An Improved E-Voting Scheme using Secret Sharing based Secure Multi-Party Computation

Divya G. Nair[1], V. P. Binu and G. Santhosh Kumar

Cochin University of Science and Technology.
e-mail: [1] *divyagnr@gmail.com*

Abstract. E-voting systems (EVS)are having potential advantages over many existing voting schemes. Security, transparency, accuracy and reliability are the major concern in these systems. EVS continues to grow as the technology advances. It is inexpensive and efficient as the resources become reusable. Fast and accurate computation of results with voter privacy is the added advantage. In the proposed system we make use of secret sharing technique and secure multi party computation (SMC) to achieve security and reliability. SMC is a field of cryptography which create methods that enable parties to compute joint function over their private inputs preserving the privacy and correctness. Secret sharing is an important technique used for SMC. Multi-party computation is typically accomplished using secret sharing by making shares of the inputs and manipulating the shares to compute a typical function of the input. The proposed system make use of bitwise representation of votes and only the shares are used for transmission and computation of result. Secure sum evaluation can be done with shares distributed using Shamir's secret sharing scheme. The scheme is hence secure and reliable and does not make any number theoretic assumptions for security. We also propose a unique method which calculates the candidates individual votes without revealing the anonymity.

Keywords: E-voting, Online voting, Security, Privacy, Secret sharing, Secure multi-party computation.

1. Introduction

The election process can be considered as an extremely critical task which functions as the key of democracy.So the security violation in any aspect will be a sensitive issue. The election system must be sufficiently robust to withstand a variety of malicious behaviors and must be sufficiently transparent and comprehensible that voters and candidates can accept the results of an election. We are exploring the field of secret sharing based SMC for the E-voting application where privacy needs to be ensured from different aspects.

In recent decades some electronic voting schemes have been proposed to make election systems more robust and accurate. The two most important properties in all of these schemes are privacy and universal verifiability. Privacy ensures that it must not be possible to track down the relation between ballots and voters (anonymity) and also voter must not be able to prove who he voted for (anti-coercion).Universal verifiability means that the correctness of election procedure must be clear for everyone. In other words it must be guaranteed that voters votes have been tallied in the same way that they have intended.

Since 1980, much work has been put on developing secure electronic voting schemes. The major design criteria was that the voting system should guarantee that voter should remain anonymous during the entire voting process and also it must be at least as secure as traditional voting systems. The EVS should also guarantee the privacy. That is the vote made cannot be traced back to the individual. Detailed description of the existing electronic voting protocols, cryptographic primitives used and their properties are given in [9]. The main properties of a voting schemes are correctness, privacy, receipt-freeness, robustness, verifiability, democracy, fairness and efficiency.

It is known that none of the existing protocols satisfy all these properties at the same time.Most of the constructions meet the major requirements depending on the type of the elections performed. The most commonly used cryptographic primitives in EVS systems are Interactive zero knowledge proofs, secret sharing techniques, homomorphic encryption, re encryption, blind signature, shuffling based schemes and secure multi-party computation. Secret sharing is used in electronic voting in order to get the voting protocol to be robust against authorities coercions. The proposed system make use of homomorphic properties of secret sharing scheme for vote tallying. It also provides secure computation since shares are used for the computation.

Development of secret sharing scheme started as a solution to the problem of safeguarding cryptographic keys by distributing the key among n participants and t or more of the participants can recover it by pooling their shares. Thus the authorized set is any subset of participants containing more than t members. This scheme is denoted as (t, n) *threshold scheme*. The notion of a threshold secret sharing scheme is independently proposed by Shamir [21] and Blakley [3] in 1979. Since then much work has been put into the investigation of such schemes. Linear constructions were most efficient and widely used. A threshold secret sharing scheme is called *perfect*, if less than t shares give no information about the secret. Shamir's scheme is perfect while Blakley's scheme is non perfect. Both the Blakley's and the Shamir's constructions realize t-out-of-n shared secret schemes. However,their constructions are fundamentally different. The proposed system make use of Shamir's secret sharing scheme. Shamir's scheme is based on polynomial interpolation over a finite field. It uses the fact that we can find a polynomial of degree $t-1$ given t data points.A polynomial $f(x) = \sum_{i=0}^{t-1} a_i x^i$, with a_0 is set to the secret value and the coefficients a_1 to a_{t-1} are assigned random values in the field, is used for secret sharing. The value $f(i)$ is given to the user i as secret share. When t out of n users come together they can reconstruct the polynomial using Lagrange interpolation and hence obtain the secret. Shamir's scheme is perfect and ideal.The knowledge of $(t-1)$ pieces make secret data completely undetermined and also the share size is same as the secret. This scheme is easily computable when necessary data is available and it avoids single point of failure. Also it increases reliability, security, safety and convenience.

