Remote Internet voting: developing a secure and efficient frontend

Vinodu George · M. P. Sebastian

Received: 11 February 2013 / Accepted: 22 June 2013 / Published online: 11 July 2013
© CSI Publications 2013

Abstract Electronic voting is an essential component of the e-governance of a country for establishing the people's choice in selecting the political leadership. Poll station voting is generally considered to be the most successful form of the election process. Both traditional and electronic versions of the poll station voting had many implementations in general elections, worldwide. The remote poll station voting scheme allows voters to participate in the election process if they have an access to any one of the poll station, no matter where they are at the time of voting. However, this scheme suffers from declining participation due to the inconvenience to the voters in reaching the poll stations. Also, this scheme needs a secure, private network for connecting the poll stations, making it very expensive. The remote Internet voting is very user convenient, which allows voting from any Internet connected computer, from anywhere. However, security, reliability and complexity issues have restricted the wider adoption of this scheme in large-scale elections. This paper proposes a secure and efficient frontend voting protocol using a trusted platform module for remote Internet voting with trusted third party authentication protocol.

Keywords Electronic voting · Real-time election · Remote Internet voting · Trusted platform module · Voter authentication · Trusted third party authentication

1 Introduction

Electronic voting could gain much attention in both technological and theoretical areas of research in the recent past. E-voting, in general, simplifies and automates the election process. Furthermore, it speeds up the process and minimizes the counting mistakes. Many schemes have been proposed in the literature for e-voting over the last two decades. Majority of these methods are applicable only with poll station or kiosk based remote poll station voting [1]. The current literature on e-voting is mainly concentrating on the privacy and security aspects of the poll station voting. The inconveniences associated with poll station voting methods often cause lower voter turnout. Remote Internet voting scheme is generally suggested as the best solution for improving the voter participation in all kinds of elections. However, concerns related to the security of the platform and reliability of the remote machines restrict their wider adoption. Internet voting systems are already in use to a small extent in the elections of United States, United Kingdom, Switzerland and Estonia [2]. The insecure implementations of the current e-voting schemes over the public Internet have resulted in many security infringements. This emphasize the need for better security protocols in the case of remote Internet voting schemes.

1.1 Need for cryptographic voting

In an Internet voting system, accessed through an Internet-enabled device, the votes cast are transmitted to a server over the public Internet. This can also be considered as a direct recording electronic (DRE) [3] voting system in which the voter's choice is directly recorded on a server. By doing so, the voters are assured that their selection of the candidate has correctly captured and all eligible votes

V. George (✉)
Department of Computer Science and Engineering,
LBS College of Engineering, Kasaragod, India
e-mail: vinodu.george@gmail.com

M. P. Sebastian
Faculty of Information Technology and Systems Area,
Indian Institute of Management, Calicut, India
e-mail: sebasmp@iimk.ac.in

are correctly tallied. Another advantage of Internet voting is prevention of coercion. It means that an outsider cannot persuade or pressurize a voter to act in a certain way, even if the voter is willing to be influenced. One of the major requirements of a coercion resistant [4] system is receipt-freeness [5] by which a voter cannot prove how he/she has voted. These contradictory requirements present a major challenge: how the voters can be assured to trust the election result, without proving to a potential coercer to whom he cast the vote. Cryptographic techniques are needed for implementing such verifiable, secret ballot elections. Only cryptographic voting scheme [6] can achieve both verifiability and ballot secrecy at the same time.

1.2 Classification of Internet voting systems

Internet voting can be classified as follows:

Poll station Internet voting: In poll station voting, voting is conducted in a controlled polling place. Each voter is assigned a polling station and he can cast the vote only from that station or kiosk. Polling officials are responsible for the conduction the election and voter authentication.

Remote poll station Internet voting: In remote poll station Internet voting, voter is permitted to cast the vote from any poll station. Major advantage of the remote poll station voting is that it does not require voters to go his hometown to vote, and thus reduces costs and improves the “convenience” of the voter.

Remote Internet voting: In a remote Internet voting, the voters can cast their vote from a remote Internet enabled computer or from any access device. In remote Internet voting, it does not require voters to go somewhere to vote and it could be conducted at any uncontrolled remote location. There will not be any poling officials to authenticate and monitor the polling. Remote Internet voting offers a lot of saving in the capital and operating expenditure. However, there are many concerns with Internet voting such as security, availability, authentication, anonymity, verifiability, transparency and secrecy. Even though it maximizes the “convenience” of the voter, remote Internet voting is the most challenging voting process. This gives scope for additional work in remote Internet voting schemes.

1.3 Modular approach for designing Internet enabled e-voting systems

A complete voting system is an integration of a set of different processes like voter registration, voter verification, casting of the vote, encrypting the vote, issue of receipt (if required), storing the vote, tallying, declaration of the result, providing proof of correctness (if required), etc.

