Volume 2, No. 5, Sept-Oct 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Secured Flawless Effective Polling and Automatic Instant Counting

Joe Prathap P. M
IT Dept,
RMD Engineering College
Kavaraipettai, Tiruvallur District, India
joeprathappm@rediffmail.com

Brindha G. R.*
ICT Dept,
SASTRA University
Thanjavur, Tamil Nadu, India
brindha.gr@ict.sastra.edu

Abstract. In a biggest republic country like India, conducting a perfect election is always a tedious work. It not only takes lot of money and man power but time too. Usually the election conducted as various phases and the counting takes place some later days. This will lead frustration among the voters and candidates, as well as impose severe security risks. To avoid, instant counting is a best way. But, practically adopting this instant counting is still a long way to achieve. Our proposed work focuses this instant counting and result display with high security measures. With the help of effective Main Polling Server, conducting election and counting votes made semi automatic in our work. Thus, it leads the reduction of money, time, security risks and people involved.

Keywords: Polling, automatic counting, polling server, security

I. INTRODUCTION

In the traditional vote and counting method for polling, the voters register (manual seal) their vote on the ballot paper which contains candidates name, party name and symbol of that party. The voters used to drop the ballot paper in the ballot box. At the end of the day, the ballot box was sealed and all the ballot boxes in that particular area were collected and stored safely in a common place. On the day of counting, all the ballot boxes were opened in the presence of higher officials and counting started in front of authorized agents by the government officers. Then and there they used to announce who was leading with the difference in vote counting. This took a whole day. But when the electronic voting machine was implemented, the ballot paper information is there in Electronic Voting Machine (EVM) itself and the voter should press the proper button against the favorite candidate. After polling, here the transportation is same as in the manual counting.

The EVM used to keep the vote counting in its memory and officers will check each and every EVM and add the votes. Though the current voting machine reduces the troublesome counting process, still each and every EVM should be checked for the total votes, and then it should be consolidated. the current voting machine reduces the troublesome counting process, still each and every EVM should be checked for the total votes, and then it should be consolidated.

Electronic voting carries to the polling booth many positives, such as enhanced turn out, convenience for impaired people, and better accuracy and speed [1]. Unluckily, its implementation in different countries has been slow and/or the cause of dispute and disagreements. This is because of its pitiable design and performance of (some of) the systems which are deployed for elections many countries, as special studies have described and established [2][3][4][5].These reports have also exposed that such systems give serious holes in specification, plan, and execution. Such weak point exposes the system, and as a result to various attacks and threats, ranging from a denial of service to alteration of the results [6].

Formal methods permit designers to establish, check, or otherwise study appealing assets of a complex procedure whose behavior is mentioned conceptually, and then interactively renew the behavioral configuration to be as close to an execution as suitable for a given guarantee level. The use of formal techniques in the voting domain is still at an early stage. Some of the reports explain and express the viability of using formal techniques on specific components, such as the cryptographic protocols used to shield and convey data [7][8][9][10].

Still, the achievement of the new generation of voting machines depends on our skill to take advantage of on the lessons we learned with and analyzing the systems presently positioned. The reliability and guarantee of a compound and safety-critical system's accurate performance with respect to configuration can be attained if good manufacturing practices are suitably developed and used. Based on this, there are many approaches to deal with (some of) the issues stated above.

Among these, the usage of formal techniques has been given to improve the quality and security of complex systems [11][12][13][14]. Some others focus on the checking points of general matters of e-voting systems [15][16][17].

Fig. 1 shows the overview of proposed polling environment. This environment contains mainly four major sections. They are controlling and counting Main Polling Server (MPS), Local Constituency Server (LCS), and Collection of polling booths and most importantly Electronic Voting Machine (EVM) and polling process.