Secure Multi-party Computation [11] is an active research area in cryptography. It allows a set of parties to compute a function of their inputs while preserving input privacy and correctness. It is often the case that mutually distrustful parties need to perform a joint computation but cannot afford to reveal their inputs to each other. This can occur, for example, during auctions, data mining, voting, negotiations and business analytics. The problem is how to conduct such a computation while preserving the privacy of the inputs. A secure multi-party computation problem deals with computing any probabilistic function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation other than that participant's input and output.

SMC is dedicated to dealing with the problem of privacy-preserving cooperative computation among distrusted participants. It was first introduced by Yao in 1982 by putting forward the famous Millionaire's problem [22]. This method is used to implement cooperative computation with all participants private data, ensuring the correctness of the computation as well as not disclosing additional information except the necessary results. If no participant in the multi-party computation can learn more from the description of the public function and the result of the global calculation other than what he can infer from his own information, the computation protocol is secure.

SMC is accomplished here by secret sharing schemes. In secret sharing, the secret is not single handed, but multi-handed so that even if any of the parties involved in the computation are malicious, the secret can be reconstructed. A secret sharing scheme is verifiable if auxiliary information is included that allows parties to verify their shares as consistent. To handle malicious parties involved in any computation, the secret sharing scheme needs to be verifiable.

We present a modified scheme which uses secret sharing based secure multi-party for vote tallying. Section 2 gives related work in this area. Section 3 and 4 explains the proposed system and algorithm. Experimental results are given in Section 5. Security analysis of the scheme is mentioned in Section 6. Section 7 is the conclusion.

2. Related Work

The first electronic election scheme was proposed by David Chaum [5] in 1981. Electronic voting systems, catering to different requirements, have been widely implemented and used.There have been several studies on using computer technologies to improve elections. In 1987 Benaloh [1] presents an election scheme based upon secret sharing and the prime residuosity assumption. Boyd *et al.* [4] in 1990 proposed multiple key cipher without a trapdoor function and presents a voting scheme as an application of said cipher. Iverson and Kenneth [14] in 1992 made proposals of application of secret sharing technique and zero knowledge technique in secure election. Fujioka *et al.* [10] suggested a practical secret voting scheme for large scale elections in 1993. In this voting scheme voting is managed by an administrator who registers and authenticates voters and a counter who tallies votes. Benaloh *et al.* [2] also proposed a receipt free election scheme in 1994. In 1997 Cranor *et al.* [7] made an implementation of the Fujiyoka *et al.* scheme. In 1998 a receipt free voting scheme for large scale election is proposed by Okamoto [18]. Publicly verifiable secret sharing and its application to e-voting is proposed by Schoenmakers [20] in 1999. Lee and Kim [15] proposed a modified scheme in 2000. Hirt *et al.* [12] used homomorphic encryption scheme.Neff and Andrew [17] in 2001 suggested verifiable secret shuffle for e-voting. Malkhi *et al.* [16] in 2003 gave constructions without cryptographic technique.The scheme uses secret sharing techniques and homomorphism. General secret sharing using Chinese remainder theorem with application to e-voting is proposed by Iftene [13] in 2007. Generalization of the Pailliar's crypto system and application to voting is proposed by Damagaard *et al.* [8] in 2010. Chen *et al.* [6] suggested a scheme based on discrete logarithm problem and secret sharing in 2014. Enhanced scheme with more confedentiality and privacy is suggested by Pan *et al.* [19] in 2014.