Fig. 1 Frontend and backend modules of a voting System

In designing such a complex system, a modular approach is the preferred method.

It is observed that the voters are directly involved only in a small number of steps in the voting process, say, voter registration, voter verification, and casting of the vote. Direct interaction of voters is not required in the remaining steps except for some verification process. Hence the modules interacting with the voters can be separated from the other modules and this part of the voting system is generally known as the frontend of the system. The remaining modules for which frequent voter interaction is not required are considered to be part of the backend of the voting system [7]. The interfacing media between the backend and the frontend can be the public Internet or a private network. Figure 1 depicts the frontend and the backend modules of a e-voting system.

The voter registration needs to be done in person, in well advance, on proper authentication. Then he/she become an eligible voter. All the mandatory information should be collected during this process, which will be verified at the time of voting. Vote casting and other associated functions are online functions. Hence this module of the frontend is responsible for interacting with the voters, producing the ballots and encrypting them. Then, the encrypted ballot will be forwarded to the backend module through the interface. The transformation from plain ballot to the encrypted ballot should be performed in a publicly verifiable manner using standard cryptographic methods.

1.4 Mandatory requirements for a cryptographic voting scheme

The factors which are crucial for a cryptographic voting scheme include the following [8, 9].

Privacy: A secret ballot system is the backbone of any modern democracy. Privacy in a voting system has two aspects. The first aspect is the privacy in vote casting,

which can be achieved by the protocols at the poll station. However, in remote poll station voting, it is the responsibility of each individual voter to ensure privacy. The second aspect of privacy is the traceability of a vote and the voter. In an ideal case, a voter should not be traced by the vote he/she cast. This aspect of privacy gains more importance in remote poll station voting. In a networked environment with the use of standard network protocols, it may be difficult to remove all the information regarding the voter from the vote cast. Moreover, this information is required for validating the vote. So the suggested solution to this problem is not to remove the trace, but to obscure it. A cryptographic method known as Mixnet [7] is one of the popular solution to deal with this problem.

Protection against malware: Voters can cast the vote from a remote computer or from any access device. There would be no polling officials to authenticate and monitor the voting device or the polling place. There is a chance of virus or other malware functions running on the remote machine, which may compromise the integrity of the vote or even can be harmful to the voting system. Hence there should be some mechanism to authenticate the remote platform that it is threat-free.

Coercion resistance: It is the inability of someone other than the voter to know how he/she has voted. The adversary may attempt to force a voter to cast the vote in a particular manner or may force the voter to abstain from the voting procedure by obtaining the voter's private key in remote Internet e-voting. There is also the possibility of the adversary to represent the original voter at any stage of the voting process. A coercion-resistant voting scheme assumes the intervention of a very powerful adversary and the voting scheme shall resist any such adversarial attack.

Catalano et al. [4] proposed the concept of coercion-resistance. According to their scheme, the coercers can have more privileges. Most of the coercion resistant voting schemes support multiple casting of the vote and only the last cast vote will be taken for the final tally. Such a voting scheme must be strong enough to prevent such adversarial attacks.

Scalability: The complexity of the underlying protocols in a voting scheme has a vital role in its adoption to a real world election. Even though the scheme is very secure, if it needs complex processing then it is not efficient and cannot be adopted for a general election. An efficient voting scheme has to be scalable in terms of storage, computation, and communication needs with respect to the number of voters. In view of the fact that the cryptographic building blocks of a voting scheme necessitate more computations, the scalability factor becomes a significant parameter in a real world election.

Integrity: The voting schemes must be tamper-proof and error-free. All the votes cast must be correctly read and

taken for the final tally. Invalid votes, if any, should be rejected. Universal verifiability mechanisms can be used to confirm the accuracy of the system.

Verifiability: Privacy and verifiability are two conflicting requirements of a voting system. So keeping the system verifiable, maintaining the privacy is a challenge. A voter would like to verify that his vote was correctly read and taken for the final tally. There are two types of verification that need to be supported. The first one is individual verifiability which enables the voter to verify whether his vote is taken for the final tally. The system must ensure that only the voter can verify his/her own vote and no one else can see or verify his/her vote. The second aspect is the universal verifiability in which any citizen can verify that all valid votes cast only were included for the final tally, and the tally process was correct.

2 Related work

Threats against remote Internet voting system range from common Internet attacks caused by viruses, malwares, Trojan horses, denial-of-service to voting system specific attacks. The ongoing research attempts on the Internet voting systems are mainly to provide the same level of reliability and trustworthiness as enjoyed by the polling booth voting systems. This section surveys the relevant research work in this direction.

One of the earlier proposals for remote Internet voting system was the secure voting in Symposiums (SVIS) [8] voting system, by Sako in 2001. This system was useful for selecting the best dissertation in workshops and symposia. SVIS collects encrypted votes and uses efficient mixing before the final tally. It was a mixnet [7] based remote voting system, meant for a small electorate. Hence the security provisions were limited, making it not suitable for a real time election. The system did not address the problems of improper influence, and malware too.

