A Secure Electronic Voting Software Application based on Image Steganography and Cryptography

Lauretha Rura

Supervised by: Dr. Biju Issac

A thesis submitted in fulfillment of the requirements for the degree of Master of Science (by Research)

at
Swinburne University of Technology (Sarawak Campus)

Faculty of Engineering, Computing and Science

2013
ABSTRACT

Various electronic voting systems have been introduced as substitutes for traditional voting systems as electronic voting systems reduces the manual labor in collection, checking, transportation and counting of votes. However, with the growth of information technology, electronic voting systems are susceptible to malicious attacks (or threats) which undermine the integrity of the voting process. Simple encryption methods are not sufficient to protect the secrecy of election data. Encrypted data can cause suspicion, making it subject to malicious attacks. This research implements a novel approach by combining cryptography schemes with image steganography techniques. Besides improving security, steganography makes the data transmission over the network during the election become less suspicious too.

After looking into the requirements of an electronic voting system, current electronic voting schemes available in published literature are examined. Next, the cryptographic and steganography techniques best suited for the requirements of our voting system are chosen and the software was implemented. They are namely Password Hashed-based Scheme, Visual Cryptography, F5 Image Steganography and Threshold Decryption Cryptosystem. Besides improving security, the research also aims to improve the quality of the voting process through user friendly interface design, reduction or elimination of additional hardware such as printers and scanners, improving the speed of the election process and end to end voter verifiability that includes a vote receipt. The last of these would increase the voter’s confidence in electronic voting.

The electronic voting system developed is based on Web-based Java EE Technology and MySQL database using the Iterative Waterfall model as the Software Development Life Cycle approach. Java EE was chosen to develop the software because of its wide range of Application Programming Interface (API) that enhanced the software development by automating many of the business and security processes of implementation. A simple quantitative analysis has been also done for data analysis after software implementation by conducting usability testing. Here, the voter’s response and opinions of the system as a whole were observed.

The studies, experiments and analysis done in this research resulted in an enhanced version of remote E2E system that provides a reliable and secure system with more flexibility and better performance compared to some reported E2E Voting Systems. Though implemented as a working prototype, the system can be implemented to support small to medium scale elections.
ACKNOWLEDGEMENT

This Master’s thesis is the result of an inspiring and challenging journey, to which many people have contributed and given their support. For that, I would like to take this opportunity to express my sincere gratitude to the following individuals:

My coordinating supervisor Dr. Biju Issac for his expert, sincere and valuable guidance and encouragement throughout my Master’s Degree. Without his constant support, I would not have learnt so much in the past years.

My associate supervisor Dr. Manas Kumar Haldar for his insightful and constructive suggestions, especially during my thesis writing. His guidance and help are very much appreciated.

My family members, Herman Rura, Nita Hartana, Melissa Rura and Chalvin Putra Rura for their unceasing love, faith and prayers. Their support for me throughout my life is invaluable. They have taught me to believe in myself when I was in doubt and inspired me to strive harder when I was not motivated. Without all of you, I would not be who I am today.

My closest friend, Winda Dorbilyanti Lie, Finie Kong, Stephan Lee and Lai Qian Jie for sincere friendship and for helping me throughout my ups and downs.

Margaret Dahliani Lim and Hanni Stella Angelica for their help and for brightening up my days and Colin Tan Choon Lin for learning the true meaning of Matthew 6:33 (Bible) with me.

Above all, I would like to extend my utmost appreciation and gratitude to Jesus Christ whose love is so evident in my life. It is only through His grace and mercy that I have been able to complete this Master’s thesis.
DEclarations

I hereby declare that the dissertation - “A Secure Electronic Voting Software Application based on Image Steganography and Cryptography” is my own work and that all the sources I have used or quoted have been acknowledged by means of complete references.

It has not already been accepted for any degree, and is also not being concurrently submitted for any other degree.

Lauretha Rura
2 December 2013
# Table of Contents

**Abstract** .......................................................................................................................... i  
**Acknowledgement** ........................................................................................................... ii  
**Declarations** ..................................................................................................................... iii  
**Table of Contents** .............................................................................................................. iv  
**List of Figures** .................................................................................................................... vii  
**List of Tables** .................................................................................................................... x  
**Glossary** ............................................................................................................................ vii  
**Chapter 1** ............................................................................................................................ 1  
  1.0 Introduction ..................................................................................................................... 1  
  1.1 Research Background .................................................................................................... 1  
  1.2 Proposed Solution ......................................................................................................... 2  
  1.3 Research Objectives ..................................................................................................... 4  
  1.4 Thesis Outline .............................................................................................................. 5  
**Chapter 2** ............................................................................................................................ 6  
  2.0 Introduction ..................................................................................................................... 6  
  2.1 Categories of Voting Systems ...................................................................................... 6  
      2.1.1 Traditional Polling Station Voting Systems ...................................................... 6  
      2.1.2 Electronic Voting Systems .............................................................................. 9  
  2.2 Information System Security for Remote Electronic Voting System ....................... 16  
      2.2.1 Cryptography ................................................................................................. 16  
      2.2.2 Steganoraphy ................................................................................................ 20  
  2.3 Voting Systems Threats ............................................................................................... 28  
      2.3.1 Threats in Traditional Polling Station Election ............................................ 29  
      2.3.2 Threats in Electronic Voting Systems .......................................................... 29  
  2.4 Summary and Conclusion ......................................................................................... 30  
**Chapter 3** ............................................................................................................................ 32  
  3.0 Introduction ..................................................................................................................... 32  
  3.1 Helios ............................................................................................................................ 33  
      3.1.1 System Overview ......................................................................................... 33  
      3.1.2 Principal Components of the System ......................................................... 35  
  3.2 Scantegrity II................................................................................................................ 38  
      3.2.1 History of Scantegrity Voting System ......................................................... 38  
      3.2.2 System Overview ......................................................................................... 39  
      3.2.3 Principal components of the system .......................................................... 40  
  3.3 Prêt à Voter .................................................................................................................... 42  
      3.3.1 System Overview ......................................................................................... 42  
      3.3.2 Principal Components of the System ......................................................... 43
# Secure E-voting Analysis and Design

## Introduction

### Iterative Waterfall Model

### Software Requirement Analysis of eVote Voting System

#### Users

#### Scope

#### Major Assumptions

#### Operating Environment and Dependencies

#### System Requirements

### Software Design of eVote Voting System

#### System Integration

#### Data Models

#### Process Models

### Summary and Conclusions

# System Implementation

## Introduction

### Technologies

#### Client-Side

#### Server-Side

### Development Tools

#### Netbeans IDE

#### Glassfish Application Server

#### MySQL Relational Database Management Server (RDMS)

### Implementation of Information Security in eVote System

#### Cryptography

#### Steganography

### System Implementation

#### eVote v1.0

#### eVote v2.0

### Summary and Conclusions

# System Testing and Consideration of Security Attacks on the System

## Introduction

### Software Testing

#### Testing Environment

#### Testing Participant

#### Software Testing Procedure

### Common Security Attacks on E2E Voting Systems

### Internal Sources

### External Sources
6.3 Comparison of E2E Voting Systems ................................................................. 111
6.4 Summary and Conclusions .................................................................................. 114

CHAPTER 7 .................................................................................................................. 116
SUMMARY OF CONTRIBUTIONS AND FUTURE WORK ......................................... 116
7.0 Introduction ......................................................................................................... 116
7.1 Research Objectives and Results ....................................................................... 116
7.2 Research Contributions ....................................................................................... 118
7.3 Future Work ........................................................................................................ 119

REFERENCES .......................................................................................................... 120
APPENDICES ............................................................................................................. 127
  Appendix A: Functional Requirements Tables ..................................................... 127
  Appendix B: Non-functional Requirements Tables .............................................. 132
  Appendix C: Data Dictionary ............................................................................... 135
  Appendix D: Activity Diagrams .......................................................................... 137
  Appendix E: Use Case Descriptions ................................................................. 146
  Appendix F: UML Class Diagrams .................................................................. 151
  Appendix G: Sample Questionnaire ............................................................... 156

LIST OF PUBLICATIONS .......................................................................................... 159
LIST OF FIGURES

Figure 2-1: Lever machines’ ballot outlines (AVM ballot – top (The New York Times 2010), Shoup machine ballot – bottom (American History 2004)) .......................................................... 8
Figure 2-2: Votomatic punch card machine (National Center for Case Study in Science) .............. 11
Figure 2-3: DataVote voting machine (Smithsonian National Museum of American History) 11
Figure 2-4: External devices for disabled voters ....................................................................... 13
Figure 2-5: Data flow diagram of visual cryptography implementation in general ....................... 19
Figure 2-6: Image steganography mechanism ........................................................................... 21
Figure 2-7: Original image ........................................................................................................ 22
Figure 2-8: Stego-image (original image which has been embedded with a hidden message) 22
Figure 2-9: Quantization process of JPEG compression .......................................................... 24
Figure 2-10: Message encoding process of F5 Steganography algorithm (Westfield 2001) .... 26
Figure 2-11: A part of color indices in a Palette-based image (marked in square) that shows color similarity in the neighboring pixels ........................................................................ 26
Figure 2-12: Encoding Process of Spread Spectrum (Marvel, Boneclet & Retter, 1999) ......... 28
Figure 2-13: Decoding Process of Spread Spectrum (Marvel, Boneclet & Retter, 1999) .......... 28
Figure 3-1: Sequence diagram of basic E2E voting system’s mechanism ................................. 33
Figure 3-2: Screenshot of vote verification page of Helios Voting System ............................... 35
Figure 3-3: Process of Invisible Ink printing (Adida and Neff 2006) ........................................ 40
Figure 3-4: Sample ballot used in Prêt a Voter Voting System ................................................. 43
Figure 4-1: Integrated eVote System ......................................................................................... 56
Figure 4-2: ERD of eVote .......................................................................................................... 58
Figure 4-3: Each component usage of MVC design pattern concept (Netbeans 2013) ........... 60
Figure 4-4: Behavior of MVC design pattern concept in a Sequence Diagram ..................... 60
Figure 4-5: Package Diagram of eVote ..................................................................................... 61
Figure 4-6: eVote’s business processes in Use Case Diagram .................................................. 62
Figure 5-1: Excerpted source code of Password Hashed-Based Algorithm implementation in eVote System (Salted Password Hashing) .......................................................... 68
Figure 5-2: Excerpted source code of PRNG function in SecureRandom Class (Oracle 2011). 69
Figure 5-3: Excerpted source code of Visual Cryptography implementation in eVote System (Visual Cryptography) .............................................................. 70
Figure 5-4: Excerpted source code of Threshold Decryption Algorithm implementation in eVote System ........................................................................................................... 71
Figure 5-5: Comparison of initial and stego-image size on different implementation of image steganography techniques .......................................................... 72
Figure 5-6: Sequence diagram of eVote v1.0 system ................................................................. 73
Figure 5-7: Sequence diagram of verifyReceipt() function ..................................................... 75
Figure 5-8: Sequence diagram of overlayImages() function ........................................75
Figure 5-9: Process flow diagram of eVote v1.0 system ........................................76
Figure 5-10: Screenshot of the email received by an eligible voter ......................77
Figure 5-11: Sequence diagram of voter registration process ...........................78
Figure 5-12: Sequence diagram of addEligibleVoter() function .........................78
Figure 5-13: Screenshot of polling officer’s homepage shown upon successful registration process .................................................................79
Figure 5-14: Excerpted source code of Password Hashed-Based algorithm implementation in eVote authentication stage (Salted Password Hashing) .......................................................80
Figure 5-15: Sequence diagram of user authentication process ...........................80
Figure 5-16: Screenshot of polling officer’s homepage .........................................81
Figure 5-17: Pseudo-code of F5 Image Steganography algorithm applied in voting stage (Provos & Honeyman 2003) ..................................................................................................................82
Figure 5-18: Screenshot of the voter’s vote receipt received by the voter .............83
Figure 5-19: Process flow diagram of voting stage ...............................................83
Figure 5-20: Sequence diagram of secret keys distribution process by the system administrator ..........................................................................................................................84
Figure 5-21: Excerpted source code of tally result list generation .......................84
Figure 6.1: Number of participants based on their age .......................................87
Figure 6-2: Number of participants based on their gender and highest level of education ........87
Figure 6-3: Screenshot of JUnit test cases implementation for Password Hashed-Based scheme ..........................................................................................................................89
Figure 6-4: Screenshot of JUnit test cases results for Password Hashed-Based scheme ..........89
Figure 6-5: Screenshot of JUnit test cases implementation for Visual Cryptography scheme ..90
Figure 6-6: Screenshot of JUnit test cases results for Visual Cryptography scheme ........90
Figure 6-7: Screenshot of image steganography secret data encoding test case and its results ..91
Figure 6-8: Screenshot of image steganography secret data decoding test case and its results ..91
Figure 6-9: Screenshot of secret keys generation test case and its results ...................92
Figure 6-10: Screenshot of voter registration page ................................................93
Figure 6-11: Screenshot of user input validation during voter registration process ................93
Figure 6-12: Screenshot of voter homepage upon successful registration ...............94
Figure 6-13: Screenshot of polling officer login page ..........................................94
Figure 6-14: Screenshot of polling officer homepage upon successful authentication ........95
Figure 6-15: Screenshot of voting page accessible only for the voters ....................96
Figure 6-16: Screenshot of ballot confirmation page ..............................................96
Figure 6-17: Screenshot of voting summary page ...............................................97
Figure 6-18: Screenshot of key distribution page for to the system administrators ..........97
Figure 6-19: Screenshot of tally summary page upon successful secret keys distribution by the system administrators .................................................................98
Figure 6-20: Screenshot of polling officer profile page ............................................................. 98
Figure 6-21: Screenshot of secret key submission in tally result page from polling officer’s screen ............................................................. 99
Figure 6-22: Screenshot tally result page upon successful secret key submission ...................... 99
Figure 6-23: Screenshot of tally result page ............................................................................ 100
Figure 6-24: Screenshot of vote verification page for the voters ................................................ 100
Figure 6-25: Screenshot of vote verification page upon successful verification ....................... 101
Figure 6-26: Evaluation of the Visibility of System Status ..................................................... 102
Figure 6-27: Evaluation of the Match between the System and the Real World ........................... 102
Figure 6-28: Evaluation of the Consistency and Standards ..................................................... 102
Figure 6-29: Evaluation of the Aesthetic and Minimalist Design ........................................... 103
Figure 6-30: Evaluation of the User Control and Freedom ..................................................... 103
Figure 6-31: Evaluation of the Help Users Recognize, Diagnose and Recover from Errors ... 104
Figure 6-32: Evaluation of the Error Prevention ...................................................................... 104
Figure 6-33: Evaluation of the Recognition Rather than Recall ............................................ 105
Figure 6-34: Evaluation of the Flexibility and Efficiency to Use .............................................. 105
Figure 6-35: eVote Perceived Ease of Used Evaluation .......................................................... 107
Figure 6-36: eVote Perceived Usefulness Evaluation .............................................................. 107
Figure D-1: Activity Diagram of Election Setup process ........................................................ 137
Figure D-2: Activity Diagram of Add Eligible Voter or Officer process ................................ 138
Figure D-3: Activity Diagram of Ballot Monitoring process ..................................................... 139
Figure D-4: Activity Diagram of Tally Checking process ....................................................... 140
Figure D-5: Activity Diagram of Profile Editing process ..................................................... 141
Figure D-6: Activity Diagram of Registration process ............................................................. 142
Figure D-7: Activity Diagram of Authentication process ...................................................... 143
Figure D-8: Activity Diagram of Voting process ................................................................. 144
Figure D-9: Activity Diagram of Vote Verification process ................................................... 145
Figure F-1: Class Diagram of ballot box package ................................................................... 151
Figure F-2: Class Diagram of cryptography package ............................................................. 151
Figure F-3: Class Diagram of controller package ..................................................................... 152
Figure F-4: Class Diagram of entity package .......................................................................... 153
Figure F-5: Class Diagram of steganography package ........................................................... 154
Figure F-6: Class Diagram of session package ....................................................................... 155
List of Tables

Table 4-1: Legend of Use Case Diagram ................................................................. 61
Table 5-1: Robustness of different image steganography methods against visual and statistical attacks ................................................................. 72
Table 6-1: Comparison of E2E Voting Systems’ Components ..................................... 112
Table 6-2: Comparison of E2E Voting Systems based on its defense mechanism against common potential threats ................................................................. 113
Table A-1: System authorization ............................................................................ 127
Table A-2: System authentication ........................................................................... 127
Table A-3: Applied technologies ........................................................................... 128
Table A-4: System documentation ........................................................................ 129
Table A-5: Data type ............................................................................................. 130
Table A-6: Data handling ....................................................................................... 131
Table B-1: System security .................................................................................... 132
Table B-2: System usability ................................................................................... 134
Table B-3: System performance ............................................................................ 134
Table B-4: System reliability ................................................................................ 134
Table C-1: eVote’s database design ....................................................................... 135
Table E-1: Use Case description of election setup process ....................................... 146
Table E-2: Use Case description of add eligible voter or officer process .................... 146
Table E-3: Use Case description of ballot monitoring process ................................... 147
Table E-4: Use Case description of tally checking process for system administrator ...... 147
Table E-5: Use Case description of tally checking process for polling officer ............. 148
Table E-6: Use Case description of profile editing process ....................................... 148
Table E-7: Use Case description of registration process ........................................... 149
Table E-8: Use Case description of authentication process ....................................... 149
Table E-9: Use Case description of voting process .................................................. 150
Table E-10: Use Case description of vote verification process .................................... 150
GLOSSARY

- **Abstain**: Formally decline to vote either for or against a proposal or motion.

- **AC Coefficients**: Part of DCT coefficients. It is the remaining 63 coefficients with non-zero frequencies.

- **Activity Diagram**: A diagram that describes various actions and their sequence in the system workflows.

- **Agile**: One of the SDLC methods that adopt iterative and incremental model for development.

- **Authentication Stage**: One stage in an election where registered voters are authenticated by logging into the system to ensure voters’ identity.

- **Ballot Box**: An equipment or tool to collect the overall cast votes.

- **Ballot Card**: A type of paper-ballot used in punch card machine.

- **Ballot Counting**: The process of counting all of the collected ballots at the centralized box.

- **Ballot Initialization**: The process of initializing and preparing an electronic ballot in an electronic voting system election.

- **Ballot Marking**: The process of marking the provided ballot by a voter in order to cast a vote.

- **Ballot Transmission**: The process of transmitting or transferring all of the collected ballots from the individual boxes at the polling station into a centralized box.

- **Ballot**: An instrument used to cast votes in an election.

- **Ballot-marking Device**: An aid (instrument) for marking a non-electronic ballot.

- **Bcrypt() Function**: An enhanced key stretching approach that is based on Blowfish encryption algorithm and uses a 128-bit salt value and encrypts a 192-bit magic value.

- **Bit Commitment Scheme**: A commitment scheme in information security where the value chosen is a bit.
- Blind Signature Scheme: A digital signature method to secure the content of a message by blinding it before it is signed.
- Blofwish Encryption Algorithm: A symmetric block cipher that can be used as a drop-in replacement for DES or IDEA.
- Braille Keypad: A special designed keyboard indicated as an aid for disabled (blind) people
- BSM: Binary Similarity Measures.
- Bulletin Board: An append-only broadcast channel to publish the necessary information of voters and their votes without revealing its value.
- Chi-square Test: One of the statistical steganalysis.
- Chrominance Data: Image pixels over part of the image.
- Ciphertext: The product of encryption performed on digital file or text using a cryptography algorithm.
- Class Diagram: A representation of a detailed view of a single business process uses case and shows the class that participate in that particular use case.
- Completeness: All valid votes are counted correctly.
- Computational Cost: The overall computation required for a system to execute its processes.
- Convenience: The system must allow voters to cast their votes quickly, in one session with minimal equipment or special skills.
- Cover Image: The media to carry the secret message in steganography.
- Crypt() Function: A key stretching approach that uses salt value to strengthen one-hash function algorithm against the attacks.
- Cryptographic Hash-Function: The algorithm often used in user authentication by encrypting the original message to cryptographic hash value: The end product of encrypted cryptographic hash-function.
- Cryptography: The art & science of preventing users from unauthorized or illegal actions towards information, networking resources and services.
- CSS: Cascading Style Sheet.
• Cut and Choose Scheme: A type of commitment scheme that based on zero-knowledge proof.

• Data Insertion: The process of embedding secret data into a digital file carrier.

• DC Coefficients: Part of DCT coefficients. It is the coefficient with zero frequency in both dimensions.

• DCT Quantization: The process of taking the remaining coefficients and dividing them individually in a block against the pre-determined set of values and then rounding the results to the nearest real number value.

• DCT: Discrete Cosine Transform.

• Decryption: The process of converting data into plaintext.

• DES: Data Encryption Standard.

• Digital Signature Scheme: A technique for non-repudiation based on the public key cryptography. It is attached to the secret message to guarantee the source and identity of that message.

• Document-based E-voting System: A type of electronic voting system where voters could cast their vote electronically by using paper ballot. It is an integration of traditional polling status voting and electronic voting system.

• DoS Attack: Denial of Service, one type of cyber-attacks that focused on disrupting availability. It is also known as Denial of Service or Distributed Denial of Service Managing DoS Attacks.

• DRE Machine: An electronic implementation of mechanical lever system.

• E2E System: End-to-End Verifiable Voting System, a type of E-voting system that ensure universal and individual (voter) verifiability.

• EJB: Entity Java Bean.

• Election Officials: A group of individuals who are authorized to coordinate an election.

• Election Trustees: A group of individuals who are entitled to ensure the integrity of an election.
• ElGamal Encryption: An asymmetric public-key encryption algorithm based on Diffie-Hellman key exchange.

• Eligibility: Only authorized voters are allowed to vote.

• Encryption: The process of converting data or information into code.

• ERD: Entity Relational Diagram.

• eVote: The name of the proposed system in this research.

• F3: The initial JPEG image steganography algorithm proposed by Westfeld. It allows the secret message to be embedded within the AC coefficient values that is equal to 1 and it does not embed the message by overwriting the LSBs of the DCT coefficients directly.

• F4: The enhancement of F3 algorithm. It eliminates the histogram attack issue by mapping negative coefficients to the inverted steganographic value.

• F5: The extended version of F4 that implements two additional features to enhance its algorithm, namely Matrix encoding and Permutative Straddling.

• Fairness: Nothing must affect the voting (No one can indicate the tally before the votes is counted).

• FK: Foreign Key.

• Forced-abstention Attack: One type of external source security attacks where the attackers would attack the system by forcing the voters to refrain from casting their votes.

• Functional Requirements: A set of tasks or functions the system is required to perform. It summarizes the intended behavior of the system.

• Glassfish: A cross-platform open-source Application Server for Java EE platform.

• GUI: Graphical User Interface.

• Helios: An electronic voting system created by Ben Adida in 2008. It is an open-source web-based voting system that offers verifiable online elections for anyone.

• Homomorphic Encryption: A scheme of securing cast votes by encrypting each of those votes. Vote tabulation is to be carried out by decrypting the combination of the encrypted votes.
• HTML: Hypertext Markup Language.
• HTTP: Hypertext Transfer Protocol.
• HTTPS: Hypertext Transfer Protocol Secure.
• Huffman Algorithm: The entropy encoding algorithm used for lossless data compression.
• HVS: Human Visual System.
• IDE: Integrated Development Environment.
• Image (spatial) Domain: Stenography technique that embed messages in the intensity of the pixels directly.
• Incoercibility: It is not possible for anyone but the voters themselves to acquire any information regarding their secret ballots, even if the voters are untrustworthy (assuming that the voting process would be done by the voter in private).
• Integration Testing: a continuation of Unit Testing that focuses on the interaction of several components or units which have been tested previously in the unit testing.
• Invisible Ink: A customized ink used as an election tool that allows the voters to retain their receipts in a secure and secret manner with the help of unique confirmation codes on each ballot that no attackers would be able to coerce.
• Iteration: One of the SDLC methods that adopt iterative and incremental model for development.
• Iterative Waterfall: The modified version of the waterfall model where the phases can be carried out with no regard to the sequences of the phase itself.
• Java EE: Java Platform Enterprise Edition.
• Java FX: Java platform for RIAs developments.
• JDBC: Java Database Connectivity.
• JDK: Java Development Kit.
• JMS: Java Message Service.
- JNDI: Java Naming and Directory Interface.
- JPA: Java Persistence API.
- JSP: Java Server Pages.
- JTA: Java Transaction API.
- JUnit: A unit testing framework used to support quick and easy generation of test suites to conduct unit testing.
- Kiosk Internet Voting System: An extended version of remote electronic voting system that limits voters’ accessibility by only allowing the voters to cast their vote in specific kiosks in particular locations without any supervision from the polling officers.
- LAN: Local Area Network.
- Lossless Compression: Image compression technique that preserves the original image’s integrity. The decompressed image's output is bit-by-bit identical to the original image input.
- Lossy Compression: Image compression technique that creates smaller files by discarding excess image data from the original image.
- LSB: Least Significant Bit.
- Matrix Encoding: Encoding algorithm implemented in F5 image steganography algorithm. It uses binary Hamming codes to encode more bits per encoding change.
- MDC: Modification Detection Code, a cryptographic hash function.
- Mechanical Lever Machine: A mechanized vote-processing tool used as an alternative for conducting an election in more efficient manner without raising security concerns.
- Mixed-net Scheme: A scheme that implements mix-network (anonymous channel) to secure the ballots by detaching the voters with their votes. This channel takes encrypted votes as inputs and shuffles the votes into a plaintext format.
- Mobility: No restrictions on the location from where a voter can cast a vote.
- MVC: Model-View-Controller.
- MySQL: One of the relational database management systems in IT industry.
- Netbeans: One of the Integrated Development Environment (IDE) as well as a generic application platform framework initially developed for Java Desktop applications.
- Non-document based E-voting system: A type of electronic voting system where voters can easily vote electronically in private with the ability to verify their vote at the end of the poll.
- Non-functional Requirements: A set of tasks or functions that shall be performed by the system. It sums up the whole operations of the system.
- Oblivious Signature Scheme: A combination of RSA and blind signature scheme.
- One-way Function: A cryptographic function with irreversible ciphertext.
- Onion: Encrypted information in Prêt a Voter's paper ballot.
- OO: Object Oriented.
- Optical Scan: A system that uses an optical scanner to read marked paper ballots to collect and count ballots.
- Package Diagram: A type of diagram that displays two or more packages and their dependencies.
- Pailier Cryptosystem: An asymmetric public-key encryption algorithm that offers additive homomorphism.
- Palette-based: A color-indexed image where its colors are stored in a pre-determined color palette list.
- Paper-ballot: An instrument or tool used by the voters to cast their votes.
- Password Hashed-based Scheme: A password’s authentication algorithm that does not require extensive computation, yet it is proven to be cryptographically secure.
- Patchwork: A statistical steganography technique that implements redundant pattern encoding to embed a secret message in an image.
- PBKDF2: Password-Based Key Derivation Function.
- PBKDF2WithHmacSHA1: An algorithm of secret keys construction found in RSA Laboratories’ Public-Key Cryptography Standard (PKCS) #5 v2.0.
• Perceived Ease of Use: The degree to which a user believes that the use of a particular system would require less effort compare with another systems.

• Perceived Usefulness: The degree to which an individual believes that a particular system would enhance his or her job performance.

• Performance Cost: The overall time required for a system to execute its processes.

• Phantom votes: blank votes.

• PK: Public Key.

• PKCS: Public-Key Cryptography Standards.