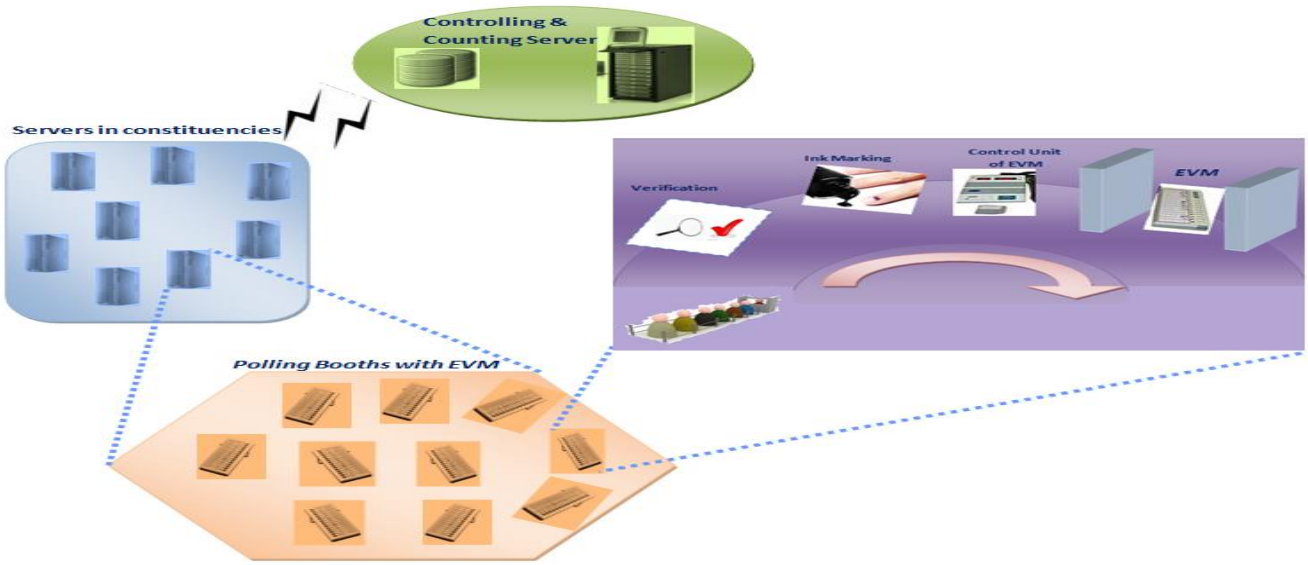

Figure 1. Overview of proposed polling environment

Controlling and counting server plays a major role in the system. It acts as a central database and share data with LCS. After polling it also receives the data from LCS for counting and display. According to living area and population, various constituencies are fixed. And for each constituency separate LCS was allotted, which can details from main server. Now every LCS contains several polling booths and each polling booth contains EVM Machines. So LCS has the control over the polling booths, which can be authorized and maintained by officials. The figure further explores the polling process in a polling booth. Inside the polling booth after the identity verification, ink marking with indelible ink was done. Now the voters are allowed to vote in EVM, after the controller pressing the ballot button, voter can press the voting button based on his/her wish.

Fig.2 shows the functional architecture of the main polling server. The MPS functions are divided into various management unit as shown below. In

EVM's data feeding management and allotment management, the data about EVM's controlling LCS and constituency name and Booth id with voter's population is fed.

Through the date and time duration management date of polling, starting time and ending time is entered. LCS setup management manages each EVM, which is under its control by checking proper working and control during transmission and signal exchange. Through the signal exchange secured transmission is enabled to activate EVM and receive the polled vote count. With the help of MPS-LCS session control management when to start polling when start transmission from EVM to LCS or any intermediate statistical data transmission required, are managed. After receiving the data from all EVMs, LCS will transmit the data to MPS in parallel, so that MPS will display the transmitted results. Then the statistical analysis management will provide the later analysis about the percentage of votes that each party got and comparative analysis of winning group alliances and opponent group.

