IMPLEMENTATION ISSUES IN SECURE E-VOTING SCHEMES

Riza Aditya[*], Byoungcheon Lee[*,**], Colin Boyd[*] and Ed Dawson[*]

[*]*Information Security Research Centre,*
Queensland University of Technology

GPO BOX 2434, Brisbane, QLD 4001, Australia
{r.aditya, b6.lee, c.boyd, e.dawson}@qut.edu.au

[**]*Joongbu University*

101 Daebak-Ro, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea
sultan@joongbu.ac.kr

ABSTRACT

As cheating is an inherent threat to voting, it is essential that an e-voting system provides a high level of security. At the moment, commercially available e-voting solutions mainly advertise their convenience, efficiency and low cost. On the other hand, cryptographically secure voting schemes in the literature are generally considered to be complex and inefficient for a real-world implementation. This paper examines implementation issues of cryptographically secure secret-ballot voting schemes. A survey of different schemes and various implementations is provided. The possibilities of hardware implementations for various cryptographic primitives are discussed. The paper provides a foundation in designing secure and practical e-voting schemes to produce a secure, efficient and publicly acceptable implementation of voting schemes in the real world.

Key Words: 18. Information Technology, 49. Information Security, 9. E-business.

1. INTRODUCTION

Voting is fundamental to any consensus-based society. It is a basic tool to reveal a group's opinion on a matter that is under consideration. National election is the most important application of voting in democratic societies. This is because the result concerns the governing of a nation, and affects the lives of its citizens. Other example applications of voting range from passing legislations in parliament, and decision making in shareholder meetings to student council elections, and reality television shows.

As the size of population and the need to cover larger constituencies increases, paper-based (traditional / manual / conventional) voting becomes cumbersome for large-scale voting. Other factors such as accuracy, efficiency (tally speed), and convenience make the transition to electronic voting inevitable.

1.1. Electronic Voting

Starting with the use of mechanical machines in voting (lever voting machines) as documented by Jones (2001), automation of voting using electronically supported devices has gained significant popularity. Voter turnout and the problems encountered during the 2000 US presidential election case in Florida have also positively influenced the acceptability of electronic voting (e-voting). Recent use in popular reality television shows (world idol, big brother, video hits) made e-voting a natural part of our everyday lives.

There are a number of different approaches to realise e-voting to date. Machine readable (create, read, count) ballot systems, Direct Recording Electronic (DRE) systems (using touch screen machines), voting using mobile/handheld devices, and Internet voting systems are all categorised as e-voting. It is important to understand that e-voting generically refers to "voting using some electronic means".

Compared to manual / paper-based voting, e-voting has the following fundamental advantages:

- Convenience
  E-voting is more convenient for voters. For national election, more polling booths can be set-up using remote connection for ballot collection. This reduces voters' travel time, and significantly increases voter turnout. For voting with a lower security requirement (e.g. reality television shows), using the Internet or mobile devices is the most convenient methods for voters to vote. Voters are allowed to vote from any location at their convenience. Voting from abroad is also possible.
- Efficiency
  Using some electronic means (e.g. optical mark sense sheets, touch screen voting, remote connection), tally stage to reveal voting result is made more efficient. Ballots tabulation and the aggregation of results from different polling locations can be done electronically. Using e-voting, voting result can be revealed in a timelier manner compared to manual counting of paper ballots.
- Accuracy
  A high profile case of manual counting inaccuracy was disclosed during the 2000 US presidential election in Florida, where a manual recount was performed. Using e-voting, human error can be eliminated in the tally stage. Ballot validity is automatically checked, and the counting is performed by software. Using certified software, the voting result obtained is more accurate compared to manual counting.
- Cost
  The use of electronic ballot removes the cost of producing a physical paper ballot. The use of some remote communication mechanisms also minimises the cost of transporting physical ballots for aggregation of voting result. Ballot counting automation using a computer program minimises administration overhead, and reduces the number of officials required for the counting process.
- Additional features
  Compared to the manual system, there are additional features that only an e-voting system can offer. These features include vote revocation, vote correction, and individual vote verification. Depending on the voting application, these functionalities might be desirable to have.

Aside from the above, an essential property required in a voting system is security. Whether it is fame, political power, financial gains or others, there is considerable motive for cheating in voting. Further discussion on security is provided in Section 2. The challenge in e-voting research is to design a system providing more functionality and security than the current manual / traditional one.

1.2. Main Contributions

This paper presents implementation issues in secure e-voting schemes. Security threats in a voting system are identified and discussed as a motivation to have a secure e-voting system. To have a secure system, security must be incorporated within the system. Three important phases (Figure 1) of having a secure system are identified as: design, development, and deployment. These phases are examined and analysed in this paper.