• Poll Site Internet Voting System: The extended version of remote electronic voting system where the election is conducted under polling officers’ supervision in allocated polling sites. Their ballot would then be transferred over the internet to the centralized tally center where all ballots are collected and tabulated.

• Polling Officer: The individuals who are coordinating the overall processes of an election.

• Polling Station Voting: An election conducted in public where Voters can mark their choice in a private booth in the polling station and put the ballot papers into a sealed ballot box.

• Permutative Straddling: Permutation technique to shuffles (permutes) all of the quantized DCT coefficients by using Pseudo Random Number Generator (PRNG) key in order to spread out the embedded data evenly throughout the cover image.

• Prêt a Voter: An electronic voting system that implements Visual Cryptography approach in an election procedure to offer vote verifiability.

• Privacy: All votes must be secret.

• PRNG: Pseudo-Random Number Generator.

• Punch Card Machine: A device used in modern day election to eliminate the mechanical lever voting machines.

• Punchscan: a precinct-read optical scan balloting system that allows the user to take their ballot with them after scanning.
• RAD: Rapid Application Development, one of the SDLC methods that adopt evolutionary prototyping model for development.

• Randomization Attack: One type of an external source security attack in an election with the intention is to nullify the voter’s vote by submitting as many forged ballots (randomly composed ballot material) as possible.

• RDBMS: Relational Database Management System.

• Receipt-freeness: Each voter can neither obtain nor be able to construct a receipt to prove the content of their ballot to anybody else.

• Registration Stage: One stage in an election where all constraints for the election are prepared.

• Remote E-voting System: Also known as remote online voting system. It is the improved and enhanced version of preliminary voting systems in order to fulfill voters' demands, where voters can cast their votes from anywhere, provided internet connection is available.

• RIA: Rich Internet Applications, platform independent.

• RIES: Rijnland Internet Election System.

• Robustness: The result reflects all submitted and well-formed ballots correctly, even if some voters and (or) possibly some of the dishonest election officials cheat.

• Rocker Paddle: A device designed to help voters with lower body paralysis.

• RPC: Randomized Partial Checking.

• RS Analysis: Regular-Singular analysis.

• Salt Value: The generated value for each password inserted by the user before hashing to randomize its hashing.

• Scantegrity II: A practical enhancement for optical scan voting systems, which achieve increased election integrity through a novel use of confirmation codes printed on ballots in invisible inks.

• Scantegrity: An improved version of Punchscan that combines the implementation of two separate sheets into one individual sheet in its ballot mechanism.
- Schnorr Identification Scheme: A scheme for an individual to prove his identity electronically without revealing his identifying information.
- Scrum: One of the SDLC methods that adopt iterative and incremental agile model for development.
- SDLC: Software Development Life Cycle, a methodology for design and implementation of an information system.
- Secret-ballot Receipt: A security technique that ensures election integrity by preventing altered votes due to vote buying or selling and vote coercion.
- SHA1PRNG: The default Pseudo-Random Number Generator (PRNG) of SecureRandom Class.
- Simulation Attack: One type of external source security attacks where dishonest voters could sell their rights to vote by giving their private keys to the attacker who would cast the vote on their behalf.
- Sip and Puff Device: A binary device designed to help voters with upper body paralysis
- Soundness: A dishonest voter cannot disrupt the voting.
- Spiral Model: One of the SDLC methods that adopt incremental, waterfall and evolutionary prototyping model for development.
- Spread Spectrum: A statistical steganography technique that hides data by spreading it throughout the cover image.
- SRS: Software Requirements Specification.
- SSL: Secure Socket Layer.
- Steganography: A branch of information security technique that has not been commonly used in E2E voting systems but is also included in the software architecture design.
- Stego-image: The product of encryption performed on digital file or text using a steganography algorithm.
- System Administrator: The individuals who are responsible to configure, support and maintain the system in an election to ensure its reliability.
- **System Testing**: A testing conducted on the final version of a system to evaluate its compliance with its specified requirements.
- **Tallying Stage**: One stage in an election where all of the collected ballots are counted.
- **TAM**: Technology Acceptance Model.
- **Threshold Decryption Cryptosystem**: A cryptographic system that secures and provides reliable key management for cryptographic system.
- **Transform (frequency) Domain**: Steganography technique where images are first transformed and then only the secret message is embedded in the image.
- **TTPI**: Trusted Third Party Internetstemmen.
- **UML**: Unified Modeling Language.
- **Unit Testing**: A type of testing performed by doing individual test on single units of the system rather than on the entire system.
- **Universal-verifiability**: The ability of an E-voting system to offer election transparency to its users.
- **Unreusability**: No voter can vote twice.
- **Use Case Diagram**: A diagram that is used to conveniently document the system activities.
- **User Acceptance Testing**: A testing conducted in order to consider not only technical factors to support the system’s performance, but also to consider the behavioral factors of the users.
- **Verifiability**: No one can falsify the result of the voting.
- **Visual Cryptography**: A method for protecting image-based secrets that does not involved any cryptographic computation in decoding its shares.
- **Vote Receipt**: A receipt used in the implementation of E-voting system to assure the voters that their intended votes are counted as cast.
- **Vote Tabulation**: The process of counting all of the collected cast votes.
- **Vote Tabulation**: The process of counting ballots after vote casting in an election.
- **Vote Tampering**: One of the examples of fraud conducted in an election.
• Vote Verification Stage: One stage in an election where the election result is verified by the voters by ensuring that their submitted ballots are counted as cast.

• Vote-Coercion: The possibility for anyone besides the voter to acquire any information of the voter's secret ballot.

• Voter: The main individuals in an election who are entitled to cast the votes.

• Voter-verifiability: The ability of an E-voting system to offer vote verifiability to the voter with the implementation of vote receipt.

• Voting Stage: One stage in an election where the voters cast their votes.

• Waterfall: The primary SDLC method with five distinct phases, namely requirement analysis, design, implementation, testing and maintenance.

• YCbCr: A family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y' is the luma component and CB and CR are the blue-difference and red-difference chroma components.

• YUV: A color model that determines a color space of Y component that corresponds to luminance (brightness), and the U and V elements.
CHAPTER 1

SIGNIFICANCE AND SCOPE OF THE THESIS

1.0 Introduction

This chapter briefly describes the overall context of the research done. It includes research background along with its objectives followed by a short insight of the research outcome. A thesis outline is also done at the end of the chapter to give an overview of the thesis.

1.1 Research Background

Election remains as a controversial subject since it has played a major role in democratic societies. The election process must maintain a balance between fairness and performance. Most voting procedures are based on paper ballot and are widely used in elections whether at government or organizational level. However, the paper ballot is costly, inconvenient, inflexible and time consuming for voters. Many people nowadays prefer a faster way to vote without neglecting the equity of the procedure. A traditional election procedure does not meet voters’ demands anymore. This might lead to a low turnout in an election. With the rapid growth of computer technology, many researchers are proposing secure, reliable, and convenient electronic voting systems as a substitute to the traditional voting procedure. Compared to the traditional voting system, the new system could offer shorter time in finishing the main operations in the election progress, such as vote casting, authentication, registration and vote tallying processes. In a traditional voting procedure, all these processes would be divided into few stages at few different stations. Implementation of the electronic voting system could help the election officials who administer the election and also minimize the cost of the election itself. If an electronic voting system is designed properly, it can also provide a more secure system than the paper-based election by providing precise data communication and preventing threats and attacks by intruders.

In recent years, researchers are focusing more on developing new technologies which can prevent coercion, provide receipts to ensure voter-verifiability and offer universal-verifiability through the implementation of bulletin board. These three aspects are considered to be the vital components of a reliable election procedure. Voting systems with such characteristics are categorized as end-to-end verifiable voting systems (E2E). Many E2E systems have been proposed and are widely used nowadays (Han Wei et al. 2007; Kiayias et al. 2007; Ryan et al.
In principle, an E2E voting system offers assurance to the voters by distributing a receipt of their vote after they have cast their votes, which can be used verification purpose from the overall tabulation of the collected votes. This receipt could not be used as a proof for vote buying or coercion but all of those encrypted receipts will be posted publicly in a read-only Bulletin Board after each voter finishes the voting process. Thus, the E2E system could still protect the voter’s privacy and ensure election’s integrity.

### 1.2 Proposed Solution

The proposed E2E voting system is based on earlier design concepts but differs from previous work because of the implementation of two distinct schemes. They are cryptography and steganography. Both components of information security are combined in a layer of data protection. In electronic voting, cryptography is a commonly used technique as it is a good defense against threats. But, steganography has rarely been used as an additional layer of security in electronic voting system. Therefore, the author believed it can offer a better solution for threats and risks that might occur. The combination of these two schemes is the proposed novel approach in this research project, mainly implemented to secure the communication between the user and the server. It is expected to produce an improved technique which could meet the voter’s demand and perform with a less performance cost in a secure manner.

In order to ensure the integrity of an election, many schemes have been implemented and proposed. In these schemes, cryptography is used to protect the data transmitted between the voter and the server to ensure that it would not be leaked to a third party. Cryptography techniques are also applied in each process in the system to make sure the authenticity of the voter, the originality of the ballots cast and collected votes, the reliability of the tallied votes and the privacy throughout the election. There are many cryptography methods that can be applied, such as blind signature scheme, homomorphic encryption, oblivious signature scheme, bit commitment scheme, Schnorr identification scheme, mixed-net schemes, digital signature scheme, secure multi-party computation, cryptographic hash-function, etc. However, in this research only a few selected schemes are used. These are applied in different voting stages to preserve the main characteristics of an electronic voting system. The selected schemes are password hashed-based scheme, visual cryptography as adopted from secret-ballot receipts proposed by Chaum (2004) and threshold decryption cryptosystem.

Steganography - a branch of information security technique that has not been commonly used in E2E voting systems but is also included in the software architecture design. It is the science of hiding information in communications, where no one other than the sender and receiver would know the existence of hidden information (Provos & Honeyman, 2003). In 2007, Hong and
Hong stated steganography pays less attention to intentional attacks since it focuses more on data insertion capability. However, as information technology evolved and more threats arose, it became necessary to develop more secure steganography algorithms. The advantage of steganography over cryptography is its ability to offer more advanced way of hiding a secret. Therefore, steganography is proposed to be used to secure the data communication in this research. This scheme provides secret communication accessible by encoding a secret message to various types of cover data such as text, images, audio, video file format. Each covered data has multiple methods to hide the secret message. Unlike cryptography, the output data of steganography (stego-object) would still look the same as its input data. As a result, it would be difficult to identify and interpret the hidden secret in the stego-object. For electronic voting system implementation, both image and text steganography are appropriate candidates. They have a higher degree of redundancy, which allow larger size data to be encoded into the cover file. Other than that, they are also unlikely to raise any suspicion because they are the most common transmitted data between the voter and the server. However, image steganography, offers a better encoding technique to be used as it can hide the secret message by securely transferring a hidden secret in a digital image file. Hence the implementation of image steganography is proposed in this research project.

There are five different stages in the system design architecture, namely the registration stage, authentication stage, voting stage, tallying stage, verification stage. The secret-ballot receipts theorem introduced by David Chaum (2004) is mainly a combination of cut and choose scheme together with a cryptography technique by Naor and Shamir (1994), visual cryptography. This scheme is applied and modified in this research project. Thus, visual cryptography will be implemented in the voting stage and verification stage as part of secret-ballot receipt implementation. Right after a vote is cast, steganography will be used throughout the system processes for data communication purpose. In the tallying stage, the threshold decryption cryptosystem will be implemented. The combined method is believed to be sufficient to provide a secure, reliable and convenient voting system. Since the proposed tool is an electronic voting system, it is necessary to assume that the voter would complete the voting process secretly.
1.3 Research Objectives

The set of studies carried out in this research can be summarized into the research objectives listed below.

(a) To improve the quality of election procedure in an electronic voting system based on its security and usability aspects by incorporating cryptography and image steganography in the system architecture.

(b) To minimize the computational and performance cost required in electronic voting system.

It is believed that a secure, reliable, convenient, and efficient voting system could be delivered if the objectives of this project are achieved. The combination of both cryptography and image steganography needs further studies in order to obtain an optimized scheme for the system. Thus, a novel and advanced approach for secure electronic voting system will be investigated, examined and introduced. This research project results in a Java EE 6 web-based application to cater a reliable, secure and efficient voting process in a small to medium scale election. Some aspects of possible security attacks and some other counter measures are taken into consideration in this project to ensure the reliability of the proposed software. They are the non-functional of the E2E verifiable voting system. Those requirements include the following (Fujioka et al. 1992; Benaloh 2006; Gritzalis 2002; Cetinkaya 2008; Kofler et al. 2003; Aditya 2005):

(a) Completeness – All valid votes are counted correctly.
(b) Soundness – A dishonest voter cannot disrupt the voting.
(c) Privacy – All votes must be secret.
(d) Unreusability – No voter can vote twice.
(e) Eligibility – Only authorized voters are allowed to vote.
(f) Fairness – Nothing must affect the voting. (No one can indicate the tally before the votes are counted)
(g) Verifiability – No one can falsify the result of the voting.
(h) Robustness – The result reflects all submitted and well-formed ballots correctly, even if some voters and (or) possibly some of the dishonest election officials cheat.
(i) Incoercibility – It is not possible for anyone but the voters themselves to acquire any information regarding their secret ballots, even if the voters are untrustworthy (assuming that the voting process would be done by the voter in private).
(j) Receipt-freeness – Each voter can neither obtain nor be able to construct a receipt to prove the content of their ballot to anybody else.
(k) Mobility – No restrictions on the location from where a voter can cast a vote.
(l) Convenience – The system must allow voters to cast their votes quickly, in one session with minimal equipment or special skills.

1.4 Thesis Outline

This thesis is divided into seven distinct chapters, apart from references and appendices. The contents of the chapters are summarized as follows:

Chapter One: Significance and Scope of the Thesis, introduces the topic of the research, describes the research background, aims and objectives, and the contributions made.

Chapter Two: Literature Review, reviews literature related to this research. It covers two distinct types of voting systems, together with the known threats encountered by each system. The main security schemes implemented are also described in detail to support the motive behind implementing those methods.

Chapter Three: Studies of Related E-voting Systems, discusses related published work in detail. In this chapter an examination of four electronic voting systems is carried out to show why a new system is proposed. Each of them have been implemented in an actual medium to big scale elections

Chapter Four: Secure E-voting Analysis and Design, elaborates the design approach of the E-voting system. In this chapter each proposed stage of the voting process is highlighted. The proposed software architecture is derived based on users’ demand and previous researchers’ evaluation and recommendations.

Chapter Five: System Implementation, documents the implementation of security techniques conducted on the system. The non-functional requirements of E-voting system stated in Chapter 5 are used as benchmark to evaluate the proposed system.

Chapter Six: System Testing and Consideration of Security Attacks on the System, evaluates the proposed system to ensure its reliability, security and efficiency. This chapter also contains discussion of some security attacks on electronic voting system and how the proposed system could counteract such attacks.

Chapter Seven: Summary of Contributions and Future Work, concludes the new knowledge which has been attained from completing this master’s research project together with the future work that may be carried out to improve the proposed system.
CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

One of the most important factors of a clean and efficient election is to have a good and secure voting system. The voting process is a set of steps that allows voting and ensures the ultimate goal of the election (Santos & Querioz, 2011). This chapter describes the related literature for this research project. First, it covers two major categories of voting systems in general; traditional polling station voting system and electronic voting system. Both these voting systems are vulnerable to threats and attacks that need to be anticipated in order to achieve its aims. A traditional voting system is the most common procedure applied for general election. As a result, its threats and attacks are commonly known to the society. The main focus of this literature review lies on electronic voting systems’ threats and attacks. It is explained in the later part of this chapter.

2.1 Categories of Voting Systems

Voting systems are divided into two main categories. Most elections are done by manually submitting paper ballots to centralized ballot boxes located in few different venues. This procedure is categorized under traditional voting systems. However, in modern day the election process is often carried out by electronic voting systems (Carback et al. 2010). Both procedures are described in more detail below.

2.1.1 Traditional Polling Station Voting Systems

Conventionally, election was conducted openly in public. They were mostly held in the town halls. In the first method, voters would need to call their chosen candidate’s name aloud in front of other voters. The second method was to write down one’s choice on pre-distributed blank papers and place them in the designated polling place. Such procedures violated users’ privacy and could easily lead to vote tampering and fraud. Thus, in order to prevent such violations in the elections, various types of voting procedures were introduced and applied. Below are the descriptions of each of these procedures.
2.1.1.1 Paper-ballot

Paper-ballot system has been in use for a long time and has been used in many elections throughout the world. It was initially implemented at the 1856 Victoria state election in Australia. In this method, election officials hand out the official paper ballot to each voter and verify their suffrage. On every paper ballot, names of candidates are listed. Voters then mark their choice in a private booth in the polling station and put the ballot papers into a sealed ballot box. The officials count these ballots manually once the polling is complete. All of these processes need to be carried out on the Election Day itself.

One of the apparent disadvantages of this system is accessibility. Voters must come to the assigned polling station in order to vote for their candidate. This would decrease the chances for eligible voters to cast their votes, resulting in low percentage of polling. Additional procedures were introduced in some countries to address this problem, such as home voting, postal voting and advance voting. However, these methods do not eliminate the threats and risks fully. For example, home voting election is held to accommodate electorates who are not able to go to the polling station. The election officials will go to their houses and provide controlled environment for voting (Volkamer 2009, p. 15). Controlled environment here means a free and secret environment for the voters to cast their vote under the officials’ supervision. Both home and postal voting still use paper-ballots issued to the eligible voters by the election officials. The security of home voting relies on trustworthiness of election officials. Postal voting gives more convenience to the voters in terms of casting their votes. Voters can do it from anywhere and post their ballot to the central polling station. Postal voting allows voters to vote without the supervision of election officials, because of which voters would not have the advantage of getting their ballot checked for common errors, for example undervote or overvotes or of getting answers to their queries from election officials. Further, without supervision, vote-coercion cannot be prevented.

2.1.1.2 Mechanical Lever Machines

More than a century after the first paper-ballot voting system was implemented, election specialists introduced voting machines. It is a mechanized vote-processing tool used as an alternative for conducting an election in more efficient manner without raising security concerns. One of the well-known voting machines is the mechanical lever machine.

Jacob H. Myers patented Mechanical Lever voting machine in 1889. It was named Myers Automatic Booth (later also known as Automatic Voting Machine or AVM) and debuted three years after in Lockport town election, New York. According to Myers, his mechanical lever voting machine would, “protect mechanically the voter from rascaldom, and make the process
of casting the ballot perfectly plain, simple and secret” (Jones 2001). Thus, the number of residual votes as well as the possibilities of vote tampering in an election could be diminished. This could be accomplished because of its automated vote tabulation process, whereby manual counting was no longer required. Due to its popularity, by 1930 mechanical lever voting system was used in elections in most of the major cities in the United States, and by 1960s more than half of the states in the United States employed mechanical lever machines. Over the years, Ransom F. Shoup made a few improvements to the AVM. One of these improvements is the ease of transporting and storing the machine. His enhanced machine is known as the Shoup machine, which could be used with or without electric power. Two separate manufacturers, The AVM Corporation and Shoup Voting Machine Corporation, supplied the two machines. The difference between the two machines lies in their tabular ballot outline displayed on the front panels. For AVM, candidates are represented in the party rows under each of its office title. On the other hand in Shoup machine, candidate list are shown horizontally in each of its office or party column. The front panels of both machines are displayed in Figure 2-1 for clearer view of these machines.

![Figure 2-1: Lever machines' ballot outlines (AVM ballot – top (The New York Times 2010), Shoup machine ballot – bottom (American History 2004))]
Although their front panels are different, both machines operate in a similar manner. The machines ensure secrecy of the ballots as no paper-ballots are used. All votes are recorded in the counters located at the rear of the machine. This part of the machine remains closed until the end of the polling process. In order to cast their votes, voters must enter a private booth where a lever machine is placed. The voting machine is enabled by the election officials. Voters flip the lever next to or on top of the chosen candidate’s name and pull the lever to the right to start the voting process. As they do this, an “x” would appear next to the candidate’s name as an indicator that the voter has selected that particular candidate. Voters could also insert write-in vote by writing it down in the allocated slot next to the candidate that they wish to vote for. Once selection is made, they pull the same lever to the left in order to cast their vote and to open the privacy curtain that was implemented to prevent overvotes. The machine records each vote, together with the number of eligible voters who have voted, automatically returns the flipped levers to their original positions and resets all write-in votes as the handle are pulled back. No audit trails (e.g. vote receipts) are provided to recount the votes. Without this, the reliability of the lever machine and hence the voters’ trust in the system will be lowered. Election officials carry out the vote tally process by accessing the rear part of the machine where all the results get displayed. The machine updates these results automatically each time a voter casts his vote. Officials need to copy these results into the tally sheets for record keeping and send the result to the central polling station to be compiled.

Over the years, the implementation of these methods has evolved owing to the voters’ demand to get more secure and efficient election experience.

2.1.2 Electronic Voting Systems

Electronic Voting Systems are divided into two major categories. They are document based electronic voting system and non-document based electronic voting system.

2.1.2.1 Document-based Electronic Voting System

The most commonly used electronic voting system to date is the document-based electronic voting system or also known as paper-based ballot voting system. Through this system, voters could cast their vote electronically which will eliminate the time needed for ballot printing by officials and ballot marking by voters. Vote tabulation and vote recording in these systems are done by the machines. Various machines are available in the market; some of them are explained in detail as follows.

(a) Punch Card

Punch card machine works as a ballot marker device for voters. Different punch card machines have different types of ballot card designs. Some ballot card designs require voters to punch the
ballot on the opposite side of the candidate names, printed on the ballot. In another kind of punch card machines, voters need to insert the ballot card into the punch card machine and punch small and perforated rectangles also known as chads on the ballot according to their choices by using the machine (Ansolabehere 2001, p. 3). In this system, candidate’s names are not listed in the ballot card; instead, they are printed in a booklet available in the voting booth. Two prominent punch card machines, VotoMatic and DataVote. Both, punch card machines and ballots are displayed in Figure 2-2 and 2-3.

A punch card machine is rather inexpensive compared to mechanical lever voting machine. Thus, sufficient amount of machines could be provided in each of the polling station to ensure the effectiveness of the election. After punching ballots, voters submit their punched ballot to the centralized ballot box. Voters can also scan their ballot into a punch card reader before the electronic ballot tabulator carries out vote tabulation. Vote results could be tallied quickly by using this machine. Like paper ballots, punch card ballots can be preserved. Because of these features, punch card technology was widely adopted. As a proof of its recognition, in the 2000 USA Presidential Election, 34.5% of the valid voters used punch card machines to cast their ballots - 31% used VotoMatic and the other 3.5% used DataVote punch card machine (Ansolabehere 2001, p. 4). Punch card voting system led to higher rates of residual voting compared to other voting technologies. Residual vote is the difference in number of valid votes cast to the number of votes cast. In other words, residual vote is a result of invalid (uncounted) votes during the votes tallying process. It might occur due to undervotes or overvotes in an election. Based on its nature, undervotes could be the major reason of residual vote in an election where punch card machine are used. Undervotes can happen due to error in ballot interpretation, due to by improper chads punching. Chad is to be assessed as improper when part of it is still attached to the ballot card, for example hanging chad. It is an event of chad’s corner hanging onto the ballot card or pregnant chad where all the four corners of a chad are still attached onto the punch card ballot, even after the chad was punched through. Overvotes occur when voters select two or more candidates from the same category. Residual votes can affect the result of an election. Help America Vote Act (HAVA) of 2002 was passed in United States of America, which required updating and upgrading of voting equipment. Each system must meet the following characteristics (Norden & Iyer 2011).

1. Notify the voter (when he/she) has selected more than one candidate for a single office on the ballot.
2. Notify the voter before the ballot is cast and counted if multiple votes have been selected for an office.
(3) Provide the voter with the opportunity to correct the ballot before the ballot is cast and counted.

![Votomatic punch card machine](image1)

**Figure 2-2**: Votomatic punch card machine (National Center for Case Study in Science)

![DataVote voting machine](image2)

**Figure 2-3**: DataVote voting machine (Smithsonian National Museum of American History)

(b) **Ballot Marking Device**

A ballot-marking device (BMD) serves as aid for marking a non-electronic ballot. It helps disabled people to vote by providing tools, such as touch screen monitor, Braille keypad, sip and puff devices and foot-pedals. Sip and puff is a binary device designed to help voters with upper body paralysis. It is usually a wand of straw, which allows the voter to inhale (sip) or exhale (puff) to navigate around the ballot, make a selection, and cast the ballot (Faherty 2006). Voters with lower body paralysis can use the food-pedal also known as rocker paddle. These devices are shown in Figure 2-4.
BMD does not tabulate ballots. It marks the ballot, for example pre-inserted empty ballot, the machine will help the voter to fill the designated area (e.g. in oval shape) and print each candidate name on the ballot. Voters can then check if all the ovals are filled once the ballot are ejected by the machine and verify their ballot manually from the printed ballot or audio output for blind voters. If there are any mistakes, the ballot would be voided and voter is to be given another chance to vote again. Else, the printed ballot can be submitted in to the ballot box or scanned with the assistance of the disabled voter’s companion or the polling officers, to be tabulated in the tallying process. In addition, BMD can also print voters’ voting summary.

a. Braille Keypad (Board of Election in the City of New York 2010)

b. Sip and Puff Device (Origins Instruments Corporation)
(c) Optical Scan

This system uses an optical scanner to read marked paper ballots to collect and count ballots. Optical Scan (OS) allows manual vote casting and tabulation if technical problems occur. As optical scan is a document-based ballot, voting system there is a physical record of votes. This can be beneficial for recounting as well. However, if the election officials are not trustworthy, the votes would be vulnerable to manipulation which can lead to electoral fraud, especially if the ballot obtained by the officer is abstain.

The system is based on a physical paper ballot on which votes are indicated by appropriate marks. An optical scanner then reads these marks when the ballot is fed into it, and votes are tabulated electronically. These marks are usually made by the voter’s hand (for example by using a pencil to fill in an oval or by drawing a connecting arrow for each contest). Because marks are made by hand, optical scan systems can be used for absentee ballots as well. Only the optical scanner can read a narrow range of ballot marks. Therefore, the system has reading errors like the punch card system. For both punch card and optical scan systems, it is possible for voters to cast invalid ballots (for example if more than one candidate is chosen for a one person race), In-precinct counting at the point of voting can warn the voter that an invalid ballot has been cast so that he or she may try again. Warnings of undervotes can also be provided. When centralized tabulation is used, the opportunity for real time error correction is lost in punch card systems. However, for optically scanned ballots, the jurisdiction can organize a committee to infer voter intent on improperly marked ballots (if permitted by state law).

2.1.2.2 Non Document-based Electronic Voting Systems

To maximize the efficiency of the election process and to increase the convenience of voting without sacrificing its integrity, researchers proposed the non-document-based electronic voting system. By using such a system, voters can easily vote electronically in private with the ability
to verify their vote at the end of the poll. This increases voters’ confidence in the system. With advances of Information Technology, non-document-based electronic voting system was introduced as an alternative. Such a system can be of two types, Direct Recording Electronic (DRE) Machine and Remote Electronic Voting System

(a) DRE Machine

DRE is an electronic implementation of mechanical lever system. It was first introduced in 1970. In a DRE, device ballots are shown on the device itself. Voters choose candidates from it. Votes are recorded directly in the machine’s non-rewritable memory once votes are cast through the touch-screen monitor in a polling booth. Once vote has been cast, voters can also print a receipt as a copy. The machine is also equipped with various features, such as multiple ballots and languages, a variety of input and display modes including those for disabled voters. Thus, DRE can provide good usability as well as accessibility, auditability and a good degree of security.