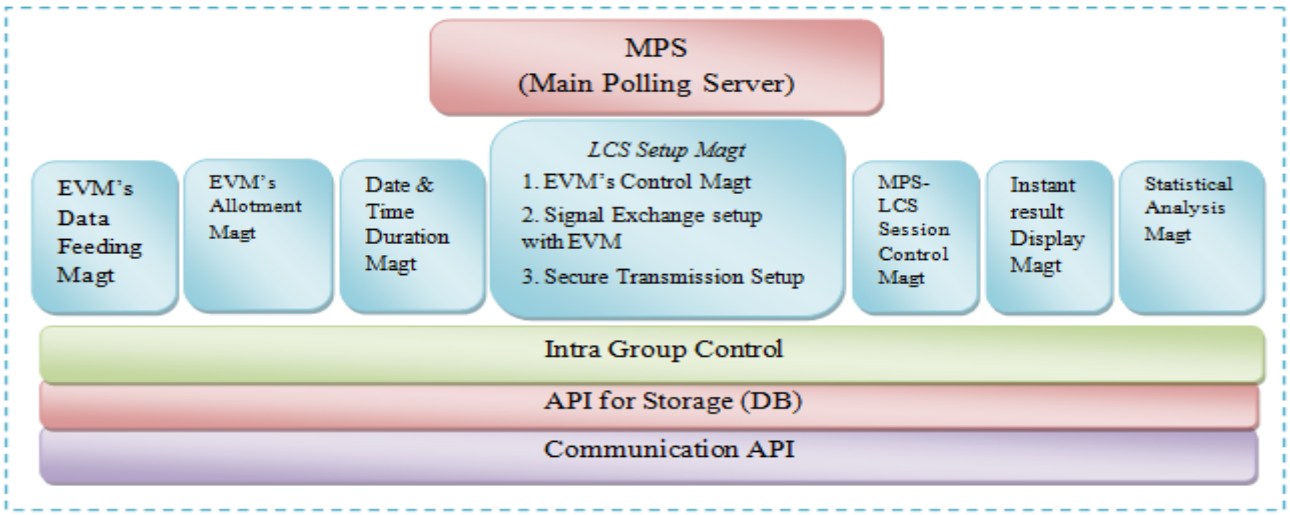

Figure 2. Functional architecture of main polling server

Fig.3 shows the information flow diagram which depicts the flow of information before polling which starts from MPS to LCS then to EVM, the top down method and after polling EVM to LCS then in turn to MPS, in bottom up approach.

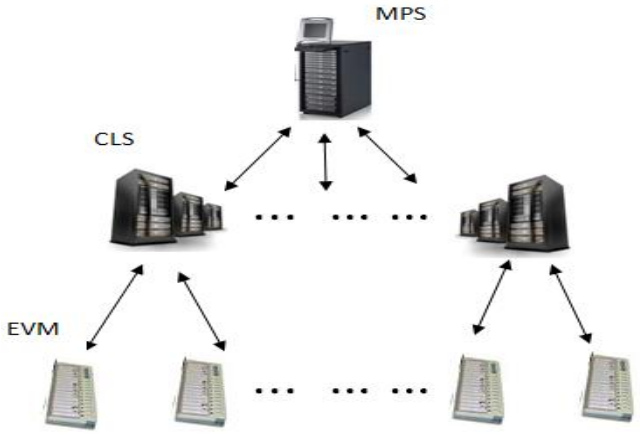

Figure 3. Information flow diagram

II. ALGORITHM

A. Processes in_EVM():

Begin

a. Before Election:

From MPS Database
{
For i= 1 to No_Of_ EVM
{
Declare Region_Name$_i$,Constituency_Name$_i$, Region_ID$_i$, Constituency_ID$_i$,Eelection_Date, Start_Time, End_Time, Number_of_Parties$_i$, Party_ID$_i$, Party_Symbol$_i$, Number_of _ Voters_Male$_i$, Number_of _ Voters_Female$_i$, Max_Count$_i$.
Declare Vote_Entry_Status as Boolean variable;
For i=1 to No_Of_Candidates
{
Assign Counter $_i$=0;
Current_Poll_Count$_i$ =0;
Assign LCS$_i$;
}}
}

b. During Election:

Begin {
If (Current_Date == Eelection_Date && Current_Time >= Start_Time)
{
Repeat //voting
If (Vote_Entry_Status==TRUE)
Detect Pressed Key
Counter$_i$ ++ ; // For Specific Parties
Current_Poll_Count++;
Store in Local Database
Vote_Entry_Status = FALSE;
Until (Current_Time == End_Time || Current_Poll_Count >= Max_Count)
} end

c. After Election:

Begin {
Declare EVM_Status = READ ONLY Mode;
Lock the DB for further modifications;
If (CLS_Request == TRUE)
{
Verify Digital_Signature;
Transfer Total_Polled_Count to CLS.
Receive ACK;
}
If(CLS_Receive ==TRUE)
{
For i= 1 to No.Of_Candidates
Tranfer Count$_i$ to CLS DB
Receive ACK.
SEAL THE EVM.
} end

B. Processes in_CLS():

Begin

a. Before Election:

For i= 1 to No._of_Constituencies
{
For i= 1 to No_Of_ EVM
{
Verify the EVM_Machine_Working_Status$_i$;
Transfer EVM to Polling_booth;
}}

b. During Election:

If (Current_Date == Eelection_Date && Current_Time == Start_Time)
{
For i= 1 to No_Of_ EVM
{
Send_Status_Signal == ACTIVE; //Activate the EVM
Periodic Status Check via Handshake Protocol with Polling_Officer;
} }

c. After Election:

Repeat
{
For i= 1 to No_Of_ EVM
{
Receive Confirmation_Signal from Authorised Polling_Officer;
Establish Secure Session with EVM ;
Verify EVM_Mode == READ ONLY;
Activate Data_Transmission
}
Until all details
If (MPS_Request == TRUE)
{
Establish Contact with MPS
Activate MPS-LCS_Session_Dialogue;
SEAL the LCS.
}