The secure electronic registration and voting experiment voting system (SERVE) [9], an Internet-based system, was built for the U.S. Department of Defense's Federal Voting Assistance Program (FVAP). The SERVE had many vulnerabilities and the project was subsequently discontinued. The major criticism against SERVE was that the Web server recorded the vote, along with the voter's identity. If the Web server had been compromised, then the voter privacy would have been broken completely.

Adder [10] is an Internet-based e-voting system developed by Kiayias et al. at the University of Connecticut. It is based on homomorphic encryption and free software released under the GNU GPL. This is a fully functional e-voting platform where the voter creates the encrypted vote which has the security properties such as robustness,

trust distribution, ballot privacy, auditability and partial verifiability. But the proposed model does not address the issues such as vote buying and selling, coercion resistance, voter verifiability, malware and other client-environment hazards.

Civitas [11] was another remote Internet voting system developed at Cornell University by Clarkson et al. This was an extension of the voting system proposed by Catalano et al. [4] that uses both re-encryption mixnets and homomorphic encryption. Civitas was claimed to be the first electronic voting system with coercion-resistant remote Internet voting. It was also the first system implemented to guard against unauthorized access. However, the tabulation and verification processes were slow, and no provision against malware.

Helios was proposed by Adida [12] as the first open-auditable web-based voting system. It was a web implementation of the Tuinstra and Benaloh [5] challenge voting system, similar to the Adder voting system. The major difference between the two is that in Helios, the encryption of vote is done by the election authority; where as in Adder, it is done by the voter. Helios was meant for the elections of small online electoral communities. In spite of the easiness in use, speed and open source nature, it was not suitable for a major election. It also did not address the issues of improper influence or of the malware.

Much literature is not available on remote electronic voting (remote Internet voting) using trusted platform module (TPM), especially with coercion-resistant electronic voting. Sadeghi et al. [13] makes a study on the properties of a trustworthy client which protects the voters from malwares and the voting system from corrupt voters. It proposes a method based on Trusted Computing in combination with a secure operating system. However, they have not proposed any protocol for a secure online voting. Fink et al. [14] proposed a method just for achieving E2E verifiability [15] by means of trusted computing.

Smart and Ritter [16] propose a method for remote Internet voting protocol where the authorities can verify the state of the voter's machine via the TPM. Their protocol allows a voter to remain anonymous, while satisfying their eligibility to vote via a novel use of the Direct Anonymous Attestation (DAA) protocol [17]. The protocol supports a coercion-resistance voting scheme to prevent any attack from a coercer. Even if the voter behaves not in the predefined way, still the protocol provides revocable anonymity. Even though the proposal meets most of the requirements of secure online voting, the complexity of the protocol makes it impractical for a real time election. For achieving the coercion resistance property, the protocol uses a method similar to that of Catalano et al. [4], which need complex computation (i.e., $O(n^2)$ in terms of the number of voters). Also for achieving the anonymity, it uses

the DAA protocol of TPM, which involves complex computation. Moreover, this method does not address the voter authentication besides the remote platform authentication.

2.1 Need for developing a secure and efficient frontend

Table 1 compares the performance of the existing popular remote Internet voting schemes. One can see that the scheme by Smart and Ritter [16] is meeting all the mandatory requirements for a voting scheme, except for scalability. In political elections, the voting system may be used by several millions of voters. Thus, scalability of the system is of paramount importance in large democracies like India. Hence there is a need for further research to develop new, scalable protocols for remote Internet voting.

2.2 The voting model

An election system consists of the following sets of entities [18]:

Authority: Denoted by A , is responsible for issuing the keying material, i.e., the encrypted credential [19] and the candidate slate (Ballot) to the voters.

Validator: Denoted by D , is responsible for the validation of all votes cast.

Talliers: The set of n_T Talliers, denoted by T , are responsible for mixing the ballots, jointly counting the votes and publishing the final tally.

Voters: The set of n_V voters, denoted by V , are the entities participating in the given election.

The Authority, Validator and Tallier are not single entities, but are a set of entities, jointly performing the responsibilities. This is to eliminate dishonest entities. If at least one member of an entity from the set is honest, then no kind of illegitimate activity is possible with the votes or voters.

2.3 The voting life cycle

The simplified life cycle model of an electronic voting scheme consists of four phases.

Initial setup: This phase initializes all the technical part of the organizational structure and the election system. Registration of voters required, in person, for availing the Remote Internet Voting facility.

Voter registration: This phase is mandatory for each election. From a trusted third party (TTP), voter has to get a certificate for the session key. During the registration, the Authority will issue an election id and an encrypted ballot.