The backbone of DRE machines lies on the computer program behind the entire user interface and the device. Poor system design of both software and hardware can lead to malicious attacks on both voting results and audit records (Aviv et al. 2008). DRE must ensure the prevention of vote tampering and interference during the progress of the election as well as power failure. Besides, it must report and record any malfunctions that occur in the system (Ansari et al. 2008). One of its major disadvantages is its inability to give full confidence to voters that their votes are correctly recorded and counted. In order to offer a solution to this problem, some DRE manufacturers make use of VVPAT technology, a ballot printer which is implemented together with the DRE machine to provide verifiable feature. In its enhanced version, Mercury method is applied where vote receipts are not issued, but they are printed and shown under a glass cover (Mercury & Neumann 2002). This is done to ensure receipt-freeness for the voting system. Another improvement of DRE is Public Network DRE, which is similar to DRE. The difference lies in the feature where vote data are transmitted over public network from the polling station to the centralized ballot box. There are several ways of transmitting the data, either as when voters cast their votes, or in bulk once the polling is complete.

(b) Remote Electronic Voting System

Due to different backgrounds of voters, the variation in needs and demands of the voters are high. In order to fulfill such demands, preliminary voting systems are being improved and enhanced. One of the significant enhancements is the remote electronic voting system, also referred to as internet (online) voting. Internet technology can give a better accessibility to voters, without compromising the security requirements of a voting system. It increases
participation in local and state government elections. Remote electronic voting also provides instant analysis of votes, thus increasing election efficiency. Besides, remote electronic voting offers better ballot-error handling and allow additional information or help to the voters. Although remote electronic voting might not meet all of the voters’ needs, it is expected to be used for elections in future.

In an election, a few standard procedures must be carried out. They are, voter registration and authentication, vote casting (ballot marking), ballot transmission, and tally process. Some elections provide ballot verification in order to gain voters’ trust. Remote electronic voting systems are equipped with standard procedures to ensure none of the security benchmarks is compromised. Voters are to complete the registration process by providing their details to the polling officers. These details would then be used by the polling officers to authenticate the registered voters. In this phase, officials may have difficulties in preventing vote coercion or fraud. Therefore, every remote electronic voting system must assume that their voters are honest, and would not sell their vote to any parties. On the other hand, vote casting includes few other voting processed such as, candidate selections, ballot marking, ballot transportation, submission and ballot counting process. Remote electronic voting is separated into few categories based on its voting setting. They are namely, poll site online voting system, kiosk internet voting system and remote online voting system.

In poll site internet voting system, the election is conducted under polling officers’ supervision in allocated polling sites. Their ballot would then be transferred over the internet to the centralized tally center where all ballots are collected and tabulated. An improved version of this voting system is the regional poll site system, which allows voters to vote from any poll site in their city or region. To make sure that each voter votes only once, computers in every poll site must be connected together through the internet. There are no significant differences in other processes of both initial poll site internet voting system and regional poll site system.

On the other hand, kiosk internet voting system somewhat limits voters’ accessibility by only allowing the voters to cast their vote in specific kiosks in particular locations. However, these kiosks or booths maybe positioned in convenient places such as post offices and shopping malls. In this system, voters can easily vote in a given period under no supervision from the polling officers. Hence, more security levels should be added into this system by implementing number of security features to ensure its reliability.

In the contrary to these three systems, remote online voting system allows voter to vote anywhere as they desire with only one limitation, voters must have a computer that is connected to the internet. With the evolution of technology, the users can to vote by using other devices.
such as mobile phones and tablets. Provided with such features, remote online voting system is still an unfeasible option for higher participation rate. Voting in general include all kinds of people in different society level. Most of the voters in rural areas would not have the devices required as well as internet connection. Even in the suburbs or city, it might be possible that the voters are not computer-literate. Thus, up to this date, elections are still carried out conventionally with polling station system. Other concerns on remote online voting vary from technical problems and malicious attacks on the internet to voter authentication and ballot fairness. Most of the threats rise due to the absence of polling officers when the election takes place. Further explanations on these threats are described in section 2.3, Voting Systems Threats.

2.2 Information System Security for Remote Electronic Voting System

Voting procedures in an electoral election relies on various information security building blocks. In this research project, two branches of information security support the procedure. They are cryptography and steganography. Both of the schemes are combined together in this research to ensure the design of a secure electronic voting system by providing a double layer of data protection. Steganography has rarely been used as an additional layer of security in electronic voting systems and it can offer better protection against threats and risks mitigation by maintaining secret communication between two parties (client-side and server-side). The combination of these two schemes are expected to produce an improved technique which could meet the voter’s demand as well as perform with less computational cost in a secure manner that is feasible to be implemented in remote electronic voting systems.

Reviews and findings of each of the information security methods applied in this research project are discussed as follows. It is separated into two parts based on its information security category, cryptography and steganography.

2.2.1 Cryptography

In remote electronic voting systems, cryptography is used due to its defense capability against threats. It is used to protect the data transmitted between the voter and the server to ensure that it would not be leaked to a third party. In this proposed system, it has been narrowed down to only a few selected schemes. Each of them would be applied in different voting stages to preserve the main characteristics of an electronic voting system. Those schemes are password hashed-based scheme, visual cryptography and threshold decryption cryptosystem. To conclude this section, secret-ballot receipts that is mainly a combination of cut and choose scheme is also explained briefly.
2.2.1.1 Password Hashed-based Scheme
Hash function is applied to protect voter’s password in the authentication process. There are various types of password’s authentication algorithms. Each has its own strengths and weaknesses in terms of providing reliable defense and strong system performance. Hash-scheme does not require extensive computation, yet it is proven to be cryptographically secure (Wagner & Goldberg 2000, p. 3).

Initially, Roger Needham and Mike Guy in 1963 Titan System (Wilkes 1968) introduced password hashed-based scheme. In this protocol, the host just needs to identify the validity of the password given by the voter. Instead of storing the password, the host only stores the result of the one-way hash function of the password. The host will then compare the result of the one-way function (password) to the value stored in the database. By applying this scheme the risk of having an intruder breaking into the host and stealing stored secret data (password) in the database can be addressed. One-way functions are easy to compute for given passwords, but the reverse process of obtaining the passwords from one-way function result is significantly harder. Hence, this protocol provides secure authentication. To enhance this protocol and make it even more difficult to be compromised (e.g. dictionary and brute force attack on stored pre-computed password) is by using “salts” in crypt() function was proposed by Morris and Thompson (Morris & Thompson 1979). Salt is a random string that is concatenated with the user’s password before computing one-way function. Thus, the salt value and the value of the computed (hashed) password given by the user are stored data in the database. However, even with the enhancement of using salt value in this approach intruder still can steal the user’s password if they have access to the system’s database. crypt() function does not entirely eliminate the threats from dictionary and brute force attacks. It only delays the attacks by eliminating the possibility of password stealing at the same time. In order for hackers to attack voter’s password a long time and powerful computers are required. Based on that, in 1999 Provos and Mazieres introduced a new approach called bcrypt() function, specifically designed for password protection. This function is based on Blowfish encryption algorithm, uses a 128-bit salt value and encrypts a 192-bit magic value. However, due to the usage of salts value in crypt() function, bcrypt() function is vulnerable to hackers attack. As a result, standardized approach of password authentication algorithm, Password-Based Key Derivation Function (PBKDF2) is introduced to protect passwords.

2.2.1.2 Threshold Decryption Cryptosystem
Shamir developed threshold decryption cryptosystem in 1979. A \((k, n)\) threshold scheme secures and provides reliable key management for cryptographic system. By having robust security and protection over the key management itself, the security of a cryptographic system
itself could be ensured. Shamir stated two goals of threshold scheme in his paper. Both of them are used to conceal data \( D \) by dividing \( D \) into \( n \) parts, where (Shamir 1979, p. 1):

(a) Knowledge of any \( k \) or more \( D_i \) pieces makes \( D \) easily computable.
(b) Knowledge of any \( k-1 \) or fewer \( D_i \) pieces leaves \( D \) completely undetermined (in the sense that all its possible values are equally likely).

Threshold decryption cryptosystem is usually implemented in conjunction with other cryptography methods. This is because even though threshold decryption cryptosystem offers robust and secure key management of cryptography system, its keys needs to be protected by another information security scheme. Threshold scheme would be implemented in the ballot decryption process to ensure that only authorized personnel can have access to the vote tallying process. In order to perform this decryption process, the private key, which has been divided and distributed to a few appointed personnel, must be merged before each of the election officials as well as the election administrator can gain access to the ballot summary list.

2.2.1.3 Visual Cryptography

Visual cryptography algorithm offers less computational cost compared to the other cryptography schemes, which use complex cryptographic algorithms used to protect a secret. According to Naor and Shamir (1994) who proposed this technology, visual cryptography is a method for protecting image-based secrets that does not involved any cryptographic computation in decoding its shares. It encrypts visual information, for example pictures, text, etc. in a particular way and produces a set of shares as the result. Visual Cryptography allows Human Visual System (HVS) to conduct direct decoding over its concealed shares. The shares consist of a printed page of ciphertext and a secret key in the form of printed transparency. In the enhanced version of Visual Cryptography scheme, they need to be stacked (combined) altogether using a visual cryptography tool to reveal the hidden secret (Chandramathi et al. 2010). Initially it was designed in such a way that the secret message can be revealed without utilizing a particular cryptography tool. Secret message could still be uncovered even though each of these shares is just plainly random noise to HVS by simply placing the transparency over the printed page of ciphertext. It can be considered as a convenient and reliable tool for secret protection or even for verification process because it is not time-consuming, low in computational cost and can be done without any external devices. Illustration on how visual cryptography works in the remote electronic voting system proposed in this research project is displayed in Figure 2-5. It is used to secure the voters’ ballots as well as to provide vote verification process to the voter. The implementation of Visual Cryptography in the proposed system of this research project is described in detail in Chapter 5, System Implementation.
2.2.1.4 Secret Ballot Receipt

The principle of secret-ballot receipt lays in the concept where privacy violation should not occur at all in the election. It ensures election integrity by preventing altered votes due to vote buying or selling and vote coercion (Vora 2005; Paul et al. 2003; Chaum 2004). This technique provides a direct assurance of each voter’s vote by offering vote verification feature. The implementation of secret–ballot receipts scheme helps to reduce the need for physical security, audit, and observation in several stages of the system, as it does not require any external hardware to complement its functionality. The initial flow of secret-ballot receipts implementation requires each voter to use an external hardware (printer) in order to retrieve and verify his or her casted ballots in a printed receipt. This scheme is adopted from Visual Cryptography scheme proposed by Naor and Shamir. In visual cryptography, a secret – in this case, a vote is hidden in two-separated layer of pixel symbols. The secret would only be revealed once those layers are overlaying one another. However, to minimize external hardware usage in the election process, instead of generating a printed receipt, the system would then generate a digital receipt in an image format.

Chaum also proposed the conjunction of mix-net scheme in the secret-ballot receipt to ensure the integrity and privacy of the tallying process (Chaum 2004). Here, the voter’s chosen layer would be passed among few trustees who would generate intermediate batches based on voter’s receipt batch as an input. The final product of this process is a tally batch in the form of ballot image.
2.2.2 Steganography

Steganography is the science of hiding information in communications, where apart from the sender and receiver, others would not know the existence of the hidden information (Provos & Honeyman 2003). There are few steganography techniques for hiding the data, such as in the use of codebooks, layout of texts, every \( n^{th} \) character, the use of whitespaces and newlines, the least significant bit of an image etc. (Johnson & Jajodia 1998). This scheme could be applied to various types of digital file formats such as text, images, audio, video, and protocol file format. They are classified as its cover-file. The best cover-file must have a high degree of redundancy. Redundancy in steganography can be defined as the bits of an object that provide accuracy far greater than necessary for the object’s use and display (Morkel, Eloff & Olivier 2005). Such bits can be altered for secret data encoding without a clear trace of the alteration. The more redundancy a cover-file has the larger size of secret data it can hide.

As the information technology evolves, more threats arise and a simple encryption method is not sufficient to protect the secrecy of data anymore. An encrypted data could easily cause suspicion since it is clearly shown as one. Unlike cryptography, the output data of steganography (stego-object) would still look the same as its input data. As a result, it would be difficult to identify and interpret the hidden secret in the stego-object. Steganography is better than cryptography in its ability to offer less suspicious way of hiding a secret. Besides that, it also offers larger data insertion capability. Therefore, steganography is proposed in this thesis to secure the data communication in the election procedure.

Two of the most well-known types of steganography are reviewed and evaluated to determine the most suitable steganography scheme to be applied in this remote electronic voting system development. Text steganography and image steganography are explained as follows.

2.2.2.1 Text Steganography

Text Steganography is the most difficult kind of Steganography because there is very little redundant information in a text file as compared to a picture or a sound file (Brassil et al. 1995). The structure of text documents is identical to what we observe, while in other types of documents such as in pictures, it is different. Therefore, in such documents, we can hide information by introducing changes in the structure of the document without making a noticeable change in the output. Using text is preferred over other media because text occupies lesser space, communicate more information and cost less to print. The use of text and hidden communication dates back to antiquity. For example, some Iranian classic poets have used this method. Today, the computer systems have facilitated information hiding in text. The applications of information hiding in text have expanded from hiding information in electronic texts and documents to hiding information in web pages. Most text Steganography methods are
for English texts and there are a few text Steganography methods for other languages. Some works have been done on hiding information in texts, such as Word Shifting, Syntactic Methods, Semantic Methods, Open Spaces, etc. In the next section, some of these methods are reviewed. Text steganography is limited to data encoding capability. Thus, they are not completely applicable in the system.

### 2.2.2.2 Image Steganography

Image is a collection of pixels displayed row by row horizontally in a grid. Each pixel consists of a color and it is often represented as bits. Most monochrome and grayscale images have 8-bit depth and RGB images are represented as 24-bit image. Image steganography provides an enhanced security technique of data encoding through digital image file as the media (cover-file) due to the insensitivity of the HVS. Figure 2-6 illustrates the encoding and decoding secret message mechanism of image steganography.

**Figure 2-6: Image steganography mechanism**

Figure 2-7 and 2-8 show the comparison of an original and stego-image with unnoticeable difference. Figure 2-7 illustrated the original cover image before encoding process is carried out, whereas Figure 2-8 shows a stego-image that have been encoded with random encrypted words as its secret message. Image steganography can be separated into two types based in its compression method, image (spatial) domain and transform (frequency) domain (Morkel, Eloff & Olivier 2005). Its difference lies on each of their message encoding mechanism. For image domain, a message would directly be encoded into a source image. Therefore, it is the simplest between the two types of image steganography. However, this type of image steganography is not suggested for the remote electronic voting system, as the stego-image would need to be sent over the internet. This might destruct the stego-image because image domain techniques are not robust against lossy compression, image filters and any other image processing practice. On the other hand, in transform domain, message would be embedded into an image in between the compression phases with both lossy and lossless compression. In general, transform domain techniques is more suitable for remote electronic voting system development compared to
image domain technique because it eliminates the possibility of message being lost during the compression process when the excess image data is removed (lossy compression).

One of the common approaches in embedding information in image domain method is Least Significant Bit (LSB). In transform domain technique, DCT Coefficient steganography is commonly used in hiding a secret besides Palette-based, Patchwork and Spread Spectrum. It is widely used in the Internet due to the small size of the image after the compression. All of these schemes are described in detail as follows.

![Figure 2-7: Original image](image1)

![Figure 2-8: Stego-image](image2)

(a) **Least Significant Bit (LSB)**

LSB can be used together with image file formats such as BMP and GIF that uses lossless compression method. LSB scheme is implemented with characteristics similar to text steganography. In text steganography, a message can be hidden in every n\textsuperscript{th} character of a passage. Likewise in LSB the secret information is embedded in the least significant bit (8\textsuperscript{th} bit) or in all of the pixels of an image (Provos & Honeyman 2003). LSB technique does not remove any information of the original image; instead, it inserts the secret data in the original image’s least significant bits of each pixel. In 24-bits PNG image, there are three different color components, red, green and blue. When a steganography method is applied to an RGB image, each of these three components is used because each of them is represented by bytes. For example, in 200 x 200-pixel image, a total number of 120000 bits or 15000 bytes of secret data can be embedded. The following example shows how a letter S can be encoded in eight bytes of three pixels in a 24-bit image.
In this example, three least significant bits were modified in order to hide the secret data. The human eye cannot detect these changes in the bits because our eyes are not sensitive enough to notice the differences in the LSB of two images.

(b) JPEG

In order to understand the mechanism of JPEG steganography, its compression needs to be explained first. JPEG compression is a commonly used method for reducing the file size of an image, without reducing aesthetic qualities to become noticeable by the naked eye (Bateman 2008). In order to compress an image using JPEG compression method, initially its pixel values in RGB color model need to be converted into YCbCR, also known as YUV color model. RGB color model is defined by Red, Green and Blue chromaticity (colors). YUV color model, on the other hand, determines a color space of Y component that corresponds to luminance (brightness), and the U and V elements, which relate to chromaticity of the image (Morkel, Eloff & Olivier 2005). Its coefficients are defined by a 2D grid of blue to yellow and red to green chromaticity on each of its axis. Once the conversion is completed, the chrominance values of the image would then need to be down sampled. Down sampling also known as subsampling, is the process of reducing the number of pixels in a digital file (Kong & Zhu 2008). Due to the HVS’s sensitivity over brightness, the quality of an image could still be maintained in the process of removing its color information. Down sampling is used in the mechanism of JPEG compression to compress the image file size. This process is followed by transforming chrominance values into frequencies. Here DCT method is applied. DCT is a mathematical transforms of cosine function calculation (typically) that converts the chrominance data (image pixels) by seemingly ‘spreading’ the location of the pixel values over part of the image (Bateman 2008). It converts the pixel values into frequency by grouping them in 8x8 pixel blocks and transforming each of the pixel value into 64 high and low frequencies (DCT coefficients). Any alteration to a single frequency value affects the whole block of image pixels. Next, these DCT coefficients values are required to be quantized. DCT quantization is the process of taking the remaining coefficients and dividing them individually in a block against the pre-determined set of values and then rounding the results to the nearest real number.
value (Bateman 2008). Figure 2-9 shows the example of the three blocks generated up to this point of JPEG conversion. The aim of quantization is to remove the high frequency values that are not perceptible to the eye to reduce the image size. The next phase of JPEG compression mechanism is zigzag ordering. It is done to group the similar value of quantized DCT coefficients together. The last step of this compression is lossless compression by encoding the DCT coefficients with Huffman algorithm to compress the image size.

Figure 2-9: Quantization process of JPEG compression

JPEG steganography allows secret data to be embedded in inconspicuous parts of the image by applying JPEG compression algorithm. From the explanation above, it can be concluded that in this compression algorithm both lossless compression and lossy compression are involved. Image compression is carried out through the DCT conversion and the quantization process. DCT is categorized as a lossy image compression due to the rounding errors (high frequency) in the chrominance data conversion into DCT coefficients. These rounding errors are caused by the repeated calculations done to determine the cosine values using limited precision numbers. Because of the insensitivity of the HVS, these errors are not noticeable to the human eye. Thus, they can be used to hide the secret message. JPEG steganography methods have adopted this technique for data encoding process, e.g. JSteg algorithm as well as Outguess 0.1 and 0.2 algorithm that replace the LSBs of the DCT coefficients with the message bits. There are two types of DCT coefficients in each of the pixel blocks - DC and AC coefficient. Any modification made to the DC coefficients value will affect the AC coefficient values. Therefore, these three steganography schemes only allow data embedding over AC coefficient values and they must not be equal to 0 and 1. Besides these schemes, Andreas Westfeld (2001) initially proposed F3 algorithm as alternative. It was then followed by its enhancement, F4 and F5 algorithm in order to eliminate its flaws. In this section, the review is focused on F5 steganography algorithm that tackles all the major threats and attacks faced by the other five algorithms. Its mechanism is described as follows.
(i) Encoding Process

Message encoding in F5 resembles the process done for F4 algorithm that was implemented to eliminate the issues of F3 image steganography algorithm. In its data encoding mechanism, F3 allows the secret message to be embedded within the AC coefficient values that is equal to 1. Besides that, F3 does not embed the message by overwriting the LSBs of the DCT coefficients directly. It implements a shrinkage mechanism, after which the message would be embedded. It first compares the message bits to the absolute value of non-zero AC coefficients. If both values do not match, F3 algorithm decrements the non-zero coefficient value by 1, else both values remain unchanged. However, this process raised up an issue that leads to less data embedding capacity for the cover image. This is caused by the existence of even-positive coefficients from both skipped AC coefficients of 0 and the reduced value generated by applying the algorithm, which leads to more embedding process of zeros than ones. Thus, this algorithm is susceptible to histogram attack (Bateman 2008). F4 algorithm eliminates this issue by mapping negative coefficients to the inverted steganographic value. Here, the even-negative coefficient and odd-positive coefficient are modified to steganographic 1, whereas the even-positive and odd-negative coefficients remains as 0. The overall mechanism of F5 is very similar to F4 algorithm. F5 is the extended version of F4 that implements two additional features to enhance its algorithm, namely Matrix encoding and Permutative Straddling. In F5 encoding process, once half of the JPEG compression is completed, Permutative straddling will be applied to spread out the embedded data evenly throughout the cover image. This technique shuffles (permutes) all of the quantized DCT coefficients by using Pseudo Random Number Generator (PRNG) key. In the next step the secret message is embedded with matrix encoding into this permuted sequence of coefficients. Matrix encoding uses binary Hamming codes to encode more bits per encoding change. By eliminating coupling of any fixed pair of DCT coefficients, its data embedding efficiency could be improved (Provos & Honeyman 2003). This also means that F5 is resistant to one of the statistical attacks, Chi-square ($\chi^2$) test. Lastly, the JPEG compression is continued by applying Huffman encoding algorithm that resulted in a stego-image file. The overall encoding process of F5 steganography algorithm is shown in Figure 2-10.

(ii) Decoding Process

Decoding process of F5 algorithm is somewhat similar to the message decoding process of F4 steganography algorithm. The stego-image file needs to be converted back with Huffman decoded to retrieve the permuted DCT coefficients. It is simply the reverse of Huffman encoding algorithm process. In order to decode the message, the permuted coefficients are
required to be permuted again to obtain its original sequence. Once this is completed, secret message can be retrieved (decoded) from the DCT coefficient blocks.

Figure 2-10: Message encoding process of F5 Steganography algorithm (Westfeld 2001)

(c) Palette-based Image

A palette-based image, such as GIF image is a color-indexed image where its colors are stored in a pre-determined color palette list (Morkel, Eloff & Olivier 2005). The color palette contains list of 256 color pixels of a Palette-based image, whereas the color indexes are pointers to the color in the color palette list (Tzeng et al. 2004). In palette-based image steganography method, secret message can be embedded directly in the image colors or even in the bits of color palette, simply because it does not correspond to the color pixels of a Palette-based image. This technique permutes the colors in an image palette in a specific order to encode the secret message with LSB encoding approach. By using this method, HVS attack over its stego-image can be avoided. For secret data encoding in the bits of color palette, the selection of cover image is crucial. For the selected cover image with very dissimilar neighboring palette pixels, any modification to its color palette could result in completely different color pixels. This problem can be avoided by sorting the color palette, adding visually similar colors to the color palette with dissimilar neighboring colors, generating sub-palettes based on the similarity of colors in palette-based image, or by using grayscale cover image because it has bigger range of similar color pixels (Morkel, Eloff & Olivier 2005). The similarity in the neighboring pixels of a Palette-based image is shown in Figure 2-11.

Figure 2-11: A part of color indices in a Palette-based image (marked in square) that shows color similarity in the neighboring pixels
(d) **Patchwork**

Patchwork is a statistical steganography technique that implements redundant pattern encoding to embed a secret message in an image. This algorithm adds redundancy to the hidden information and then scatters it throughout the two marked areas (patches) over the cover image (Morkel, Eloff & Olivier 2005). These marked areas are selected with PRNG. One area is lightened while the other is darkened to manage the average luminosity of the image during the message encoding process. Thus, any alteration to the marked patch is imperceptible (Morkel, Eloff & Olivier 2005). Due to the redundant pattern encoding characteristic, its stego-image would be more resistant to image processing, such as image rotation and cropping. This however, depends on the size of the secret message. The message can be embedded repeatedly throughout the cover image only if it is small enough. Messages with larger size might not be embedded properly in multiple patches.

(e) **Spread Spectrum**

Similar to patchwork, spread spectrum method hides data by spreading it throughout the cover image as well (Johnson & Jajodia 1998). Both methods are suitable for application in encoding a small amount of secretive information. In other words, it can be defined as the process of the spreading the bandwidth of a narrowband signal across a wide band of frequencies (Johnson & Jajodia 1998). In its message encoding process, the secret message is stored in the noise of the original image, resulting in the construction of a Spread Spectrum stego-image. As an image is not likely to have noise, the embedded message in the stego-image will not be noticeable by the human eye or by computer analysis without access to the original image (Morkel, Eloff & Olivier 2005). The encoding process of spread spectrum is shown in Figure 2-12 and the process is shown below (Marvel, Boneclet & Retter, 1999):

1. Create encoded message by adding redundancy via error-correcting code.
2. Add padding to make the encoded message the same size as the image.
3. Interleave the encoded message.
4. Generate a pseudorandom noise sequence, \( s \).
5. Use encoded message \( m \), to modulate the sequence, generating noise, \( n \).
6. Combine the noise with the original image, \( i \)
Spread spectrum communication relates to time and frequency. Similarly, in image processing, spread spectrum relates to space and data rates. The steps of spread spectrum decoding process are as below (Marvel, Boncelet & Retter, 1999) and illustrated in Figure 2-13:

1. Filter the stego-image, to get an approximation of the original image, \( i' \).
2. Subtract the approximation of the original image from the stego-image to get an estimate of the noise \( n' \), added by the embedder.
3. Generate the same pseudorandom noise sequence, \( s' \).
4. Demodulate by comparing the extracted noise with the regenerated noise.
5. Deinterleave the estimate of the encoded message \( m' \), and remove the padding. Use error-correcting decoder to repair the message as needed.

There are many other potential threats to the integrity of an election. Some are not related to voting system’s technology, but are caused by the approaches or technologies used in the voting system itself, including platforms and devices. Threats may be fraud in voters’ authentication and registration, vote coercion, phantom votes, poor ballot design and its transportation and storage, poor usability design and maintenance of required devices, improperly constructed election procedures and other technical problems. Voting system threats are discussed under
two categories of election procedures - traditional polling station election and electronic voting system. In this project, the potential threats focus is on those during the voting and tabulation process.

2.3.1 Threats in Traditional Polling Station Election

In a traditional polling station election, all of the election processes are done manually by polling officers. There are settings, ballot preparation, votes tallying and troubleshooting devices. Therefore, trustworthy officers are most important in a traditional polling station. However, one cannot fully ensure that the polling officers will not cheat or cannot be bribed. Other issues such as, complicated ballot design or voting procedures, multiple voting by a voter, possible damaged ballots during ballot preparation or transportation etc., may also impair traditional voting system. To reduce these problems, mechanical voting machine was proposed.

