Secure E-Voting With Blind Signature

Subariah Ibrahim, Maznah Kamat, Mazleena Salleh, Shah Rizan Abdul Aziz

Faculty Of Computer Science & Information Technology
University Technology Of Malaysia
81310 Skudai, Johor Bharu, Johor, Malaysia

Abstract - With a rapid growth in computer networks, many people can access the network through the Internet and therefore an electronic voting can be a viable alternative for conducting an election. Electronic voting system must attempt to achieve at least the same level of security as ordinary elections.

We have developed an electronic voting system, E-Voting for a general election. E-Voting system employs cryptographic techniques to overcome the security issues in the election process. In this system, voter's privacy is guaranteed by using a blind signature for confidentiality and voter's digital signature for voter's authentication.

E-Voting is implemented by employing Java socket technology and BouncyCastle cryptography provider. The provider, which is an open source library, is used to provide the secure communication channel. The voter's private key for digital signature is protected by using password-based encryption with SHA and Twofish-CBC algorithm so that only valid voter can use it.

1. Introduction

Election is a fundamental instrument of democracy that provides an official mechanism for people to express their views to the government. Traditionally, the process of voting is quite cumbersome because voter must come in person to vote. This problem results in the low participation rate of voting. Vote-by-mail can cater for certain voters such as those who live in sparsely populated areas and who work far away from the voting centers. However, this method is time-consuming and cumbersome for the authority to manage since it requires extra work to send, collect and count the ballots manually [1]. Electronic voting system or EVS can overcome those problems. EVS is expected to make our modern social life more convenient, efficient and inexpensive. By using EVS in national election, a voter can vote from his home or office.

EVS must meet security requirments such as confidentiality, integrity, authentication, and verifiability. This is because EVS is more vulnerable than traditional voting due to the nature of digital processing of election data which can be easily manipulated, hence may result in widespread fraud and corruption [1,2].

In this research, we have implemented a prototype of an EVS, called E-Voting, that satisfies four security requirements for a safe election. This is achieved by designing some protocols that guarantee those requirements. We believe that E-Voting can reduce human error in voting process by providing easy-to-use user interface.

The rest of this paper is organized as follows: In Section 2, security requirements for an EVS are reviewed. Security mechanisms employed in E-Voting are discussed in Section 3. In section 4, we discuss the general architecture of our system and explain the protocols in each voting stage. In section 5, the implementation of the system is discussed. Finally, concluding remark will follow in Section 6.

2. Security Requirements

Many extensive researches on electronic voting have been conducted and therefore there are now an extensive list of security requirements available. Without these security requirements, numerous opportunities for a widespread fraud and corruption may exist. In order to overcome these problems, an election should have the following requirements [3, 4]:

(i) Confidentiality
(ii) Integrity
(iii) Authentication
(iv) Verifiability

2.1 Confidentiality

The voter's ballot should be kept confidential. In addition, a voting protocol must not allow any voter to prove that he or she voted in a particular way. This is important to avoid opportunity of vote buying and extortion.

2.2 Integrity

In electronic voting, there is no physical ballot. The digital ballot can be tampered with ease, given the insecure nature of current computer networks. Hence, with digital ballot system, the integrity of receipt by the computer of the voter's choice is a great issue. The published results should measure how the eligible voters actually vote. To achieve the integrity of the EVS, the protocol must ensure only valid votes are counted in the final tally and no one can change anyone else's vote without being discovered.

2.3 Authentication

During voting, EVS must provide an authentication mechanism that ensures a voter who is allowed to vote must be an eligible voter and he is a person he claims to be. Therefore during registration, some form of credential or ticket must be given to a voter so that it can be used to authenticate him during voting. Another issue that must be considered here is that the notion of one-person one vote must be preserved. So, EVS must provide a mechanism to check this notion.

2.4 Verifiability

Verifiability can be categorized into two: individual and universal verifiability. Individual verifiability means a voter can check that his vote was properly received and has been taken into account in the final tally. While a universal verifiability means anyone can verify at a later time that the election was properly performed [5]. Universal verifiability allows auditing by the public and therefore anyone can verify that all votes have been counted correctly. In EVS, to achieve verifiability, the results and the collected ballots are published for public viewing.

3. Security Mechanisms in E-Voting

In this research, we developed an EVS, E-Voting, which applies security mechanisms in order to achieve the four security requirements needed for any election process. In this system, voter's privacy is guaranteed by using a blind signature for confidentiality and voter's digital signature for voter's authentication.

3.1 Digital Signature

Digital signature is used to authenticate that the message comes from a particular sender. This is done by attaching a code that acts as a signature. This signature guarantees the source and the integrity of the message [6].