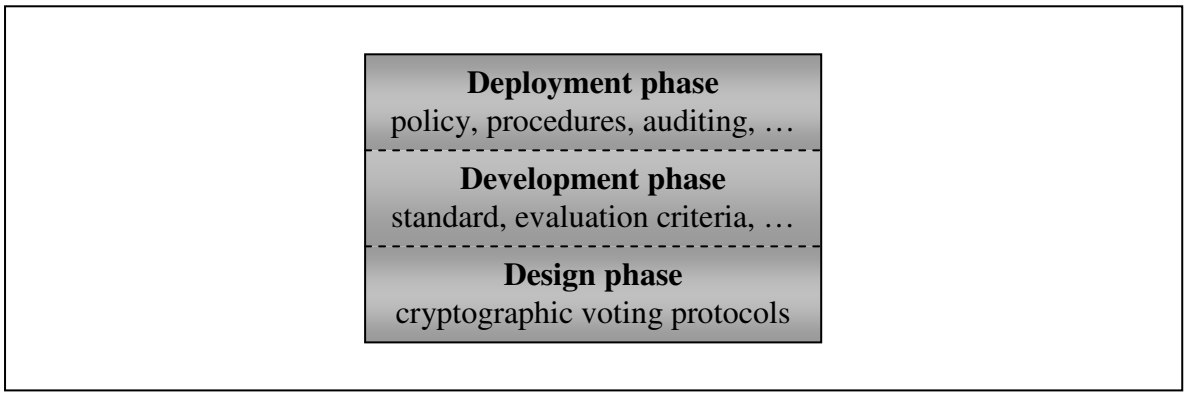

Figure 1. A Framework to Create Secure E-Voting Systems

A review of cryptographic voting schemes is provided as a fundamental design to build a secure voting system. Problems in the development phase and their possible solutions are identified. Further issues in the deployment phase and their possible solutions are also identified. The paper provides a foundation for creating a secure, efficient, and publicly acceptable e-voting system to be used in the real world.

1.3. Organisation of the Paper

The remainder of the paper is organised as follows. Section 2 offers background knowledge, security threats and security requirements of a voting system. Section 3 reviews two fundamental schemes in e-voting cryptographic protocol designs. Section 4 and 5 present issues and possible solutions in the development and deployment phase of a voting system respectively. Section 6 is a conclusion.

2. VOTING AND SECURITY

2.1. Overview

A voting scenario typically consists of four stages. They are:

1. Set-up stage

   During this stage, voting parameters are initialised. They include candidates, voters and authorities' eligibility criteria, voting procedures, ballot validity rules, and counting rules. Eligible candidates register themselves, and the registration and tally authorities are selected in this stage. Afterward, the voting parameters, candidates and authorities are made public such that they can be publicly known and verified.

2. Registration stage

   Voters are allowed to register themselves to the registration authorities during this stage. Their eligibility is determined by the criteria set in the previous stage, where ineligible voters are not allowed to register and participate in voting. Afterward, the list of registered voters is published for public verification.

3. Voting stage

   During the voting period, registered voters are then allowed to cast their votes as follows:

   a. Voter authentication: each voter is authenticated according to the list of registered voters in the previous stage, and those who are not found in the list are not allowed to participate in this stage.

   b. Vote registration: each of the authenticated voters then receives an empty ballot, and registers his/her vote in the ballot inside a physically private and secure location to avoid coercion/intimidation.

   c. Ballot casting: the ballot is then anonymised such that the "voter-vote" relationship is kept secret; in paper-based voting, this is achieved by using a sealed ballot-box where ballots are anonymised inside the ballot-box.

4. Tally stage

   In this final stage, all ballots from the previous stage are processed to obtain the voting result as follows:

   a. Ballot collection: the ballots from the voting stage are collected; in paper-based voting, ballots are obtained after the sealed ballot-boxes are opened by tally authorities.

   b. Ballot verification: each of the ballots is verified to be valid or invalid according to the rules set during the set-up stage, where invalid ballots are not included for tabulation.

   c. Vote tabulation: valid ballots are counted and tabulated as per the counting rules; the results from each polling location are aggregated, and the voting result is revealed and made public.

2.2. Voting Systems

In terms of elections, there are various voting systems used by different countries. Each of them has its own advantages and disadvantages. Some might be considered to be fairer than others. The Handbook of Electoral System Design by Reynolds and Reilly (1997) lists the categorisation of these systems as below.

- Plurality-majority systems
  In plurality-based systems, the candidate with the highest number of votes wins (no threshold of votes). In majority-based systems, the candidate with a majority of votes wins (above a certain threshold).
- Proportional representation systems
  Voting results under this system are proportional to the number of votes received. For example, if a major party wins 40% of the votes in an election, the party will be allocated approximately 40% of the seats, and a minor party with 10% of the votes will also gain 10% of the parliamentary seats. This is to reduce the disparity between a party's share of the national vote and its share of the parliamentary seats.
- Semi-proportional systems
  This type of system is that which inherently translate votes cast into seats won in a way that falls somewhere between the proportionality of proportional representation systems and the majoritarianism of plurality-majority systems.

The most popularly known voting system is the plurality-majority type. In plurality-based system, there are typically a small number of candidates to be chosen, and the candidate receiving most (not necessarily a majority) votes wins. An alternative to this system requires that a candidate need to obtain majority of votes to win. Majority is defined as more than half of the number of the voters or ((½ *n*) + 1), where *n* denotes the number of voters. This is often regarded as a fairer system, since the result reflects the will of the majority of voters.