C. Processes in_MPS():

Begin

a. Before Election:

Collect & Feed Authenticated data to MPS. // all relevant details for entire polLing starting from voters list to inner constituency details.

```
For i= 1 to No_Of_ EVM
{
RESET;
ASSIGN New Details;
Verify Details
Transmit to Corresponding Physical Locations
}
```

b. During Election:

```
For i= 1 to No_Of_ CLS
 Periodic Status_Check;
```

c. After Election:

```
Begin
 For i= 1 to No_Of_ CLS
      {
      Establish Handshake_Session with CLS
       Digital Signature based data transmission
      Parallel Computation
      Update the Central DB Corresponding Column
      Automated Report Generation
      Instant_Result _Declaration
      }
End
```

III. ALGORITHM EXPLANATION

The proposed algorithm contains three major sections. They are EVM section, CLS section and MPS section. In each section the actions are further divided into three categories as follows: before Election, during Election and after Election.

A. EVM Section:

EVM plays a major role in polling. Only the EVM machine has the direct contact with the voters. Since each EVM machine has own unique ID & name we can easily recognize it. Before election, data feeding into the EVM machine is a major task. Care should be taken while entering the candidates list & their corresponding symbol, because these details will differ for various constituencies. Before feeding DB to the EVM, it should be assigned to the corresponding polling booth. That is the MPS Server and LCS Server knows every detail about EVM machine locality assignment in advance. This will help the system not only to do communication during and after polling, but also for instant counting and displaying results. The EVM machine is updated with other details like number of voters (male & female) in a particular polling booth, which eventually used to avoid overflow as well as fake voting in some extent. Particularly for every candidate assign a unique counter variable with initial value as 0. Whenever a voter polls his vote, the corresponding counter value will be increased by one.

The EVM machine not only synchronizes with MPS but also with corresponding LCS. Once the data feeding over into the EVM, the machine sealed and securely transmitted to the corresponding constituency. From that before the Election Day it transferred to allotted polling station. On Election Day, the chief polling officer for that particular polling station, open the sealed machine and based on direction from LCS, activate the EVM machine. The machine itself already set with timer and election date. After the closing time automatically the machine's database goes to READ ONLY status. After polling period over, the polling officer waits for comments from LCS. Once we got the LCS REQUEST signal, the secure handshake link between EVM and LCS established. Initially the EVM sent the total polling count to the LCS. After receiving ACK from LCS, upon further request EVM transfer the count values for individual candidates. The LCS will verify the count by adding individual count and compare it with the total count. If it matches, then data properly transmitted. Now LCS will sent the special acknowledge message to EVM that it received the data successfully. Now, again the EVM sealed and transferred securely to constituency head. The data are still kept in the machine it may use for later recounting if necessary.

B. CLS Section:

This part is actually acts as a supervisor to the particular constituency. It has the direct communication with MPS & EVM. LCS has the full control over the allotted EVM machines to it. Before and during election, it periodically gets the status about the machine functionality via the authorized polling officers. After election the secure handshake protocol established between EVM machines and LCS for data transfer. After getting data from machines from all the polling stations EVM, now LCS waits for the MPS Signal. After that MPS will take details from LCS via secure handshake protocol.

C. MPS Section:

The MPS is the heart of this polling system. It has a large database which contains all the details necessary for polling ranges from voters list to candidate list. The data's are clustered based on the constituency. And for each constituency the data further clustered as polling stations. Thus forming a tree like structures as mentioned earlier (Figure 3). Before election the data feeding with LCS and EVM was done under the supervision of MPS. During election it periodically contacts with LCS Server about polling status and check for any problems. We know that our election scheme is not a fully automated one, so higher officials also play a major roles here. After election the counting is take place. Since the MPS is a high end server, in parallel it can contact with more than one LCS at a time. It first checks the readiness of the LCS server. Once the session between LCS and MPS started means, the other applications (if any) happen between the networks will be disabled and the acknowledged based transmission (automatic) of information from LCS will happen. Once the session over, LCS kept in inactive mode with content are turned as read only. Actually in the MPS the data are received as per the pre allotted columns in database. So computation also made as automatic one based on the contents in various columns the results instantly appeared in various analysis format. With that the results will be announced instantly to all.

IV. USE CASE RELATIONSHIPS

The following section describes the various use case scenarios in the polling setup. Fig. 4 represents the communication between voters, candidates and the MPS administrator. Voter can register and withdraw, similarly the candidates can register their candidacy or withdraw.