Voting: Votes are cast in this phase. Along with the vote, the voter has to submit the trusted platform information and the secret information, bound with the platform to the

Table 1 Performance comparison of the existing remote Internet voting schemes

Property scheme	Privacy	Protection against malware	Coercion resistance	Scalability	Integrity	Verifiability
SVIS	No	No	No	No	Yes	No
SERVE	No	No	No	No	Yes	No
Adder	Yes	No	No	Yes	Yes	No
Civitas	Yes	No	Yes	No	Yes	Yes
Helois	Yes	No	Yes	No	Yes	Yes
Smart and Ritter [16]	Yes	Yes	Yes	No	Yes	Yes

Validator. The voter's digital ballot gets authenticated maintaining the ballot secrecy.

Validation: This phase validates all votes, the invalid ones are discarded. Validation is done with the help of platform information submitted with the encrypted vote (provided by the voter) and by comparing the information stored with Validator.

Tallying: This phase finalizes the result of election from the valid votes.

3 A scalable frontend protocol for remote Internet voting

Our objective is to propose a scalable, secure and efficient frontend voting module for remote Internet voting. For verifying the voter and his platform, the proposed method uses the Trusted Third Party (TTP) authentication model of TPM [20] instead of the DAA [21]. In DAA, the identity of the user or of the specific TPM will not be revealed. The user of the platform proves the authenticity anonymously. However, the prover and verifier require complex computational exchanges for proving the authenticity. Table 2 compares between the DAA and TTP attestation methods. Figure 2 shows the steps for TTP Authentication. For keeping the anonymity of the voter in TTP Authentication,

Table 2 Comparison between TTP and DAA attestation methods

Trusted third party attestation	Direct anonymous attestation
With the help of a privacy Certification Authority (CA)	No need of any privacy Certification Authority
Voter details will be available to the attesting authority	Voter privacy will be retained from the attesting authority
Attestation is done with PKI of CA and the Voter	Complex attestation procedure (Using the Camenisch-Lysyanskaya signature scheme)
Needs only one interaction between Voter and CA say request for certificate and issue of certificate	Need to run two separate protocols: join and sign, each of which need multiple interactions between the Voter and the verifier (Authority)

the tallying of votes is isolated from the Validator. The Tallier and Validator are two separate entities, where the Validator forwards the validated votes to the Tallier for final tallying. So the Validator sees only the ballots which are encrypted with the Tallier's key and the Tallier gets only the encrypted ballots without any mapping information to the voters.

3.1 The trusted platform module

The TPM is defined as a hardware instantiation designed to provide trusted features as specified by the Trusted Computing Group (TCG) of the Trusted Computing Platform Alliance (TCPA) [22] specification. It is used for the secure generation of the various cryptographic keys, and is capable of doing remote attestation and storing information in a secure sealed manner. This chip is normally attached to the computer system for authenticating that platform. At the time of manufacturing, an Endorsement Key (EK), which is a private/public key pair, is created and the private part is securely concealed within the chip which cannot be extracted from the chip.

Like EK, some other keys are also integrated to TPM. The verifier or the user has to trust these entities implicitly and there is no method to measure them explicitly. These entities actually provide the Roots of Trust to a TPM. There are three roots of trust which include Roots of Trust for Measurement, Roots of Trust for Reporting and Roots of Trust for Storage [21].

3.2 The implementation approach for the frontend protocol

The frontend protocol has three phases, the Setup phase, the Registration phase, and the Voting phase. The initial setup, which is a one-time process, includes all the formal procedures executed by the voter and the Authority for the voter to avail the remote Internet voting facility. The functions in this phase include submission of application by the voter for availing the voting facility, and proper in-person physical verification and authentication of the

Fig. 2 Trusted third party authentication

1. In Person, one time registration with Authority
2. After proper authentication and registration, issue a live CD contains secure programs, biometric code etc.
3. Request for the certificate of session key from TTP
4. Get session key certificate from TTP
5. Send election registration request to the Authority
6. Obtain secure ballot and election id from Authority after proper registration
7. Cast the vote and submit to Authority

Fig. 3 The remote Internet voting sequence

voter by the Authority. After successfully completing these steps, the Authority will issue a live CD for each voter which contains all the software for enabling him/ her to cast the vote through the Internet. This completes the first phase.

The second phase of the protocol makes a voter register for an election. The voter needs a session key and a certificate issued from a TTP for each election using the election framework (this certificate proves the authenticity of the keys of the voter). After verifying the authentication documents, the Authority issues the encrypted ballot with the election id to the voter. This completes the second phase. The third phase is the voting phase where the encrypted vote along with the other required information is sent to the Validator for proper validation. After validating the vote and the voter, the Validator detaches the encrypted vote from the attached voter information and sends only the encrypted vote to the Tallier for the final Tally.

The user and platform authentications are done through binding the user data with the platform which can be done with the help of TPM. The live CD and the TPM eliminate the chances for introduction of malware or any other

unauthenticated program into the election framework by a voter. Authenticity of a voter's key can be proved with a TTP. Figure 3 depicts the sequence for remote Internet voting using a TPM.