One majority type voting strategy is preferential voting. Each voter is required to provide an order of preference for the candidates. If no candidate receives a majority, the candidate with the lowest first preference vote is eliminated. Votes of the eliminated candidate are redistributed to the remaining candidate according to the second preference. Repeatedly, more candidates are eliminated until one reaches a majority. Election in the Australian House of Representatives is an example of such a system.

2.3. Security Threats

Security is an integral part of voting as there are considerable motives to cheat in voting and the result of voting affects many people. There are numerous security threats to a voting system including vote buying/selling, coercion/intimidation, unauthorised voting, double voting, corrupt authorities, and rigging. We provide classification of the threats based on the attacker, and some examples of the threats as follows:

Developers/vendors

Moving into e-voting, there are other problems highlighted by Mercuri (2000), Harris (2003), and Jefferson, *et al*. (2004). The requirement of having a voter verifiable paper audit trail was first studied by Mercuri (2000) since she does not trust the softwares used for voting. This is because developers/vendors may be corrupt. Specific examples of alleged cheating in e-voting are provided by Harris (2003). She states that vote tampering is made easier using e-voting as the process can now be automated. The work was then published academicly by Kohno, *et al*. (2004). Following from the work by Mercuri (2000), Harris (2003) argues that without proper vote audit trail, it is even easier to tamper with the voting result. In this category, secure programming principles, and independent testing and certification authorities are required to prevent cheating by the developers/vendors.

Authorities

Although physical security measures are in place, the conduct of voting still requires voting authorities. These officials range from security guards, registration authorities, election judges in polling sites to tally authorities. A security threat in this category is corruption of an authority. A corrupt authority can basically tamper with the voting result, where the difficulty level to tamper with the result depends on the trust level of the authority. An authority given a high level of trust can more easily manipulate the voting result compared to an authority given a low level of trust. To prevent corrupt authorities, they are chosen based on their reputation, and a set of regulations (codes of conduct, procedures, fines, and punishments) and auditing are enforced to deter the officials from being corrupt. More than one authority with the same role can also be chosen, such that it is not possible for one authority to corrupt the voting result.

Voters

Acknowledged by Jones (2001), voting has not always been private. Prior to the use of official ballots, coercion/intimidation and corruption were common among voters. Thus, voting results did not reflect true opinion of the voters. To eliminate this problem, private voting was enforced with voting conducted privately using paper ballots inside a polling booth. However, this led to vote buying/selling where buyers hand out filled-in ballots outside

polling booths for the voters to cast into the ballot-box. Voters then produce an empty ballot to the buyer afterward.

The use of official ballot printed and distributed by the government was then enforced to alleviate this problem. It later became known as the Australian ballot or secret-ballot, since it was first used in the states of Victoria and South Australia in 1856 (Australian Electoral Commission 2000) to enforce compulsory secrecy in voting.

External

An external attacker might disrupt or manipulate voting for either personal, financial, or political gain (e.g. terrorist organisation). An example of external threat is to physically block the polling sites such that voters are unable to vote during the voting period, or coerce/intimidate voters not to vote or to vote according to the attacker's choice (not the voter's choice). Other possible attacks in this category include hackers compromising voting machines, tally machines, or performing a denial of service attack, such that the votes are unable to be transported for counting. Some of these attacks are listed by Jefferson, *et al*. (2004). Some precautions to prevent such threats include placing security guards in polling places and only using secure private networks for voting.

The work by Jefferson, *et al*. (2004) analysed the security of an experimental Internet voting system to be used by the US citizens to vote from overseas. The report highlights the importance of security for voting, and the infeasibility of ensuring security on Internet-based voting. The Internet consists of numerous different networks belonging to different administrative domains, and thus it is infeasible to strictly control the security of data travelling through the public Internet. It is also impractical to enforce physical security and ensure the security of each personal computers used by the voters. Thus, we only recommend the use of polling-booth voting.

Equipment failures (glitches)

Threats in this category include equipment failures, glitches, or malfunction. This might occur accidentally, or by sabotage. By thorough inspection prior to the voting period and the use of sealed or tamper-resistant devices and backup procedures, these types of threats can be minimised/eliminated.

Threats can also originate from a combination of attackers. External entities can corrupt the authorities, developers/vendors, or voters, and a voter may accidentally break a voting machine. As voting result affects many people, the risk is too high for a voting system to be compromised. Compared to current online commercial transactions, a different and higher security level is required for an e-voting system.

2.4. Security Requirements

Addressing the threats in the previous subsection, a number of security requirements need to be formulated. Comprehensive and often complex security requirements are fundamental to ensure secure voting. These security requirements are as follows:

- Accuracy: as a basic property of voting, the voting result must reflect correct tabulation of the individual ballots.
- Privacy: voter-vote relationship must be kept private to ensure that voters express their true opinions without fear of being intimidated.
- Receipt-freeness: introduced by Benaloh and Tuinstra 1994, voters must neither be able to obtain nor construct a receipt which can prove the content of their vote to a third party. This is to prevent vote buying/selling, such that voters are not used as a proxy to cast votes.
- Eligibility: only authorised voters are allowed to vote, preventing fraudulent votes from being counted in the tally stage.