Mechanical voting machine was not a completely a solution for the problems. Voters are expected to know the way the machines function. Otherwise, ballot damage and phantom vote can occur. Others issues also arise, for example faulty machines, human error during vote collection process, unreliable vote transport and vote coercion. Moreover, there are two well-known mechanical lever machines, each with different user interfaces and ballot design. If both types of machines are used, voters may be confused. Power failure was one of the problems faced by mechanical voting machines before the implementation of Shoup Machine, which can be used even without electricity. Due to its strong dependency on the presence of the polling officers, researchers in this field are introducing electronic voting systems and remote electronic voting systems.

2.3.2 Threats in Electronic Voting Systems

There are five common stages in an electronic voting system. Threats exist at each stage of the system. They are described as follows.

(a) Registration and Authentication Stage

In this stage, two kinds of issues might arise. The first kind arises from errors made by the polling officers or system administrator as the voters register and authenticate their identity. These errors can be avoided if machine is used for authentication, for example by reading voter’s identity card through card scanners or by using biometric means. The authentication stage might have another problem if the machine malfunctions and voters’ identity cannot be authenticated.
(b) **Ballot Initialization**

Electronic voting machine and remote electronic voting systems have two distinct ballot initialization processes. Ballot initialization and preparation in a remote electronic voting system rely on the voting machine itself. These processes are much simpler when using electronic ballot instead of paper ballot. However, voters cannot cast electronic vote in the event of power failure. This is the main reason why a number of electronic voting machines still use paper ballot, even though paper ballots are vulnerable to weather and improper ballot storage. Hence, the possibility of damage paper ballots is higher compared to electronic ballot.

(c) **Voting Stage**

These threats are most likely to occur when a ballot is invalid either because of lack of understanding of the voting machine or because of technical problems such as malfunctioning of the voting machine and power failure. These threats are also present in paperless (electronic) ballots where malicious attacks on computer can be mounted to tamper on block (Denial of Service attack) votes.

(d) **Vote Verification Stage**

The E2E Verifiable Voting procedure was implemented for verification of votes by issuing vote receipts. This procedure cannot eliminate vote tampering and coercion; it can only reduce them. The procedure of issuing vote receipts can affect ballot secrecy. The supporter of a candidate can find out if a voter has voted against the candidate.

(e) **Tallying Stage**

When the ballots are transmitted (transported) from the polling station to the central repository, a few risks can occur. Paper ballots and electronic voting machines can be removed or tampered. Computer-based voting systems are also subject to malicious attacks unless they are secured with information security schemes.

2.4 **Summary and Conclusion**

This chapter reviewed the overall information and evolution of voting systems from the related literature. As this research focus is on the development of a secure remote electronic voting system, E2E voting system, to be specific, most of the discussion in this chapter emphasizes on this type of voting system. Two distinct types of voting systems were discussed, namely traditional polling station voting system and electronic voting system. It covered detailed explanation of machines and developed systems for each voting systems. Furthermore, the potential threats found in the implementation of both systems were discussed, as well as different information security schemes that address them.
From the review and findings gathered on this chapter, the feasibility of developing a remote E2E voting system through the integration of cryptography and image steganography techniques is concluded. This approach is believed to be adequate in fulfilling the main objectives of electronic voting system identified in the previous chapter.
CHAPTER 3

STUDIES OF RELATED E-VOTING SYSTEMS

3.0 Introduction

E-voting systems have the ability to complete voting process faster than the paper ballot voting procedure. However, this matter alone cannot satisfy the users’ demand, especially on the area of security. E-voting systems must be able to earn users’ trust and confidence by providing enhanced security feature(s) without affecting its efficiency and flexibility. The system should offer some level of transparency to the user without allowing any breach of trust and privacy.

In order to fulfill this condition, E-voting systems must provide both individual and universal verifiability. Such systems are categorized under E2E Voting System (Adida 2008). In principle, E2E voting system offers assurance to the voters over their cast vote. This is done by distributing vote receipt of encoded cast vote to each of the voters for verification purpose. By using it, they can check if their vote was correctly cast and included in the final tally. However, the vote receipt cannot be used as a proof in vote buying or vote coercion because it is encoded. To support this verification process, E2E systems implemented Bulletin Board, a secure append-only broadcast media where each of the encoded votes would be posted once the voters completed the voting process. In order to verify their cast votes, they need to match the encoded value on their receipt against the values shown on the bulletin board. As a result, E2E voting system would protect the voter’s privacy and supports Incoercibility that preserves the integrity of the election result. Its mechanism is illustrated in Figure 3-1.

E2E voting systems vary based on their security and flexibility levels. In this chapter, four different types of E2E voting systems that have been applied in medium to large scale-real-world elections will be discussed in detail in order to give a better understanding of E2E voting systems’ mechanism. They are Scantegrity II for Takoma Park Municipal Election, Rijnland Internet Election System (RIES) for public election in Netherland, the implementation of Helios voting system at Recteur election of Université Catholique de Louvain in Belgium and lastly student council election at Princeton University that make use of Prêt à Voter system (Carback et al. 2010).
Each of these systems has two types of principal components - System Users and System Stages. The four E2E systems are elaborated as follows.

3.1 Helios

Helios Voting System is an electronic voting system created by Ben Adida in 2008. It is an open-source web-based voting system that offers verifiable online elections for anyone. It was designed to ensure a clean election setting where coercion is not its main concern (Adida 2008). Ballot secrecy and election integrity are still applicable through open-audit election. Both of these aspects cannot be achieved in a typical traditional election, where only the election officials are entitled to do the observation throughout the election process. The current version of this system, v.3.1 is accessible at http://heliosvoting.org. The system overview of Helios is described in detail below.

3.1.1 System Overview

Helios is equipped with a number of practical features to enhance its efficiency, namely authentication, integration with multiple web-services such as Facebook, Gmail and Twitter so that potential voters can directly log in with one of the chosen web-services without having to...
register their credentials in a separate process. Helios provides bulk voters registration by using the pre-existing electoral rolls and officers administering election (Cortier & Smyth 2011). The procedure consists of two main stages, ballot preparation and ballot casting. These stages cannot be compromised because of an end-to-end verifiability that consists of correct ballot preparation and recording and a secure tally procedure. To ensure this, Helios implemented a ballot verification feature called Ballot Tracking Center where users can verify if their votes have been received and tallied correctly. Voter authentication is carried out during ballot casting phase. This is to offer individual and universal verifiability. The verifiers include the voters, election officials (observers) and even first time testers. To access its entire functionality, user can register to its website, then set up and run an election or they can also to audit the system. In ballot preparation stage, users are allowed to view the ballot, make their selection from the pre-defined candidate list, secure it by encrypting their selection and verify the encrypted selection. After all these processes are done, the voter would be asked to authenticate for final ballot casting.

This open-audit voting system was inspired by the Simple Verifiable Voting Protocol proposed by Benaloh in 2006 that implemented Sako-Killian mix-net scheme in its approach. In the mix-net scheme, multiple servers of trustees or election officials are introduced. It was designed in this way to increase the integrity of an election by assuming that the trustees will remain truthful. Helios architecture initially provided only one trustee, that is, the Helios server. The latest version of Helios Voting System offers a better approach to protect system’s privacy by appointing multiple trustees. Each of these trustees has to decrypt the final tally of the election by using advanced cryptographic techniques. One of the techniques adapted consists of Threshold Decryption Cryptosystem.

The system needs more features to offer a secure, convenient, accurate and efficient open-audit voting system. Thus, its architecture and user interface will be more complex. Users need a simple user interface that is easier to be understood and utilized. Weber and Hengartner (2009) as well as Karayumak et al. (2011) analyzed and evaluated both initial and current version of Helios Voting System and proposed its improved versions. The weakness of the initial version of Helios Voting System lays on its inefficient and complex features in different stages of the election procedures. Most of those features led to skill gap among the users. The features are designed in such a way that only a majority of technical users would understand its functionality clearly. One of the examples is its ballot encryption and decryption processes. Users have very little or no knowledge over cryptography. Thus, the additional features of cryptography methods applied would make very little difference for them compared to the technical users. One of the examples of these features is the display of vote receipt (ciphertext format) as shown in Figure
3-2. Besides that, the format of its vote receipt is somewhat inconvenient to be kept by the users.

Figure 3-2: Screenshot of vote verification page of Helios Voting System

3.1.2 Principal Components of the System

The principal components of the system used are as follows.

3.1.2.1 System Users

Helios must ensure voter verifiability and universal verifiability. Voter verifiability is fulfilled through the implementation of vote receipt. Universal verifiability on the other hand, is ensured by recruiting election trustees. The other user type in Helios is the voters. Their roles could be extended to an administrator. Thus, Helios Voting System has three types of users - administrator, voters and the election trustees. These users type are to be described as follows.

(a) Administrator

Apart from casting and verifying his vote, Helios allows users to set up and run an online election once they have logged in to the system. Such users are considered as administrator. In Helios, an administrator has four main responsibilities - to set up ballot, manage voters, freeze
the election which means to mark the start of the election and lastly to compute the election tally (Adida 2008). These processes are elaborated in detail in the next section, *System Stages*.

(b) Voters

The roles of voters in Helios voting system are similar to voters’ roles in any other E2E voting system. They are capable of registering and authenticating themselves as well as casting their vote and verify it. A detailed explanation about this is given in section 3.1.2.2, *System Stages*.

(c) Election Trustees

In its latest version, Helios implemented an enhanced feature by adding the number of trustees in a particular election. As a result, Helios server is not the only trustee of the system. The actual voter as well as the voter who acted as an administrator can also be an election trustee. Trustees are responsible for decrypting the election result. Each trustee generates a key pair and submits the public portion to Helios. When it's time to decrypt, each trustee needs to provide his secret key. The implementation of election trustees guarantees privacy over the tally process of Helios system. The process is described in the next section.

3.1.2.2 System Stages

In Helios Voting System, users participate in four distinct stages of the system. These four stages are explained in detail below.

(a) Election Setup

For the voters to cast their vote in the voting stage, first an election environment must be setup. Another voter who holds the administrative role carries out this process. Such voter is known as administrative voter. Election setup is divided into several processes. They include:

   (i) Set up Ballot

   Administrator is permitted to conduct this process by preparing, reviewing and editing the ballot questions. They are also allowed to update the ballot question at any time before the election is freeze in case they have made any mistakes.

   (ii) Manage Voters

   Helios administrator can also manage the eligible voters. He can add (register), update, remove and contact the voters to pass any information of the election through email. In the latest version, Helios is still equipped with bulk voter upload feature where administrator can add an average number of 40 voters at a time by using Comma-Separated Values file format (Adida 2008).
(iii) **Freeze the Election**

Once the administrator has set up the ballot and added eligible voters, he needs to freeze the election. By doing this, voter list, the election dates (start and end) and the ballot details would be absolute and cannot be modified (Adida 2008). This determines the beginning of the election.

(iv) **Compute Tally**

To compute the tally result, Helios administrator must first end the election by initiating the tally process. Once this is completed, all the trustees must submit their pre-generated secret key in order to decrypt the tally and release the result. Helios administrator could be one of the election trustees and Helios system itself is a trustee by default.

(b) **Voting, Verification and Authentication**

Initially, eligible voters would receive an e-mail invitation to vote in a particular election once they have registered by the administrator. By clicking the link provided in the e-mail, voters are directed to Helios webpage (voting booth) where clear and concise information will be displayed to give a better understanding of the system. Next, the voters can start the voting process. As the voters select their candidates, their votes are encrypted locally and the ciphertext are shown to the voters. It is also known as the verification code. Here, there are two options that the voters can choose, to verify their votes or to simply trust Helios and cast their votes directly. Both options will encrypt their vote to generate vote receipt or also known as encrypted ballot together with its fingerprint, called as the smart ballot tracker. If they choose to verify their votes, voters would need to utilize the ballot verifier feature. This feature allows the voter to verify their encrypted ballot by examining the election fingerprint and the smart ballot tracker of the encrypted ballot. In order to utilize this feature, the voters need to submit their vote receipt. This step could be repeated by the voters as many times as they wish. Once the voters are satisfied with their verified vote and its accuracy, they could then proceed to cast their vote by logging in to the system. Their identities would be authenticated and their vote would be cast by the system.

(c) **Tally**

During the initiation of tally computation by the administrator, the election would be ended and voters’ collected encrypted ballots are shuffled automatically. Once the ballots are shuffled, trustees would then need to decrypt all the shuffled ballots by submitting their secret key. Except for Helios system, any other authorized trustees must generate the key pairs themselves and keep their secret key secure prior to this ballots decryption process. As soon as the election ends, Helios server will post the encrypted tally on the bulletin board. This bulletin board is also
referred to as Helios Ballot Tracking Center. The information displayed on the bulletin board includes voters’ name and their respective smart ballot tracker. The administrator needs to gather all the election data, verify the shuffle, decryption as well as tally and then only publish the final result of the tally. Administrator is to notify the voters regarding the result publication in the end of this stage.

(d) Audit

Helios provides two verification features - vote verification and election verification. Vote verification has been described previously in *Voting, Verification and Authentication Stage*. Election verification can be carried out by utilizing Helios Election Verifier feature. Voters could access this feature through the link provided in the tally released email sent by the administrator. This feature takes election URL as the input. For every valid input submitted, all the information of that particular election will be retrieved and shown on the webpage. The information includes verification details of the election, ballots, trustees, tally and final result of the election.

3.2 Scantegrity II

Scantegrity II is a practical enhancement for optical scan voting systems, which achieve increased election integrity through a novel use of confirmation codes printed on ballots in invisible inks (Chaum et al. 2008). This section covers the review of Scantegrity II voting system and the history behind its development.

3.2.1 History of Scantegrity Voting System

Punchscan and Scantegrity were the predecessors of Scantegrity II. All of them are categorized as optical voting system due to the implementation of optical scanner in their systems. Given below is the system overview for each of these systems.

(a) Punchscan

This voting system is a precinct-read optical scan balloting system that allows the user to take their ballot with them after scanning (Popoveniuc & Hosp 2010). This ballot acts as vote receipt that can be used by the voters to verify the accuracy of their votes. However, it does not violate the receipt-freeness requirement of electronic voting system as the receipt would not be able to reveal the actual value of the ballot cast. This is due to its ballot mechanism.

A Punchscan ballot consists of two parts of sheets - top sheet and the bottom sheet. On the top part of both sheets, a random ballot number is written. This number is used by the voter to track his ballot in the vote verification process. The top sheet contains a list of the candidate names
with a letter or symbol on each of the names. It also has holes on it that shows the printed letter on the bottom sheet. Voters could then mark their choices on these designated holes by using bingo dauber (Clark, Essex & Adams 2007). The same letter or symbol is also printed on the bottom sheet. The letter or symbol’s order on both sheets is never the same.

For its tabulation procedure, Punchscan was designed with anonymity network often referred to as the Punchboard. It joins binding commitments table and Randomized Partial Checking or also known as RPC (Adida & Neff 2006) to permute the votes in order to hide the link between voters and their votes once the ballot is cast.

(b) Scantegrity

Scantegrity is an improved version of Punchscan that combines the implementation of two separate sheets into one individual sheet in its ballot mechanism. It provides a strong security protection without disrupting the voters’ convenient. It also ensures the existing requirements of electronic voting system that Punchscan fulfilled. It offers minimal cost for wide-scale deployment by implementing a new ballot design (Chaum et al. 2008). Scantegrity introduced a new element in its ballot mechanism called the ballot chit, which is a perforated corner of the ballot that contains a serial number that can be used to verify voter’s vote (Chaum et al. 2008). Scantegrity has the same cryptography techniques that Punchscan voting system used.

3.2.2 System Overview

The physical ballot of Scantegrity II consists of a voting portion and a receipt portion. Just as the traditional paper ballot voting procedure, the voters are given conventional paper ballot where they need to mark their chosen candidate with a special tool (pen) that uses invisible ink. This technology allows the voters to retain their receipts in a secure and secret manner with the help of unique confirmation codes on each ballot that no attackers would be able to coerce. The confirmation codes on voters’ ballot are kept secret and will only be visible to the voters when they cast their votes. No information regarding the confirmation codes would be accessible to anyone before the votes are cast. Due to this feature, Scantegrity II can earn more trust and confidence of the voters. The mechanism is illustrated in Figure 3-3. After the voters mark their ballots, the confirmation codes are shown on the paper ballot itself and can be used for verification purpose. Every voter who has cast his vote is entitled to verify its accuracy by checking the generated confirmation code against the list on the blackboard. This results in individual verifiability by the voters themselves. Besides, the system also provides universal verifiability for everyone to (re)confirm the computation of the tally, and ensures that votes are not altered or deleted for manipulating the final tally of a particular election. The implementation of vote verification feature however, should not reduce its usability level.
Scantegrity II is firstly designed to accomplish three main usability goals (Chaum et al. 2008); such as there should not be any major modification on its marking and casting processes from the conventional optical scan ballot. Secondly, the system should be able to generate receipt from its ballot without any disruption on its voting procedure. Lastly, its vote verification feature should not be compulsory to the voters.

Through the implementation of invisible ink, Scantegrity II could prevent some of the issues raised by the Punchscan and Scantegrity. One of the attacks is randomization attack. It is an attack that results in the nullification of the voters’ choice, by forcing the voters to cast irregular votes that appear to be the same as the balloting materials randomly (Juels, Catalano & Jakobsson 2005). The implementation invisible of inks also eliminate the possibilities of phantom votes.

3.2.3 Principal Components of the System

Described as follows are the principal components of the Scantegrity II optical voting system.

3.2.3.1 System Users

There are three types of users in Scantegrity II voting systems. This information is derived from the actual election of Takoma Park Municipal Election. Each user holds different roles. Their roles are described in detail as follows.

(a) Scantegrity Team

The Scantegrity team members need to make sure that the election has been prepared properly. In order to do so, they need to set up the election environment in a particular place and prepare the necessary materials, such as the printed ballots. Other than this, they also need to train and educate the eligible voters on the election procedure.

(b) Election Officials

Election officials in this voting system are responsible to assist the voters throughout the election process. They are also required to assist the voter in scanning the marked ballot by
using the optical scanner and marking the detached vote receipt based on the ballot status of the voter; “Ballot Voted”. Election officials also need to create an audit ballot; a void ballot generated for auditing the correctness in receipt printing. They are also in charge of supervising the whole election to maintain its stability level and they have to complete a survey given by the Scantegrity team in order to get users’ feedback about the system.

(c) Voters

Voters' roles in this voting system do not have much difference compare to the roles of the users in other E2E voting systems. Their roles include vote casting, creating a vote receipt, detaching the receipt from the original ballot, casting the ballot and filling the survey form from the Scantegrity team.

3.2.3.2 System Stages

The system stages of the Scantegrity II are as follows.

(a) Registration and Authentication

In this stage once the voters’ credentials have been authenticated, the voters would be given a voter authority card which indicates their legitimacy as eligible voters in that particular election. When the voters present their voter authority cards to the polling officers, they will receive their ballot and verification card. This verification card can be used as a reference (receipt) to verify their vote in the later stage of the election. Therefore, it is important for the voters to keep their verification card safely.

(b) Ballot Marking

The voters are to mark their votes by using the pen provided on the designated polling booths located at the election compound. With the integration of invisible ink mechanism, voters would obtain a confirmation code on the reserved bubble beside the selected candidate(s) when they mark their ballots. There are different confirmation codes for each candidate on each ballot. They are randomly generated. Thus, voter’s secrecy could be ensured.

(c) Generate Receipt

Based on their preference the voters could choose to complete this stage. It is optional for them to generate a receipt for their marked ballot. If the voters would like to verify the accuracy of their votes, then they would then need to transcribe the obtained confirmation codes together with the online verification number to the verification card given to them previously. With this information they are able to verify if their votes have been casted, collected and counted properly.
(d) Ballot Casting

By using the optical scanner device available at the polling compound, the voters can cast their votes. They should insert their marked ballot to the scanner where their votes will be recorded electronically and the physical ballot inserted should be destroyed.

(e) Online Verification

After the voters have casted their ballots into the scanner, they are able to verify their votes online by accessing the stated election website on their verification card. Once accessed, the voters would need to submit their ballot verification number and check if the confirmation codes listed on the website (bulletin board) match with the one in their verification card.

3.3 Prêt à Voter

The development of Prêt a Voter was initially motivated by Chaum’s work of implementing Visual Cryptography approach proposed by Naor and Shamir (1994) in an election procedure (2004) to offer vote verifiability. It implements the same concept as Chaum’s secret-ballot receipt scheme with rather simpler methods. This system was introduced to provide more accurate and faster tallying process to cut unnecessary election cost and to increase voter participation. Prêt à Voter offers assurance from its election auditability.

3.3.1 System Overview

Prêt à Voter’s election auditability feature allows any of the system’s users including the audit teams to evaluate its integrity by checking distinct stages of voter authentication, ballot preparation and vote processing. This could be categorized as universal verifiability. Similar to other E2E voting systems, Prêt à Voter assure the voters that their votes have not been altered. It was collected and counted correctly in the tally by giving each of the voters a unique encrypted receipt. This receipt will not leak out the ballot; it can only be used to check the vote status against the read-only Bulletin Board. With the support of some security components, vote verifiability could be ensured. Those security techniques give a better security protection where internal sources of threats could be anticipated and handled properly. Those security components include encryption schemes such as RSA, ElGamal and Paillier and few other cryptographic methods, for example threshold decryption cryptosystem, zero-knowledge proofs, homomorphic encryption, etc. Thus, the vote would remain secret and the possibilities of election fraud could be averted.
3.3.2 Principal Components of the System

The principal components of the Prêt a Voter are as follows.

3.3.2.1 System Users

There are four distinct types of users in this system. They are the voters, election authority, auditors and the help organizations (Ryan et al. 2009). Each one of them is described more detail as follows.

(a) Voters

The uniqueness of an E2E voting systems is its vote verifiability feature. Prêt à Voter offers this feature as well. This feature enables voters to reconfirm that their votes have been cast. As this feature is optional, it needs to be made as simple as possible in order to encourage voters to make use of it to increase their level of confidence and trust in the election system. Thus, the number of voters can be increased.

(b) Election Authorities

The role of election authorities may vary in different elections based on their requirements and settings. In general, their roles cover the preparation of ballot forms, authenticating voter’s eligibility, and supervision of the election, conducting a secure tallying process and publishing the official tally result. In Prêt à Voter there are two main election authorities, they are mix servers and the tellers [prêt a voter]. Both of them are entitled to decrypt encoded ballots in order to count the final tally.

(c) Auditors

In E2E voting systems, auditors could be IT experts or other appointed individuals who are trustworthy and responsible to provide expert opinion on evidence of proper functioning of the electronic voting system, by checking or auditing the published information on the Bulletin Board (Ryan et al. 2009).
(d) Help Organizations

This type of users acts as a helpdesk to assist voters during the election. Similar to the conventional voting system, help organizations hold an important role in E2E voting systems even though they are not essential. Most of the time, remote E2E voting systems do not require helping organizations as voters can easily cast their votes based on their preference. Assistance from the help organizations is offered via internet or through the help feature on the system. On the other hand, in most paper-ballot based voting systems where voters vote in the polling booth, election authorities’ help is provided for assistance during the election.

3.3.2.2 System Stages

Prêt à Voter introduced another new approach of vote encoding. Users’ votes are encoded in randomized candidate list. There are four stages in its system design, such as: ballot generation, vote capture, vote processing and auditing (Ryan et al. 2009). Each of these stages is explained below.

(a) Ballot Generation

In generating their ballot, Prêt à Voter uses detachable paper ballot of candidate list in random order and the corresponding boxes where the users can indicate their vote(s) (Winckler et al. 2009). This detachable paper ballot will then be destroyed to ensure users’ privacy. The ballot itself consists of candidate names in random order together with boxes for voters to mark their chosen candidates. The other part of the ballot will be kept by the users for verification purpose as their vote receipt. In both parts of the paper ballot there is encrypted information named ‘onion’. It is applied to reconstruct the shuffled candidate list for each ballot in the proper order as the other part of the ballot is scanned (cast) after the vote capture stage. The reconstruction of the randomized candidate list or also known as onion decrypting process and can only is carried out by appointed officials. It cannot be done by one single election official. This is where mix-net decryption scheme is implemented to ensure ballot secrecy and voter privacy.

(b) Vote Capture

By using the receipt, voters are able to verify that their votes are correctly captured, collected and counted in the next stages. In order to be able to make use of this feature, the voters first need to scan their receipt to be saved into the system’s database. This action is performed to ensure the accuracy of the tally. Here there are no cryptographic techniques applied but digital signatures are used as additional layer of protection to the scanned receipts. The voters could confirm their vote status by checking the receipt against the Bulletin Board once all the scanned receipts saved in the database are published.
(c) Vote Processing

In this stage there are three main tasks that need to be performed by Prêt à Voter system. They are mixing the ballots, decrypting them and tallying the result. All of these tasks are carried out to transform the set of encrypted votes into a set of unencrypted votes without allowing anyone including those who are involved in the decryption process to perform end-to-end matching (Ryan et al. 2009). To ensure all the votes have been shuffled correctly and the connection between voters and their votes are detached (voter anonymity), PV implements mix-net decryption scheme. A set of collected ballots are passed around few authorized mix-servers to shuffle and decrypt the ballots (Provos & Honeyman 2003). Once the decryption process is done and the candidate list also the users’ votes have been retrieved, votes can be counted.

(d) Auditing

The system must make sure that all the votes cast have been counted. In order to ensure this, each task such as vote mixing, decrypting and tallying must be audited to verify their correctness. One of the methods used to complete this task, specifically to verify the accuracy of mix-net decryption scheme is Randomized Partial Checking (RPC).

3.4 RIES

RIES is a voting system that implements multiple types of technologies. It was used for the first time in 2004 Water Boards Election at Rijnland and De Dommel (Hubbers, Jacobs & Pieters 2005). Herman Robers initially designed RIES for the completion of his master’s thesis (Robers 1998). It was implemented at a local election in Delft University of Technology. Hoogheemraadschap van Rijnland, a local water management authority in Netherland, then continued its development. In the next section, the system overview of RIES voting system and its principal components is described.

3.4.1 System Overview

RIES allows eligible voters to cast their votes in two distinct techniques - either by mail or electronically. Based on this key feature RIES allows its users to independently verify the election’s result. Similar to other E2E voting system, RIES was developed in order to increase the actual number of voters participating and to decrease unnecessary cost of the conventional election via mail (Hubbers, Jacobs & Pieters 2005). It provides internet voting scheme in a simple, straight forward and transparent way without sacrificing the system’s reliability, performance and maintenance cost. RIES voting system, which was implemented in Water Boards Election, differs from the initial system by Robers because of the implementation of few features. They are the elimination of multifunction smartcard, which is used to authenticate the
voters and the supplementary feature of vote by regular mail integration and additional users’ type in the system.

### 3.4.2 Principal components of the system

Explained below are the principal components of the system.

#### 3.4.2.1 System Users

In RIES voting system, there are four types of users. They are the system administrator, election board, RIES vote server or also referred to as SURFnet and the voters. Each of them is described in detail below.

(a) Administrator (TTPI)

In most electronic voting systems, administrators’ roles are very significant during the election preparation time. The main administrator for this election is a company called TTPI. It stands for Trusted Third Party Internetstemmen which consists of the architect and the main developer of RIES (Hubbers, Jacobs & Pieters 2005). Their roles vary from creating the secret keys, publishing the reference tables, merging the mail votes with the internet votes and computing the final tally of the election. In short, TTPI must absolutely understand RIES voting system.

(b) Election Board

The election board of RIES is the board of Hoogheemraadschap van Rijnland. They are the target user of RIES system development. The board consists of a large number of potential voters, but rather a smaller number of eligible voters (Hubbers, Jacobs & Pieters 2005).