The next three sections will give the attestations process, trust measurement of entities and verification of log information in the remote platform for remote Internet voting.

3.3 The remote attestation process

A remote voting platform (computer system) needs authentication by the voting Authority, before a voter can use it for voting. The whole system (including the hardware platform, booting, operating system and voting software) should be measured and proven to the voting server. The "Remote attestation" creates a hash key summary of the current software and hardware configuration. This allows the Authority to verify that the voting software has not been modified. If the voter installs malicious software on his/her machine then he/she will not be able to cast the vote because the voting server will reject his/her ballot due to the wrong values in the Platform Configuration Registers (PCRs) [23]. Isolation of the voting process runs on the voter's virtual machine will ensure the prevention of malware functions or eavesdroppers running in the voting virtual machine.

3.4 The core root of trust measurement (CRTM) process

The core root of trust for measurement is the BIOS boot block code. This piece of code is considered trustworthy [22]. It measures the integrity value of other entities in the system in a reliable manner, and preserves it during the lifetime of the attached platform. It is an extension of BIOS, which will run first to measure the other parts of the BIOS block before passing control to the BIOS. The BIOS then measures the hardware platform and the boot loader, and then passes control to the boot loader. The boot loader kernel measures the OS and then passes control to the OS. Thus, the OS can load the application program and control

Fig. 4 CRTM sequence of Execution

can then be transferred to the application program. After loading the application or during the load process, one can check for the PCR values to see if it is running on a good configuration. Any change in any of these entities will result in a new PCR's value and the Authority can decide whether to continue or not with the new configuration.

Figure 4 shows the sequence of execution. “1”, “2”, “3” and “4” represent the fingerprint measurements. “a”, “b”, “c” and “d” represent transfer of control of executions. “i”, “ii”, “iii” and “iv” represent the storing of fingerprints in the PCRs. The sequence of execution is “1”, “i”, “a”, “2”, “ii”, “b”, “3”, “iii”, “c”, “4”, “iv”, and “d”. Consider an application “P” which wants to protect some secret information such that no malicious software can access it. We assume that the BIOS (*B*), the boot loader (*L*) and the operating system (*O*) have all been modified to support sealed storage [21]. When the computer boots, the ROM code measures (computes a hash) the BIOS (*B*) and invokes $PCRE_{extend}$ with a canonical PCR index, e.g., 1, as $PCRE_{extend}(1, BIOS)$

As a result, the TPM computes

$$PCR1 \leftarrow H(0 || BIOS)$$

Then the BIOS will be executed by the ROM code. The BIOS performs its usual verification and initialization routines and then loads the measurement of the hash value of the boot-loader into the TPM. It could choose the same PCR as

$$PCR1 \leftarrow H(H(0 || BIOS) || Boot_Loader)$$

After the $PCRE_{extend}$ operation, the BIOS can launch the boot loader. Similarly, the boot-loader will extend a measurement of the OS (*O*) into the TPM before starting to execute it. Finally, the OS will extend a measurement of the hash value application program (*P*) into the TPM and then launches the application program as

$$PCR1 \leftarrow H(H(H(H(0 || BIOS) || Boot_Loader) || O_S) || P)$$

The application generates the secret data *C* and seals it under the current value of PCRs by invoking

$$Seal(1, E(Credential)_{PK^{Av}}) \rightarrow (C, MAC_{Kroot}(1, h))$$

Where *h* is the current content of *PCR1*. If any code of these software changes, then the *Unseal* operation fails. The properties of the hash function *H* guarantee that it is extremely difficult to find two entities with the same hash function, and hence if an attacker invokes the *Unseal* operation, the TPM will refuse to decrypt the secret information *C*.

3.5 Stored measurement log (SML) verification process

Measurement of an entity in a PC platform is done by hashing the entity with a hash function like SHA-1 [24]. The result is the measured value or fingerprint of that entity. An entity in a PC platform could be an application executable, a configuration file or a data file.

Consider a situation with two entities *A* and *B* such that Entity *A* measures the hash value of entity *B*, and we get *B*'s “fingerprint” which is processed as follows:

1. This fingerprint is stored in a Stored Measurement Log (SML), which is stored outside, not protected by a TPM unit.
2. Entity *A* then inserts *B*'s fingerprint into a PCR (via the PCR's *extend* operation).
3. Control is then passed to *B*.

A stores *B*'s fingerprint to a PCR before passing the control to *B*. The benefit of following this order is that *B* cannot hide its existence (that it had been running). Imagine that *B* is a malicious program; it tries to hide itself by removing its fingerprint in the SML. But, *B* cannot remove its fingerprint from the PCR, because the PCR is protected at hardware level. No part of the system can write directly to the PCR. It is computationally infeasible to find another program whose hash value is the same as *B*. Though this integrity measurement mechanism does not prevent an entity from misbehaving or being malicious, its presence is logged in by the SML, which is guaranteed by

Fig. 5 Data flow in the registration sequence

Fig. 6 Data flow in the voting sequence

the TPM, creating an unforgeable record. This is true for all entities that have been loaded. The Authority can decide whether to trust or not the system based on this record.