- Prevention of double voting: this ensures that all voters are allowed to vote only once, such that each voter has equal influence in the voting result.
- Fairness: no partial tally is revealed before the end of the voting period to enforce privacy and ensure that all candidates are given a fair chance.
- Robustness: the system must be able to tolerate certain faulty conditions and manage some disruptions.
- Verifiability/accountability: correct voting process must be verifiable to prevent incorrect voting result. A stronger notion is universal verifiability, where everyone (including observers and outside parties, and not just those taking part in voting) can verify that voting was conducted correctly, and that the result is not corrupted.

A secure e-voting implementation must address all the security requirements. This is difficult and complex to implement. The US Federal Election Commission (2002) issued a final Voting System Standard in 2002. However, although it specifies definitive requirements, the standard lacks in build and implementation specifics. Another work in voting standard is currently performed by IEEE with project number 1583 titled "Voting Equipment Standards"[1], and project number 1622 titled "Standard for Voting Equipment Electronic Data Interchange"[2]. However, they are still work in progress.

To have a secure working e-voting system, security must be incorporated from start to finish: from its design phase, throughout its development phase and final deployment phase. We discuss these phases in more detail in the next three sections.

3. CRYPTOGRAPHIC DESIGN OF E-VOTING SYSTEMS

To build a secure system, its foundation must also be secure. In comparison to paper-based voting, e-voting provides better security as it allows the use of cryptographic protocols. This is important as security of the system can be analysed formally, and verifiability can also be provided.

Privacy is one of the most important security requirements in voting. Each vote must be private to its corresponding voter. This is to prevent vote buying/selling or intimidation such that the voting result reflects the true opinion of the voters, and not a corrupted one.

In manual/traditional voting, a voter typically records his/her vote in a ballot inside a private and physically secure location (e.g. polling booth). Afterwards, the ballot is kept secret and anonymised by a ballot-box. Finally, the anonymised ballots are tabulated to reveal the voting result.

In cryptographic voting schemes, ballots are modelled as a tuple of (*ID*, *Vote*) containing the identity of the voter *ID* and its corresponding vote *Vote*. The *ID*-*Vote* relationships must be kept private according to the privacy requirement. This is possible by either preserving the confidentiality of the vote as (*ID*, *Conf*(*Vote*)) or by preserving the confidentiality of the voter's identity as (*Conf*(*ID*), *Vote*), where *Conf* is a function providing confidentiality service. These are two categories that can be used to realise a cryptographic voting protocol. The first is by exploiting the homomorphism property of the underlying cryptosystem, and the second is by simulating a ballot-box using mix-network (mixnet).

3.1. Homomorphic Encryption-based Voting Schemes

In schemes using the (*ID*, Conf(*Vote*)) approach, protection is offered against the voting strategy of a particular voter. An encryption function *E* is used to encrypt the vote, such that the integrity and confidentiality of the vote are preserved.

---

[1] The URL for the group is http://grouper.ieee.org/groups/scc38/1622/, last accessed 7 October 2004.
[2] The URL for the group is http://grouper.ieee.org/groups/scc38/1583/, last accessed 7 October 2004.

For n voters, and $i = 1, 2, \ldots, n$, voter V_i forms his/her ballot c_i by encrypting his/her vote v_i using a homomorphic encryption function E as $c_i = E(v_i)$. Ballot c_i is then submitted to the tally authorities. After the voting period has ended, tally authorities reveal the voting result from decrypting the combination of n ballots, where individual ballot c_i is not decrypted. Tabulation of ballots is possible by exploiting the homomorphism property of the encryption function. For example:

$$E(v_1) \times E(v_2) \times \ldots \times E(v_i) \times \ldots \times E(v_n) = E(v_1 + v_2 + \ldots + v_i + \ldots + v_n) \quad (1)$$

ElGamal (1986) and Paillier (1999) proposed two examples of homomorphic cryptosystem. The homomorphism is inherited from the use of exponentiation in the encryption process. Voting schemes in the literature using this approach include Benaloh and Tuinstra (1994), Benaloh (1996), Cramer, *et al.* (1996), Cramer, *et al.* (1997), Baudron, *et al.* (2001), and Lee and Kim (2002).

Using this approach, the universal verifiability property is also satisfied, since the ballots are published, and everyone can check whether there are any ballots excluded from the tabulation.

3.2. Mixnet-based Voting Schemes

In schemes using the (*Conf*(*ID*), *Vote*) approach, anonymity service is offered to the voter by simulating a ballot-box cryptographically as illustrated in Figure 2.