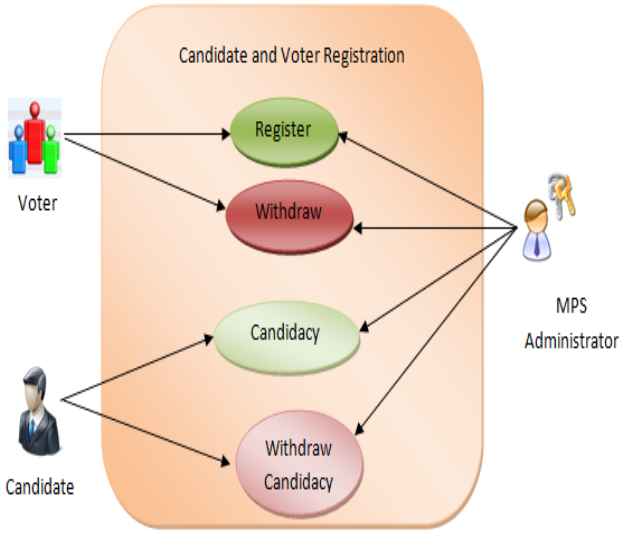

Figure 4. Candidate and Voter Registration Use case

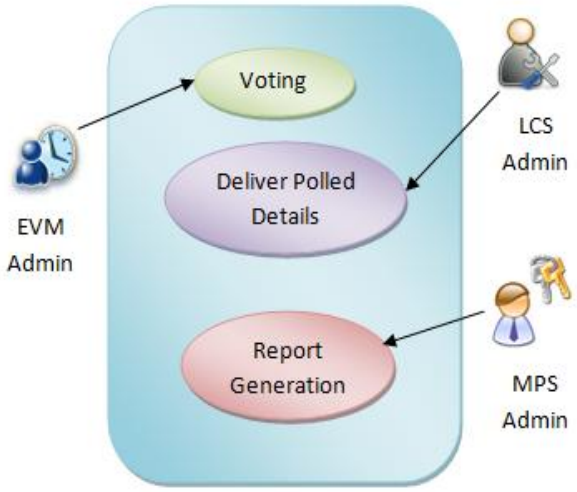

Figure 5. Voting Process Use case

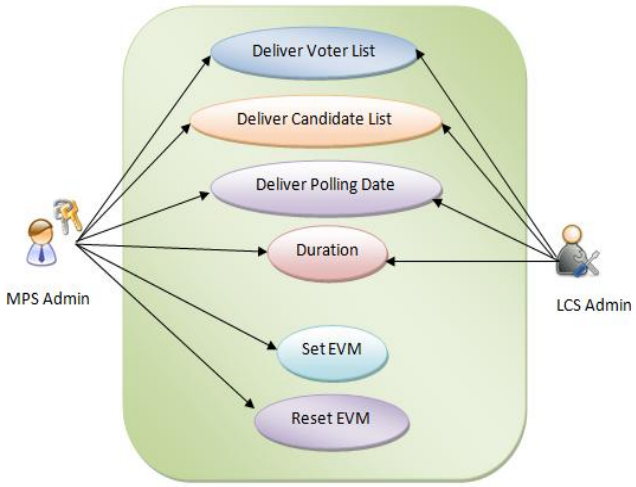

Figure 6. Relationship between MPS and LCS

The voting process use case contains mainly three actors; they are EVM admin, LCS admin and MPS admin. Their roles are clearly specified in Fig. 5.

The communication between MPS and LCS play major role with so many tasks. The various tasks are listed out in Fig. 6.

V. EXPERIMENTAL SETUP

A simulation environment was setup with the following ingredients. Assume that election is for a particular region or state.