3.6 The frontend protocol

A complete voting process involves initial setup, voter registration, voting, validation and tallying. As our protocol is for the frontend, we concentrate on the initial setup, voter registration and voting phases. This frontend is compatible with any backend system where the validation and tallying are done separately.

Setup: The initial setup is the setting up of the technical part of the election system and the organizational structure. The different key pairs associated with each entity are generated here. PK^{Av} , PK^T and PK^{VD} are the public key of the Authority, Tallier and Validator, respectively.

The initial in-person voter registration for availing the remote Internet voting facility is also done in this phase. Initial registration of the voter includes in-person verification of the voter details, reading of his/ her biometric code, generation of the key pairs, and issuing of the live CD for that voter. After getting the live CD, the initial configuration should be done on the user machine by the voter using the application program available in the CD. Algorithm 1 describes the steps for the initial configuration.

Registration: Registration needs to be done for each election. The voter has to get a certificate for the session key from a *TTP*. During the registration, the Authority will verify the authenticity of the programs run on the remote

```

1. In person registration with Authority
2. After proper authentication, issue a live CD containing extended O.S. for enabling the TPM service, application for voting, encrypted credential using authority's public key and biometric code of the voter as part of the application program.
3. Boot from the live CD and do the verification using the PCR registers through the application in the CD
   PCREextend(1, BIOS);
   PCREextend(1, Boot_Loader);
   PCREextend(1, O_S);
   PCREextend(1, Appl_Pr);
   As a result, the TPM computes
   PCR1 ← H(0||BIOS);
   PCR1 ← H(H(0||BIOS)||Boot_Loader);
   PCR1 ← H(H(H(0||BIOS)||Boot_Loader)||O_S);
   PCR1 ← H(H(H(H(0||BIOS)||Boot_Loader)||O_S)||Appl_Pr);
   PCR5 ← H(0||Bio_Code);
   Seal(5, Bio_Code) → (CB, MACKroot(5, Bio_Code));
   Seal(1, E(Credential)PKAv) → (CCr, MACKroot(1, h));
   where h = H(H(H(0||BIOS)||Boot_Loader)||O_S)||Appl_Pr)
  
```

Algorithm 1 Initial configuration

```

1. Boot from the live CD and load the OS & Application program
   PCREextend(1, BIOS);
   PCREextend(1, Boot_Loader);
   PCREextend(1, O_S);
   PCREextend(1, Appl_Pr);
2. if Unseal(CCr, MACKroot(1, PCR1))
3. then Read the Biometric Code
   B_Code ← Read Biometric_Code;
   PCREextend(5, B_Code);
4. else exit
5. if Unseal(CB, MACKroot(5, PCR5))
6. then Create session key AIK for the communication with Authority
   Get certificate from TTP for the created AIK;
   Send to Authority
   (E(E(E(Credential)PKAv, PCR1, PCR5, SML)AIK, E(AIKcert)TTP)PKAv);
7. else exit
  
```

Algorithm 2 Voter registration sequence for each election

```

1. Do the proper verification
   D(E(E(Credential)PKAv, PCR1, PCR5, SML)AIK, E(AIKcert)TTP)PKAv
   D(AIKcert)TTP
   D(E(Credential)PKAv, PCR1, PCR5, SML)AIK
   a. Verify the AIK certificate with the TTP public key
   b. Use AIK to verify the signature on the E(Credential)PKAv and PCR values
   c. Recalculate this value from the fingerprint list within SML (By applying the PCR's extended operation on these fingerprints)
   d. Compare the calculated value with PCR's value. If the PCR value and SML do not match, it implies that the SML had been tampered, and the verifier should decide not to trust the attester
   e. If they do match, the verifier goes through the fingerprint list in SML and looks for any unapproved entity. This can be done by a whitelist or blacklist
   f. If no bad entity is found, the challenger can decide that the local party is trustworthy.
2. Attach the candidate slate Bt and Election id Eid with E(Credential)PKAv and encrypt with AIK
   E(E(Credential)PKAv, Bt, Eid)AIK
3. Send the packet to voter
  
```

Algorithm 3 Steps followed by the authority for the registration of voters for each election

platforms using Roots of Trust for Measurement. The authenticity of the user platform can be verified by the *TTP* authentication protocol. After authenticating the program and the platform, the voter authentication is done with the biometric code, integrated with the SML value of the platform. On successful verification, the Authority will issue an election id E_{id} and the encrypted ballot B^t . Algorithm 2 describes the steps followed by the voter for registration and Algorithm 3 shows the steps followed by the authority for the voter registration for each election. Figure 5 shows the flow of data in the registration process.