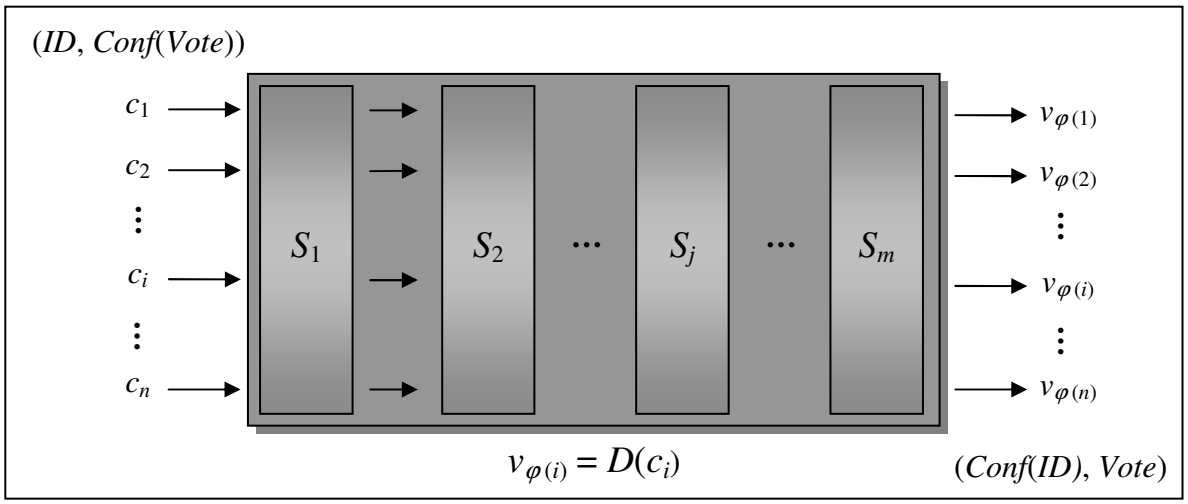

Figure 2. Mixnet-based Voting Scheme

There are m mix servers (mixers) working sequentially in the mixnet, randomly permuting the ordering of the ballots. For m mixers, n voters, $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$, each voters V_i submits their ballots $c_i = E(v_i)$ to the first mixer S_1. The first mixer S_1 then processes[3] the ballots c_i, and output the ballots in a random order as $c_{\varphi(i),1}$. Afterwards, $c_{\varphi(i),1}$ are forwarded as inputs to the second mixer S_2, repeatedly until the final output $c_{\varphi(i),m}$ are obtained. The function $\varphi(i)$ denotes a random permutation function from $i = 1, 2, \ldots, n$, to $i = 1, 2, \ldots, n$. The final outputs of the last mixer S_m are then decrypted using a decryption function D as $v_i = D(c_i)$. If at least one mixer keeps the random permutation secret, then the anonymity of the votes output by the mixnet can be guaranteed. This is a straight-forward design solution to satisfy the privacy requirement.

[3] Normally, the process include re-encryption of the ballots. Another mode of operation is by using decryption chain. However, decryption chain mixnet lacks robustness when one of the mixer fails (Golle, *et al.* 2002).

The mixnet can also be viewed as a confidentiality translation service from (*ID*, *Conf*(*Vote*)) to (*Conf*(*ID)*, *Vote*). Pioneered by Chaum (1981), schemes using this approach include Sako and Kilian (1995), Okamoto (1997), Jakobsson, *et al*. (2002), Golle, *et al*. (2002), Lee, *et al*. (2003), Chaum (2004), and Aditya, *et al*. (2004). Tabulation is straight-forward as in the traditional/manual voting using votes output by the mixnet.

3.3. Cryptographic Primitives

ElGamal (1985) and Paillier (1999) are the two commonly used cryptosystems to construct an e-voting protocol. Aside from the homomorphism property, they are public-key based cryptosystems and allow ciphertext re-encryption.

Eligibility can be satisfied as digital signatures can be implemented using a public-key based cryptosystem. Digital signature is formed by encrypting the hash of a message using the private key of a user. The signature can be publicly verified as it is decrypted using the public key of the user and compared to the original hash of the message claimed by the user. Double voting is also prevented by using digital signatures, as each ballot can be publicly checked whether it belongs to one voter.

Receipt-freeness can be realised as the ciphertext can be re-encrypted to a different ciphertext and still decrypt to the same plaintext. This is possible as the cryptosystem uses randomisation in the encryption process, such that one plaintext can be encrypted to more than one ciphertexts. The re-encryption process simply "updates" the random value used in the ciphertexts. Since the user does not hold the knowledge of the random value in the ciphertext, he/she cannot prove the content of the ciphertext to be as what he/she claims. Note that an untappable channel is required in some schemes to realise receipt-freeness, for a voter to submit the ballot to be randomised by an authority before the ballot is made public. An untappable channel is a physical assumption used in the design, and can be realised in the implementation using a physically secure private communication channel.

Other cryptographic primitives are required to satisfy all the security requirements, and complete the protocols based on Section 3.1 and Section 3.2. As there are many primitives that can be incorporated in a scheme, we only discuss two primitives we consider being important:

- Threshold decryption: the power of decryption is normally shared amongst *m* number of tally (decryption) authorities using a *t*-out-of-*m* threshold scheme (Shamir 1979), such that at least *t* number of the decryption authorities are required to successfully decrypt the ballots in mixnet-based voting schemes, or to decrypt the combination of ballots (and not to decrypt the individual ballots) in homomorphic encryption based voting schemes. This primitive allows trust to be distributed between a number of authorities. Using this primitive, privacy is preserved, and fairness and robustness are satisfied.
- Zero-knowledge protocol: this protocol allows a prover to convince a verifier that the prover holds a secret without revealing the secret itself. An example protocol by Chaum and Pedersen (1992) allows a prover to prove his/her knowledge of a plaintext inside a ciphertext without revealing the plaintext to the verifier. Authorities can be sure that the ballots are formed properly in homomorphic encryption based voting schemes, and voters can be sure that the shuffled ballots are not modified in the mixnet in mixnet-based voting schemes. Using this primitive, accuracy and verifiability are satisfied.