3. Proposed System

The above mentioned E-voting schemes work on different technologies used in the process of E-voting. The proposed system focuses the generation of secret shares, secure distribution of shares and secure computation of votes obtained for each candidate. The algorithm works on a bitwise-pattern representation of votes.

The secrecy of vote is a sensitive issue which needs to be addressed with ultimate care. In the current Electronic Voting System, when a vote is casted, only that particular candidate's data (vote) is getting modified and it can be easily tracked. So in this paper we try to address this problem using SMC. Here

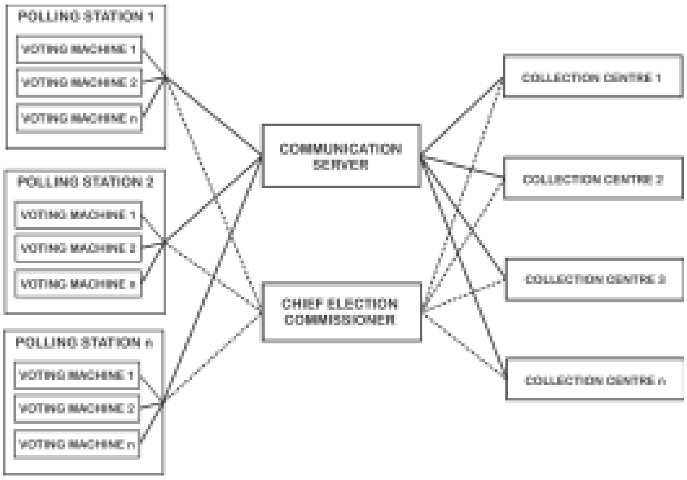

Figure 1. E-voting: System architecture.

when a vote is casted, rather than updating that particular candidate's data, the entire candidates data is getting updated which will make it difficult to track the vote.

In E-voting, each vote is considered as a secret and anonymization of voters identity should be protected. Each casted vote is divided into shares and these shares are distributed to multiple parties. The individual parties do not have any information about the vote casted by the voter by analyzing only the random share given to them. The individual shares are perfectly secure. The proposed algorithm also makes use of bitwise representation of votes which is distributed using Shamir's threshold secret sharing scheme and reconstructed using the coalition of specified number of parties.

The voting system makes use of 4 important modules to accomplish the secure voting process.

- Polling Station
- Communication Server
- Chief Election Commissioner
- Collection Centre

The System Architecture is shown in Figure 1 and the detailed architecture of each module is shown in Figure 2. Polling Station provides the interface for voting purpose. A polling station may contain many voting machines. It has a voting panel which contains the list of all candidates and their party symbols. Voting panel is loaded with this candidates information from a setup file which is managed by the Chief Election Commissioner. The vote casted by a voter is given to share generator module which contains the encoding and Shamir share generator module. The Encoding module will emcode the vote using a bitwise encoding algorithm as explained in section 4.1. Share Generator uses Shamir's secret sharing scheme for generating shares of the encoded votes. The number of shares generated is based on number of collection centers. This provides both security and trust which is implemented using Shamir's (k, n) threshold scheme in which any k shares of total n shares can be used for the reconstructing the original votes. The shares generated in the share generator module is sent to the collection centres through the Communication Server which manages the communication and coordination among all the other modules. This module handles Voting Machine Manager, Communication Manager and a