1. Receive the packet from Authority and decrypt with proper decryption key for getting E_{id} and the ballot B^t

$$D(E(\text{Credential})_{PK^{Av}}), B^t, E_{id})_{AIK}$$
2. If the voter selects a candidate B^k from the B^t (send by the Authority), then reset the write-in ballot bit
3. Encrypts the vote B^k with the public key PK^T of the Tallier.
4. Otherwise, sets the write-in ballot bit and casts his/her vote in favor of his/her own candidate, using PK^T (not in the candidate list).
5. A voter V_i thus generates and send the vote to Validator the vote as

$$E(E(B^k)_{PK^T}, TS, E_{id}, WBB, E(\text{Credential})_{PK^{Av}})_{PK^{VD}})_{AIK}, E(E(AIK_{cert})_{TTP})$$

Algorithm 4 Steps followed by the voter for casting the vote

Voting: The casting of vote is done in this phase. Along with the candidate selection B^k , the voter has to submit the voter's credential, the time stamp TS , the write-in ballot bit WBB , the election id E_{id} , and the certificate from the TTP to the Validator. The trusted platform information and the secret information, bound with the platform are also to be submitted. The voter's digital ballot needs to be authenticated maintaining the ballot secrecy. Algorithm 4 shows the steps followed by the voter for casting the vote. Figure 6 shows the flow of data in the voting process.

The proposed frontend protocol can be tailored with any required backend protocol subject to the condition that tallying and validation should be done by separate entities [14]. Now the proposed frontend voting protocol, together with the underlying backend protocol (with separate entities for tallying and validation), will be satisfying all the mandatory requirements of a scalable, secure and efficient voting system.

4 Efficiency analysis

Even though the efficiency of a voting system mainly depends on the backend system, an efficient frontend system also directly affects the voting process. Major functions involved in the frontend are initial setup, registration for each election and voting. Initial setup involves in person registration, collection of all voter details, issue of voting software and initial setup of the remote platform. After the offline process, all these functions can be done in linear time. Initial setup of the remote platform consists of initialization of PCR registers and sealing of biometric code and voter credential which can be done in linear time.

Registration for each election by the voter includes obtaining the certification from TTP and submission of that certificate to Authority for obtaining encrypted ballot. Since the TTP is trustworthy entity, the Authority is free from further verification. Otherwise the Authority is responsible for verifying the authenticity of all the voters through DAA method. Finally voting also can be done with few encryption operations which also can be done in linear

time. However these frontend operations are performed on the access device at the user end and executed only for casting the vote. Backend system is responsible for verification of the voter and storing the vote cast. Hence the efficiency of the backend system has a vital role in the overall performance.

5 Security analysis

The proposed system is vulnerable to many attacks varying from generic Internet attacks to system specific attacks, as the voting is done on the Internet from a remote platform. There are many methods to defend against the generic Internet attacks. This paper concentrates only on the analysis of system specific attacks.

In general, the vulnerabilities on a remote Internet machine can be placed under the following categories [13].

- Threat from malicious codes such as viruses, Trojan horse, etc.
- Threat from an untrustworthy voter.
- Unauthorized access to the voting system.
- Access to the voting software from an unauthorized platform.
- Untrustworthy authorities.

5.1 Threat from malware

This threat is mitigated by the roots of trust for measurement (RTM) feature of TPM . By using the log information of all programs running on the machine, the authority can ensure that only authenticated programs are running in a predefined sequence on the remote machine. If this sequence is found to be violated during the registration or voting phase, access will be denied for that machine.

5.2 Threat from malicious voters

An authorized voter can access the system only through the software issued by the authority. If the voter manipulates the software for his selfish purposes, then it may not behave in the predefined way. The authenticity of the software is always verified before the registration and voting process. This is done by comparing the fingerprint or unforgeable hash code of the programs running on the remote platform. Since this software is issued by the Authority, the original fingerprints are available with the Authority. This is compared with the fingerprints of the programs running on the remote machine, which is available in the log file submitted by the remote machine.

5.3 Unauthorized access to the voting system

Since the system is Internet based, there may be a tendency for unauthorized people to access the voting system. This is prevented by the authentication using the biometric code of the voter, which is collected at the time of the initial in-person registration of the voter with the Authority. The biometric code of the voter will be read by the application program and sent to the authority along with the other information. This is matched with the original biometric code and the authenticity of the voter is established.

5.4 Access to the voting software from an unauthorized platform

The *TPM* ensures the platform authenticity of the remote machine. This is achieved by the Roots of Trust for Measurement and Roots of Trust for Reporting futures of the *TPM*. Thus, the access to the voting system from any unauthorized platform is prevented.

5.5 Threat from untrustworthy authorities

All the Authorities mentioned here like registration Authority, Validator and Tallier are a group of Authorities rather than a single Authority. All of them have their own key shares and together will do the functions like registration of voters, validation and tallying of votes, etc. The authorities can cheat the system only if all of them collude, and we assume that there will be at least one honest authority in each of the groups.