(c) RIES Vote Server

In RIES, the national internet service provider for universities in the Netherland, called SURFnet, runs the vote server. Its main role is to manage the voters’ ballot which includes transmitting voters’ votes to TTPI.

(d) Voters

Eligible voters are allowed to cast and verify their vote by using RIES system. In the initial design of the system, voters were authenticated using a multifunction smartcard. Its implementation was rather costly. RIES use the secret key to authenticate the voters as a substitute. The other responsibilities of RIES voters are explained in the next section, System Stages.
3.4.2.2 System Stages

In its initial design by Robers, as well as in its actual development by Hoogheemraadschap van Rijnland, RIES election procedure is divided into three distinct stages. They are before, during and after the election stage (Hubbers, Jacobs & Pieters 2005). In this section, each of these stages is renamed with an intention to give a better understanding to the readers. Initial stage corresponds to before election stage, sequentially; during and after election stage is associated with voting and tally stage. The three stages are as follows.

(a) Initial Stage

As stated before, TTPI is required to generate secret keys. It includes the generation of unique election ID for all eligible voters and secret key for each corresponding voter. TTPI also needs to generate Voter ID for each voter by using the secret keys, which had been printed on the ballots. The collection of these ballots constructs a reference table or also refers to as pre-election table. This table contains of every possible vote for every voter. Each of the votes is in MD5 key-less hashes value (MDC). Once this table is created, it would be published on the internet for voters’ verification purpose. As the secret keys are not required anymore by the TTPI, soon after the reference table is published, the secret DES keys need to be destroyed by the authorized TTPI personnel.

(b) Voting Stage

Besides the voters, RIES introduced another user to assists the election stage; the RIES vote server, which is also known as SURFnet. Both these users would then communicate with each other in order to complete the voting process. As stated above, each of the users’ secret DES key is printed on the ballot given to them. In order to retrieve the candidate list and cast their vote, the users must insert their correct keys to the system. Upon successful insertion of keys, the users are able to generate ‘technical vote’. It is an encrypted vote that is sent by the voters to the vote server through the Socket Secure Layer (SSL) over the internet. Once the vote server obtains the technical vote, it would then proceed to generate and send a receipt confirmation back to the voter and continue to save their technical votes in a secured DB. The voters are required to keep both confirmation codes and technical vote for vote verification process in order to verify the accuracy of the technical votes sent and stored in the server. The voters also need to destroy the paper ballot in their possession to eliminate any possibility of vote-coercion.

After upon successful ballot submission, the system would calculate hash value by using voter’s personal secret key (MAC) and construct post-election table. Individual and universal verifiability are offered by this system through checking the validity of votes. It can be determined by computing MDC of all the votes cast in the post-election table and checking it...
against the hash value in the pre-election table (Hubbers, Jacobs & Pieters 2005) without disclosing the voter’s receipt. This process is described in detail in the tally stage.

(c) Tally Stage

In this stage, all the users need to participate to complete the tally process. Once the voters have cast and submitted their votes, TTPI and the vote server together with the election officials need to work together to collect, compute and publish the election result. Due to the two different ways of voting in RIES, the votes are processed in two different ways. SURFnet as a vote server are only obliged to collect all the technical votes cast by the voters. TTPI on the other hand are required to convert the collected mail votes into technical votes in order to compute an MD5 hash values of the collected votes and to publish them on the post-election table. The votes are computed in order to make sure the accuracy and authenticity of the votes collected on the vote server. If the hash values of the votes can be found on the pre-election table or the reference table by TTPI, then their validity status are accepted, if not they will be marked as invalid votes. Voters could check and confirm the validity of their votes based on their preference by checking their votes in the post-election table. In most of the electronic voting procedures, this particular process is carried out in one separate stage. However, RIES does not emphasize voter verifiability as much as other E2E voting system. Instead, it is focusing more on universal verifiability instead. Thus, the voters could optionally make use of the voter verifiability feature based on their needs. It is available to be used by the voters with a condition. First, they must obtain and keep the system generated receipt once they have casted their votes. Secondly, in order to verify their votes, the voters must look through the post-election table for their votes. By executing these steps, the voters are able to verify their votes’ statuses. After all the necessary procedures are completed, the election officials will count the final tally. They need to count the number of the valid votes displayed on the reference tables for each of the candidates and finally publish the outcome to the public.

3.5 Summary and Conclusion

This chapter covered the study of four E2E voting systems, namely Helios open-source web-based voting system, Scantegrity II optical voting system, Prêt à Voter and RIES voting system. Its aim was to introduce various approaches implemented in E2E voting systems. It included the system overview and the principal components of each system such as system users and system stages. These systems were chosen because of their stability. Each was used in actual governmental and organizational elections, two of which are elections in educational institutions.
In conclusion, all E2E voting systems were designed to fulfill two main objectives of E2E voting systems, to provide individual-verifiability (also known as voter-verifiability) and universal-verifiability. The four voting systems reviewed in this chapter are equipped with both features. Ironically, the implementation of these features caused the initiation of some known attacks, for example the randomization attack. This matter will be discussed further in Chapter 6, System Testing and Consideration of Security Attacks on the System.
CHAPTER 4  

SECURE E-VOTING ANALYSIS AND DESIGN

4.0 Introduction

In order to achieve the aims and objectives of this project, Software Development Life Cycle (SDLC) was followed to deliver eVoted; a secure end-to-end verifiable electronic voting system proposed in this research project. SDLC is a methodology for design and implementation of an information system (Whitman & Mattord 2012). It consists of a set of structured system development project phases from preliminary development analysis to post-development software testing and evaluation. These phases are conducted in a sequential manner. Each of these phases has its own output documentation or programs and is used as guidance by the developers to plan and control the entire system development.

Various SDLC methods are available for project development based on their compatibility with the project. Initially there was one particular SDLC that was and is still adopted in software development throughout the world. Winston W. Royce originally proposed it in 1970. With the on-going changes in system development methodologies, many enhanced variants of the method, such as Agile method, Rapid Application Development (RAD), Spiral model, Iteration, Incremental Development, Scrum, etc. have been developed. The primary SDLC method is also known as the waterfall method. The improved version of this method, the iterative waterfall model has been used for eVoted system development because of a significant reasons, that is it allows system developers and designers to correct any mistakes made at any of the stages during the system development.

The iterative waterfall model is divided into five distinct phases, namely requirement analysis, design, implementation, testing and maintenance. Each phase is described briefly in the next section. The remaining section of this chapter describes the requirement analysis and design phases of the method applied to eVoted system design. The next two phases are described in Chapter 5 and 6.
4.1 Iterative Waterfall Model

This method is a modified version of the waterfall model. The waterfall model is used to explain a feasible large-scale software development practice based on Royce’s personal experience. It consists of a linear series of software development process that needs to be completed sequentially in order to complete the project. This linear series is a collection of phases that needs to be executed in sequence and monitored closely. To monitor the process, project managers examine the allocated time set to complete each phase. The completion status of these phases would help to determine the progress of the project. However, this method carries an intrinsic flaw. The execution of each stage is dependent on the completion of the previous stage(s). This might result in delay on the project development process. In order to solve this problem, an amended method is introduced, namely the Iterative Waterfall model. Its characteristics are similar to the conventional waterfall model. Their difference lies at the mechanism of the phases. While the phases are executed in sequence for waterfall method, in iterative waterfall model, the phases can be carried out with no regard to the sequences of the phase itself. This is to allow the software designers to check and perform corrections on previous stages they completed. The five distinct phases mostly found in iterative waterfall model are described below.

(a) Requirement Analysis

This phase is called the requirement analysis phase. In order to develop computer software, all of its requirements need to be gathered and documented properly as Software Requirements Specification (SRS). SRS is a detailed description of the software under development defining its interaction with the users, hardware, and optionally with the other software. It is generated as a guidance and benchmark for project managers and developers to enhance effectiveness and efficiency in software development. It helps them to prioritize their tasks and most importantly to give a clearer perspective of the overall software design and development.

There are two types of requirements in SRS - functional and non-functional requirements. Functional requirements are usually user-related behavior of the software, which consists of purpose, scope, perspective, functions, software attributes, user characteristics, functional specifications, interface requirements and database requirements. On the other hand, non-functional requirements refer to the design and operation of the software. They include reliability, scalability, testability, availability, maintainability, performance and quality standards. There are no fixed parameters as each software has different requirements.
(b) Design

Software design phase consists of designs and schema mostly in diagram or model form which elaborates the software’s infrastructure. These models and diagrams are used to describe the solution of requirements collected and analyzed in the previous phase for efficient and timely implementation of the software. Due to its technical nature, this phase is usually completed by technical personnel, such as software developers, designers and project managers. The solutions to distinct criteria, constraints, limitations and requirements are implemented in software architecture design which generally consist of graphical user interface design, data structure and algorithm design and database schema (Entity Relational Diagram).

(c) Implementation

Based on the algorithms and designs produced on the previous phase, software developers write, compile, execute and deploy codes. In this phase business requirements and design specifications are built into software that generates the expected output(s). In order to ensure a sustainable end product, the developers must make sure that they follow all of the standards provided through the SRS and the overall architecture design of the software. Good coding techniques and programming practices from the developers is important in this stage. It could shorten the maintenance phase at the end of the SDLC.

(d) Testing

Once the software is built, its quality must be checked by conducting software verification and validation. Verification means evaluation of software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Validation on the other hand is a process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements (IEEE 1991). Besides, the testing phase also involves software debugging to mitigate errors identified during the verification and validation process or even for errors made at the initial phases of the software development process. Usually testing phase requires assistance from the intended users and needs to be completed before the software is finally deployed.

(e) Maintenance

During maintenance phase, software is stable and does not need much attention compared to the previous phases. Authorized personnel analyses its performance periodically to ensure its quality using some pre-set parameters. However, this process could also be extended to software environment adaptation, assistance of new user requirements and additional software reliability if any modifications need to be made on the system.
4.2 Software Requirement Analysis of eVote Voting System

A system has certain requirements that need to be satisfied for it to be able to function properly. This section states the requirements that eVote voting software must satisfy. It consists of few major sections that describe the logical grouping of functions and the overall modules of the system as well as the detailed functional and nonfunctional aspects of the system.

4.2.1 Users

eVote as a remote E2E voting system not only offers secure and reliable voting system, it also provides a flexible platform for the election officials to set up and maintain based on their own needs. System users are divided into three distinct types (levels) as follows.

(a) Voters

In every electoral system, may it be traditional or electronic; voters are one of its primary users. They are the main motives behind electronic voting system’s development. Voters expect and request a secure system that is time efficient and easy to use. With the evolution of Information Technology, it is possible to develop such a system. Voters’ needs are presented in the next section of this chapter.

(b) Polling Officers

Adopted from traditional voting system’s procedure, polling officers are needed in kiosk electronic voting system or in any voting system that provides voting at polling booths (stations). Remote electronic voting system normally does not require assistance from the polling officers. Electronic Voting systems, in order to add another layer of security, polling officers are employed for protection from possible threats.

(c) System Administrators

Every system is required to have system administrators to manage and maintain the system. This is true in Electronic Voting systems as well. In eVote, administrators can set up an election, add eligible voters and officers, edit registered voters’ and registered officers’ information, assign different voters to their respective polling officers, etc. Even with the highest level of access, it is not possible for eVote’s system administrators to cheat or to execute some known attacks of electronic voting system. These known attacks are to be elaborated in more detail on Chapter 6, Security Attacks on E2E Voting systems.
4.2.2 Scope

eVote is an improved version of the existing End to End Verifiable voting system. While the existing E2E voting systems cater to different scales of election scale, eVote is intended to assist voting process in small to medium scale election. The various features supported by eVote are subject to the user type, with the system admin as the highest user level. For the time being, the latest version of eVote is only available in one language; English. Further development(s) of eVote might include its availability in multiple languages and wider election scales to cater more voters.

4.2.3 Major Assumptions

During and before the development process, following are the assumptions and constraints on eVote.

- Users are computer literate (They are able to perform daily webpage tasks such as locating and accessing web software applications and are familiar with basic hardware and software terms and vocabulary)
- Users are familiar with election-related vocabulary and skills and therefore do not require eVote to assist them on this.
- Users have satisfactory level of fluency in English language.
- Users do not have any physical disability which prevents them from using a computer.
- Users, specifically the system administrator (electoral commission) must provide the required infrastructure for eVote to work. For more details on the required infrastructure, refer to the next section, Operating Environment and Dependencies.
- Users are trustworthy.

4.2.4 Operating Environment and Dependencies

In order to work properly, eVote requires the following components.

- Microsoft Windows 7 Operating System (64-bit) and above.
- Intel Core 2 Duo 1.66 GHz CPU
- 4 GB of RAM.
- Approximately 500 GB Hard Disk capacities.
- Keyboard and Mouse.
- Client and server side computers must be connected to the same Local Area Network (LAN).
- Sun Microsystems’ Java Runtime Environment 1.6.0 and above.
- NetBeans IDE, Java EE bundle V6.8 and above.
• Java Development Kit (JDK), V6.
• Glassfish Application Server V3 and above.
• MySQL Database Server V5.1 and above.
• MySQL Connector/J V5.1.25.
• Web browser application.
• Read and Write permissions on the source, destination and working platform for automatic and manual data import and export purposes.

These supporting resources are discussed in detail in the next chapter, System Implementation.

4.2.5 System Requirements

System requirements consist of three sections: requirement ID, summary description, requirement definition and its priority. Requirement ID is a unique identifier that enables identifications of each requirement uniquely. Without an ID, a requirement cannot be managed properly. No two requirements can have the same ID. Requirement definition is the description, in free-form text, that formally defines, both functional and non-functional. On the other hand, its priority states how important the requirement is. There are 3 priority levels - High, Medium and Low.

• High: The system is not fit for deployment without satisfying these requirements.
• Medium: These requirements can be satisfied with respect to developers’ competency and constraints.
• Low: These are optional requirements that need not be developed.

Lastly, it includes the summary description that sums up the requirement briefly. This is to improve the readability of the specification and to search a requirement.

To have clearer software requirement analysis and design, the system requirements are subdivided as functional and non-functional requirements. Some of the requirements might have additional properties for further information. All of the requirements are presented in tabular form in Appendix A and B.

4.2.5.1 Functional Requirements

Functional requirements are a set of tasks or functions the system is required to perform. It summarizes the intended behavior of the system. The functional requirements of eVote include system authorization, system authentication, applied technologies, system documentation, its data type and data handling. They are listed in Table A-1 to A-6 in Appendix A.
4.2.5.2 Non-Functional Requirements

Non-Functional requirements are a set of tasks or functions that shall be performed by the system. It sums up the whole operations of the system. eVote’s non-functional requirements consist of its security, usability, performance and reliability. Each of these non-functional requirements is displayed in Table B-1 to B-4 in Appendix B.

4.3 Software Design of eVote Voting System

The software design of eVote is separated into few parts, system integration, data models and process models. System integration elaborates the overall integration between the components of eVote voting system. It is described in more detail in graphical notation of data models and process models. They are discussed as follows.

4.3.1 System Integration

One of the requirements that eVote must meet is mobility requirement. In order to fulfill this requirement, eVote Voting system must be built as a web application. Hence, votes can be cast from anywhere the users prefer, provided stable internet connection is available. Due to its low platform dependency as well as other characteristics such as security, robustness and scalability, Java EE 6 has been chosen as the main platform of eVote’s system architecture. Java EE infrastructure is a set of specifications implemented by different containers (Goncalves 2010). Containers themselves are Java EE runtime environments that support a set of Application Programming Interface (API) and provide various services to their hosted components. Two types of containers implemented in this research are Web Container and EJB Container which run on Glassfish Application Server.

![Figure 4-1: Integrated eVote System](image)

**Figure 4-1**: Integrated eVote System
Web Container offers services for managing and executing web components. In this research they are the modules of eVote. Each module carries out a task that can include or extend other tasks the system performs. In the Use Case Diagram as shown in Figure 4-6, modules are represented in actions. Web Containers are also required to instantiate, initialize and invoke servlets and supporting HTTP and HTTPS protocols (Goncalves 2010). EJB Container on the other hand, manages and executes enterprise beans that contain eVote’s business logic. It creates and manages the instances of EJBs and JPAs and provides a number of services for eVote, such as distributed transaction between network hosts, security, database access, naming and directory service (Goncalves 2010). The implementation of Java EE technology will be described in detail in the Chapter 5, System Implementation. Its integration with the client machine, Java application server and database server is shown in Figure 4-1.

4.3.2 Data Models

Data models are used to capture the data information and relationship in a system design. In this research, two types of data models are used, Entity Relationship Diagram (ERD) and Data Dictionary. Both are discussed below.

4.3.2.1 ERD

ERD is a model that shows the logical relationships and interactions among system entities (Shelly, Cashman & Rosenblatt 2007). It is one of the data models used to elaborate data requirements (database) and assumptions in the system. Figure 4-2 displays the ERD of eVote System.

4.3.2.2 Data Dictionary

Data Dictionary is a central storehouse of information about the system’s data (Shelly, Cashman & Rosenblatt 2010). It provides a detailed description of each table in the user or designer’s pre-constructed database. Such description includes attribute (field) names of the table, the characteristics (data type) for each field of the table, and its relationship to the other tables (Coronel, Morris & Rob 2011). As shown in Table 11, the relationship of each table is defined by the table constraints. There are two types of keys used to enforce data integrity in the database tables - Primary Key (PK) and Foreign Key (FK). Primary Key consists of table column(s) whose value uniquely identifies the characteristic for each row of the table. Thus, each table must have a Primary Key. The other constraint, Foreign Key, indicated the relationship of one table to another table through the referential constraint. It consists of one or more columns in a particular table whose value in one record uniquely identifies another record of another table. The detailed information of eVote Voting system’s database design is illustrated in Table C-1 in Appendix C.
4.3.3 Process Models

In a system design, analyzed business processes are presented in process model. For eVote’s system design, its election procedure is described in few Unified Modeling Language (UML) models as follows.

4.3.3.1 UML Models

UML is a family of graphical notations, backed by single meta-method which is a more comprehensive methods used to explain a set of methods. It helps in describing and designing software systems, particularly software systems built in the Object-Oriented (OO) style (Fowler 2004). In order to explain eVote’s system design, two types of UML models are elaborated, namely Structural Modeling and Behavioral Modeling.
(a) **Structural Modeling**

System overview that emphasizes on the structure of its objects as a whole and defines its static architecture can be classified as a structural modeling. This type of modeling is used to model objects’ elements such as classes, interfaces, components (packages) and nodes. It is also utilized to show in detail relationships and dependencies between the objects’ elements. Described below are some of eVote’s structural models, which gives a clearer view, and understanding of its system architecture design.

(i) **Package Diagram**

This type of diagram displays two or more packages and their dependencies. Package Diagram helps in organizing model elements, e.g. classes or use cases into groups as well as overviewing user requirements and system design (Ambler & Sadalage 2005). In eVote’s architecture design, its classes are grouped based on the design pattern concept implemented in this research project, the Model-View-Controller (MVC) architecture. This concept was originated by Smalltalk programmers, Trygve Reenskaug in 1979. By implementing MVC concept, the system and its tasks are separated and handled independently by three interoperable components, namely model, view and controller. Each of its usage in a system is described in general in Figure 4-3, while its behavior is shown in Figure 4-4. The three components of MVC concepts are distributed into different packages in eVote system model as shown in Figure 4-5 with an exception for model component, which it is implemented in a different part of the system due to its nature as a User Interface controller. The integration of its packages with the other packages that supports eVote’s functionality and security aspects is the backbone of this system.

(ii) **Class Diagram**

In process modeling, class is used to define object types and its characteristic, and an object is an instance of a class. UML Class Diagram represents a detailed view of a single business process use case and shows the class that participate in that particular use case. Other than that, it also documents the relationship among the classes often refer as cardinality (Shelly, Cashman & Rosenblatt 2007). In other words, it shows the building blocks of eVote voting system, which is designed as an object-oriented system. The components of eVote’s Class Diagram including its attributes and behavior, and its correlation to other class in the package are illustrated in detail in Appendix F. Each of them corresponds to the entities in data dictionary.
Model
- Encapsulates application state
- Responds to state queries
- Exposes application functionality
- Notifies views of changes

View
- Renders the models
- Requests updates from models
- Sends user gestures to controller
- Allows controller to select view

Controller
- Defines application behaviour
- Maps user actions to model updates
- Selects view for response
- One for each functionality

State query
- User gestures

Change notification
- View selection

Figure 4-3: Each component usage of MVC design pattern concept (Netbeans 2013)

Controller Model View
handleEvent
service
update
getData

Figure 4-4: Behavior of MVC design pattern concept in a Sequence Diagram
(b) Behavioral Modeling

Requirements of a new system together with its proposed functionality can be defined and described through Use Case Model in Behavioral Modeling. It represents the interaction between a user and the system in a clearer perspective and helps to identify the business process activities of the system. Use Case Model includes Use Case Diagrams and Descriptions, Activity Diagrams and optionally Sequence Diagrams. To describe eVote's processes in more detail, Use Case Diagrams Description and Activity Diagrams are provided in the next subsections.

(i) Use Case Diagram

This diagram could be used to conveniently document the system activities. It shows the roles of eVote Voting system and how the system implements those roles (Satzinger, Jackson & Burd 2008), as shown in Figure 4-6. In order to create Use Case Diagram, few symbols are needed. Table 4-1 elaborates the description of each symbol used in the Use Case Diagram.
Table 4-1: Legend of Use Case Diagram

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>The shape describes a set of actions performed by the actor. A Use Case may extend its process to another Use Case, which automatically inherits its behavior. Besides that, a Use Case may include the functionality of the extended Use Case to its own process as well.</td>
</tr>
<tr>
<td><img src="image" alt="Actor" /></td>
<td>Actor is the user of the system. It uses Use Case to perform some processes of the system. The overall role of an actor in the system and the scope of actor’s action are defined by the Use Cases accessible to that particular actor.</td>
</tr>
<tr>
<td><img src="image" alt="Line" /></td>
<td>The line that connects the actor to the Use Case(s) identifies the relation and accessibility between the action and actor.</td>
</tr>
</tbody>
</table>

Figure 4-6: eVote’s business processes in Use Case Diagram
(ii) **Activity Diagram and Use Case Description**

In a system, a sequence of steps to process a business transaction or user request is generally
known as system workflow. It depends on the business processes. Activity diagrams describe
various actions and their sequence in the system workflows. An activity diagram is a form of
system model associated with the object-oriented approach that has no development
approach restriction (Satzinger, Jackson & Burd 2008). As shown in Figure 4-6 below, the
election procedures of eVote voting system are namely, election setup, add eligible voter or
officer, ballot monitoring, tally checking, profile editing, registration, authentication, voting
and vote verification. In figure D-1 to D-9 in Appendix D, each of these business processes
is elaborated in Activity Diagrams. Use Case description of each processes are elaborated in
Table E-1 to Table E-10 in Appendix E for further explanation of each use case.

4.4 **Summary and Conclusions**

The proposed electronic voting system in this research is called eVote. eVote was designed with
SDLC model as guidelines in its implementation. Iterative Waterfall model was chosen due to
its flexibility characteristic, where the system designer is allowed to make changes at any stages
of the SDLC that they have completed. This chapter discussed the first two stage of the SDLC
in detail, Requirement Analysis and Design.

Most of the requirements are gathered from literature review. In chapter 1, some of the non-
functional requirements of electronic voting systems were discussed. The second part of this
chapter extends the requirements, which cover the functional requirements and few other non-
functional requirements. Besides that, some additional information including the system users,
its scope, major assumptions and its operating environment and dependencies were also
discussed.

In the last part of this chapter, eVote system design was reviewed. In eVote, election procedure
is divided into five different stages, namely registration, authentication, voting, tallying and vote
verification. eVote is developed using object-oriented design. Thus, each of the election users
and processes are represented as objects. Its business processes are presented in different
diagrams and tables to explain the system in detail, namely, ERD, data dictionary and several
types of UML diagrams. The integration of each component described in this chapter is
explained further in the next chapter.
CHAPTER 5

SYSTEM IMPLEMENTATION

5.0 Introduction

The main idea of this chapter is to describe the other phases of iterative waterfall model namely - different levels of system implementation and its related technologies. It includes the review of various types of technologies and development tools used to support its development and the implementation of cryptographic and image steganographic approaches discussed in Chapter 2, Literature Review. The implementation is divided into two parts based on the Java platform used - eVote v1.0 and eVote 2.0. In the development of its initial version, eVote v1.0 utilizes the APIs provided by Java Standard Edition (SE). This first version of the system is developed purely to evaluate the stability and accuracy of different information security techniques. Therefore, it does not require advance technologies in its development. In contrast to v1.0, eVote v2.0 make use of the various services offered by another Java Platform; Java Enterprise Edition (EE). eVote v2.0 is the finalized version of the voting system in this research project. It is equipped with required features of remote E2E voting system. Therefore in this chapter, the review only emphasizes on technologies related to Java EE platform.

5.1 Technologies

eVote as a web application assist the communication between two parties over the HTTPS protocol - the client and the server. All requests made by the user are sent over to the server by the client which is a web browser. Once the server receives the requests, it must respond accordingly. These responses will then be displayed by the client to the respective users. Because of this, the review of applied technologies in this section is divided into two sub-sections - client-side and the server-side. These client-server technologies are described as follows.

5.1.1 Client-Side

The client-side technologies were applied to enhance eVote’s performance and usability. With the implementation of these technologies, web page formatting could be done more efficiently with less complexity for the system developer. These three technologies are discussed as follows.
(a) **HTML**

Hyper Text Mark-up Language (HTML) is a mark-up language which is a set of mark-up tags used to describe web pages. HTML represents the content as well as the visual aspect of a static web page. However, to add additional features to the layout of web pages, eVote utilizes another client-side technology; Cascading Style Sheets (CSS).

(b) **CSS**

CSS is a style language that is designed primarily to enable the separation of document content from document presentation, including elements such as layout, colors, and fonts. CSS on the other hand enables the separation of content and presentation of the system through its syntax. It’s most common application is to style web pages written in HTML and XHTML.

(c) **JavaScript**

JavaScript is a scripting language used for client-side development which allows the developer to add some additional functions to the web page dynamically (Goncalves 2010). JavaScript offers an immediate compilation and response time. Besides that, JavaScript syntax is loaded on the client-side which reduces the load on application server.

5.1.2 **Server-Side**

Client-side scripting or J2SE APIs alone cannot deliver the requirements of remote E2E voting system. Server-side technologies must be used in eVote’s development to ensure the fulfillment of its functional features. These technologies are explained as follows.

(a) **Servlet and Java Server Pages (JSP)**

Similar to JavaScript, JSP was developed to create web pages dynamically as response to client request. Clients access eVote through any available web browser via HTTPS connection. Those generated web pages are compiled on the server-side as servlets. Both technologies can utilize any resources on the server-side, including EJBs, databases, web services and other resources.

(b) **Enterprise JavaBeans (EJB) Technology**

In J2EE, business logics that are encapsulated into server-side components called the EJBs. EJBs provide a range of transaction and persistence services and access to Java EE service and communication API. Thus, EJBs could be integrated with other Java SE and Java EE technologies, e.g. JPA, JDBC, Java Transaction API (JTA), Java Messaging Service (JMS), Java Naming and Directory Interface (JNDI), etc. According to the application complexity there are three types of EJB, Stateless, Stateful and Singleton. The maintenance of the values (state) for each beans instance variables is performed across multiple method calls. eVote operations
can occur in only a single method call, meaning any instance could be used for any clients. Therefore, stateless session beans are applied in this research project.

