illustrates the opening stage.

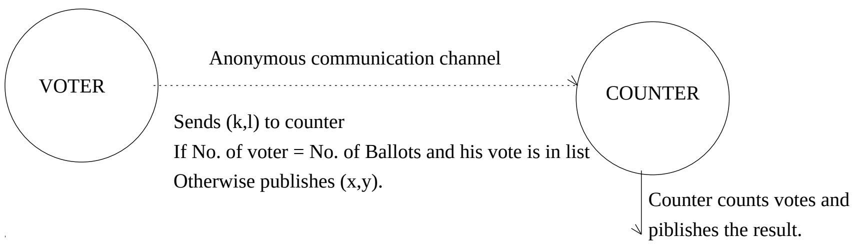

Figure 11: Third stage : opening

4.6.3 Analysis of Protocol.

Completeness. If every participant is honest, the result of the voting is trustworthy.

Privacy. Even if the administrator and the counter conspire, no relation can be found between a voter and his vote.

This is true because the relationship between voters and their ID's are hidden by the blind signature scheme that is used, and the ballot and the key are sent through anonymous channels and cannot be traced.

Un-reusability. True, if the blind signature is secure.

Eligibility. Assume that the ordinary signature scheme is secure then the dishonest person cannot vote. Since the voter must be eligible. This means that the voter must be on the list of the eligible voters.

Fairness. Counting is done after the voting is completed, so does not effect the voting.

Verifiability. Assumptions: no voter abstains from voting, and no one forges the ordinary digital scheme.

Therefore if the administrator and the counter conspire, they cannot change the result of the voting.

The assumption , "No voter abstains from voting", is impractical. Since always there is the possibility that a voter registers but does not vote, therefore if the administrator and the counter conspire, they can add false vote to the list and even worse the whole election will be disrupted since every voter checks this fact that the announced number of eligible voters by administration should be the number of ballot in the list published by the counter. Therefore regarding the previous assumption, always there is the possibility that the check fails.

However this scheme satisfies a considerable number of the criteria of a secure voting scheme and seems to be more practical than the previous proposals. But it needs more refinement and improvement in order to have a secure voting scheme from practical point of view.

5 Conclusion.

There is still a need for the improvement of voting schemes in order to ensure that we have a secure and practical voting scheme. The concern should lie in the issues and details that have not yet been explored or studied in depth. Clearly, further research and reflection is demanded to answer the major problems arising in the Electronic Election.

References

[1] J. Benaloh and M. Yung. Distributing the power of a government to enhance the privacy of votes. *Proceedings of the 5th ACM Symposium on Principles of Distributed Computing*, pages 52–62, August 1986.

[2] Colin Boyd. A cryptographic scheme for computerized general elections. In *Advances in Cryptology, Proceedings of EUROCRYPT'89, Lecture Notes in Science 434*, pages 617–625. Springer-Verlag, 1990.

[3] Hildebrand Carol and Hoffman Thomas. Campaign '92: High-tech politics. *Computerworld*, 26(27):12, July 1992.

[4] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[5] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[6] D. Chaum. The dinning cryptographers problems: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[7] D. Chaum. Elections with unconditionally secret ballots and disruption equivalent to breaking rsa. In *Advances in Cryptology, Proceedings of EUROCRYPT'88, Lecture Notes in Science 330*, pages 177–182. Springer-Verlag, 1988.

[8] D. Chaum. Blind signature for untraceable payments. In *Advances in Cryptology, Proceedings of CRYPTO'82, D. Chaum, R. L. Rivest, and A. T. Sherman Eds.*, pages 199–203. Plenum, NY, 1992.

[9] J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographically secure election scheme. *26th Annual Symposium on Foundations of Computer Science, IEEE*, pages 372–382, October 1985.

[10] Blankenhorn Dana. At& t to service democratic convention. *Newsbytes*, July 1992.

[11] W. Diffie and M. E. Hellman. New direction in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[12] Karen A. Frenkel. Computer and elections. *Communications of The ACM*, 31(10), 1988.

[13] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *Advances in Cryptology, Proceedings of AUSCRYPT'92*. Springer-Verlag, 1992.

[14] K. R. Iversen. A cryptographic scheme for computerized general elections. In *Advances in Cryptology, Proceedings of CRYPTO'91, Lecture Notes in Science 576*, pages 405–419. Springer-Verlag, 1992.

[15] Suleiman Kassicieh, Glen H. Kawaguchi, and Len Malczynski. Security, integrity and public acceptance of electronic voting: Managing elections in the 1990s. *Journal of Systems Management (JSM)*, 39(12):6–10, December 1988.

[16] Jacobson Louis. Let your fingers do the voting. maybe. (telephone voting). *The Wall Street Journal*, August 1992.

[17] M. Naor. Bit commitment using pseudo-randomness. In *Advances in Cryptology, Proceedings of CRYPTO'89, Lecture Notes in Science 435*, pages 128–136. Springer-Verlag, 1990.

[18] Charles P. Pfleeger. *Security In Computing.* Prentice-Hall,Inc., 1989.

[19] Mercuri Rebecca. voting-machine risks. *Communications of the ACM*, 35(11):138, November 1992.

[20] Roy G. Saltman. Accuracy, integrity and security in computerized vote-tallying. *Communications of the ACM*, 31(10), 1988.

[21] Loudermilk Stephen. Ncr keeps democratic convention in touch. (use of ncr corp.' 7054 integrated touch screen electronic voting system at the 1992 democratic national convention). *PC Week*, l9(28):18, July 1992.

[22] Hoffman Thomas. Electronic vote tabulation saves cash. *Computerworld*, 26(38), September 1992.