3.4. Comparison

Different types of cryptographic voting schemes are suited to different types of counting systems. Aditya, *et al*. (2003) shows that it is impractical to realise preferential voting systems

using homomorphic encryption based schemes. Voters are required to perform more computationally expensive tasks in homomorphic encryption based schemes, as they are required to prove that the ballots were formed properly. Also, the bandwidth requirement is higher since voters are required to communicate the ballot and the proof. Thus, homomorphic encryption based schemes are not scalable in terms of the voting strategy used. Only simple structured vote (yes/no voting, and 1-out-of-*m* voting) can be accommodated using this type of scheme; while write-in ballots can not be realised. On the other hand, it is an elegant approach to provide privacy to voter-vote relationship. Ballot tabulation is more efficient since the voting result is obtained automatically by combining the individual ballots. Receipt-freeness and public verifiability is also straight-forward as individual ballots are kept secret but made public.

The concept of mixnet-based voting scheme is easier to understand as it mimics the use of ballot-box in traditional/manual voting. Ballot submissions incur less computational cost for voters as they are not required to prove the validity of the ballots. Bandwidth requirement is also less than what is required for homomorphic encryption based schemes as voters only need to communicate the ballot to the mixnet. This type of voting scheme can accommodate flexible vote structure, even write-in ballots. However, ballot validity checking and tabulation are still performed manually. Mixers bear computational cost for proving that their shuffling was correct. Also, voters typically require untappable channel to communicate the ballot to the mixnet for receipt-freeness. Public verifiability is more complicated as ballot shuffling need to be verified.

Recent research by Kiayias and Yung (2004) combined the approach of homomorphic encryption and mixnet by using a flag on the ballot. Ballots containing a particular vote are flagged to be processed using the homomorphic approach, and write-in ballots are processed using the mixnet approach. This seems to be the most promising direction since it tries to overcome the disadvantages, but also combines the advantages of both approaches.

3.5. Security vs. Performance

Batch verification (Bellare, *et al*. 1998, Boyd and Pavlovski 2000, Aditya, *et al*. 2004) is a useful cryptographic primitive to check a group of proof instances based on discrete logarithms simultaneously. It exploits the homomorphism property in the underlying proof system used. As exponentiation is a computationally expensive task in computing, the efficiency improvement is possible by combining the exponents first, and performs one exponentiation on the combination of the exponents afterward (more detail is available from Bellare, *et al*. 1998, Boyd and Pavlovski 2000, Aditya, *et al*. 2004). Thus, a performance gain can be achieved since only one exponentiation is performed using the batch verification technique instead of multiple exponentiations on each of the proof instances.

However, security is generally proportional to computational cost in cryptographic processes. Not everything in the cryptographic voting schemes can be batched. Higher computational cost requires longer computational time to complete. On the other hand, performance can be sacrificed to achieve better security (and vice versa) according to the voting application.

Better performance can be obtained by lowering the complexity of the scheme, which translates to reduced level of security. In national election, all the security requirements must be satisfied, while performance is more important in reality television shows. Security is a trade-off to performance.

4. DEVELOPING AN E-VOTING SYSTEM

There are currently problems with e-voting and trust issues in e-voting vendors in the US. E-voting vendors in the US are allowed to be private companies, where the certification

authority has some connections to the vendors (as documented by Harris, 2003). This raises concern as the manufacturer of the machines may be biased to a particular candidate.

On the other hand, e-voting in India was conducted successfully with very minor problems compared to the US one. Development of e-voting in India was performed partially by the government. The difference between the two is fundamentally a trust issue. Note that this issue also exists in the paper-based voting system.

Building a sound and secure e-voting implementation requires a sound and secure design. The design of an e-voting system must use cryptographic protocol as in Section 3. The cryptographic scheme chosen for a real system must be built properly by consulting with the designer. Proper development, testing and certification, are keys to public confident for a secure system.

4.1. Trust Issues

Building a system as per the design specification is not often straight-forward. As the security requirements are complex, the cryptographic protocol design is also complex. The complexity is then transferred to the development phase, and the complication made it difficult to perfectly implement a system according to its design. Furthermore, there are trust issues to voting developers/vendors as shown by Mercuri (2000) and Harris (2003). Incompetent programmer can introduce software bugs in their program, or even create a faulty one. Corrupt developer/vendor can deliberately place a backdoor to voting software to later manipulate voting result.

An obvious solution is to use a trusted developer/vendor to build a secure voting system according to the secure design. However, even a good programmer can accidentally introduce software bugs in the program. Also, it is weak to measure security of a system by the trustworthiness of the developer/vendor. A better method to ensure the security of a system is by following best practice in the development process.

4.2. Development Process

Security engineering principles and coding standards (e.g. GNU Coding Standards[4]) must be followed in developing the system to minimise the number of software bugs and to ease system inspection. Also, the developer must consult regularly with the system designer to check whether their program is developed and used appropriately according to the design.