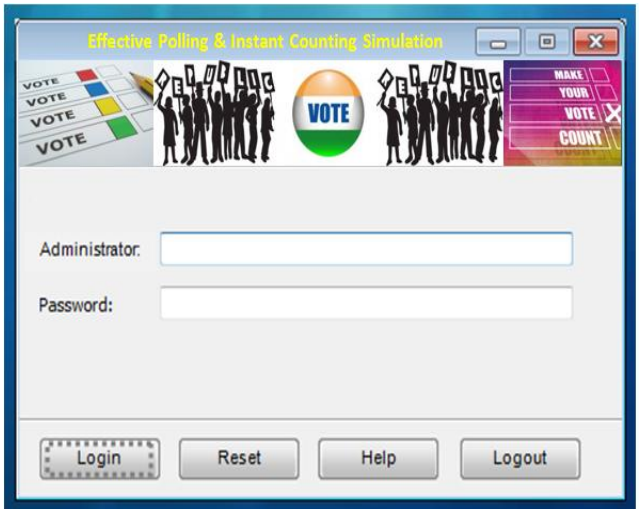

Figure 7. Administrator login page

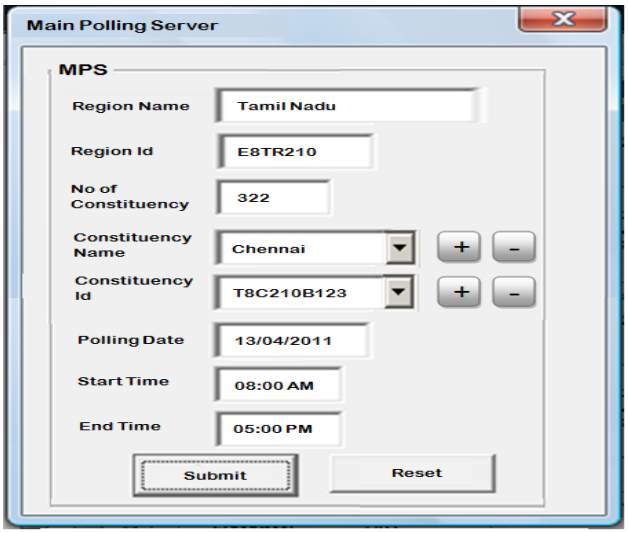

Figure 8. Main Polling Server

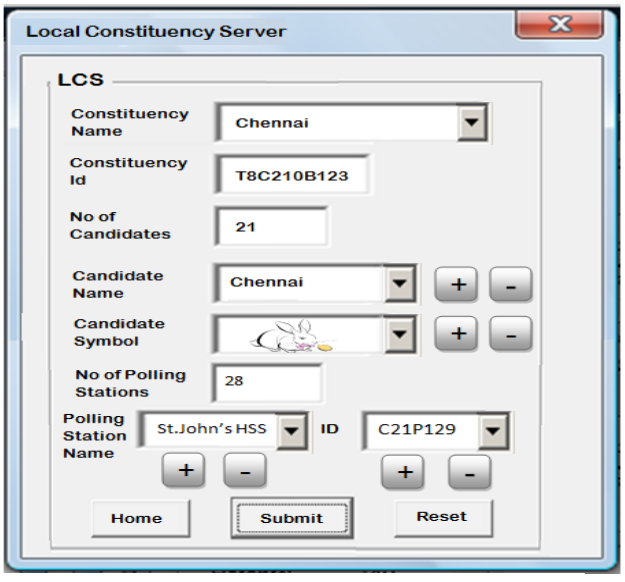

Figure 9. LPS for a Polling Station

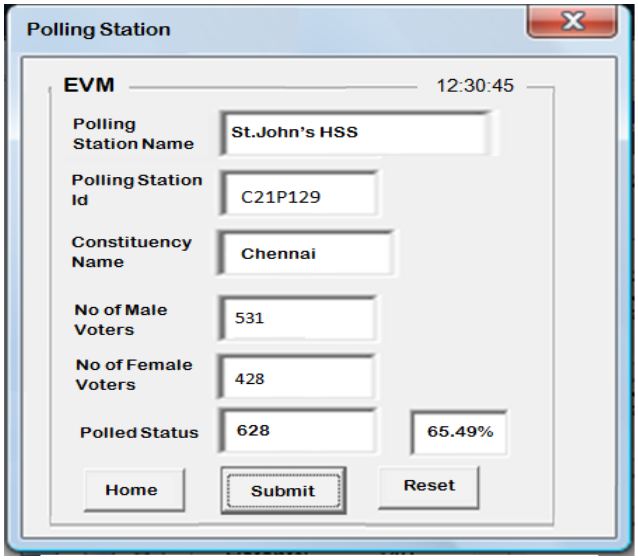

Figure 10. EVM for a Polling Booth

An authenticated login page was created with that only authorized people can enter into the system and can conduct the polling. Sample login page is depicted in Fig.7. In the MPS page the region name, region ID, total number of constituencies, Constituency list & its ID are uploaded. Moreover the election date, start time and end time of election also uploaded in the database. These details are represented in Fig. 8.

The necessary details required in LCS server side is uploaded with the help of this LCS page as shown in Fig 9, which include assigning proper polling station name and ID. Based on the total number of constituencies the individual data's about constituency was uploaded. In the EVM page polling station name, ID, corresponding constituency name, No. of male voters and female voters are loaded as shown in Fig. 10.In addition to that the number of polled in a particular time and percentage will be displayed.