E-Voting employs RSA encryption. RSA is the first public-key encryption algorithm, and is widely considered the best to date [7]. It is an asymmetric cipher, utilizing two different keys, the public and the private, to perform encryption and decryption. The key pair is generated as follows: two prime numbers, p & q are generated and multiplied together to get the modulus, n. $(p-1)$ is multiplied by $(q-1)$ to get $\Phi(n)$. The public exponent, e which is less than and relatively prime to $\Phi(n)$ is chosen. Private exponent, d is calculated using the formula, $d = e^{-1} \bmod \Phi(n)$. The public key is $\{n,e\}$ and the private key is $\{n,d\}$. The equation for encryption and decryption are as follows:

$$\text{Encryption: } C = M^e \bmod n$$
$$\text{Decryption: } M = C^d \bmod n$$

In E-Voting, digital signature is created by using RSA encryption. The process begins with the hashing of the message, M, to produce a message digest, H. The digest is then encrypted using the sender's private key $\{n,d\}$ to produce the signature, S.

$$S = H^d \bmod n$$

To verify the message, the receiver will hash the message, M by using the same digest function. At the same time, the signature, S is decrypted using the receiver's public key.

$$H = S^e \bmod n$$

The results of the two processes are then compared. If they are equal then the message is authenticated and the integrity of the message is maintained.

3.2 Blind Signature

A blind signature is similar to a digital signature except that it allows a person to get another person to sign a message without revealing the content of a message. It is the most popular cryptographic technique in EVS by providing confidentiality of the voter's ballot [4][5][8][9]. The signature is used to authenticate the voter without disclosing the content of a ballot. Hence the authority whose function is to verify the eligibility of a voter will not know whom a voter votes for.

In E-Voting, a ballot is blinded in order to achieve its confidentiality requirement. A voter is required to get the signature of a validator when he votes. To ensure the secrecy of his ballot, a voter casts a ballot, B, blinds a ballot using a random number and sends it to the validator. Let (n,e) be the validator's public key and (n,d) be his private key. A voter generates a random number r such that $\gcd(r,n) = 1$ and sends the following to the validator:

$$B' = r^e B \bmod n.$$

The random number r conceals the ballot from the validator. The validator then signs the blinded ballot after verifying the voter. The signed value is as follows:

$$S' = (B')^d = r\, B^d \bmod n$$

After receiving the validated ballot, the voter unblinds the ballot, to get the true signature, S of the validator for the ballot, by computing,

$$S = S'\, r^{-1} \bmod n = B^d$$

3.3 Other Cryptographic Schemes

Two other cryptographic schemes employed in E-Voting are Diffie-Hellman key exchange and password-based encryption (PBE). The key exchange enables two users to exchange a key securely that can then be used for subsequent encryption of messages. It provides a method to create a shared secret key by exchanging public keys. Two users, who want to communicate, must have global public elements: prime number p and base g. User A generates key X_A by selecting a number less than p. Its public key, Y_A is calculated using a formula: $g^{XA} \bmod g$. User B also does the same thing producing its private key X_B and public key Y_B. Each side sends their public keys to each other. Then each side does the key agreement process. For User A, it generates the secret key, K

using the formula $(Y_B)^{XA}$ mod q. For user B, the same secret K is generated by $(Y_A)^{XB}$ mod q.

The RSA private key is too long for people to remember. Therefore it is normally stored in a medium like diskette or smart card. To preserve its confidentiality, the key need to be encrypted. E-Voting encrypts the private key with a user password. The password is hashed using a message digest algorithm, SHA-1 and the resulting digest is used to construct a binary key for Twofish-CBC. A salt, random number, is added to the algorithm to make the key difficult to break [7].

The private key, salt and user password are passed to the PBE cipher to produce a cipher text, C. The salt and the cipher text are combined together, $\{salt, C\}$.

When user wants to retrieve the private key, user must provide the password. The cipher text, *C*, the salt and the password are passed to the cipher to produce back the private key.

4. E-Voting System Architecture

In this section, we describe the overall architecture and the voting stages of E-Voting system. Figure 1 illustrates E-Voting architecture. It consist of seven basic entities: voter, administrator, tallier, validator, registrar, E-Voting database and national registration database. In E-Voting, only a registered voter can vote, register himself as a candidate or becomes a nominator.

Administrator of E-Voting is responsible in setting the dates of registration and voting. Besides that, administrator registers voter to become a candidate. A Validator is responsible to validate the ballot sent by voters during voting. He also verifies the eligibility of the voter and produce the ballot id and the list of candidates to vote. At the end of voting period, Tallier will count the ballots. During voting, Tallier verifies the ballot sent by voter.