Independent and impartial testing and certification must be performed to the system. Open source is one method to achieve this. An experimental Australian e-voting system has received praise from the community since their code is made public by the use of open source. Although it has only been used for a trial election, the code for EVACS by Software Improvements[5] is available for public scrutiny. Thus, it has a higher confidence level than a closed-source e-voting system. Others following the approach of open source to develop a voting system include open voting consortium[6], and open vote foundation[7].

It is difficult to certify that the implementation is compliant with the requirements while also making sure that all the mechanisms implemented work properly together. A thorough check is required to verify that all the mechanisms and all of their possible interactions in an e-voting systems are working as specified. Thorough inspection on the machine code must be performed to ensure that the voting system is working as specified, and that there are no malicious codes, backdoor, or visible software bugs in the program. One method is to certify the system using an international evaluation standard such as common criteria[8].

---

[4] The standard is available from http://www.gnu.org/prep/standards_toc.html, last accessed 7 October 2004.
[5] Their official website is http://www.softimp.com.au/evacs.html, last accessed 7 October 2004.
[6] Their official website is http://www.openvotingconsortium.org, last accessed 7 October 2004.
[7] Their official website is http://www.open-vote.org, last accessed 7 October 2004.
[8] The URL for the project is http://www.commoncriteriaportal.org, last accessed 7 October 2004.

Machines used for voting must also be checked to be secure. A trusted computing platform[9] can be used as the hardware inside the machine use encryption to communicate to each other, such that the authenticity, confidentiality and integrity of the communication can be ensured. Furthermore, trusted systems can be set such that only signed applications that have been comprehensively inspected to be secure by an independent and impartial testing and certification authority are allowed to be installed and executed in the trusted computing platform. Such systems can prevent malicious unchecked or unauthorised codes to be run in the machine.

5. E-VOTING SYSTEMS DEPLOYMENT

Although the system has a sound design and was built, tested and certified properly, improper use of the system might compromise its security. Thus, the deployment of an e-voting system in the real world must follow a set of procedures to maintain its security. Note that this problem also exists in the manual/traditional system.

5.1. Physical Security

To maintain the integrity of the system, machines built for e-voting must also be tested and certified thoroughly. Fragile seals can be used to easily indicate tampering. Tamper-resistant or tamper-evident devices can also be used to ensure security of the system. Physical security measures must also be taken, such as storing the machines in a secure location prior to the voting period, employing security personnel to guard the location, and implementing access control measures (physical locks, username/password, token based systems, biometric) to control access to the machines and the system. This is to prevent the machine from being faulty due to accident (high temperature environment, flooding, etc) or by malicious tampering.

Audit trails, and auditing must be implemented to account for abnormal situations should they occur, to identify the source of a fault, or to identify and trace back an attacker. Backup measures must also be implemented to ensure robustness in the event that something wrong should happen. Such measures include providing redundant machines as backup machines, or to revert back to manual paper-ballot in the extreme case.

5.2. Other Issues

Other issues in using an e-voting system in the real world include political issues, public acceptance, and user awareness. An example of political issue is where a conservative party opposes the use of an e-voting system because of cost factor. The initial cost of purchase, roll-out and preparing authorities to use the system nation-wide might be high. However, it is inevitable to use e-voting in the future with the scalability limitation of current paper-based voting system. Also, e-voting should offer lower operational cost in the long run.

While younger voters might prefer the use of e-voting as using new technology, some mature age voters might prefer the use of manual/traditional voting system as familiarity reasons. Public education must also be provided, such that the technology is understood and accepted by the voters. Phased upgrade can be implemented as both manual/traditional and e-voting systems are used together for a period of time before completely changing to e-voting system entirely.

Better e-voting standard containing requirements and implementations specifics need to be developed. Policies, procedures, and legislation are also required to enforce secure e-voting.

[9] The URL for the group is http://www.trustedcomputinggroup.org, last accessed 7 October 2004.

6. CONCLUSION

Implementing a sound and secure e-voting system is not as straight-forward as simply employing a counting software. Accuracy, privacy, receipt-freeness, eligibility, prevention of double voting, fairness, robustness and verifiability/accountability are security requirements that an e-voting system must address.

Issues in three phases of design, development, and deployment of secure e-voting systems have been discussed. To satisfy the security requirements, a voting system must be designed (Section 3), developed (Section 4) and deployed (Section 5) properly. A sound and secure design using cryptographic protocols is required for a sound and secure system. The system must be built according to the design, following development best practices, and thoroughly tested and certified. This is to obtain public confidence that the system built is not corrupted. Furthermore, proper procedures and security measures are required to maintain the security of such system during its deployment. Our future work will focus on examining the security of a specific e-voting implementation.

The move to electronic voting is inevitable. The technology for e-voting already exists, and voting machines have been used in some part of the world. It is just a matter of public acceptance and time for total deployment of e-voting systems.

ACKNOWLEDGEMENTS

We acknowledge the support of the Australian government through ARC Discovery 2002, Grant No: DP0211390, and ARC Linkage International fellowship 2003, Grant No: LX0346868.