Thus we uploaded the necessary details for the successful conduction of polling system. Based on the input values, random data's are generated for each polling station and the instant counting was taken place as per the simulation screenshots shown above.

VI. GENERATED REPORTS

The various reports are generated based on the needs and the winners were announced with all statistical analysis. This section provides some important screens generated by our simulation environment.

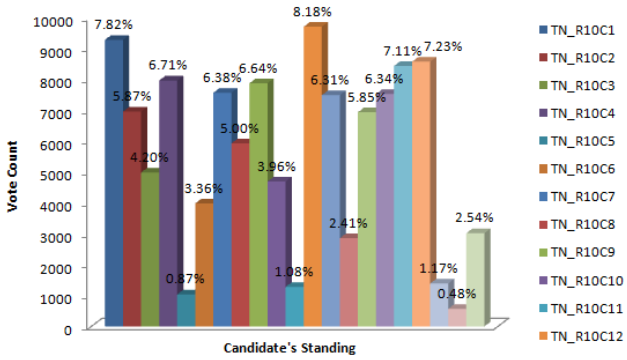

Figure11. Candidates standing in a particular constituency

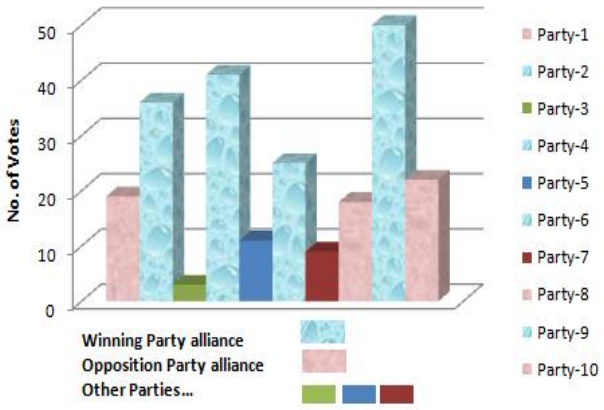

Figure12. Status chart of each party alliance

Fig. 11 shows the comparative analysis for the number of constituencies won by each parties and their percentage in total. Fig. 12 depicts the comparative analysis of each individual candidate standing. Fig. 13 illustrates the comparative analysis of each party alliance

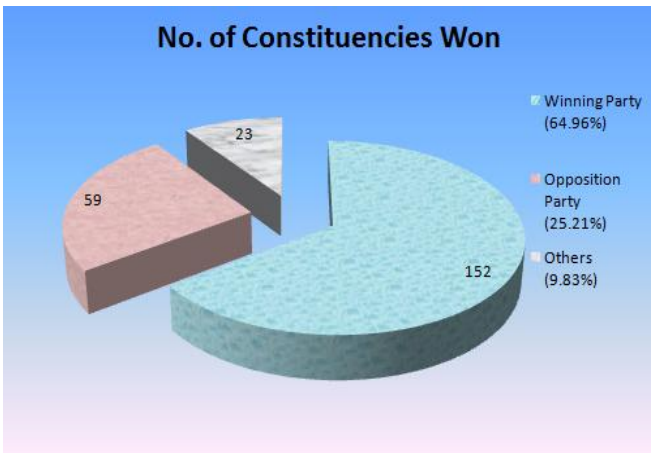

Figure13. Winning party & others position

VII. CONCLUSION & FUTURE WORK

Perfect polling system is a mandatory one for any republic and independent countries like India. Lots of money, man power security mechanism was used for the secure polling method. Always there is a great research in optimality in this issue. Our proposed work not only save the time, but also reduce lot of manpower. With this instant counting, candidates/voters are no need to wait for a long period in order to know their results. If the counting done separately after polling (as in the existing system) there is a lot of security issues, for example, stealing EVM machines, altering the contents and also possibilities of illegal result grabbing, etc. To avoid this official need to spent lot of measures for protection. These all difficulties are gracefully avoided in our proposed system. But still our method also has some flaws but that can be avoided by further refinement in the algorithm. The current algorithm doesn't mention details about postal votes. Here the entire system relies on central Main Polling Server. Even though it is highly secure there is a chance of single point of failure.