These multi-authority voting system can overcome problems like phantom voters, casting a vote for abstaining voters, registered voters cast more than once or the ballots miscounting [1].

4.1 Election Stages

There are four stages in E-Voting. They are Voter Registration, Candidate Registration, Voting and Counting. The following sections will descibe the protocols and events in each of the four stages. Prior to these stages, all entities key pairs except for voters should be generated.

4.1.1 Voter Registration

In any election, an individual must register to be an elligible voter. This is done before the voting period. Voter registration for E-Voting is done as follows:

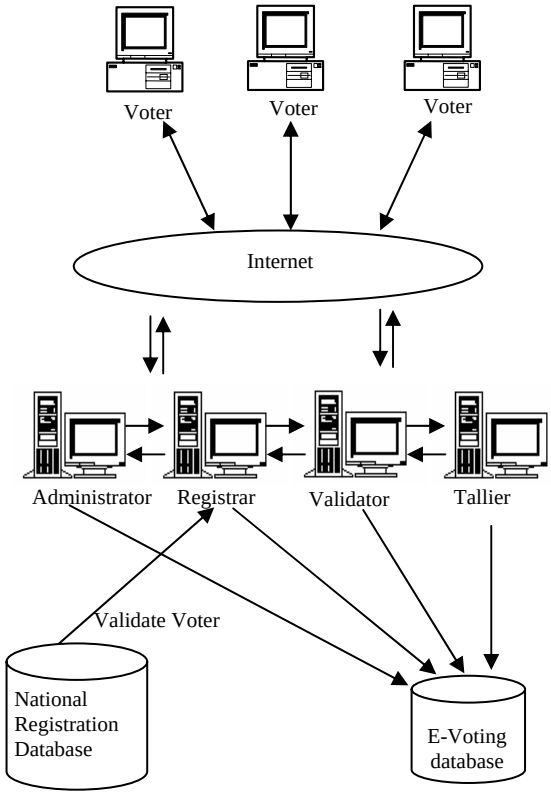

Figure 1: E-Voting System Architecture

i. Voter sends his name and national identity card (*Nric*) number to the registrar server. The message is encrypted before transmitting through the network.
ii. Registrar server checks the user's particulars with the national registration database to determine the eligibility of a voter and his precinct.
iii. If a voter is eligible, the system will generate RSA key-pair. The public key is stored in the E-Voting database while the private key is stored in a voter's diskette protected by his chosen password.

4.1.2 Candidate Registration

Election candidates must be nominated by two people and all of them (candidate and nominators) must be registered voters. Registration of candidates are as follows: a registered voter and the two nominators send their names and Nric numbers to the Administrator server. These information is individually signed by the respective voters and encrypted by the Adminisrator's public key.

4.1.3 Voting

In voting stage, a voter must send a ballot to both Validator and Tallier. The process of voting is described below:

i. Voter sends his name and Nric to the Validator.
ii. Validator checks the eligibility of the voter and whether he or she has voted before. If the voter is

a valid voter, Validator will obtain the precinct number of the voter and send him the ballot. Each ballot has a unique id.

iii. The voter casts his ballot and the ballot is then blinded, signed, encrypted and sent to the Validator to be validated.
iv. Validator signed the blinded ballot after verifying the voter. The signed blinded ballot is sent back to the voter.
v. Voter checks the integrity of the ballot by unblinding the validated ballot and compare it with the original one.
vi. The validated ballot and the original ballot are sent to the tallier. The communication is protected by a session key (BlowFish) that has been agreed when the connection is set up.
vii. The Tallier checks the validity of the ballot using Validator's public key. The validated ballot is stored in the E-Voting database.
viii. Tallier acknowledges voter by sending the ballot *id*, time and date of voting and is signed with Tallier's private key, *td*.

4.1.4 Counting Protocol

When voting period is over, Tallier will automatically count the ballot and determine the counts for each candidate and save it in the database. The result is verified by the election authority before publishing it through the web.

5 Implementation

In order to implement this E-Voting system efficiently, we use JBuilder5 program. This program could speed up the development of this system because it has facilities to draw forms and to add library easily.

5.1 Cryptographic Library

The E-Voting system is developed using Java language version jdk1.3.1_01. In addition, Java Cryptography Architecture (JCA) and Java Cryptography Extension (JCE) are also being used in providing cryptographic functions to all the E-Voting components. JCE is not provided by default in jdk1.3.1_01, therefore JCE must be downloaded. Two advantages of using JCA and JCE are that the encryption functions need not to be redeveloped and the applications will be portable to different environments using Java libraries. BouncyCastle JCE provider is chosen because it is licensed as open source and the most complete cryptographic provider [7].