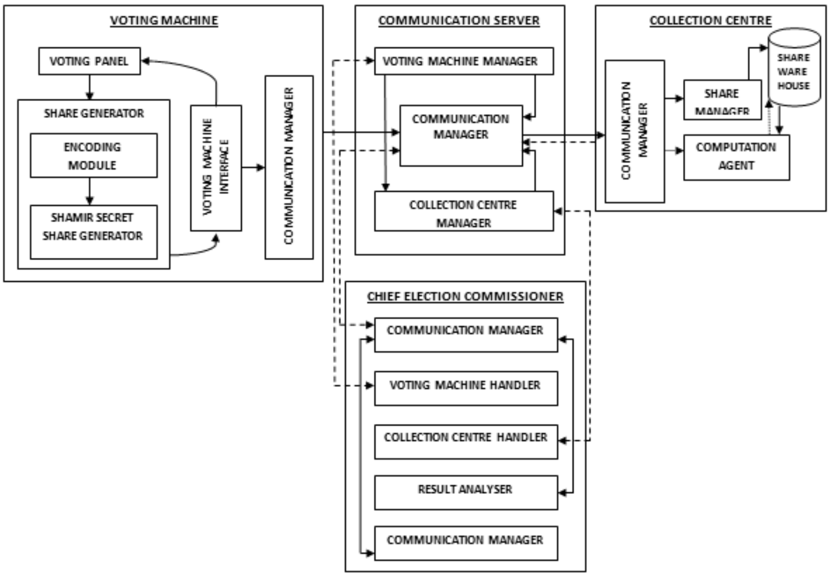

Figure 2. Detailed architecture.

Collection Centre Manager. Chief Election Commissioner module is working in an administrative role which manages the other modules. The Voting Machine Handler manages a set up file containing the list of candidates and their party symbols. Any modification made in the set up file will be reflected in the voting panel interface. The Collection Centre Handler manages the collection centres. For reconstructing the sum of votes for each candidate, k collection centres need to be selected randomly based on (k, n) threshold scheme. Collection Centre Handler randomly selects any of the k collection centres while reconstruction. Authorization of Collection centres is also managed by Collection Centre Handler. The Result Analyzer analyses the whole Election Process and declare the result.

Collection Centre (CC) manages all the shares and provides a local database for holding the shares. Usually a group of authorized parties behave as collection centres. Each collection centre will be having a local database which receives one share for every vote casted. Number of collection centres ((n) depend on the number of shares generated for each vote which in turn depends on the chosen threshold (k, n) scheme. Each collection centre CC_i gets i^{th} shares of all the votes. By getting a share, collection centre does not have any idea regarding either the vote casted or the voter. The Computation Agent performs summation of all the shares it received in its local database and it is used as a partial sum in the multi party computation. When the Collection Centers are selected for the final result computation, the partial sum is passed to Collection Center Handler module in the Chief Election Commissioner module. The Result Analyzer compute the result by reconstructing the encoded secret using Lagrange Interpolation in Shamir's scheme. The decoding algorithm is performed then, which will reveal the individual sum of votes for each candidate.

4. Proposed Algorithm

The algorithm for Share generation, distribution and result reconstruction is explained in Algorithm 1 EVS.

Algorithm 1: EVS

```
Input: Casted Vote
Output: Individual votes obtained by each candidate
m=no_of _voters()
n=no_of _candidates()
Set the bit pattern and candidate bit block of sizes $n(1 + lgm)$ and $(1 + lgm)$ respectively
for each voter i=1: m do
    encoded_vote = set_lsb (Candidate bit block)
    $V_i$ = bin_decimal (encoded_vote)
    Generate $k - 1$ random numbers $r_1, r_2, r_3 \cdots r_{k-1}$
    Construct the polynomial $f(x) = V_i + r_1 x + r_2 x^2 + \cdots + r_{k-1} x^{k-1}$
    for j= 1:n do
        Generate share $X_{ij} = f(j)$, where $X_{ij}$ is the $j^{th}$ share of $i^{th}$ vote
        Send the share $X_{ij}$ to $CC_j^{th}$ collection centre through the secure communication manager
for each Collection Centre $j = 1$:n do
    Sum of shares $SCC_j = \sum_{i=1}^{m} X_{ij}$
for each randomly chosen Collection Centre $CC_k$ do
    retrieve $SCC_k$ by the Collection Centre Handler
Use the k sum of shares $SCC_i$ (i:1 to k) interpolate and obtain the secret polynomial
Decoding the constant term in the polynomial will give the individual candidates total votes
```

4.1 *Encoding and Decoding of Votes*

The proposed system uses bitwise representation of votes for the computation of sum. The number of bits required is based on the number of voters and also the number of contesting candidates.

The encoding process is explained below with an example. Consider we have 8 voters and 3 candidates. Each time a vote is casted, equivalent binary pattern will be generated by the encoder module. 4 bits are required in this case for each candidate and a total of 12 bits since there are 8 voters and 3 candidates. The 12 bit vote pattern $b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0$ is initially set to 0. When a voter votes for candidate 1, bit b_0 is set to 1, vote for candidate 2 is represented by setting bit b_4. Similarly For candidate 3 bit b_8 bit is set.