6 Conclusion

A secure and efficient frontend protocol for remote Internet voting is proposed in this paper. It provides true, trustworthy authentication of the involved parties and remote machine using a trusted platform module. The attestation by the TTP for authenticating the remote platform keeps its anonymity. The anonymity of the voter is maintained by performing the validation and tallying of the vote cast by separate Validator and Tallier. The Validator validates the vote without knowing the vote content and the Tallier tallies the vote without knowing the voter identity. Only the validated vote will be forwarded to the Tallier. It also allows a voter to verify that his/her vote is counted in the final tally. By combining with an appropriate backend protocol, the resulting voting scheme could serve for any kind of elections. A limitation of this protocol is the need for the voter to keep his/her machine free from all kinds of malware to cast the vote. Even though the current proposal is for the authentication of a single voter, it can be extended

for a group of voters (for example, family members). This is suggested as a topic for further research.

References

1. Mercuri R (1992) Voting machine risks. *Communications of ACM*. 35(11): 138
2. ACE Project (2012) e-Voting on countries. [Online]. <http://aceproject.org/ace-en/focus/e-voting/countries>. Accessed 8 Dec 2012
3. Mercuri R (2002) A better ballot box. *IEEE spectrum*, pp 46–50, Oct 2002
4. Catalano D, Jakobsson M, Juels A (2005) Coercion resistant electronic elections. In: 4th ACM workshop on privacy in electronic society (WPES'05), ACM Press, New York, pp 61–70
5. Tuinstra D, Benaloh J (1994) Receipt-free secret-ballot elections. In: Proceedings of the 26th annual symposium on theory of computing (STOC'94), ACM Press, New York, pp 544–553
6. Sampigethaya K, Poovendran R (2006) A framework and taxonomy for comparison of electronic voting schemes. *J Comput Secur* 25:137–153
7. Popoveniuc S (2009) A framework for secure mixnet-based electronic voting. Ph.D. thesis, George Washington University
8. Sako K (2007) On svis project [Online]. <http://kathrin.dagstuhl.de/files/Materials/07/07311/07311.SakoKazue.Slides.ppt>. Accessed 31 March 2007
9. Rubin A, Simmons B, Wagner D, Jefferson D (2004) A security analysis of the secure electronic registration and voting experiment (SERVE). Technical report. <http://servesecurityreport.org/>. Accessed 15 May 2011
10. Korman M, Walluck D, Kiayias A (2006) An Internet voting system supporting user privacy. In: 22nd annual computer security applications conference (ACSAC '06), Dec 2006, pp 165–174
11. Clarkson MR, Chong S, Myers AC (2008) Civitas: toward a secure voting system. In: IEEE symposium on security and privacy, May 2008
12. Adida B (2008) Helios: Web-based open audit voting. In: Fourteenth USENIX security symposium (USENIX Security 2008), July 2008
13. Sadeghi A-R, Schulz S, Volkamer M, Alkassar A (2006) Towards trustworthy online voting. In: First Benelux workshop on information and system security (WISec '06)
14. Fink RA, Sherman AT, Carback R (2009) TPM meets DRE: reducing the trust base for electronic voting using trusted platform modules, (2009) IEEE transaction on information, forensics, and security no. Special issue on voting, 2009
15. Essex A, Carback R, Clark J, Popoveniuc S, Sherman AT, Poorvi V, Chaum D (2008) Scantegrity: end-to-end voter verifiable optical-scan voting. *IEEE security and privacy*, May/June 2008
16. Ritter E, Smart M (2010) True trustworthy elections: remote electronic voting using trusted computing. University of Birmingham, Technical Report CSR-10-12
17. Trusted Computing Group (2011) TPM main part 1, 2, 3, specification version 1.2 level 2. Revision 1161, March 2011
18. Acquisti A (2004) Receipt-free homomorphic elections and write-in ballots. <http://eprint.iacr.org/>. International Association for Cryptologic Research, Cryptology ePrint Archive Report 2004/105
19. Brands S (2008) A technical overview of digital credentials. Technical Report. 2002. [Online] <http://www.xs4all.nl/~brands/overview.pdf>. Accessed 17 Jan 2008
20. Bajikar S (2002) Trusted platform module based security on notebook PCs. Trusted Computing Group, White Paper

21. Parno B (2007) The trusted platform module (TPM) and sealed storage. RSA Laboratories, Technical Reports
22. TPM main part 1, 2 & 3 specification (2011) 1.2.2.1.1.6.1st ed.: Trusted Computing Group Publication
23. Trusted platform module (TPM) (2007) Intel Corporation, Quick Reference Guide
24. Van Oorschot P, Vanstone S, Menezes A (1997) Handbook of applied cryptography. CRC Press, Boca Raton