During voter registration, secure communication between voter and registrar is achieved by encrypting data sent using a session key. The session key is created using Diffie-Hellman Key Exchange algorithm. The main issue of using this algorithm is the way to choose global parameter. Both parties must have the same global parameter. Therefore, we use a standard for that modulus and base, known as SKIP (the Simple Key management for Internet Protocols). SKIP has established values for 512, 1024, and 2048-bit keys. 1024-bit SKIP spec is chosen for E-Voting.

5.2 Servers

Server with large memories and high speed processor is recommended. This is because the server may run a large number of threads to service large numbers of voters at one time.

5.3 Network Connection

The E-Voting system is based on a client-server model and it has been implemented using socket-based communication. Socket-based communication enables applications to view networking as if it were file I/O, whereby a program can read from a socket or write to a socket as simple as reading from a file or writing to a file. Java provides socket-based programming through classes in package java.net. Stream socket, which use Transport Control Protocol (TCP) is chosen to provide communication channel between client and server application. TCP protocol offers connection-oriented service, which ensures reliable data delivery [10].

5.4 Programs

The are seven application programs developed to form the whole system. The server programs represent the server entities: Registrar, Administrator, Validator and Tallier. Respectively, the server programs are RegistrarServerApplication, AdministratorServerApplication, ValidatorServerApplication and TallierServerApplication. Client applications are Voting, VoterRegistration and CandidateRegistration.

A web site is developed to allow voter to see the election result and check their ballots for verification.

6 Concluding Remarks

We have implemented a system prototype that implements security protocols that meet the security requirements of an EVS. With this system, electronic ballots are generated automatically. The participation rate of voting is expected to increase because voters do not have to line-up in a long queue anymore. With electronic ballots, results can be obtained faster than the traditional voting. Besides that, it satisfies the universal verifiability since the public can verify the election results.

Below are improvements that can be made on the system:

a) E-Voting requires users to download and install the client application. It will be easy for computer literate users but will incur difficulty for those who are not. This can be overcome by implementing web-based voting. In order to do these, applets will be used to implement the security part of the system. Applet is a

downloadable program code and can be executed in a voter's web browser which supports Java. Therefore a voter and a candidate do not need to download any code ahead of time and install it in their client computers.

b) The system developed employs public-key cryptosystem. When a key-pair is generated for a voter during registration, the public-key is kept in a database while a private-key is stored in a diskette provided by a user. Instead of using a diskette, smart card technology can be employed to do this. It is also possible to enhance the work by incorporating E-Voting with public-key infrastructure (PKI) which provides a means to verify that a public-key belongs to a specific voter.

c) Security of the system can be greatly improved if biometric authentication system such as fingerprint or face is employed. However, this type of authentication system needs extra hardware.

7 References

[1] Subariah Ibrahim, Mazleena Salleh and Maznah Kamat, "Electronic Voting System: Preliminary Study," *Jurnal Teknologi Maklumat*, Vol. 12, pp. 31-40,2000

[2] Subariah Ibrahim, Mazleena Salleh and Maznah Kamat, "Design of a Secure Web-Based Electronic Voting System," in *Proceedings of Malaysian Science and Technology Congress*, 1999.

[3] R. Cramer, R. Gennaro, and B. Schoenmakers, and M. Yung, "Multi-Authority Secret-Ballot Elections with Linear Works." *Eurocrypt '96, LNCS 1070*, pp 72 – 83, 1996.

[4] L.R. Cranor, and R.K. Cytron, "Design and Implementation of a Practical Security-Conscious Electronic Pollind System," *Washington University: Computer Science Technical Report*, 1996

[5] M. J. Radwin, "An Untraceable, Universally Verifiable Voting Scheme," in *Seminar In Cryptology*, 1995.

[6] Stalling, W., Cryptography and Network Security, 3rd Edition, Prentice Hall, New Jersey, 2003.

[7] J. Garms, and D. Somerfield, *Professional Java Security*, Wrox Press Ltd., 1st ed. Birmingham, UK, 2001.

[8] Davenport, Ben, Alan Newberger and Jason Woodard, "Creating a Secure Digital Voting Protocol for Campus Elections," April 2000. http://www.princeton.edu/ usgvote/technical/paper.html

[9] B.W. DuRette, "Multiple Administrators for Electronic Voting." 1999. http: //theory.lcs.mut.edu/~cis/voting/voting.html.

[10] Deitel & Deitel, *Java How To Program*, 3rd ed., Prentice Hall, New Jersey, UK, 1999.