The representation of vote is shown in Table 1. The share generator module will generate shares corresponds to the decimal equivalent of this binary pattern. Each time a vote is casted, a random polynomial is constructed and the shares corresponds to the secret vote is generated and sent to the Collection Centers. For the counting and result generation process, the Collection Center Manager receives k sums of shares.The Result Handler then apply Lagrange Interpolation formula to generate the polynomial $f(x)$ whose constant portion gives the sum of all votes. Decoding of this is then performed by taking each 4 bit combination of the binary pattern. The decimal equivalent of this is the candidates individual vote.

The Table 2 gives the list of all shares generated for this example using a (3,5) scheme in which 5 shares are generated out of which any 3 can be used for reconstructing the secret sum and obtain the candidates individual sum of votes.

5. Experimental Results

The implementation of the algorithm is done in java. The interfaces are developed with Netbeans IDE. Results based on 5 voters, 3 candidates and 5 Collection centers are considered. The shares generated based on Shamir's (3,5) scheme is shown Table 2. CC1, CC2 and CC3 are chosen for the computation of the result. SCC1, SCC2, SCC4 obtained are. The Result Analyser uses these values for interpolation. The

Table 1. Voting system: Representation of votes.

Voter	Candidate	Representation	Secret
Voter1	Candidate1	0000 0000 0001	1
Voter2	Candidate3	0001 0000 0000	256
Voter3	Candidate1	0000 0000 0001	1
Voter4	Candidate2	0000 0001 0000	16
Voter5	Candidate1	0000 0000 0001	1

Table 2. Voting system: Share generation.

Collection centre	CC1	CC2	CC3	CC4	CC5
Voter1	(1,91)	(2,269)	(3,535)	(4,889)	(5,1331)
Voter2	(1,327)	(2,498)	(3,769)	(4,1140)	(5,1611)
Voter3	(1,70)	(2,251)	(3,544)	(4,949)	(5,1466)
Voter4	(1,113)	(2,278)	(3,511)	(4,812)	(5,1181)
Voter5	(1,167)	(2,475)	(3,925)	(4,1517)	(5,2251)

polynomial obtained is $275 + 238x^1 + 255x^2$. The constant term 275 represent the sum of votes. Decoding of this will result in 0001 0001 0011. Each 4 bit represent the individual votes obtained by candidates.

6. Security Analysis

In traditional elections most ideal security goals such as democracy, privacy, accuracy, fairness and verifiability, are assured to a certain level given physical and administrative premises. The task of meeting the security goals is quite difficult in online elections.Another controversial pair of security properties in E-voting schemes are privacy and eligibility. It is difficult in online elections to unequivocally identify and check the credentials of a voter, while at the same time protecting the privacy of his/her vote. Computerized voting will never be used for general elections unless there is a protocol that both maintains individual privacy and prevents cheating.

A good voting system should satisfy number of generic voting principle. The authentication mechanism should ensure that only eligible persons can vote and should not allow any one to vote more than once. The proposed method satisfies the fundamental requirement of a secure voting protocol. No one can determine for whom anyone else voted. Even the authorities will not be able to determine this because the information is not stored anywhere. For each vote casted the shares are send to all the collection centres and the partial sum is updated. The shares generated using Shamir's scheme is information theoretically secure and no information about the vote casted is obtained from the shares. The consistency of the result obtained can be verified with k different set of shares.

7. Conclusions

The voting scheme in this paper employs a new tallying method and helps to compute the individual candidates vote easily. It achieves the highest efficiency when the number of candidates is small and guarantees strong vote privacy and reliability. We applied Shamir secret sharing scheme for Secure Computation of E-voting results. The algorithm is promising and it ensures secrecy, integrity and efficiency in the process of electronic voting. The share generation and reconstruction are based on the bitwise representation of shares and the proposed method have several merits compared with existing cryptographic methods which are computationally complex. The bits required for the representation of votes is directly proportional to the number of candidates and voters. Each vote is encoded and random shares are generated which are information theoretically secure. We have not mentioned the authentication of users and collection center in this paper which will be managed by the Chief election commissioner module. The shares can also be sent to the collection center through different channels for ensuring more security. The research is still challenging in the cryptographic community to design more powerful and secure schemes.