REFERENCES

Aditya, R. and Boyd, C. and Dawson, E. and Viswanathan, K. (2003), Secure E-Voting for Preferential Elections, *Proceedings 2nd Annual International Conference on Electronic Government*, 246-249.

Aditya, R. and Peng, K. and Boyd, C. and Dawson, E. and Lee, B. (2004), Batch Verification for Equality of Discrete Logarithms and Threshold Decryptions, *Proceedings 2nd Applied Cryptography and Network Security Conference*, 494-508.

Australian Electoral Commission (2000), Australian Electoral History, *Online*, Available from http://www.aec.gov.au/_content/when/history/history.htm, last accessed 7 October 2004.

Baudron, O. and Foque, P. A. and Pointcheval, D. and Stern, J. and Poupard, G. (2001), Practical Multi-Candidate Election System, *Proceedings 26th ACM Symposium on Principles of Distributed Computing*, 274-283.

Bellare, M. and Garay, J. A. and Rabin, T. (1998), Fast Batch Verification for Modular Exponentiation and Digital Signatures, *In Advances in Cryptology - EUROCRYPT 98*, 236-250.

Benaloh, J. (1996), Verifiable Secret-Ballot Elections, *PhD Thesis*, Faculty of Graduate School, Yale University.

Benaloh, J. and Tuinstra, D. (1994), Receipt-Free Secret-Ballot Elections, *Proceedings 26th ACM Symposium on the Theory of Computing*, 544-553.

Boyd, C. and Pavlovski, C. (2000), Attacking and Repairing Batch Verification Schemes, *In Advances in Cryptology - ASIACRYPT 00*, 58-71.

Chaum, D. (1981), Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM, 24(2), 84-90.*

Chaum, D. and Pedersen, T. P. (1992), Wallet Databases with Observers, *In Advances in Cryptology - CRYPTO 92*, 89-105.

Cramer, R. and Franklin, M. and Schoenmakers, B. and Yung, M. (1996), Multi-Authority Secret-Ballot Elections with Linear Work, *In Advances in Cryptology - EUROCRYPT 96*, 72-83.

Cramer, R. and Gennaro, R. and Schoenmakers, B. (1997), A Secure and Optimally Efficient Multi-Authority Election Scheme, *In Advances in Cryptology - EUROCRYPT 97*, 103-113.

ElGamal, T. (1985), A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms, *In Advances in Cryptology - CRYPTO 84*, 10-18.

Federal Electoral Commission (2002), Voting System Standards, *Online*, Available from http://www.fec.gov/pages/vssfinal/vss.html, last accessed 7 October 2004.

Golle, P. and Zhong, S. and Boneh, D. and Jakobsson, M. and Juels, A. (2002), Optimistic Mixing for Exit-Polls, *In Advances in Cryptology - ASIACRYPT 02*, 451-465.

Harris, B. (2003), *Black Box Voting: Ballot Tampering in the 21st Century*, Plan Nine Publishing.

Jakobsson, M. and Juels, A. and Rivest, R. L. (2002), Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking, *Proceedings 11th USENIX Security Symposium*, 339-353.

Jefferson, D. and Rubin, A. and Simons, B. and Wagner, D. (2004), A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), *Online*, Available from http://www.servesecurityreport.org/, last accessed 7 October 2004.

Jones, D. (2001), A Brief Illustrated History of Voting, *Online*, Available from http://www.cs.uiowa.edu/~jones/voting/pictures/index.html, last accessed 7 October 2004.

Kiayias, A. and Yung, M. (2004). The Vector-Ballot E-Voting Approach, *Proceedings 8th Financial Cryptography Conference*, 72-89.

Kohno, T. and Stubblefield, A. and Rubin, A. and Wallach, D. (2004), Analysis of an Electronic Voting System, *Proceedings IEEE Symposium on Security and Privacy*, 27-40.

Lee, B. and Boyd, C. and Dawson, E. and Kim, K. and Yang, J. and Yoo, S. (2003), Providing Receipt-Freeness in Mixnet-Based Voting Protocols, *Proceedings 6th Information Security and Cryptology Conference*, 245-258.

Lee, B. and Kim, K. (2002), Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer, *Proceedings 5th Information Security and Cryptology Conference*, 389-406.

Mercuri, R. (2000), Electronic Vote Tabulation Checks & Balances, *PhD Thesis*, School of Engineering and Applied Science, University of Pennsylvania.

Okamoto, T. (1997), Receipt-Free Electronic Voting Schemes for Large Scale Elections, *Proceedings Security Protocols, 5th International Workshop*, 25-35.

Paillier, P. (1999), Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *In Advances in Cryptology - EUROCRYPT 99*, 223-238.

Reynolds, A. and Reilly, B. (1997), *The International IDEA Handbook of Electoral System Design*, International Institute for Democracy and Electoral Assistance.

Sako, K. and Kilian, J. (1995), Receipt-Free Mix-Type Voting Scheme: A Practical Solution to the Implementation of A Voting Booth, *In Advances in Cryptology - EUROCRYPT 95*, 393-403.

Shamir, A. (1979), How to Share a Secret, *Communications of the ACM*, 22(11), 612-613.