(c) **Java Persistence API (JPA)**

JPA is the Java standard for object-relational mapping technique (Paniza 2011). It provides the integration of Java object and relational models for data persistence among applications. This enables data access from relational database to Java containers in the application in object oriented manner. There is quite a number of persistence providers available, EclipseLink persistence provider were used for this research.

(d) **Java Database Connectivity (JDBC)**

JDBC is a Java API that provides efficient and easy connectivity between Java EE application and a wide range of data sources by utilizing the appropriate driver. It allows application components to perform connection and authentication to the database server, manage transactions, execute stored procedure and communicate to database engine in order to transfer and retrieve results of SQL queries (Singh, et. al. 2002).

### 5.2 Development Tools

This section covers all necessary tools used in the development of eVote voting system. Each of them is reviewed as follows.

#### 5.2.1 Netbeans IDE

Netbeans is one of the Integrated Development Environment (IDE) as well as a generic application platform framework initially developed for Java Desktop applications. Thus, Netbeans is equipped with built-in support for Java SE, Java EE, Java ME and Java FX. Other than that, it offers many additional tools and support that could maximize developer’s productivity and efficiency in building Java EE applications. One of the tools used to developed eVote is JUnit. JUnit is an open source system development tool designed to support system testing of a particular system developed in Java programming language. It allows the developers to write and test their source codes in easier and more efficient way. Based on these reasons, Netbeans IDE 7.3 had been chosen as our primary development tools.

#### 5.2.2 Glassfish Application Server

Glassfish is a cross-platform open-source Application Server for Java EE platform. It was initially developed by Sun Microsystems (now acquired by Oracle Corporation). On its latest version, Glassfish v3 provides fully-featured implementation of Java EE 6. The same version of Glassfish Application Server was used project because of its status as Oracle’s own Application
Server, which makes Glassfish the most compatible Application Server to be used with Netbeans IDE 7.3 and Java EE 6.

### 5.2.3 MySQL Relational Database Management Server (RDMS)

Relational Database Management System (RDBMS) is a database management system based on its relational model. RDBMS data is structured in database tables, fields and records. Each RDBMS table consists of database table rows and each row consists of one or more database table fields. The data are stored in a collection of tables, which might be related to common fields or also known as database table columns. RDBMS provide relational operators to manipulate the data stored into the database tables. Most RDBMS use Structured Query Language (SQL) as database query language. MySQL is among the most popular relational database management system in IT industry. MySQL 5.6 was selected as the primary RDBMS in this research projects based on the aspects listed below.

- Open source.
- Support different type of Operating System.
- Works well with Java programming language.
- Allow multiple users accessing the database at the same time.
- Reliable and stable rather than other database.

In eVote system’s development, MySQL database is used to store all of the election data. However, a particular JDBC driver is required to access MySQL database. MySQL Connector/J is the official MySQL JDBC driver.

### 5.3 Implementation of Information Security in eVote System

Security is very important in the development of a Remote Electronic Voting System. Based on the literature reviews and user requirements, two types of information security have been used in this research project. They are described as follows.

#### 5.3.1 Cryptography

There are three distinct cryptography techniques that were implemented in this research, namely, Password Hashed-based Scheme, Visual Cryptography and Threshold Decryption Cryptosystem. All three cryptography techniques have elaborated in detail on Chapter 2, Literature Review.

Password Hashed-based Scheme is applied to secure user’s password in registration and authentication stage. Hashed-based Algorithms are one way functions, meaning its ciphertext cannot be reversible into a plaintext back. Therefore no decrypting computation is needed in the server-side. The server is only required to compare the ciphertext from user input with the
ciphertext stored in the database to authenticate the user. This makes Hashed-based Algorithms the most suitable technique for password cryptosystem. As previously mentioned in Chapter 2, salt value and PBKDF2 are implemented alongside Hashed-based Algorithms to support Password Hashed-based Scheme for protection against known attacks. Salt values are generated for each password inserted by the user before hashing to randomize its hashing. To generate a completely secure salt value, reliable Pseudo-Random Number Generator (PRNG) is used. In this research project, Java EE SecureRandom Class was used with 24 bytes of salt value. The other Java EE Class that was utilized is SecretKeyFactory Class for key stretching. This class constructs secret keys by using PBKDF2WithHmacSHA1 algorithms found in RSA Laboratories’ Public-Key Cryptography Standard (PKCS) #5 v2.0 (Oracle Corporation 2011). Figure 5-1 shows the utilization of these algorithms in the system.

```java
public static String createHash(char[] password) throws NoSuchAlgorithmException, InvalidKeySpecException {
    // Generate a random salt
    SecureRandom random = new SecureRandom();
    byte[] salt = new byte[SALT_BYTES];
    random.nextBytes(salt);
    saltVal = salt;
    // Hash the password
    byte[] hash = pbkdf2(password, salt, PBKDF2_ITERATIONS, HASH_BYTES);
    // format iterations:salt:hash
    return PBKDF2_ITERATIONS + ":" + toHex(salt) + ":" + toHex(hash);
}

private static byte[] pbkdf2(char[] password, byte[] salt, int iterations, int bytes)
    throws NoSuchAlgorithmException, InvalidKeySpecException {
    PBEKeySpec spec = new PBEKeySpec(password, salt, iterations, bytes * 8);
    SecretKeyFactory skf = SecretKeyFactory.getInstance(PBKDF2_ALGORITHM);
    return skf.generateSecret(spec).getEncoded();
}
```

**Figure 5-1:** Excerpted source code of Password Hashed-Based Algorithm implementation in eVote System (Salted Password Hashing)

Visual Cryptography was implemented in the system development in order to prevent vote buying or selling as well as vote coercion by providing direct assurance to each of the voters through digital vote receipt. The plain text, in this case the ballot, will be encrypted to two shares of ciphertext. The ciphertext has two separated layers of pixel symbols. For additional layer of security of the ciphertext shares, Java EE SecureRandom Class was applied in secret message distribution over the shares. SHA1PRNG is the default PRNG of SecureRandom Class.
as illustrated in Figure 5-2. In the end, one share will be given to the voter as their vote receipt; the other share is to be stored in the database. To decrypt these shares, Visual Cryptography decryption algorithm is required to be executed. Figure 5-3 and 5-4 respectively display the process flow of encryption and decryption in Visual Cryptography. This mechanism was adopted from Chaum’s secret-ballot receipt (Chaum 2004). However, a simple amendment made to the applied mix-net scheme used by Chaum (2004) due to its extensive process. In this research, Threshold Decryption Cryptosystem will be used.

Even with the implementation of cryptography security approaches, attacks and threats still cannot be averted. There are enormous numbers of possible attacks in a Remote Electronic Voting System. Besides using Visual Cryptography and Password Hashed-based Scheme, eVote implemented Threshold Decryption Cryptosystem as an additional layer of security. This decryption scheme offers robust and secure key management of cryptography system. It is implemented in Tallying Stage to protect election’s ‘Ballot Box’. A few authorized personnel (system administrator excluded) are appointed to hold a key. The combination of these keys would then be used to unlock the ‘Ballot Box’. Figure 5-4 describes the secret key distribution mechanism.

```java
private void getDefaultPRNG(boolean setSeed, byte[] seed) {
    String prng = getPrngAlgorithm();
    if (prng == null) {
        prng = "SHA1PRNG";
        this.secureRandomSpi = new sun.security.provider.SecureRandom();
        this.provider = Providers.getSunProvider();
        if (setSeed) {
            this.secureRandomSpi.engineSetSeed(seed);
        }
    } else {
        try {
            SecureRandom random = SecureRandom.getInstance(prng);
            this.secureRandomSpi = random.getSecureRandomSpi();
            this.provider = random.getProvider();
            if (setSeed) {
                this.secureRandomSpi.engineSetSeed(seed);
            }
        } catch (NoSuchAlgorithmException nsae) {
            throw new RuntimeException(nsae);
        }
    }

    if (getClass() == SecureRandom.class) {
        this.algorithm = prng;
    }
}
```

**Figure 5-2:** Excerpted source code of PRNG function in SecureRandom Class (Oracle 2011)
public void generateShares() throws FileNotFoundException, IOException {
    if (fSrcFile == null || !fSrcFile.exists()) {
        throw new FileNotFoundException("Source file does not exist: "+ fSrcFile.getName());
    }
    BufferedImage imgSrc = Crypting.loadAndCheckSource(fSrcFile, 0, 0, false);
    imgKey = Crypting.generateKey(imgSrc.getWidth(), imgSrc.getHeight());
    imgEnc = Crypting.encryptImage(imgKey, imgSrc);
    try {
        ImageIO.write(imgEnc, "png", new File(encImgFile));
        ImageIO.write(imgKey, "png", new File(keyImgFile));
    } catch (IOException e) {
        throw new IOException("IOException occurs interface saving " + imgKey.toString() + " and " + imgEnc.toString());
    }
}

public String decryptShares(File fKeyFile, File fSrcFile) throws FileNotFoundException, IOException {
    if (fKeyFile == null || !fKeyFile.exists() || fEncrFile == null || !fEncrFile.exists()) {
        throw new FileNotFoundException("Source file does not exist: "+ fKeyFile.getName() + " and " + fEncrFile.getName());
    }
    imgKey = loadAndCheckEncrFile(fKeyFile);
    imgEnc = loadAndCheckEncrFile(fEncrFile);
    imgOverlay = overlayImages(imgKey, imgEnc);
    try {
        ImageIO.write(imgOverlay, "png", overlayImgFile);
    } catch (IOException e) {
        throw new IOException("IOException occurs interface saving " + imgOverlay.toString());
    }
    return imgOverlay.getName();
}

**Figure 5-3:** Excerpted source code of Visual Cryptography implementation in eVote System (Visual Cryptography)
5.3.2 Steganography

In Chapter 2 two types of steganography were discussed. Both types support secure data communication between multiple parties by providing secure data hiding algorithms. The difference between cryptography and steganography lies in the encrypted data. Cryptography encryption system generates cipher, while Steganography produces a stego-object which is not perceptible by HVS or statistically detectable by statistical attack, for example chi-square test. Due to its characteristics, such attacks can be categorized as steganalysis. Steganalysis is the art of identifying stegograms that contain a secret message; it does not consider the extracted message (Bateman, 2010). Commonly, there are three types of steganalysis; active, passive and malicious. Each is classified based on its distinct behavior in identifying the secret message in the communication channel. While in active steganalysis, the attacker or also known as the Warden intended to destroy the secret message, in passive steganalysis Warden would only block the detected secret message so that the receiver would not be able to receive the message. The last type of steganalysis is malicious. Malicious steganalysis is more dangerous compared to the previous two steganalysis. It identifies the secret message and tries to impersonate the sender by doing a workaround on the secret message to get its embedding scheme and replicating the secret message. In this system, however, it is assumed that all of the data are transmitted in a private secure channel, which means the channel is not accessible by any Warden. Thus, the implementation of steganography in this system could improve eVote’s security defense significantly.

Based on the studies, Image Steganography, in specific F5 Steganography Algorithm, is considered to be more appropriate for eVote secure data transmission compared to text steganography (and other image steganography schemes). This is because there is less
information redundancy in text cover file compare to image cover file which makes data hiding less effective. Further results obtained from the pre-evaluation of image steganography shows that F5 Steganography technique has more stable and decent characteristics compared to the other Image Steganography Algorithms namely, LSB, Palette-based and Spread Spectrum. One of the evaluations conducted was the comparison of initial and stego-image size for different image steganography techniques as shown in Figure 5-5. Shown in Figure 5-5, the stego-image of F5 method does not increased. As a matter of fact, the stego-image size decreased. Thus, it can be one of the advantages in transmitting the embedded stego-image from the client server to the election server. The other comparison is carried out to examine the robustness of each image steganography techniques against visual attack and statistical attacks namely, Regular-Singular (RS) analysis and Binary Similarity Measures (BSM) test. Based on the results displayed in Table 5-1, where the robustness of each image steganography methods are ranked from low to high, it can be seen that F5 is not highly subject to Visual Attack. Besides that, F5 also eliminates the possibility of the $\chi^2$ attack as explained in Chapter 2, Literature Review. Hence, it can be concluded that F5 resistance over these attacks is somewhat better compare to the other techniques.

Table 5-1: Robustness of different image steganography methods against visual and statistical attacks

<table>
<thead>
<tr>
<th>Image Steganography Algorithm</th>
<th>Visual Attack</th>
<th>Statistical Attack</th>
<th>Steganalysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSB</td>
<td>Low</td>
<td>Yes</td>
<td>RS Analysis</td>
</tr>
<tr>
<td>Palette-based</td>
<td>Low</td>
<td>Yes</td>
<td>BSM Test</td>
</tr>
<tr>
<td>F5</td>
<td>Medium</td>
<td>Yes</td>
<td>BSM Test</td>
</tr>
<tr>
<td>Spread Spectrum</td>
<td>High</td>
<td>Yes</td>
<td>BSM Test</td>
</tr>
</tbody>
</table>

Figure 5-5: Comparison of initial and stego-image size on different implementation of image steganography techniques
5.4 System Implementation

eVote system implementation is separated into two parts based on the distinct Java platform used - eVote v1.0 and eVote v2.0. Both are described as follows.

5.4.1 eVote v1.0

eVote v1.0 is not equipped with registration, authentication, tallying and publishing stages. It was mainly developed to examine the reliability of the implemented primary security approaches - image steganography and visual cryptography. It is not equipped with registration, authentication, and tallying stages. Thus, as stated in the earlier part of this chapter, eVote v1.0 was developed over Java Standard Edition (SE) platform with no connection to the Database Server. A few JFrame GUIs were implemented to perform the main components of eVote’s business process at a single user level to accommodate quality assurance of the selected security approaches. Figure 5-6 shows the process flow of voting stage and vote verification stage in eVote System in a sequence diagrams.

Figure 5-6: Sequence diagram of eVote v1.0 system
The stages are briefly described as follows.

(a) Voting Stage

To secure the vote transmission between the client and server and to assure the user of the integrity of the votes, both F5 image steganography algorithm and visual cryptography scheme are implemented in this stage. Once Netbeans IDE successfully compiles and runs the program, the first window will show the list of candidates for each category. Short information of each candidate selected is also provided on the right-hand side of the window pane. This feature allows users to have clear information on the candidate they are about to elect. As voters submit their votes, Embed class will be called. This class generates JpegEncoder Object which will create the stego-image file from the cover image file provided in the system. Stego-image is created in Compress() function of JpegEncoder class. After the generation of stego-image, it is saved to the client machine automatically. Ballot management in Voting Stage also applies; visual cryptography. The system must create a compatible source image containing the ballot for Visual Cryptography encryption by passing the secret message derived from the decrypted stego-image. As a result of the visual cryptography encryption, half of the shares need to be obtained by the voters. They are prompted to complete this task by saving their vote receipt into the folder they specify.

(b) Vote Verification

In a traditional paper-based voting, once the tally process is done, authorized personnel will announce the result of the election. However, voters will not be able to verify their own votes. As a result, voters cannot be assured that their submitted ballot is counted as cast. This may affect the turnout in subsequent elections. In order to solve this problem, voter receipt is implemented in the system development of E2E voting system. This receipt is not revealed in their ballots. It can be used by each voter to ensure that the ballot cast is properly formed by the system. Each user can only obtain one share of the visual cryptography encrypted image. The other half of the shares is automatically saved in the database. The combined shares would be used to retrieved and verified voter’s ballot. Voters of such a system can verify their own votes by submitting the vote receipt sent to their email accounts. The verification feature is supported by visual cryptography Scheme. The vote receipt submitted by the voters will be matched (decrypted) against the other half of the encryption share saved in the database during the voting stage. Figure 5-7 and 5-8 show the sequence diagram of the function called to decrypt the images.
The result of this decryption mechanism is the voter’s ballot. In this way, most of Remote Electronic Voting System’s requirements, such as incoercibility, receipt-freeness, universal-verifiability, etc. are ensured as neither the polling officers nor the system administrator have access to identify the collected ballots. Only the voter themselves have access to them.
5.4.2 eVote v2.0

eVote v2.0 is an enhanced version of eVote v1.0. eVote v2.0 is developed based on the preliminary studies conducted in this research project. It is developed using Java EE 6 framework with the integration of implemented technologies and development tools listed in the beginning of this chapter. As explained on the previous chapter, the system is equipped with three types of user levels, voters, polling officers and system administrators. eVote v2.0 offers full implementation of the system with features to meets the requirements listed on Chapter 4, Secure E-voting Analysis and Design. Figure 5-9 displays the overall process flow in eVote system stages. In this section, vote verification stage will not be discussed anymore, unlike voting stage which will be reviewed again. The processes of vote verification in both versions of eVote are the identical. However, the voting process of eVote v.10 and v2.0 are different due to the implementation of database.
Further explanation of each stage in eVote System is given as follows.

(a) Registration Stage

This stage is also known as the preparation stage. In this stage all constraints for the election are prepared. The registration stage is completed by the voters and polling officers. Prior to this stage, system administrators must prepare the election setup by inserting the details of the election and the candidates’ details for each category in the election. Besides, the system administrators also need to add eligible voters’ and polling officers’ record in the database. This record includes their username, Identification Card (IC) number and their valid email address.

Upon successful attempts, eligible users will receive an email from system administrator notifying their eligibility to register into eVote system as shown in Figure 5-10.

![Figure 5-10: Screenshot of the email received by an eligible voter](image)

Voter’s registration can be carried out only by eligible users. By accessing the link provided in the email, eligible voters and officers can now register themselves in the system. In order to register themselves in the system, users are required to provide their details and submit it to the system. These are then matched with users’ details saved by the administrators in the database to ensure the accuracy of the details given by the users. As another layer of security, users’ passwords will be cryptographically secured by applying Password Hashed-based Scheme described in the beginning of this chapter. By the implementation of this cryptography technique, only the hashed password together with its salt value is saved into the database. Its implementation will be explained further in the next stage of eVote System, Authentication Stage. Figure 5-11 and 5-12 show the registration process flow of the voters in two distinct sequence diagrams. There’s a slight difference between voters’ and officers’ registration processes. Each voter attempting to register in the system will be randomly assigned to a polling officer. This is done as an additional layer of protection over the database records which will be described further in the tallying stage. After successful registration process, users will be directed to their respective homepage by the system as displayed in Figure 5-13.
Figure 5-11: Sequence diagram of voter registration process

Figure 5-12: Sequence diagram of addEligibleVoter() function
(b) Authentication Stage

In a Remote Electronic Voting System, the implementation of this stage is mandatory. The objective of this stage is to ensure voters’ identity. Registered voters are authenticated by logging into the system. They will be prompted to enter their self-defined username and encrypted password for security purpose. As mentioned in the previous stage, users’ passwords are not saved in the database. Only its hashed and salt values are saved. It is because Hashed-based Algorithms are one way functions. It cannot be converted back to a plain text. To authenticate users, administrators are required to compare the ciphertext from user input with the ciphertext stored in the database. Shown in Figure 5-14 the screenshot of function used to authenticate users using Password Hashed-based Algorithm. Figure 5-15 elaborates the process of user authentication in a format of sequence diagram. Once a user has been identified as a registered voter and has successfully logged into the system, he will see a welcome screen which states the user account status and a menu panel where a user can navigate through features offered depending on the user level. The scheme is shown in Figure 5-16.
public static boolean validatePassword(char[] password, String goodHash) 
        throws NoSuchAlgorithmException, InvalidKeySpecException {
            // Decode the hash into its parameters
            String[] params = goodHash.split(":");
            int iterations = Integer.parseInt(params[ITERATION_INDEX]);
            byte[] salt = fromHex(params[SALT_INDEX]);
            byte[] hash = fromHex(params[PBKDF2_INDEX]);
            // Compute the hash of the provided password, using the same salt,
            // iteration count, and hash length
            byte[] testHash = pbkdf2(password, salt, iterations, hash.length);
            // Compare the hashes in constant time. The password is correct if
            // both hashes match.
            return slowEquals(hash, testHash);
        }

    private static boolean slowEquals(byte[] a, byte[] b) {
            int diff = a.length ^ b.length;
            for (int i = 0; i < a.length && i < b.length; i++) {
                    diff |= a[i] ^ b[i];
            }
            return diff == 0;
    }

Figure 5-14: Excerpted source code of Password Hashed-Based algorithm implementation in eVote
authentication stage (Salted Password Hashing)

Figure 5-15: Sequence diagram of user authentication process
Figure 5-16: Screenshot of polling officer’s homepage

(c) Voting Stage

This stage can only be completed by the voters. They are the only type of users who are eligible to cast the votes. In some of the E2E system, this stage is carried out by marking their chosen candidate(s) on the paper ballot, and then it needs to be scanned with the optical scanner or by using DRE machine to be submitted for the tally process. However in eVote, this stage is carried out by forming a secured ballot electronically and sending it to the election server where all the ballots would be collected and stored.

After completing the two stages mentioned above, voters can then logged on to the system and access the voting page. They can cast their vote by selecting their desired candidates for each category listen on that particular page. Besides that, the users could also review, reset and reselect candidates before they submit their vote. Voter’s ballot is generated every time the chosen candidates are reviewed.

During ballot generation, F5 Image Steganography Algorithm would be applied. F5 algorithm, as elaborated (pseudo-code) in Figure 5-17 is implemented to secure data communication in between of the voter and the election server, even before the ballot is cast. Voter’s chosen candidates would be encrypted in a stego-image format as a camouflage for their ballot. Attackers would not have any idea that the client is actually sending their ballot to the server in a jpeg file format image. This ballot will later on be sent over to the tally server. Once received by the server, the ballot package would be decrypted to reveal the candidate names hidden on
the ballot before then encrypted again with visual cryptography as an additional security level to earn voter’s direct trust by providing vote receipt. This ensures that voters’ votes have been collected as casted.

Figure 5-17: Pseudo-code of F5 Image Steganography algorithm applied in voting stage (Provos & Honeyman 2003)

The integration of both information security schemes is displayed in Figure 5-6 in a sequence diagram. Decrypted stego-image (ballot) would be encrypted with visual cryptography technique by splitting the vote into few shares. In this system the shares would be limited to two. Stand-alone share would not reveal any information to anyone, but once the shares are overlaid (combined) using a visual cryptography decryption algorithm, the voter's casted vote value would be revealed. Basically, each voter would be given one layer (share) of the image as their receipt which will be sent to their respective email account as displayed in Figure 5-18, while the other separated layer of the vote would be kept or saved by the administrator for ballot counting purpose as well as to discard the relation of each voter with their own ballot. Therefore, the voters would still be able to verify their votes and will have a better trust and confidence in the system. In eVote Voting System, the voting stage process is finished when the voting summary page is shown. The overall process flow of eVote’s voting stage is elaborated in Figure 5-19.
Figure 5-18: Screenshot of the voter’s vote receipt received by the voter

Figure 5-19: Process flow diagram of voting stage

(d) Tallying Stage

Tallying stage follows the voting stage. After votes are cast, ballot is securely stored in the database. Users cannot access the ballots before the completion of the tallying stage. The tally determined at this stage is obtained by polling officers with help from system administrators. Each polling officers holds a unique secret key to retrieve ballot records from the database. These keys are pre-distributed by the system administrators during the election setup. System
administrators generate these keys by utilizing UUID. Figure 5-20 elaborates this process in a sequence diagram.

Figure 5-20: Sequence diagram of secret keys distribution process by the system administrator

```java
if (today.before(endDate)) {
    request.setAttribute("electionEnded", false);
} else {
    request.setAttribute("electionEnded", true);
    if (distributed) {
        officerList = officerFacade.findAll();
        for (Officer of : officerList) {
            if (!of.getDecryptStatus()) {
                errorDecryptFlag = true;
            }
        }
        if (!errorDecryptFlag) {
            tallyList = tallyFacade.findAll();
            Random rand = new Random();
            Collections.shuffle(tallyList, rand);
            Map tallyMap = new HashMap();
            List<Candidate> candidateList = candidateFacade.findAll();
            for (Candidate c : candidateList) {
                tallyMap.put(c.getName(), 0);
            }
            for (Tally t : tallyList) {
                Iterator iter = tallyMap.entrySet().iterator();
                while (iter.hasNext()) {
                    Entry mEntry = (Entry) iter.next();
                    if (t.getCandidate().getName().equals(mEntry.getKey())) {
                        mEntry.setValue((Integer) mEntry.getValue() + 1);
                    }
                }
            }
            session.setAttribute("tallyMap", tallyMap);
            session.setAttribute("tallyList", tallyList);
        }
    }
    session.setAttribute("distributed", distributed);
}
```

Figure 5-21: Excerpted source code of tally result list generation
In order to access the tally list, polling officers must perform ‘decryption’ process by merging their secret keys. This method is called the threshold decryption cryptosystem (Shamir 1979). Only after each of the polling officers have submitted their secret keys, the tally result list (bulletin board) is accessible to the system administrators and the polling officers for monitoring, as shown in the source code snippet in Figure 5-21 above. This tally list is only readable and does not show any relation between the ballot and its voter. Threshold scheme is implemented in the ballots decryption process to ensure that only the authorized personnel can count the vote.

5.5 Summary and Conclusions

This chapter discussed the integration of eVote voting system’s components. eVote was implemented in two phases on different Java platforms. The first version, eVote v1.0 was developed in Java SE platform to examine the stability and reliability of applied cryptography and image steganography techniques in voting and vote verification stages. eVote v2.0 was developed as a web application, extending the functionalities of eVote v1.0 using services provided by Java EE platform. It is followed by discussion of supporting technologies of Java EE platform and development tools used in the development of eVote.

This chapter also covered the implementation review of each security schemes applied in eVote. These security schemes were previously described in detail in Chapter 2, Literature Review. Password-based scheme is used to protect users’ passwords from known security attacks in authentication process, data transmission in voting stage is secured through the implementation of F5 image steganography, Visual cryptography is used to fulfill individual-verifiability through vote verification feature, and threshold decryption scheme was applied to support clean tallying process. No one can alter the election results, even if the system administrators and polling officers are not trustworthy. Each scheme is applied in different stages of the election procedure to ensure a clean and fair election.
CHAPTER 6

SYSTEM TESTING AND CONSIDERATION OF SECURITY ATTACKS ON THE SYSTEM

6.0 Introduction

Even though Remote Electronic Voting was developed to increase the efficiency and effectiveness of the election, the reliability of the system must be accessed. This could be done by considering all the necessary requirements of an election. Therefore this chapter describes extensive test of the software to see if each part of the software performs as expected. The software testing consists of four parts – unit testing, integration testing, system testing and user acceptance testing. These are described in detail in this chapter.

Integrity is central to an election system. For this reason, security is most important. The security aspects are thoroughly examined in software testing. At the same time, the system should be user friendly. This feature is examined in user acceptance testing part of software testing through the use of questionnaires.

Finally, the four E2E voting systems discuss in Chapter 3 are compared with the electronic voting software proposed in this research, in terms of vulnerabilities to external threats.

6.1 Software Testing

This section evaluates the three main components that need to be ensured by Remote Electronic Voting System, namely effectiveness, efficiency and reliability. It is done by conducting software testing. In this software testing, there are 15 participants involved.

6.1.1 Testing Environment

Testing is done by using user’s local machine with minimum specifications listed below. The set of software and resources listed in Chapter 4; under section 4.2.4 Operating Environment and Dependencies must be installed and set up properly for usability testing.

6.1.2 Testing Participant

15 representative individuals from different demographic groups participated in eVote Voting System testing. They are recruited based on the consideration of few significant aspects, such as
computer literacy, some basic knowledge of information security, usability and voting age. The users are also chosen to meet the minimum age requirement by Malaysian law. The summary of participants’ demographic information is shown in Figures 6-1 and 6-2.