Once the MPS crashed means it is very difficult to conduct proper polling. Again our work is not fully automatic. Here voting done with the help of polling officers, whereas the counting and result analysis done as

fully automatic one. In future we can create a fully automated polling system by avoiding the man power requirements. Instead of setting poll booth, we can also prefer E-Voting for the election system. Now in already grown countries like America they adopted E-voting successfully. But growing countries like India not yet adopted this E-voting because of some practical difficulties. Thus our proposed algorithm will be a new milestone for conducting the optimal instant polling (especially for our country election).

VIII. REFERENCES

[1]. L.F Cranor, Electronic Voting: Computerized Polls May Save Money, Protect Privacy, Crossroads 2 (4) 12–16, ISSN 1528-4972,1996.

[2]. T. Kohno, A. Stubblefield, A.D Rubin, D.S. Wallach, Analysis of an Electronic Voting System, IEEE Symposium on Security and Privacy 0- 27,2004.

[3]. A. Aviv, P. Cerny, S. Clark, E. Cronin, M. Sherr , M. Blaze, Security Evaluation of ES&S Voting Machines and Election Management System, in: EVT: Proceedings of the conference on Electronic voting technology, USENIX Association, Berkeley, CA, USA, 1–13, 2008.

[4]. D. Balzarotti, G. Banks, M. Cova, V. Felmetsger, R.A. Kemmerer, W.K. Robertson, F. Valeur, G. Vigna, An Experience in Testing the Security of Real-World Electronic Voting Systems, IEEE Trans. Software Eng. 36 (4) , 453–473, 2010.

[5]. S. Wolchok, E. Wustrow, J.A. Halderman, H.K. Prasad, A. Kankipati, S.K. Sakhamuri, V. Yagat, R. Gonggrijp, Security analysis of India'selectronic voting machines, in: Proceedings of the 17th ACM conference on Computer and communications security, CCS '10, ACM, New York, NY, USA, 1–14,2010.

[6]. Richard A Kemmererb, Adolfo Villafioritaa a Foundation Bruno Kessler,via Sommarive, Formal Analysis of an Electronic Voting System: An experience Report Komminist Weldemariama, 18, TN 38123, Trento, Italy bDepartment of Computer Science,University of California Santa Barbara, CA 93106-5110.

[7]. A.Juels, D. Catalano, M. Jakobsson, Coercion-resistant Electronic Elections, in: WPES: Proceedings of the ACM workshop on Privacy in the electronic society, ACM, New York, NY, USA, 61–70, 2005.

[8]. S. Kremer,M.D Ryan, Analysis of an Electronic Voting Protocol in the Applied Pi-Calculus, in: M. Sagiv (Ed.), Programming Languages and Systems — Proceedings of the 14th European Symposium on Programming (ESOP'05), Lecture Notes in Computer Science, Springer, Edinburgh, U.K., 186–200, 2005.

[9]. S. Campanelli, A. Falleni, F. Martinelli, M. Petrocchi, A. Vaccarelli, Mobile Implementation and Formal Verification of an e-Voting System, in: Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, IEEE Computer Society, Washington, DC, USA, 476 – 481, 2008.

[10]. S. Delaune, S. Kremer, M. Ryan, Verifying Privacy-Type Properties of Electronic Voting Protocols, J. Comput. Secur. 17 (4), 435–487, 2009.

[11]. R.A Kemmerer, Integrating Formal Methods into the Development Process, IEEE Software 7 (5) 37–50 1990.

[12]. D. Xu, K.A Nygard Threat-driven Approach to Modeling and Verifying Secure Software, in: Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering, ASE '05, ACM, New York, NY, USA, 342–346, 2005.

[13]. M. Lowry, D. Dvorak, Analytic Verification of Flight Software, IEEE Intelligent Systems 13 (5) 45–49,1998.

[14]. C.L. Heitmeye, M.M Archer, E.I Leonard, J.D. McLean, Applying Formal Methods to a Certifiably Secure Software System, IEEE Transactions on Software Engineering 34 (1).

[15]. B.I. Simidchieva, M.S. Marzilli, L.A. Clarke, L.J. Osterweil, Specifying and Verifying Requirements for Election Processes, in: Proceedings of the international conference on Digital government research, Digital Government Society of North America, 63–72, 2008.

[16]. A. Villafiorita, K. Weldemariam, R. Tiella, Development, formal verification, and evaluation of an E-voting System with VVPAT, IEEE Transactions on Information Forensics and Security 4 (4) , 651–661, 2009.

[17]. C. Sturton, S. Jha, S.A. Seshia, D. Wagner, On Voting Machine Design for Verification and Testability, in: Ehab Al-Shaer and Somesh Jha and Angelos D. Keromytis (Ed.), ACM Conference on Computer and Communications Security, 463–476,2009.