References

[1] Josh Benaloh, Verifiable Secret-ballot Elections, PhD thesis, PhD Thesis, Yale University, (1987).

[2] Josh Benaloh and Dwight Tuinstra, Receipt-free Secret-ballot Elections, In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 544–553, ACM, (1994).

[3] George Robert Blakley, Safeguarding Cryptographic Keys, In *Managing Requirements Knowledge, International Workshop on*, pp. 313. IEEE Computer Society, (1899).

[4] Colin Boyd, A New Multiple Key Cipher and an Improved Voting Scheme, In *Advances in Cryptology Eurocrypt 89*, pp. 617–625, Springer, (1990).

[5] David L. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM*, vol. 24(2), pp. 84–90, (1981).

[6] Chin-Ling Chen, Yu-Yi Chen, Jinn-Ke Jan and Chih-Cheng Chen, A Secure Anonymous E-voting System based on Discrete Logarithm Problem, *Applied Mathematics & Information Sciences*, vol. 8(5), (2014).

[7] Lorrie Faith Cranor and Ron K. Cytron, Sensus: A Security-Conscious Electronic Polling System for the Internet, In *System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on*, vol. 3, pp. 561–570. IEEE, (1997).

[8] Ivan Damgård, Mads Jurik and Jesper Buus Nielsen, A Generalization of Pailliers Public-key System with Applications to Electronic Voting, *International Journal of Information Security*, vol. 9(6), pp. 371–385, (2010).

[9] Laure Fouard, Mathilde Duclos and Pascal Lafourcade, Survey on Electronic Voting Schemes, *Supported by the ANR Project AVOTÉ*, (2007).

[10] Atsushi Fujioka, Tatsuaki Okamoto and Kazuo Ohta, A Practical Secret Voting Scheme for Large Scale Elections, In *Advances in Cryptology Auscrypt'92*, pp. 244–251. Springer, (1993).

[11] Shafi Goldwasser, Multi Party Computations: Past and Present, In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 1–6. ACM, (1997).

[12] Martin Hirt and Kazue Sako, Efficient Receipt-Free Voting based on Homomorphic Encryption, In *Advances in Cryptology EUROCRYPT 2000*, pp. 539–556. Springer, (2000).

[13] Sorin Iftene, General Secret Sharing based on the Chinese Remainder Theorem with Applications in E-voting, *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, (2007).

[14] Kenneth R. Iversen, A Cryptographic Scheme for Computerized General Elections, In *Advances in Cryptology CRYPTO91*, pp. 405–419. Springer, (1992).

[15] Byoungcheon Lee and Kwangjo Kim, Receipt-free Electronic Voting through Collaboration of Voter and Honest Verifier, In *Proceeding of JW-ISC2000*. Citeseer, (2000).

[16] Dahlia Malkhi, Ofer Margo and Elan Pavlov, E-voting without Cryptography, In *Financial Cryptography*, pp. 1–15. Springer, (2003).

[17] C. Andrew Neff, A Verifiable Secret Shuffle and its Application to E-voting, In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 116–125. ACM, (2001).

[18] Tatsuaki Okamoto, Receipt-free Electronic Voting Schemes for Large Scale Elections, In *Security Protocols*, pp. 25–35. Springer, (1998).

[19] Haijun Pan, Edwin Hou and Nirwan Ansari, Enhanced Name and Vote Separated E-voting System: An E-voting System that Ensures Voter Confidentiality and Candidate Privacy, *Security and Communication Networks*, (2014).

[20] Berry Schoenmakers, A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting, In *Advances in Cryptology CRYPTO 99*, pp. 148–164. Springer, (1999).

[21] Adi Shamir, How to share a secret, *Communications of the ACM*, vol. 22(11), pp. 612–613, (1979).

[22] Andrew Chi-Chih Yao, Protocols for Secure Computations, In *FOCS*, vol. 82, pp. 160–164, (1982).