![Number of Participants Based on Age](image)

**Figure 6-1:** Number of participants based on their age

![Number of Participants Based on Gender and Highest Level of Education](image)

**Figure 6-2:** Number of participants based on their gender and highest level of education

### 6.1.3 Software Testing Procedure

The process of eVote’s software testing was conducted in two distinctive parts. At the initial stage, software testing was carried out by the system developer through the conduction of unit testing as well as integration testing. The testing was done extensively for every module to
ensure the system’s performance and stability. Secondly, the testing for this research project was performed by the participants. Each of them is required to complete three sets of tasks assigned to them and also to fill in a questionnaire in order to judge the eVote Voting System’s effectiveness, efficiency and reliability. Each participant spent not more than half an hour to complete all the tasks and to fill their feedback in the questionnaire provided. The sample of questionnaire is attached in Appendix 2. Based on this questionnaire, two types of software testing were carried out. They are system testing, which in this case is usability testing and user acceptance level based on the Technology Acceptance Model introduced by Davis in 1989.

The three sets of tasks that need to be executed by the participant are based on the three user levels of eVote - voter, polling officer and system administrators. They are listed below.

(a) **Voter:**
- Voting
- Vote verification
- Registration.

(b) **Polling Officer:**
- Ballot monitoring
- Secret key submission for tally result
- Authentication.

(c) **System administrators:**
- Election set up
- Voters management (add and delete voter and also editing voter profile)
- Key distribution for tally result

Each participant is only allocated one set of tasks out of the three sets. Thus, one set of tasks was completed by 5 participants each. They were assigned randomly to each of the set tasks. Participants’ responses and opinions from the questionnaires were analyzed to give preliminary result for the latter part of the overall tests - the system testing and user acceptance testing.

### 6.1.4 Software Testing Results and Discussion

Four distinct types of software testing were conducted in this research. Their results are discussed in detail as follows.

#### 6.1.4.1 Unit Testing

This test was performed by doing individual test on single units of the system rather than on the entire system (Link 2003). Those units are mostly represented in object-oriented classes. Components of these classes were then evaluated to ensure their functionalities are working
correctly. In this research project, a unit testing framework, the JUnit is used to support quick and easy generation of test suites to conduct unit testing as described in Chapter 5, System Implementation.

The unit testing for this research project was conducted to evaluate the four techniques of information security that were implemented in order to secure the system – Password Hashed-based Scheme, Visual Cryptography, F5 Image Steganography and Threshold Decryption System. The processes tested under those techniques are namely salt value generation, password encryption and decryption for Password Hashed-based Scheme, source image and shares generation as well as receipt verification for Visual Cryptography, secret keys distribution to support Threshold Decryption Cryptosystem and secret data encoding and decoding for F5 Image Steganography. Figures 6-3 to 6-7 illustrate the JUnit testing results of these processes respectively.

Figure 6-3: Screenshot of JUnit test cases implementation for Password Hashed-based scheme

Figure 6-4: Screenshot of JUnit test cases results for Password Hashed-based scheme
These figures clearly illustrate that all of the cryptography and steganography schemes have been properly implemented by eVote. This can be determined from reviewing the JUnit results shown on the figures. Other than that, the response time for each of these main processes could also be reviewed. The longest process that eVote needs to support is the stego-image decoding process which took slightly more than 2 seconds. The salt value and secret key distribution processes require the shortest amount of time to be completed. In order to check the accuracy of the codes written in Java classes, JUnit suites need to be run for a few times. In this case, it was run for three consecutive times as shown in Figure 6-4 and Figure 6-9. Salt values are associated to users’ passwords. Thus, the salt values generated in each of the test must be different to avert dictionary attack caused by the occurrence of identical passwords in the database. The results of these test cases were then reviewed. Its accuracy can only be ascertained if the results of these three test cases are different from one another as displayed in Figure 6-4 for salt value generation and in Figure 6-9 for the generation of secret keys.

Figure 6-5: Screenshot of JUnit test cases implementation for Visual Cryptography scheme

Figure 6-6: Screenshot of JUnit test cases results for Visual Cryptography scheme

Figure 6-5 and 6-6 elaborate the result of unit testing for the applied Visual Cryptography scheme. The shares generation process took slightly shorter time compared to stego-image decoding process in Figure 6-8. However it took much longer time compared to the stego-image encoding shown in Figure 6-7. When the encoding-decoding processes of both schemes are
evaluated, the overall process of Visual Cryptography implementation is much lengthier compared to F5 Image Steganography technique. Therefore, the votes’ transmission process in the voting stage is carried out by transmitting stego-image instead of visual cryptography share.

Figure 6-7: Screenshot of image steganography secret data encoding test case and its results

Figure 6-8: Screenshot of image steganography secret data decoding test case and its results
6.1.4.2 Integration Testing

Integration testing is a continuation of Unit Testing. This level of software testing focuses on the interaction of several components or units which have been tested previously in the unit testing (Link 2003). Each test case defines a business process of a particular system. While unit testing is connected to the system implementation phase of the implemented SDLC method; Iterative Waterfall Model, integration testing corresponds to its system design phase. By performing integration testing, the correctness of the business process could be evaluated. In this research project the system integration testing was only carried out for eVote’s primary business processes. They include modules that were initiated in the five stages of eVote Voting System described in Chapter 5. Those modules are the registration process for eligible voters and polling officers, user authentication process, voting process that can only be carried out by the valid voters, tally checking by polling officers and system administrators and lastly vote verification. The overall process of each stage has been described in detail in Chapter 5. Thus, this section will only focus on the end product of the processes illustrated in Figures 6-10 to 6-25. They are described as follows.

(a) Registration

The registration processes for eligible voters and polling officers are the same. Each voter is required to fill their required information in places provided on the signup page, as displayed in
The user input values will be validated by using Javascript on the client side as displayed in Figure 6-11. This eliminates the unnecessary communication process between the client machine and the system server in server-side validation.
The successfully registered polling officers would be directed into their respective homepage, as shown in Figure 6-12. The same registration process applied for the eligible voters.

**Figure 6-12:** Screenshot of voter homepage upon successful registration

(b) Authentication

This is the common process that type each of the users would have to go through. While system administrators are authenticated based on the pre-set access control done with the support of Glassfish application server, voters and polling officers are authenticated by using their self-defined username and password as shown in Figure 6-13.

**Figure 6-13:** Screenshot of polling officer login page
As stated before, voters’ and officers’ credentials are securely saved in the database through the implementation of Password Hashed-based Scheme that does not support password decryption. Therefore, the accuracy of this particular module corresponds to the JUnit test cases of Password Hashed-based Scheme previously completed in the unit testing.

![Screenshot of polling officer homepage upon successful authentication](image)

**Figure 6-14**: Screenshot of polling officer homepage upon successful authentication

### (c) Voting

Voting process integrates the two types of distinct information security test cases done in the unit testing. The two types are visual cryptography and F5 image steganography JUnit test cases. In order to support the integration of these two security schemes few important things must be completed. First, the voter needs to make their selection based on the list of candidates provided as shown in Figure 6-15. Second is ballot generation. In eVote voting system, ballot is generated as stego-image format. As stated before, it is to ensure secure data transmission between the voter and the server during vote casting. Once the user is satisfied with their ballot review, they can directly submit the ballot by clicking ‘Confirm’ as displayed in Figure 6.16. By doing this, the user allows the system to send his ballot securely to the server in stego-image format. Once the server have received this ballot, it will then decrypt the stego-image to retrieve the ballot. Note that, the ballot will not be saved neither in the database nor in the client machine to eliminate some possible security attacks. Besides that, the system server is also responsible to implement Visual Cryptography technique to generate vote receipt and its key. This key will be stored securely in the database for vote verification purpose. All of these
processes are included in as vote casting. Upon successful vote casting, the user will be notified of successful voting process and vote receipt generation as shown in Figure 6-17.

Figure 6-15: Screenshot of voting page accessible only for the voters

Figure 6-16: Screenshot of ballot confirmation page
To secure the tally result of the election, the Threshold Decryption Cryptosystem has been applied into the tallying process. In order to apply this technique, the secret key distribution and submission must be integrated. System administrators and the polling officers need to work together in its implementation. The system administrators first need to distribute the secret keys to the election officials once the election has ended. This feature is shown in Figure 6-18 and 6-19.

(d) Tally

To secure the tally result of the election, the Threshold Decryption Cryptosystem has been applied into the tallying process. In order to apply this technique, the secret key distribution and submission must be integrated. System administrators and the polling officers need to work together in its implementation. The system administrators first need to distribute the secret keys to the election officials once the election has ended. This feature is shown in Figure 6-18 and 6-19.
In order for system administrators and polling officers to access the tally result, polling officers need to submit the pre-distributed secret keys by system administrator. In other words, both parties are accountable to one another in this process. Thus, threats from the election officials could be averted. Figure 6-20 to 6-6-23 elaborate this process in system screenshots.
**Figure 6-21:** Screenshot of secret key submission in tally result page from polling officer’s screen

**Figure 6-22:** Screenshot tally result page upon successful secret key submission
(e) Vote Verification

Unlike the traditional voting system, eVote Voting System is E2E Verifiable through the vote receipt feature. The success of integration testing depends on the accuracy of vote receipt (visual cryptography shares) in the unit testing done in the previous level of the software testing. The screenshots of eVote’s vote verification process is shown in Figure 6-24 and 6-25.

**Figure 6-23:** Screenshot of tally result page

**Figure 6-24:** Screenshot of vote verification page for the voters
6.1.4.3 System Testing

For system testing in this research project, usability testing has been chosen. Usability testing considers users’ concerns over the system. This cannot be obtained by performing unit testing and integration testing which examine the system’s functionality and run-time errors. In usability analysis, a number of evaluators are presented with an interface design and asked to comment on it (Molich & Nielsen 1990). The comments on the questionnaire are then examined. The questionnaire consists of the following components.

- Visibility of system status
- Match between the system and the real world
- Consistency and standards
- Aesthetic and minimalist design
- User control and freedom
- Helps user recognize, diagnose and recover from errors
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency to use

From the observation of usability testing carried out by 15 participants, collections of data were obtained. Illustrated in Figures 6-26 to 6-34 are its derived results in a simple statistical analysis.
Figure 6-26: Evaluation of the Visibility of System Status

Figure 6-27: Evaluation of the Match between the System and the Real World

Figure 6-28: Evaluation of the Consistency and Standards
Figure 6-29: Evaluation of the Aesthetic and Minimalist Design

The graphs presented in Figure 6-26 to 6-29 illustrate users’ perspective regarding the accessibility and overall design of the system interface. Based on the three sets of task given to them, they evaluate the navigation process of eVote. Users give a positive feedback for their first experience to use eVote, as shown in Figure 6-29. They are able to understand the system and navigate through different processes in the three distinct user levels easily. The neat layout with simple, consistent and understandable menu arrangement is one of the factors that supported this, as derived from these four figures. Other than that, all the voters are assumed to have good computer literacy. Thus, the participants are only taken from this demographic group. In the future, another testing can be carried out to evaluate eVote’s system interface by involving another group of people with less computer literacy.

Figure 6-30: Evaluation of the User Control and Freedom
The other factor that needs be ensured by a system to support users’ accessibility is a proper error handling. It must be constructed properly to meet the user requirements. eVote provides a number of error-handling mechanisms. JavaScript handles some of them in the client-side, while others are supported in the server-side. The 15 participants have fairly satisfactory experience with its error-handling mechanism. Based on the comments gathered from the questionnaires, eVote does not prevent error well enough. This is as illustrated in Figure 6-32.

The last factor that the users evaluate is their effectiveness in using eVote for the second time. This is the main reason why they are asked to complete three different set of tasks from each user level of eVote. Due to its similar layout design and the straightforward functionalities it offered, users agree that eVote are able to provide efficient election procedures. The result of the observation made is displayed in Figure 6-33 and Figure 6.34 below.
6.1.4.4 User Acceptance Testing

This test was conducted in order to consider not only technical factors to support the system’s performance, but also to consider the behavioral factors of the users. In an attempt to understand user acceptance level of a system, the Technology Acceptance Model was applied in this research project (Davis 1989). Such a model is able to provide a robust indication of eVote’s user acceptance. Davis claimed that these are two distinct cognitive appraisals of users’ attitudes. They are the design features and the affective response to the system. In this testing, two types of the design features proposed by Davis were applied to examine the user acceptance...
level. Those design features include perceived usefulness (extrinsic motivator) and perceived ease of use (intrinsic motivator). Perceived usefulness is the degree to which an individual believes that a particular system would enhance his or her job performance. On the other hand, perceived ease of use is the degree to which a user believes that the use of a particular system would require less effort compared with another system. Figure 6-35 shows result collected based on the evaluation of perceive ease of use.

Based on the evaluation conducted, eVote’s perceive usefulness aspect is shown in Figure 36. The 15 participants mostly prefer to cast their vote by using a remote E2E voting system, compared to casting their votes in the polling booth provided in a traditional voting system. According to the participants, the implementation of vote receipt is more reliable and offers more assurance to them, compared to the implementation of indelible ink commonly practiced in the traditional voting system. There are many ways counterfeit votes can be cast using indelible ink due to the involvement of many parties in its implementation. However, the implementation of vote receipt only requires the involvement of system administrators and polling officers who are assumed to be trustworthy. Thus, it can be concluded that the perceived ease of use affects the outcome of the perceived usefulness evaluation. Perceived usefulness itself is the core factor of user acceptance level over the system. From the result displayed in Figure 6-36, a decent result of eVote’s perceived usefulness aspect could be predicted.

**Figure 6-35: eVote Perceived Ease of Use Evaluation**
User acceptance level could also be identified by evaluating the user requirements gathered in the requirements phase of the software development. Each requirement was grouped based on their characteristics and analyzed as follows.

(i) Mobility and Convenience

This is one of the aspects that most E2E system could not provide. Many of them use one or more election technologies that forced the users to vote in a particular venue. eVote offers secure remote electronic voting system with vote verification feature which allows voters to vote anytime and anywhere, provided their PC is connected to the same LAN as eVote’s server. The current development of eVote limited its mobility due to the limited time and funds to host the system in the internet. Further development of eVote may extend this requirement, where voters can vote anytime and anywhere as long as internet connection is available.

(ii) Completeness

In a traditional voting procedure this requirement is simply catered by comparing the number of voters and the counted votes. This is merely time consuming and the chances of human-error are high. On the other hand, E2E system could complete this task within seconds. Polling officers and system administrators are responsible for monitoring the ballots and the overall tallying process, including the tally result. However, vote coercion can still occur. eVote voting system reduces this by eliminating direct access to the database. Users can only access the database...
server from the system. It ensures that the completeness of eVote voting system would not be violated.

(iii) Eligibility and Unreusability

These requirements are connected to one another as unreusability is a subject to voters’ eligibility. Only those voters who have been registered and authenticated through their unique information and eligibility status can access the system. They will be given only one chance to cast their votes.

(iv) Privacy, Soundness, Verifiability and Robustness

The deliverance of these three aspects is ensured in eVote voting system’s design by implementing image steganography. Vote transmission during voting process in eVote is secured beforehand by encoding the cast ballot with F5 image steganography algorithm. As the privacy of the votes could be guaranteed, the soundness and verifiability could be carried out as well because no one can obtain any information on the votes except the voters themselves.

(v) Incoercibility and Receipt-freeness

As visual cryptography receipt is included in eVote’s system architecture, these two requirements are supported. It is used in conjunction with Visual Cryptography in order to ensure the integrity of the election and to prevent improper influence in the election process. The receipt is secure and in the format of Visual Cryptography Image (share). Untrustworthy voters are not able to prove to a coercer that he has voted in a particular way. Thus, incoercibility and receipt-freeness are supported in eVote.

(vi) Fairness

In tallying stage, Threshold Decryption Scheme was applied to keep the vote secure. The read-only tally result is only accessible polling officers and system administrators after the election has ended and the distributed secret receipts have been submitted by all of the polling officers to the system’s server. Because of this no one can gain any information regarding the tally result before the election ends.

6.2 Common Security Attacks on E2E Voting Systems

In this research project only some of attacks reported for E2E voting systems will be considered. Some of these attacks do not required skills, knowledge and access over the system as a whole, whilst for some other attacks they are required. The attacks are divided into two parts - internal and external sources (Regenscheid & Hasting 2008).
6.2.1 Internal Sources

Attacks or threats by individual or groups who have access to the voting system equipment or its processes such as network communication are classified as attacks from internal sources (Regenscheid & Hasting 2008). Such attacks include unintentional and deliberate attacks (Pardue, Yasinsac & Landry 2010) that threaten electronic electoral integrity. Unintentional attacks accidentally occur due to poor system design and architecture or risks mitigation plan. The parties who are hold responsible for these attacks are the individuals whom have contributed in the development process of the system. Deliberate attacks on the other hand are often done by untrustworthy users or other insiders (Regenscheid & Hasting 2008), such as:

(a) Valid Voters

Even with limited level of access to the voting system’s resources, valid voters are still capable of carrying out some attacks on the system. Untrustworthy voters could misuse their authorized right of casting a vote by submitting invalid ballots into the system which will lead to undervotes, overvotes or even phantom votes.

(b) Polling Officers

Polling officers are entitled to higher authorization access to the system compared with the voters and lower authorization access compared to the system administrators. In general, polling officers are allowed to access and monitor voters’ and ballots’ information as well as the finalized data of the tally result. Thus, they might take inappropriate actions such as announcing collected result before the election ends.

(c) System administrators

Voting System’s administrators have the highest authorization level. They are responsible for election (software) installation to manage and monitor the overall users and processes of the election. Besides, they must also make sure that the whole operations of the election must be running properly without any disruption. Due to their high authorization level, actions of untrustworthy system administrators can be catastrophic to the system. Most of the voting systems assume that their administrators are trustworthy. The same goes to eVote Voting System.

(d) Other insiders

The other insiders who might be a source of threat to the Voting System are those individuals or organizations who have access to the resources of the Voting System before, during or after the election. They might include the Voting System manufacturers, integrators or its support staff.
6.2.2 External Sources

Threats from external sources includes threats or attacks by individual or groups with no particular level of authorized access to the voting system equipment or the supporting infrastructure over the system itself (Regenscheid & Hasting 2008). Even without special access and privileges, external sources are still capable of facilitating malicious attacks. The possible risks are even higher for open-source electronic voting system as anyone with a certain technical and skills could analyses the whole election’s procedure of an electronic voting system and determine its weaknesses and flaws. Once they are obtained, attackers could easily hack into the system mostly to manipulate the tally and disrupt the election processes including interfering data communication between the voters and the system’s server. Some other attacks on E2E voting systems that have been reported include randomization attacks, forced-abstention attack, simulation attack, denial of service (DoS) attack, etc. They are discussed in more detail as follows.

(a) Randomization Attacks

Schoenmakers detected this attack over Hirt and Sako scheme back in 2004 (Juels, Catalano & Jakobsson 2005). Its intention is to nullify the voter’s vote by submitting as many forged ballots (randomly composed ballot material) as possible. Even if the attackers are not able to obtain the votes cast of the voters, they have coerced the voters’ votes successfully. The integrity of complete paper-based ballot E2E system is jeopardized by its physical ballots. These ballots are susceptible from randomization attacks as attackers could easily replace the authorized ballots with the forged ones.

(b) Simulation Attack

The E2E verifiable voting system must ensure the prevention of coercion and vote’s verifiability, by offering receipt-freeness. However, it actually does not eliminate the possibility of coercion to occur during the election period. Dishonest voters could sell their rights to vote by giving their private keys to the attacker who would cast the vote on their behalf. This is called the simulation attack. Simulation attack often threatens those systems without polling officers’ supervision and assistance. This kind of attack mostly happens to remote electronic voting systems. Thus, eVote Voting System could be a subject to such attack if the voters cast their vote without the election officials’ supervision. Therefore in Chapter 4, Secure E-voting analysis and Design, an assumption is made that the voters are trustworthy and will not cheat during the election even without the supervision of polling officer.
(c) **Forced-abstention Attack**

Here, the attackers would attack the system by forcing the voters to refrain from casting their votes. By using the bulletin board as its reference list, attackers are able to identify the voters who are yet to cast their ballots and coerce them. This attack could be eliminated by restricting the bulletin board. Only eligible voters and other authorized election personnel are allowed to access and retrieve information from the bulletin board. This attack occurs due to the system’s verifiability property. Each of the E2E system must have a standard version of bulletin board to publish the voter’s ballots in order to ensure the accuracy of the centralized ballot box. By supporting such feature, forced-abstention attack could take place. To eliminate or to decrease the possibility of its occurrence, the system’s developers must restrict the visibility of the bulletin board. It should not be open to public, but only to eligible voters and the authenticated election officials. eVote offers this feature where only system administrators and polling officers are allowed to access the bulletin board (read-only). The voters can verify their vote through the vote verification feature.

(d) **DoS Attack**

Denial of Service attack is one type of cyber-attacks that focused on disrupting availability (Denial of Service / Distributed Denial of Service Managing DoS Attacks). It varies from physical IT environment destruction to make use of the system’s flaws and overload the network capacity (Denial of Service / Distributed Denial of Service Managing DoS Attacks). DoS attacks arise in all E2E voting systems. It is rather difficult to prevent such attack; however they are easy to be detected. A very strong risks mitigation plan could help to avert the occurrence of DoS attacks in E2E voting systems. Without a very strong defense mechanisms and risks mitigation plan as well as the proper implementation of software design and system requirements deliverance, it would be very difficult for the E2E voting systems to prevent or to control this attack.

### 6.3 Comparison of E2E Voting Systems

In Chapter 3, Studies of Related Voting Systems, four different types of E2E voting systems are discussed. The author has also explained in detail each of the principle of components and also their system overview. In this sub-section, those E2E voting systems are compared based on two aspects to conclude the studies of those related E2E voting systems. This sub-section also includes a discussion of eVote’s weaknesses. All of them are elaborated as follows.
6.3.1 Systems’ Components

The discussed electronic voting systems are supported by distinct types of components. They vary based on the functionality and purpose for those particular systems. All aim the same thing, to support the main functionality that the system intends to offer to the user. As mentioned before the main objective (functionality) of an E2E voting system is to provide a full End-to-end auditable election experience to the user. Thus, all of the systems from the four systems taken as a case study support universal as well as individual verifiability. Most of the differences as shown in Table 6-1 lay on their offered feature. Most of the systems do not offer one of the essential requirements of electronic voting system, especially at the current time with the growth of internet application, which is mobility. Other than that, most of the systems are also dependent to one or more external hardware. Not only costly for the deployment phase, such dependencies are also costly for further maintenance. All of these can be eliminated by implementing an internet-based E2E voting system, such as Helios and eVote.

Table 6-1: Comparison of E2E Voting Systems’ Components

<table>
<thead>
<tr>
<th></th>
<th>Helios</th>
<th>Scantegrity</th>
<th>Prêt a Voter</th>
<th>RIES</th>
<th>eVote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Verifiability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Individual Verifiability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobility</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>External Hardware</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Main Implemented Technology</td>
<td>Mix-net Scheme and Threshold Decryption</td>
<td>Optical Scan and Anonymity Network</td>
<td>Visual Cryptography and Chaum's Secret-ballot Receipt</td>
<td>Cryptographic Hash Function</td>
<td>F5 Image Steganography and Visual Cryptography</td>
</tr>
<tr>
<td>Bulk Registration</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Administrator</td>
<td>Yes</td>
<td>Yes (referred as Scantegrity team)</td>
<td>Yes (referred as election authorities</td>
<td>Yes (referred as TTPI)</td>
<td>Yes</td>
</tr>
<tr>
<td>Voter</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (including the election board)</td>
<td>Yes</td>
</tr>
<tr>
<td>Election Trustees or Officer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (referred as auditors and help organizations)</td>
<td>N/A</td>
<td>Yes</td>
</tr>
</tbody>
</table>
6.3.2 Systems’ Vulnerability

Based on the potential internal and external sources of threats the vulnerability of four different E2E voting systems are compared with the vulnerability of eVote. The four E2E Voting Systems were discussed in detail in Chapter 3. They have been applied in medium to large scale real-world elections, namely Scantegrity II for Takoma Park Municipal Election, RIES public election in Netherland, the implementation of Helios voting system at Recteur of Universite Catholique de Louvain and lastly student council election at Princeton University that make use of Prêt à Voter system (Carback, et. al. 2010) One of the common assumptions made by the voting system’s designers or developers is to assume that the system users including voters, polling officers, system administrators, etc. are all trustworthy.

Results presented in Table 6-2 conclude that generally E2E systems are not resistant to two external threats attacks used for the comparison purpose. They are forced-abstention attack and DoS attack. However, eVote can successfully avert forced-abstention attacks by restricting the bulletin board access only to administrators and polling officers. Both individuals are assumed to be trustworthy. eVote also eliminates randomization attack through the utilization of secured computerized ballots. Here, we assumed that the only individual who is admitted to maintain the system is the system developer, who is the author herself. Thus, no one has access to eVote’s source code and they cannot recreate the exact ballot used by the voters. Hence, randomization attack could be avoided. The data shown in Table 6-2 also indicates the capability of eVote to avert more threats compared to the other four E2E systems. This also proves the fitness of the implemented approach in eVote system.

Table 6-2: Comparison of E2E Voting Systems based on its defense mechanism against common potential threats

<table>
<thead>
<tr>
<th>External Threat Sources</th>
<th>Helios</th>
<th>Scantegrity II</th>
<th>PV</th>
<th>RIES</th>
<th>eVote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomization Attack</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Simulation Attack</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Forced-abstention Attack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DoS Attack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
6.3.3 eVote’s Weaknesses

As derived from the series of testing’s and users feedback, the author concluded three major weaknesses of eVote electronic voting system. They are listed and discussed as below.

(a) Increase the accessibility of eVote by deploying the system on an internet web server.

The most apparent weakness of eVote is the inability of users to access the system outside the Local Area Network (LAN). With the growth of Information Technologies, internet is the main communication channel for most systems on the market. Using an internet hosting server and a domain name allows users to access the system even when they are not in the same LAN.

(b) Optimize the security approaches that have been implemented in eVote; either in algorithm or code manner.

Security vulnerabilities could compromise the integrity of eVote voting results. Hence thorough testing has to be done on every module of the system. Codes and algorithms should be refined and simplified to provide a strong security layer on the system.

(c) Extend its platform to other types of software application.

To further meet the need of users, eVote must be made compatible for access from smart phones, tablets and other handheld devices. The display content must be formatted for viewing on mobile devices without compromising on its functionality and security.

6.4 Summary and Conclusions

In this chapter, distinct types of software testing were carried out to evaluate the effectiveness of eVote v2.0 in supporting a clean and fair election. They are unit testing, integration testing, system testing and user acceptance testing. The developer conducted unit testing and integration testing to detect and correct errors found in the systems. She evaluates all features of the system to make sure each of them is properly developed and error-free. eVote voting system not only emphasizes on the security of a voting system. It also ensures the quality of system performance and usability by conducting system testing and user-acceptance testing. Testing involved 15 participants from different demographic groups. Based on the survey and observation done, the user requirements gathered in Chapter 4 is evaluated to identify the performance and usability of the system. As result, 80% of the participants prefer to cast their votes by using eVote.

This chapter is concluded with review of common security attacks faced by E2E voting system. From this review, eVote v2.0’s vulnerability was compared with the E2E voting systems described in Chapter 3, Studies of Related E-voting Systems. eVote, as most other voting systems, are threatened by DoS attack. However, eVote successfully averted one of the common
attacks faced by the other E2E systems called Forced-abstention attack through dissimilar mechanism of its bulletin board.
CHAPTER 7

SUMMARY OF CONTRIBUTIONS AND FUTURE WORK

7.0 Introduction

In this thesis, the overall research project is concluded by reviewing the research objectives, summarizing the main scientific contribution of this research work to the development of E2E verifiable voting system and lastly, discussing directions for future work of this research.

7.1 Research Objectives and Results

This thesis has studied, designed, implemented and tested a remote electronic voting system called eVote. It is equipped with a few selected cryptography and steganography techniques. The aim of this research is to design and develop an effective, efficient and reliable voting system in an electoral procedure. It can be achieved by fulfilling two main research objectives identified in Chapter 1, Significance and Scope of the Thesis as follows:

(a) To improve the quality of election procedure in an electronic voting system based on its security and usability aspects by incorporating cryptography and image steganography in the system architecture.

The implementation of cryptography and steganography effectively secures the system from known attacks. However, it must not affect the system usability. It is one of the major user requirements that system must achieved. eVote voting system ensures this by conducting two types of testing, system testing (usability testing) and user acceptance testing.

(b) To minimize the computational and performance cost required in electronic voting system.

eVote reduces the need of external hardware, such as printer, scanner and DRE. Its primary device is a client PC, which is connected to the same LAN as its server. Besides that, eVote implements Visual Cryptography scheme in its receipt mechanism. Its implementation has low computation and performance cost. Thus, the overall election cost can be reduced.
This research project focuses on the development of remote electronic voting system in conjunction with E2E voter verifiability voting system. Incoercibility is believed to be the primary aspects that a voting system should possess. eVote is developed based on Password hashed-based Scheme, Visual Cryptography as adopted from Secret-ballot receipt scheme, Threshold Decryption Cryptosystem and F5 Image Steganography for security which do not have high computational cost as stated before. These cryptography and steganography methods are believed to be able to support the implementation of a voting system that satisfies the main objectives of this research project. Securing vote transmission at the end of the voting stage was accomplished with F5 image steganography technique. Voter and Universal verifiability feature to its intended users was offered through the application of vote receipt and bulletin board. Here, the validation of votes should be computationally feasible for anyone, without the requirement of external hardware to carry out the validation process. The research objectives was implemented after by gathering users’ requirements, reviewing related works of remote electronic voting system as well as conducting software testing. This way, software developer will get clearer insight from the users’ perspectives, which should be taken into considerations for system optimization.

eVote voting system was designed using Iterative Waterfall model. It is implemented utilizing Java EE 6 framework due to its APIs that could be used to support the construction of this voting system. The business process execution of eVote voting system is supported by the integration of Java EE 6 platform, Glassfish Application Server and MySQL Database Server on the server PC. These assisting third-party servers must be properly installed and set up beforehand. eVote is run locally (localhost) on the client server and must be interconnected to the system server through the LAN. It is not hosted on the internet. The implemented system is therefore only support partial mobility and suitable for small to medium scale elections in organizations.

Software testing with four levels of testing methods was conducted. The four testing methods are unit testing, integration testing, system testing (which in this case was performed by completing a usability testing) and finally, user acceptance testing. Based on the observation, some results were gathered successfully from both technical and behavioral factors. eVote’s robustness and performance could be evaluated technically by reviewing the results gathered from the unit testing as well as integration testing. On the behavioral side, one of the results gathered was the supporting outcomes of electronic voting system’s research and development, where most of the voters prefer to cast their vote through the voting system rather than submitting their vote manually to the ballot box at the polling station. The other result collected shows that, most of the participants in user acceptance testing favor digital vote receipt in visual
cryptography image format compare to the vote receipt in the ciphertext format. This is described in detail in the next section, *Research Contributions*.

### 7.2 Research Contributions

The development of eVote system addresses the common problems that arose in the traditional voting procedure. In such electoral system, voters are obligated to trust the result of the election as it is. They do not have the option to verify the authenticity of their cast ballot. Here, the election officials must be trustworthy. However, in most of the cases, their credibility is doubtful. As a result of this, with the evolution of information technologies, researchers have proposed implementation of electronic voting system as a feasible solution; E2E verifiable voting system. This voting system is classified into two types based on their ballot mechanism, namely Paper-based E2E Voting Systems and Electronic E2E Voting Systems. As a receipt-based voting system, both voting systems allow their voters to verify the accuracy of their vote to ensure that their votes have been collected as cast and counted as collected by using the vote receipt. Thus, incoercibility is offered in such voting systems. This however, creates other vulnerabilities to the system in paper-based E2E voting system. They are caused by the dependency over few aspects, for example possible errors made from the integration of the voting system with an additional external hardware (e.g. DRE machine) and also human-errors during the vote tabulation. On the other hand, even though the implementation of vote receipt in the Electronic E2E Voting Systems eliminates this problem. The vote receipt feature offered must be practical and convenient. It must meet the basic user-requirements. However, these aspects are often overlooked. Both of these issues were addressed in eVote voting system development.

The main contribution of this research project lies in the simplicity and user-friendliness it offers without compromising system security and usability. They are described as follows.

(a) **Eliminate the use of external devices in the implementation of E2E voting system.**

Instead of using some external hardware, such as printer, scanner and DRE, normal PC was used as the primary device. It will not only speed up the overall time needed to complete an electoral procedure but also reduce a number of significant expenses for device procurement, installation, training, upgrade and maintenance.

(b) **Provide improved vote receipt mechanism of E2E voting system.**

The vote receipt in eVote was presented in visual cryptography image format. This type of vote receipt format is more practical and convenient compared to the vote receipt in ciphertext format. This solution has no effect on the system security. This system has been designed and developed to provide a more secure voting system compared to the other E2E verifiable voting systems.
systems, be the paper-based or electronic E2E voting systems, by providing precise, yet secure data communication and also prevention over threats and attacks by the intruders.

7.3 Future Work

The results of this thesis point to several improvements for future work listed below.

- The implementation of the biometric device, like fingerprint recognition device to support enhancement of user authentication in order to provide improved security.
- To host eVote voting system on the internet to facilitate mobility for the users. Thus, eVote users need not be restricted to access the system from within the LAN.
- To provide larger set of participants in future software testing to obtain more reliable statistical analysis.

Lastly, with the constant advancement of information technologies, more optimized information security techniques could be introduced. The study and development of E2E verifiable voting system should further optimize remote E2E verifiable voting system.


Ambler, SJ & Sadalage, PJ 2006, Refactoring Databases: Evolutionary Database Design (paperback) (Addison-Wesley Signature Series (Fowler)), 1st edn, Addison-Wesley Professional.


Salted Password Hashing, crackstation.net, viewed 27 May 2013, <http://crackstation.net/hashing-security.htm#javasourcecode>.


APPENDICES

Appendix A: Functional Requirements Tables

**Table A-1: System authorization**

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User Access Level</td>
<td>The three eligible types of users can only access the system at their respective level.</td>
<td>High</td>
</tr>
</tbody>
</table>

**Table A-2: System authentication**

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Eligible Users</td>
<td>Users’ eligibility is to be re-examined during registration process. The validity of this process depends on their status.</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>User Login</td>
<td>The system must authenticate users upon login at different levels of the system. Only registered users (except system administrators) are allowed to log into the system levels according to their category.</td>
<td>High</td>
</tr>
</tbody>
</table>
### Table A-3: Applied technologies

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Windows Operating System</td>
<td>The software shall run on Windows Operating System</td>
<td>High</td>
</tr>
<tr>
<td>5</td>
<td>Java Virtual Machine</td>
<td>The virtual machine shall support the execution of software</td>
<td>High</td>
</tr>
<tr>
<td>6</td>
<td>Network</td>
<td>The system must be connected to a reliable network connection</td>
<td>High</td>
</tr>
<tr>
<td>7</td>
<td>J2EE</td>
<td>The software shall be programmed in J2EE programming language.</td>
<td>High</td>
</tr>
<tr>
<td>8</td>
<td>NetBeans IDE</td>
<td>NetBeans will act as the Integrated Development Environment.</td>
<td>High</td>
</tr>
<tr>
<td>9</td>
<td>Javadoc</td>
<td>Javadoc will be used for code documentation purposes.</td>
<td>Medium</td>
</tr>
<tr>
<td>10</td>
<td>MySQL</td>
<td>Database management system will be handled by using MySQL.</td>
<td>High</td>
</tr>
<tr>
<td>11</td>
<td>Glassfish</td>
<td>Application server shall be supported by glassfish server.</td>
<td>High</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>User Manual</td>
<td>A user manual shall be prepared to improve user’s familiarity with the system, its features and different components. It should also provide help on how to use the system functions.</td>
<td>High</td>
</tr>
<tr>
<td>13</td>
<td>Installation guide</td>
<td>An installation guide shall be created to assist the user in installing the software properly.</td>
<td>High</td>
</tr>
<tr>
<td>14</td>
<td>Code Commenting</td>
<td>All source code shall be commented to a professional standard, sufficient to permit maintenance by developers other than its original developer.</td>
<td>High</td>
</tr>
<tr>
<td>15</td>
<td>Javadoc</td>
<td>Every Java source file shall contain meaningful java annotations for every class, interface, method and public variable</td>
<td>High</td>
</tr>
<tr>
<td>16</td>
<td>System Design</td>
<td>The design of the system shall be documented.</td>
<td>High</td>
</tr>
<tr>
<td>17</td>
<td>Error Message Explanation</td>
<td>There shall be a set of explanations of error messages. It shall contain an explanation of the meaning of each error message that even an intelligent but inexperienced user should be able deduce the full meaning of the error. Where appropriate, each explanation shall also describe how to correct or otherwise respond to the message, and (if possible) identify the system component in which the error originated.</td>
<td>Medium</td>
</tr>
</tbody>
</table>
### Table A-5: Data type

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Alphabetic Characters</td>
<td>Software shall be able to process the non-numeric data from sources of input which are in form of English alphabet letters and the following characters: %, @, ^, *, ),(@,!, comma(,) and dot (.)</td>
<td>High</td>
</tr>
</tbody>
</table>
| 19 | Numbers                      | System shall be able to process the following type of numbers:  
**Natural Numbers:**  
From 0 to 2147483647 | High     |
| 20 | Date                         | Software reads, writes and accepts dates of the Gregorian calendar in dd/mm/yyyy numeric format with fore slashes separating the day, month and year numbers.  
**Day number:** Natural Number from 1 to 31  
**Month Number:** Natural Number from 1 to 12  
**Year Number:** Natural Number from 2013 to 2099  
Software will not be responsible for any incorrect date entry nor is it responsible for correcting them. | High     |
<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>PNG file input</td>
<td>There shall be an interface that imports PNG formatted files as data input source.</td>
<td>High</td>
</tr>
<tr>
<td>22</td>
<td>PNG file output</td>
<td>There shall be an interface that exports the results in PNG format.</td>
<td>High</td>
</tr>
<tr>
<td>23</td>
<td>JPEG file input</td>
<td>The system must be able to retrieve JPEG formatted files as data input source.</td>
<td>High</td>
</tr>
<tr>
<td>34</td>
<td>JPEG file output</td>
<td>The system must be able to export JPEG formatted files as data output source.</td>
<td>High</td>
</tr>
</tbody>
</table>
## Appendix B: Non-functional Requirements Tables

### Table B-1: System security

<table>
<thead>
<tr>
<th>ID</th>
<th>E-voting Requirements Type</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Eligibility</td>
<td>Eligible Users</td>
<td>Users’ eligibility is to be examined on registration process. The examination depends on the category of the user status.</td>
<td>High</td>
</tr>
<tr>
<td>26</td>
<td>User Login</td>
<td>User Login</td>
<td>The system must authenticate any user upon login at different level of the system. Only eligible and registered users are allowed to enter the system.</td>
<td>High</td>
</tr>
<tr>
<td>27</td>
<td>Privacy</td>
<td>Vote Secrecy</td>
<td>All the ballots cast must be transmitted and kept secret</td>
<td>High</td>
</tr>
<tr>
<td>28</td>
<td>Unreusability</td>
<td>Users’ Voting Status</td>
<td>Each eligible voter can only vote once</td>
<td>High</td>
</tr>
<tr>
<td>29</td>
<td>Fairness</td>
<td>Database Security</td>
<td>No one can access the database and indicate the tally before all collected votes are counted</td>
<td>High</td>
</tr>
<tr>
<td>30</td>
<td>Verifiability</td>
<td>Integrity of the Bulletin Board</td>
<td>The result of the election displayed on the bulletin board cannot be amended by any one</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Robustness</td>
<td>System Defense against Randomization Attack</td>
<td>The system must be protected against randomization attack</td>
<td>High</td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>System Defense against DoS Attack</td>
<td>The system must be protected against DoS attacks</td>
<td>High</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>Ballot Generation</td>
<td>All votes cast must be correctly generated before vote submission. Phantom votes are to be prevented.</td>
<td>High</td>
</tr>
<tr>
<td>34</td>
<td>Incoercibility and Receipt-freeness</td>
<td>Vote Receipt Mechanism</td>
<td>A voter can neither obtain any secret information of their ballot nor be able to construct a vote receipt to prove the content of their cast ballot to a third party at any moment</td>
<td>High</td>
</tr>
</tbody>
</table>
### Table B-2: System usability

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>Default Language</td>
<td>The system shall have English as its default language. Every element of the user interface shall be present in English</td>
<td>High</td>
</tr>
<tr>
<td>36</td>
<td>Mobility</td>
<td>There must be no restrictions on the location from where a voter can cast a vote</td>
<td>High</td>
</tr>
</tbody>
</table>

### Table B-3: System performance

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>Convenience</td>
<td>System must allow voters to cast their votes quickly, in one session, and with minimal equipment or special skills</td>
<td>High</td>
</tr>
</tbody>
</table>

### Table B-4: System reliability

<table>
<thead>
<tr>
<th>ID</th>
<th>Summary</th>
<th>Definition</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>Completeness</td>
<td>System must count all the valid votes correctly</td>
<td>High</td>
</tr>
<tr>
<td>39</td>
<td>Soundness</td>
<td>Data transmission for any processes of the system must be secure, such that even dishonest voters and polling officers cannot disrupt voting</td>
<td>High</td>
</tr>
</tbody>
</table>
Table C-1: eVote’s database design

<table>
<thead>
<tr>
<th>Table</th>
<th>Field</th>
<th>Data Type</th>
<th>Constraint</th>
<th>Nullity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Election</td>
<td>Name</td>
<td>VARCHAR(45)</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>start_date</td>
<td>TIMESTAMP()</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>end_date</td>
<td>TIMESTAMP()</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>distribute_keys</td>
<td>BIT</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>Category</td>
<td>id</td>
<td>TINYINT</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>VARCHAR(45)</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>Candidate</td>
<td>id</td>
<td>TINYINT</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>VARCHAR(45)</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>description</td>
<td>VARCHAR(150)</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>category_id</td>
<td>TINYINT</td>
<td>FK</td>
<td>NO</td>
</tr>
<tr>
<td>Voter</td>
<td>id</td>
<td>TINYINT</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>VARCHAR(45)</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>username</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pwd</td>
<td>VARCHAR(200)</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>salt_pwd</td>
<td>BLOB</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>icPassport</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>email</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>address</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>phone</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>officer_id</td>
<td>TINYINT</td>
<td>FK</td>
<td>NO</td>
</tr>
<tr>
<td>Officer</td>
<td>id</td>
<td>TINYINT</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>Field</td>
<td>Data Type</td>
<td>Constraint</td>
<td>Nullity</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>---------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>Officer</td>
<td>username</td>
<td>VARCHAR(45)</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>pwd</td>
<td>VARCHAR(200)</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>salt_pwd</td>
<td>BLOB</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>secret_key</td>
<td>TINYINT</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>icPassport</td>
<td>VARCHAR(45)</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>email</td>
<td>VARCHAR(45)</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>address</td>
<td>VARCHAR(45)</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>phone</td>
<td>VARCHAR(45)</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>decrypt_status</td>
<td>BIT</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>Ballot</td>
<td>id</td>
<td>TINYINT</td>
<td>PK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>img_key</td>
<td>BLOB</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>img_dec</td>
<td>BLOB</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>date_created</td>
<td>TIMESTAMP()</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>voter_id</td>
<td>TINYINT</td>
<td>FK</td>
<td>NO</td>
</tr>
<tr>
<td>Tally</td>
<td>ballot_id</td>
<td>TINYINT</td>
<td>PK &amp; FK</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>candidate_id</td>
<td>TINYINT</td>
<td>PK &amp; FK</td>
<td>NO</td>
</tr>
</tbody>
</table>
Appendix D: Activity Diagrams

Figure D-1: Activity Diagram of Election Setup process
**Figure D-2:** Activity Diagram of Add Eligible Voter or Officer Process
Administrator / Officer

System

Access voter status page

Retrieve voter’s data from the database

List of voters whom had cast their vote shown

Check their ballot status

Figure D-3: Activity Diagram of Ballot Monitoring process.
Click distribute secret keys

Generate random numbers for each of the officers

Send and notify each officer

Decrypt the keys collected

Enable access to tally result

Tally result displayed

Retrieve and insert the secret key

Tally result displayed

Figure D-4: Activity Diagram of Tally Checking process.
Figure D-5: Activity Diagram of Profile Editing process.
Figure D-6: Activity Diagram of Registration process.
Figure D-7: Activity Diagram of Authentication process
Figure D-8: Activity Diagram of Voting process
**Figure D-9: Activity Diagram of Vote Verification process**
### Appendix E: Use Case Descriptions

**Table E-1:** Use Case description of election setup process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Election Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>System Administrators</td>
</tr>
</tbody>
</table>
| Flow of Events | 1. Navigate to election setup page.  
2. Fill in election details (election name, start date and end date, as well as category and candidates information)  
3. Add in officers details (name, IC/Passport No and email address).  
4. Add in voters details (name, IC/Passport No and email address).  
5. The system returns successful message. |
| Pre-Conditions | Administrators have been successfully authenticated and can access the admin homepage |
| Post-Conditions | Administrators have successfully set up an election |

**Table E-2:** Use Case description of add eligible voter or officer process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Add Eligible Voter or Officer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>System Administrators</td>
</tr>
</tbody>
</table>
| Flow of Events | 1. View Eligible Voters or Officers List.  
2. Add the desired voter or officer by clicking add button on the top right of the page.  
3. Input voter or officer information accordingly.  
4. Save changes to the database by clicking add. |
| Pre-Conditions | Administrators have been successfully authenticated and can access the admin homepage |
| Post-Conditions | Updated list shown based on the changes made by the administrators |
**Table E-3:** Use Case description of ballot monitoring process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Ballot Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Participating Actor(s)</strong></td>
<td>System Administrators, Polling Officer</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. Navigate to voter status page.  
2. List of voters who had successfully cast their vote is shown.  
3. Check each of their ballot status clicking their respective name. |
| **Pre-Conditions** | System administrators have been successfully authenticated and can access the admin or officer homepage |
| **Post-Conditions** | Both administrators and officer can check the ballot status of each voters |

**Table E-4:** Use Case description of tally checking process for system administrator

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Tally Checking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Participating Actor(s)</strong></td>
<td>System Administrators</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. Administrators distribute the secret keys to all of the officers for security purpose.  
2. Administrators are notified once all the officers have inserted their respective keys.  
3. Tally results are accessed by the system administrators (read-only). |
| **Pre-Conditions** | System administrators have been successfully authenticated and have access to the admin homepage and election has ended |
| **Post-Conditions** | Tally result are accessible by the administrators (read-only) |
### Table E-5: Use Case description of tally checking process for polling officer

<table>
<thead>
<tr>
<th><strong>Use Case Name</strong></th>
<th>Tally Checking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Participating Actor(s)</strong></td>
<td>Polling Officer</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. Retrieve each of their secret keys distributed by the administrators.  
2. Insert their respective secret key.  
3. Officers would be able to access the tally result once all the officers have inserted their respective keys. |
| **Pre-Conditions** | Polling officer have been successfully authenticated and have access to the homepage with all the distributed keys by administrators submitted and inserted by each officer |
| **Post-Conditions** | Tally result are accessible for the polling officer (read-only) |

### Table E-6: Use Case description of profile editing process

<table>
<thead>
<tr>
<th><strong>Use Case Name</strong></th>
<th>Profile Editing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Participating Actor(s)</strong></td>
<td>System Administrators, Polling Officer, Voter</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. Access particular user’s information for editing  
2. Modify the information accordingly.  
3. Save the user’s updated information. |
| **Pre-Conditions** | System administrators, polling officer and voters have been successfully authenticated and can access their individual homepage |
| **Post-Conditions** | Updated information saved in the database and displayed |
### Table E-7: Use Case description of registration process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Registration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>Polling Officer, Voter</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. Access their respective signup page.  
2. Fill in their details.  
3. Save the information in the database by clicking save button. |
| **Pre-Conditions** | Polling officer and voter are eligible to register themselves in eVote |
| **Post-Conditions** | Eligible Voter or Officers are registered and saved in the database |

### Table E-8: Use Case description of authentication process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>System Administrators, Polling Officer, Voter</td>
</tr>
</tbody>
</table>
| **Flow of Events** | 1. User navigates to the login page.  
2. Enter their login credentials.  
3. The system authenticates their eligibility based on the saved data in the database. |
| **Pre-Conditions** | Users have been registered in eVote |
| **Post-Conditions** | User can access their individual homepage |
### Table E-9: Use Case description of voting process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>Voter</td>
</tr>
<tr>
<td>Flow of Events</td>
<td>1. Navigate to the voting page.&lt;br&gt;2. List of candidates is displayed.&lt;br&gt;3. Selects the chosen candidates.&lt;br&gt;4. User reviews the choices.&lt;br&gt;5. Confirm choices by clicking confirm button.&lt;br&gt;6. Ballot is saved into the database and system sends the voting receipt to the corresponding voter’s email account.&lt;br&gt;7. Voting summary is shown.</td>
</tr>
<tr>
<td>Pre-Conditions</td>
<td>Each voter has been successfully authenticated and can access their individual homepage</td>
</tr>
<tr>
<td>Post-Conditions</td>
<td>Voters successfully cast their vote</td>
</tr>
</tbody>
</table>

### Table E-10: Use Case description of vote verification process

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Vote Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participating Actor(s)</td>
<td>Voter</td>
</tr>
<tr>
<td>Flow of Events</td>
<td>1. Voter access the vote verification page.&lt;br&gt;2. Select and upload the vote receipt which was received before.&lt;br&gt;3. Vote receipt is verified.&lt;br&gt;4. Vote verification summary is displayed.</td>
</tr>
<tr>
<td>Pre-Conditions</td>
<td>Voter cast their vote and need to verify it by presenting the given receipt.</td>
</tr>
<tr>
<td>Post-Conditions</td>
<td>Voter’s cast vote is successfully verified.</td>
</tr>
</tbody>
</table>
Appendix F: UML Class Diagrams

Figure F-1: Class Diagram of ballot box package

Figure F-2: Class Diagram of cryptography package
Figure F-3: Class Diagram of controller package

```
ControllerServlet
+init(servletConfig : ServletConfig) : void
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void

AdminController
-userPath : String
-balletList : List = new ArrayList()
-count : int = 0
-tempId : Integer
-endDate : String
-startDateTime : String
-electionName : String

#processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+sqlQuery(query : String) : void
+sqlQueryInsert(query : String, customer_order_id : int, product_id : int, quantity : int) : void
+getChart() : void
+getEmails(s : InputStream) : List<String>

OfficerServlet
-userPath : String
-balletList : List = new ArrayList()
-voterList : List = new ArrayList()
-orderedProductListEnc : List<String> = new ArrayList()
-orderedProductListDec : List<String> = new ArrayList()

#processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
#doGet(request : HttpServletRequest, response : HttpServletResponse) : void
#doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+sqlQuery(query : String) : void
+sqlQueryInsert(query : String, customer_order_id : int, product_id : int, quantity : int) : void
+getChart() : void
```
Figure F-4: Class Diagram of entity package
Figure F-5: Class Diagram of steganography package
Figure F-6: Class Diagram of session package
## Appendix G: Sample Questionnaire

### Survey for “eVote” Voting System

#### Demographic Information

1. Gender (Please tick):
   - [ ] Male
   - [ ] Female

2. Age: _____

3. Highest Level of Education (Please tick):
   - [ ] SPM
   - [ ] STPM / A-Level
   - [ ] Bachelor Degree
   - [ ] Others: _____

#### Heuristic Evaluation

Please rate the usability of eVote Voting System
- Circle the most appropriate answer for the following questions.

4. Visibility of system status:
   a) I can understand the interface easily.
   **Strongly disagree** 1 2 3 4 5 **Strongly agree**
   
b) The system is easy to use:
   **Strongly disagree** 1 2 3 4 5 **Strongly agree**

5. Match between the system and the real world:
   a) I can understand the function of each icon/button:
   **Strongly disagree** 1 2 3 4 5 **Strongly agree**
   
b) I am able to follow the order of the system:
   **Strongly disagree** 1 2 3 4 5 **Strongly agree**

6. User control and freedom:
   a) When operating the system and make unexpected mistake, I can leave the page (e.g go to the menu or home page) easily:
   **Strongly disagree** 1 2 3 4 5 **Strongly agree**
7. Consistency and standards:
   a) The layout of the interface is arranged in a logical order:
      **Strongly disagree**  1  2  3  4  5  **Strongly agree**
   
   b) The icons/buttons in the interface are confusing:
      **Strongly disagree**  1  2  3  4  5  **Strongly agree**

8. Help users recognize, diagnose, and recover from errors:
   a) If I make mistake while using the system, I can handle it easily and quickly:
      **Strongly disagree**  1  2  3  4  5  **Strongly agree**

9. Error prevention:
   a) I always make errors when I am using the system:
      **Strongly disagree**  1  2  3  4  5  **Strongly agree**
   
   b) If there is an error, I can deal with it:
      **Strongly disagree**  1  2  3  4  5  **Strongly agree**

10. Recognition rather than recall:
    a) The system has all the functions that I expect it to have:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**
    
    b) I can recognize the function rather than recall:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**

11. Flexibility and efficiency to use:
    a) I think that inexperienced user also capable to use the system:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**
    
    b) I believe that I can operate the system more effectively at the second time:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**

12. Aesthetic and minimalist design:
    a) I found that there is irrelevant information in the system:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**
    
    b) The design gives user-friendliness experience to me:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**
    
    c) The layout of the interface is arrange in a proper way:
       **Strongly disagree**  1  2  3  4  5  **Strongly agree**
13. Do you have any other comment(s) about the system? (Please write your comment on the space provided).

eVote:

____________________________________________________________________
____________________________________________________________________
____________________________________________________________________

Security

15. I trust eVote Voting System to accurately count my vote:

**Strongly disagree**  1  2  3  4  5  **Strongly agree**

16. I trust eVote Voting System will keep my vote secure:

**Strongly disagree**  1  2  3  4  5  **Strongly agree**

17. Do you prefer to use E-voting System to cast your vote? Why? (Please tick)
If yes, please go to number 18.

□ Yes
□ No

____________________________________________________________________
____________________________________________________________________
____________________________________________________________________

18. Which of the following vote receipt format would you prefer as your vote receipt? Why? (Please tick)

□ Image (Visual cryptography encoded)
□ Ciphertext

____________________________________________________________________
____________________________________________________________________

- Thank you for your time and cooperation -
LIST OF PUBLICATIONS

During the tenure of my MSc research, the following papers have been presented and/or published:


