**DEFENCE SERVICES ACADEMY**

**DEPARTMENT OF COMPUTER SCIENCE**

**CRYPTOGRAPHIC PROTOCOLS FOR SECURE VOTING RESULTS IN ELECTION VOTING SYSTEM USING FULLY HOMOMORPHIC ENCRYPTION (FHE)**

**BY**

**AUNG MYO KYAW**

**M.C. Sc Thesis**

**MARCH, 2025**

**PYIN OO LWIN**

DEFENCE SERVICES ACADEMY

DEPARTMENT OF COMPUTER SCIENCE

**CRYPTOGRAPHIC PROTOCOLS FOR SECURE VOTING RESULTS IN ELECTION VOTING SYSTEM USING FULLY HOMOMORPHIC ENCRYPTION (FHE)**

By

Captain AUNG MYO KYAW

THESIS

SUBMITTED TO THE DEFENCE SERVICES ACADEMY

IN FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF COMPUTER SCIENCE

(M.C. Sc)

March, 2025

PYIN OO LWIN

**DEFENCE SERVICES ACADEMY**

This is to certify that we have examined, and recommend to the supervision committee for the M.C. Sc program for acceptance the thesis entitled **“REAL-TIME WEB-BASED GOLF TOURNAMENT AND PERSONAL SCORE MANAGEMENT SYSTEM”**, submitted by**Captain Zwe Wai Yan Phyoe,** Roll No**. CS-11 (March, 2024)** in fulfillment of the requirements for the degree of Master of Computer Science.

**Board of Examiners:**

1. Lt.Col. Soe Win Myint

Professor

Department of Computer Science ---------------------------Defence Services Academy (Chairman)

1. Maj. Ye Wint Maung Maung

Assistant Lecturer

Department of Computer Science ---------------------------Defence Services Academy (Supervisor)

1. Maj. Yar Zar Doe

Assistant Lecturer

Department of Computer Science ---------------------------Defence Services Academy (Member)

1. Maj. Aung Myo Thu

Assistant Lecturer

Department of Computer Science ---------------------------Defence Services Academy (Member)

1. Dr. Nyo Nyo Yee

Professor and Head

Department of Information Science --------------------------

University of Technology (External Examiner)

(Yatanarpon Cyber City)

1. Lt.Col. Myat Nyein Moe

Lecturer

Department of Information Technology --------------------------Defence Services Technological Academy (External Examiner)

**DEFENCE SERVICES ACADEMY**

**DEPARTMENT OF COMPUTER SCIENCE**

**CRYPTOGRAPHIC PROTOCOLS FOR SECURE VOTING RESULTS IN ELECTION VOTING SYSTEM USING FULLY HOMOMORPHIC ENCRYPTION (FHE)**

**Presented By**

**Captain AUNG MYO KYAW**

**Approved by**

**Co-Supervisor**

**Lieutenant Colonel Saw Myint Soe Aung**

**Ph.D. (Computer Science)**

**Assistant Lecturer**

**Department of Computer Science**

**Supervisor**

**Lieutenant Colonel Wanna Kyaw Kyaw**

**Ph.D. (Computer Science)**

**Assistant Lecturer**

**Department of Computer Science**

**March 6, 2025**

# ACKNOWLEDGEMENTS

I would like to express my profound gratitude to **Major General Soe Myat Htut**, Commandant of the Defence Services Academy for allowing me to attend the Master of Computer Science, program at the Defence Services Academy.

I would special thanks to the Rector of the Defence Services Academy, **Brigadier General Thein Lwin**, for his kind permission and encouragement for my Master’s Degree in Computer Science.

I wish to extend my thanks to the person, **Lieutenant Colonel Soe Win Myint**, Head of the Department of Computer Science at the Defence Services Academy, for his close support in doing thesis work, attending seminars, and giving valuable advice and guidance for the preparation of this thesis.

I am highly indebted to **Lieutenant Colonel Zar Nay Lin,** who is responsible for the Master of Computer Science at the Defence Services Academy, for his immense patience and for providing an incredible amount of guidance.

I must particularly express my thankfulness to my coaches, **Major Ye Wint Mg Mg**, Assistant Lecturer and supervisor, and **Major Myo Min Hein,** Assistant Lecturer and co-supervisor, Department of Computer Science at Defence Services Academy, for valuable advice, their close supervision, kind guidance and support, and cooperation of master degree program, and I realized the valuable suggestions, understanding, and generous encouragements during a long period of study for thesis paper. I am also grateful to **Captain Aung Myint Myat**, Department of Computer Science, for his helping with editing. Last but not least, many thanks to all persons who directly and indirectly contributed toward the completion of this thesis.

ABSTRACT

The integrity, transparency, and confidentiality of voting systems are critical to the democratic process. However, achieving secure and verifiable election systems while maintaining voter privacy poses significant challenges. This thesis explores the application of Fully Homomorphic Encryption (FHE) to design cryptographic protocols that ensure secure voting and result tabulation in election systems. FHE enables computations to be performed directly on encrypted data, ensuring data confidentiality throughout the voting and tallying processes. This work proposes a novel cryptographic framework that leverages FHE to achieve end-to-end security in electronic voting systems. The framework ensures the privacy of individual votes, prevents tampering during data transmission, and allows accurate tallying without decryption. Security analysis and performance evaluations are conducted to demonstrate the feasibility of the proposed protocols in real-world scenarios. The results indicate that the use of FHE in election systems can provide robust protection against threats such as vote manipulation, voter coercion, and system breaches, while preserving voter anonymity. By addressing key security and efficiency concerns, this thesis contributes to advancing the state-of-the-art in cryptographic solutions for secure election systems, paving the way for trustworthy and scalable electronic voting implementations. This system is implemented by using PHP and AJAX (Asynchronous JavaScript and XML).

**Keywords**: AJAX (Asynchronous JavaScript and XML), Handicap Calculation, Leaderboard, Mobile devices, Real-time Updates

# TABLE OF CONTENTS

**Page**

[ACKNOWLEDGEMENTS i](#_Toc160799646)

[ABSTRACT ii](#_Toc160799647)

[TABLE OF CONTENTS iii](#_Toc160799648)

[LIST OF ABBREVIATION vi](#_Toc160799649)

[LIST OF TABLES viii](#_Toc160799650)

[LIST OF EQUATIONS ix](#_Toc160799651)

**CHAPTER** **TITLE**

1 [INTRODUCTION 1](#_Toc160799653)

[1.1 Problem Statement 2](#_Toc160799654)

[1.2 Motivation 2](#_Toc160799655)

[1.3 Aim of thesis 3](#_Toc160799656)

[1.4 Objectives of thesis 3](#_Toc160799657)

[1.5 Overview of the System 3](#_Toc160799658)

[1.6 Organization of Thesis 4](#_Toc160799659)

2 [BACKGROUND KNOWLEDGE 5](#_Toc160799661)

[2.1 History of World Wide Web 5](#_Toc160799662)

[2.2 Web Page 6](#_Toc160799663)

[2.2.1 Static Web Page 6](#_Toc160799664)

[2.2.2 Dynamic Web Pages 8](#_Toc160799665)

[2.3 Hypertext Transfer Protocol (HTTP) 8](#_Toc160799666)

[2.4 Relational Database Management System 16](#_Toc160799673)

[2.5 Electronic Voting Systems in Elections 16](#_Toc160799673)

[2.6 Proportional Representation (PR) system in Elections 16](#_Toc160799673)

[2.7 Cryptography 16](#_Toc160799673)

Table of Contents (Continued)

[2.8 Fully Homomorphic Encryption (FHE) 16](#_Toc160799673)

[2.9 Homomorphic Encryption in Voting 16](#_Toc160799673)

[2.10 Summary 26](#_Toc160799682)

3 [SYSTEM DESIGN AND IMPLEMENTATION 27](#_Toc160799684)

[3.1 Work Flow of the System 27](#_Toc160799685)

[3.2 Overall Design of the System 28](#_Toc160799686)

[3.3 Detail Design for Admin 29](#_Toc160799687)

[3.4 Detail Design for User 30](#_Toc160799688)

4 [CONCLUSION AND FUTURE WORK 45](#_Toc160799698)

Table of Contents (Continued)

[4.1 Conclusion 45](#_Toc160799699)

[4.2 Benefits of the System 46](#_Toc160799700)

[4.3 Limitation 46](#_Toc160799701)

[4.4 Further Extension 47](#_Toc160799702)

[REFERENCES](#_Toc160799703)

[APPENDICES](#_Toc160799704)

# LIST OF ABBREVIATION

DBMS Database Management System

HTML Hyper Text Markup Language

HTML5 Hyper Text Markup Language revision 5

HTTP Hyper Text Transfer Protocol

HTTPS SSL encrypted version of Hyper Text Transfer Protocol

MVC Model View Controller

PHP Hypertext Preprocessor

RDBMS Relational Database Management System

SQL Structure Query Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

W3C World Wide Web Consortium

WWW World Wide Web

CHAPTER 1

# INTRODUCTION

Elections are a cornerstone of democratic societies, serving as a mechanism for citizens to express their will and select representatives. In recent years, electronic voting systems have emerged as a promising solution to enhance the efficiency, accessibility, and scalability of the voting process. However, the adoption of such systems has been hindered by concerns surrounding security, privacy, and trustworthiness. Ensuring the confidentiality of individual votes, the integrity of election results, and the transparency of the voting process remains a significant challenge for researchers and practitioners alike.

Traditional cryptographic approaches have been employed to secure voting systems, but they often involve trade-offs between security, privacy, and performance. Fully Homomorphic Encryption (FHE), a groundbreaking advancement in cryptography, offers a transformative approach to these challenges. FHE enables computations to be performed directly on encrypted data, eliminating the need for decryption at intermediate stages. This unique capability opens new possibilities for designing secure election systems that preserve voter privacy while maintaining the verifiability and correctness of results.

This thesis aims to leverage FHE to design cryptographic protocols that address the critical security requirements of election systems. These requirements include vote confidentiality, integrity, authenticity, and resistance to coercion. By enabling the tallying of encrypted votes without exposing individual votes, FHE-based protocols can mitigate common vulnerabilities associated with traditional electronic voting systems.

1. Problem Statement

The adoption of electronic voting systems introduces opportunities for more efficient and accessible elections but also amplifies concerns about security, privacy, and trustworthiness. Voters need assurance that their choices remain confidential and free from external manipulation, while election administrators require guarantees of system integrity and result accuracy. However, traditional cryptographic methods used in voting systems often face challenges such as trade-offs between voter privacy, transparency, and resistance to tampering.

Current electronic voting systems are particularly vulnerable to:

1. Vote confidentiality breaches: The risk of unauthorized access to voter data, which compromises anonymity.
2. Result integrity challenges: Susceptibility to vote manipulation or alteration during transmission or tallying.
3. Transparency and verifiability issues: Difficulty in providing proof of correctness without exposing sensitive information.

While existing solutions attempt to address some of these vulnerabilities, they often rely on decryption during the tallying process, exposing data to potential attacks. Moreover, these approaches are either computationally expensive or lack the scalability required for large-scale elections.

Fully Homomorphic Encryption (FHE) presents a promising cryptographic tool to mitigate these challenges. However, its application to voting systems has not been fully explored or optimized for practical use. The core problem lies in designing a secure, efficient, and scalable voting protocol that leverages FHE to ensure the confidentiality of votes, integrity of results, and transparency of the process, while addressing performance and implementation barriers.

This research seeks to bridge this gap by developing and evaluating cryptographic protocols that utilize FHE for secure electronic voting, aiming to provide a robust solution to the challenges facing modern election systems.

1. Motivation

In democratic societies, the integrity of the voting process is fundamental to public trust and the legitimacy of elected governments. Traditional paper-based voting systems, while relatively secure, are often criticized for being slow, resource-intensive, and susceptible to human error. Electronic voting systems have emerged as a modern alternative, promising enhanced efficiency, accessibility, and scalability. However, these systems introduce new vulnerabilities that, if unaddressed, can undermine voter confidence and the credibility of election outcomes.

The motivation for this research stems from the following considerations:

1. **Rising cybersecurity threats**: With the increasing digitalization of critical infrastructure, including voting systems, cyberattacks such as data breaches, vote tampering, and denial-of-service attacks have become more prevalent. Secure election systems must be resilient against such threats to maintain trust in the democratic process.
2. **Privacy and confidentiality**: One of the most critical aspects of any voting system is the assurance that individual votes remain private. Breaches of voter anonymity can lead to coercion, vote-buying, and a loss of confidence in the electoral process. Ensuring privacy without compromising transparency is a significant challenge that modern cryptographic techniques can address.
3. **Demand for verifiable election results**: Voters and stakeholders increasingly demand transparent and verifiable systems that provide proof of correctness without exposing sensitive information. Achieving this balance is critical for fostering trust in electronic voting systems.
4. **Advancements in cryptography**: Fully Homomorphic Encryption (FHE) represents a significant breakthrough in cryptographic research. By enabling computations on encrypted data, FHE offers the potential to revolutionize electronic voting by eliminating the need for decryption during tallying. This capability addresses many vulnerabilities associated with traditional cryptographic methods.
5. **Global trends in e-governance**: As governments worldwide embrace digital transformation, there is a growing emphasis on secure, scalable, and trustworthy systems for public services, including elections. Developing robust cryptographic protocols for voting aligns with this broader push towards secure e-governance.

This research is driven by the urgent need to address the vulnerabilities of existing electronic voting systems while leveraging state-of-the-art cryptographic advancements. By exploring the application of FHE to secure voting protocols, this study aims to contribute to the development of secure and reliable election technologies, ultimately strengthening public trust in the democratic process.

## 1.3 Aim of thesis

The primary aim of this thesis is to develop a cryptographic framework for electronic voting systems that ensures secure voting and accurate result computation using Fully Homomorphic Encryption (FHE). This framework seeks to address the critical challenges of vote confidentiality, data integrity, and system transparency, thereby enhancing the trustworthiness and reliability of election processes.

## 1.4 Objectives of thesis

The objectives of system are as follows;

1. To design a secure voting protocol:
2. To ensure voter anonymity and result integrity
3. To provide verifiability without compromising privacy
4. To evaluate performance and scalability

## 1.5 Overview of the System

The proposed electronic voting system is designed to ensure vote confidentiality, result integrity, and system transparency by leveraging Fully Homomorphic Encryption (FHE). The system comprises key components such as a voter registration module for secure authentication and anonymity, an encrypted vote submission mechanism, and a tamper-resistant repository for storing encrypted votes. Using FHE, the system performs computations directly on encrypted data, enabling secure tallying of votes without requiring decryption. This ensures that voter privacy is maintained throughout the process, while the aggregated results are decrypted and verified by authorized entities before publication.

The system also incorporates features like end-to-end encryption, tamper resistance, and public verifiability using zero-knowledge proof mechanisms, enabling stakeholders to verify election results without exposing sensitive data. The workflow spans three phases: pre-election voter registration and key initialization, secure voting during the election phase, and post-election tallying and result verification. Optimized for scalability, this system addresses the challenges of traditional and electronic voting systems, offering a robust solution for secure, transparent, and trustworthy elections.

## 1.6 Organization of Thesis

This system is organized as follows: In chapter one, introduction and objectives are described. Chapter two presents the literature review and also presents all of background theories for our proposed system. In chapter three, the proposed systems present. Chapter four describes the conclusion and future works.

# CHAPTER 2

# BACKGROUND KNOWLEDGE

In this chapter, the history of world wide web, web page, Hypertext Transfer Protocol (HTTP), XAMPP, Hypertext Markup Language (HTML), Hypertext Preprocessor (PHP) are presented. This thesis is based on Model-1 Architecture.

## 2.1 History of World Wide Web

The World Wide Web, commonly known as the web, emerged in the late 20th century as a revolutionary system for accessing and sharing information over the internet. It was conceived by British computer scientist Tim Berners-Lee in 1989 while he was working at CERN, the European Organization for Nuclear Research. Berners-Lee's vision was to create a distributed information system that would allow researchers to share documents and data more efficiently. In 1990, Berners-Lee developed the foundational technologies for the web, including Hypertext Markup Language (HTML) for creating web pages, Uniform Resource Identifiers (URIs) for identifying resources, and Hypertext Transfer Protocol (HTTP) for transmitting data over the internet. He also created the first web browser/editor (World Wide Web) and web server (httpd) to demonstrate the capabilities of the web.

The web gained widespread popularity in the mid-1990s with the release of graphical web browsers like Mosaic and Netscape Navigator, which made it easier for users to navigate and interact with web pages. This period, known as the "dot-com boom," saw exponential growth in the number of websites, online businesses, and internet users, transforming the way people communicate, access information, and conduct commerce worldwide.[12]

## 2.2 Web Page

A web page is a fundamental unit of information on the World Wide Web, consisting of content organized within a single HTML document. It serves as a digital document accessible through a web browser and is often part of a larger website or web application. Web pages can vary widely in content and purpose, ranging from static pages displaying text and images to dynamic pages with interactive elements and multimedia content. Structurally, a web page typically includes HTML markup to define the structure and layout of the content, CSS styles to control its presentation and appearance, and JavaScript code to add interactivity and functionality.

The content of a web page can be diverse, encompassing textual information, images, videos, audio files, forms, and interactive elements such as buttons and links. Web pages may also include metadata such as page titles, descriptions, and keywords to optimize search engine visibility and accessibility. Additionally, modern web pages often incorporate responsive design techniques to ensure compatibility and usability across various devices and screen sizes. Web pages are accessed by users through a web browser, which retrieves the page from a web server and renders it for display. Users can navigate between web pages by clicking on hyperlinks or using navigation menus provided within the page. With the proliferation of the internet, web pages have become an integral part of daily life, serving as a primary means of accessing information, conducting business transactions, communicating with others, and engaging with online content. [16,14]

There are two basic types of web pages:

1. Static Web Pages
2. Dynamic Web Pages

### 2.2.1 Static Web Page

Static Web Pages are made of “fixed code,” and unless the site developer makes changes, nothing will change on the page. Static sites give a lot of the same type of information that the user could get from a brochure, but it can’t just change itself. In order to do this, someone has to create a new page. That’s why static websites are sometimes referred to as brochure sites. [15]

Nothing is stored but the actual pages of a static site. There are:

1. No users
2. No comments
3. No blog posts
4. No interactivity

A static website is delivered to a user exactly the way it’s stored. That means that nothing on the page will change by the user or even the site administrator unless there’s a redesign of the site, or the site administrator goes directly into the code to change it. A static site is the most basic kind of website, and the easiest to create. It requires no server-side (also called back-end) processing, only client-side. Client-side technologies are [HTML, CSS, and JavaScript](https://www.pluralsight.com/paths/building-websites-with-html-css-and-javascript).

No programming languages, including JavaScript, are required to make a static site. However, if a site utilizes JavaScript, but no [PHP](https://www.pluralsight.com/paths/php-development-fundamentals) or any other programming language, it’s still considered a static site (since JavaScript is a client-side language). So, if the user wants a site only to give information that doesn’t need to be updated regularly, creating a static website is a simple and effective way to go.[15] The static web page is shown in Figure 2.1.

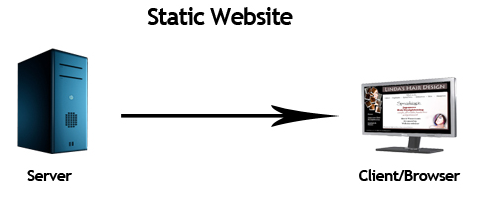


Figure 2.1 Static Web Page

### 2.2.2 Dynamic Web Pages

A dynamic web page is a type of webpage that displays different content each time it's viewed. Unlike static pages, which have fixed content, dynamic pages are generated on the fly, usually by pulling data from a database or other external source. This allows for personalized content tailored to each user, as well as real-time updates and interactions. Dynamic pages often utilize technologies like HTML, CSS, JavaScript, and server-side scripting languages such as PHP, Python, or Ruby on Rails. Examples of dynamic web pages include social media feeds, news websites with constantly updating articles, and e-commerce sites that display products based on user preferences. Their flexibility and interactivity make dynamic web pages essential for modern internet experiences.[18]

There’s a simple way to determine if a site is dynamic. If the user can interact with it, it’s a dynamic site. For example, dynamic sites allow the user to create a user profile, comment on a post, or make a reservation.



Figure 2.2 Dynamic Web Page

## 2.3 Hypertext Transfer Protocol (HTTP)

HTTP stands for Hypertext Transfer Protocol. It is the foundation of data communication on the World Wide Web. When the user access a website through the web browser, the browser sends an HTTP request to the web server hosting the website. The web server processes the request and sends back the requested web page or other resources, such as images or scripts, using HTTP.

HTTP is a protocol used for transmitting data over the internet. It is an application layer protocol, which means it focuses on the communication between web browsers and servers. HTTP operates on top of the TCP/IP protocol, which ensures that data is delivered accurately and reliably across networks. HTTP is designed as a stateless protocol, meaning that each request from a client to a server is treated independently without any knowledge of previous requests. However, web applications often require maintaining state. To address this, technologies like cookies and sessions are used to enable stateful communication between clients and servers. HTTP has undergone several versions over the years. The most commonly used versions are HTTP/1.0 and HTTP/1.1. HTTP/2 is a more recent version that offers improved performance and security features. HTTP/3, based on the QUIC protocol, is also emerging as a more efficient and faster alternative for web communication.[8]

## 2.4 Relational Database Management System

The relational model was invented by Edgar Codd as a general model of data, and subsequently promoted by Chris Date and Hugh Darwen among others. In The Third Manifesto (first published in 1995) Date and Darwen attempt to show how the relational model can allegedly accommodate certain "desired" object-oriented features. A relational database is a database model that stores data in tables. The vast majority of databases used in modern applications are relational, so the terms "database" and "relational database" are often used synonymously. Likewise, most database management systems are relational database management systems.[6]

A Relational Database (RDB) is a collective set of multiple data sets organized by tables, records and columns. RDBs establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data search ability, organization and reporting. RDBs use Structured Query Language (SQL), which is a standard user application that provides an easy programming interface for database interaction. RDB is derived from the mathematical function concept of mapping data sets and was developed by Edgar F. Codd. Stop Ransomware Mid-Flight Techopedia explains Relational Database (RDB). RDBs organize data in different ways. Each table is known as a relation, which contains one or more data category columns. Each table record (or row) contains a unique data instance defined for a corresponding column category. One or more data or record characteristics relate to one or many records to form functional dependencies. These are classified as follows:

1. One to One: One table record relates to another record in another table.
2. One to Many: One table record relates to many records in another table.
3. Many to One: More than one table record relates to another table record.
4. Many to Many: More than one table record relates to more than one record in another table.[2]

## 2.5 Electronic Voting Systems in Elections

Electronic voting systems (e-voting) refer to the use of digital technology to cast and count votes in an election, as opposed to traditional paper-based methods. These systems aim to enhance the efficiency, accessibility, and accuracy of voting, while also reducing the risk of human error in vote tallying. E-voting can take several forms, including optical scan voting systems, direct-recording electronic (DRE) voting machines, and internet-based voting platforms. The adoption of e-voting has been driven by the increasing demand for more secure, transparent, and efficient election processes, as well as the need to accommodate larger populations and a growing trend toward digital governance.

A key advantage of e-voting systems is their ability to expedite the voting and counting processes, allowing election results to be tabulated quickly and accurately. However, challenges related to security, privacy, and trust persist. The primary concerns include vote manipulation, hacking of voting machines or systems, denial-of-service attacks, and ensuring voter anonymity while providing transparency in the process. To address these concerns, cryptographic methods, such as public-key cryptography and homomorphic encryption, have been proposed and tested. These cryptographic techniques ensure the confidentiality and integrity of votes, preventing unauthorized access and alterations during the transmission and counting of votes.

Several countries have experimented with e-voting in national elections or referenda, with varying degrees of success. For example, Estonia has implemented a nationwide internet voting system, and countries like Switzerland and India have used electronic voting machines in specific regions. Despite these advancements, concerns over voter privacy, accessibility, and security continue to challenge the widespread adoption of e-voting systems, particularly in large-scale national elections. As a result, improving the security, verifiability, and scalability of these systems remains a critical area of research. [16]

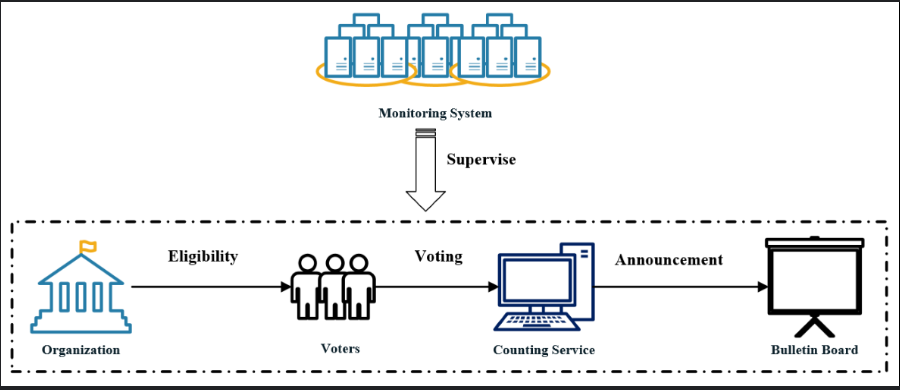


Figure 2.3: A Publicly Verifiable E-Voting System

## 2.6 Proportional Representation (PR) system in Elections

The Proportional Representation (PR) system is a voting method aimed at ensuring that political parties or groups receive seats in proportion to the number of votes cast for them. Unlike the majoritarian or plurality voting systems (such as First-Past-The-Post), where the candidate with the most votes wins, PR systems are designed to provide a more accurate representation of voter preferences in the elected body. The PR system is widely used in many democracies around the world, especially for parliamentary elections, to reflect the diversity of opinions within the electorate and promote more inclusive governance.

PR systems typically involve multi-member electoral districts, where more than one representative is elected from each district. This contrasts with single-member districts found in plurality systems. One of the key variations of the PR system is the Closed-List PR, where voters choose a party, and the seats are allocated to parties based on the percentage of votes they receive. Another variant is Open-List PR, where voters have some influence over which candidates within the party list are elected. Mixed-Member Proportional (MMP) and Single Transferable Vote (STV) are other variations that combine aspects of majoritarian systems with proportionality.

The PR system aims to achieve fairness by giving smaller political parties and minority groups a more substantial representation in the legislature than they would receive under a system like First-Past-The-Post. This is especially significant in countries with diverse populations, where smaller groups may otherwise be underrepresented. However, the PR system can also lead to coalition governments, which may involve complex negotiations between parties to form a stable government. Critics of the PR system argue that it may fragment the political landscape, leading to instability or weak coalitions. Despite these challenges, PR remains a widely used and advocated system for creating representative, proportional outcomes in elections. [17]

## 2.7 Cryptography

Cryptography is the practice of securing communication and data from unauthorized access, ensuring confidentiality, integrity, and authenticity of information. It involves the use of mathematical algorithms to encrypt and decrypt messages, transforming readable data (plaintext) into an unreadable format (ciphertext) and vice versa. Cryptography is a cornerstone of modern security protocols, enabling secure communications across various platforms, such as online banking, email, and voting systems.

Cryptography can be broadly classified into two categories: symmetric-key cryptography and asymmetric-key cryptography. In symmetric-key cryptography, the same key is used for both encryption and decryption, making it faster and more efficient for encrypting large volumes of data. However, the challenge lies in securely distributing the key to both the sender and the receiver. Common symmetric-key algorithms include the Advanced Encryption Standard (AES) and Data Encryption Standard (DES).

Asymmetric-key cryptography, also known as public-key cryptography, uses a pair of keys: one for encryption (public key) and another for decryption (private key). This method solves the key distribution problem and is essential for secure communications over the internet. RSA and Elliptic Curve Cryptography (ECC) are prominent examples of asymmetric-key cryptographic algorithms. Asymmetric cryptography is often used for tasks such as digital signatures, secure key exchange, and authentication.

An essential aspect of cryptography is hashing, which involves converting input data of arbitrary size into a fixed-size string, typically used for ensuring data integrity. Hash functions like SHA-256 are commonly used to verify that data has not been altered, as even the smallest change in the input will produce a completely different hash.

In the context of electronic voting, cryptography plays a vital role in ensuring the security of votes and protecting voter privacy. Techniques such as homomorphic encryption and zero-knowledge proofs are employed to allow for secure vote tallying without compromising the confidentiality of individual votes or the integrity of the voting process. Cryptographic protocols also facilitate verification and accountability in electronic voting systems, ensuring that the election results are accurate and transparent while preventing tampering and fraud. [18]

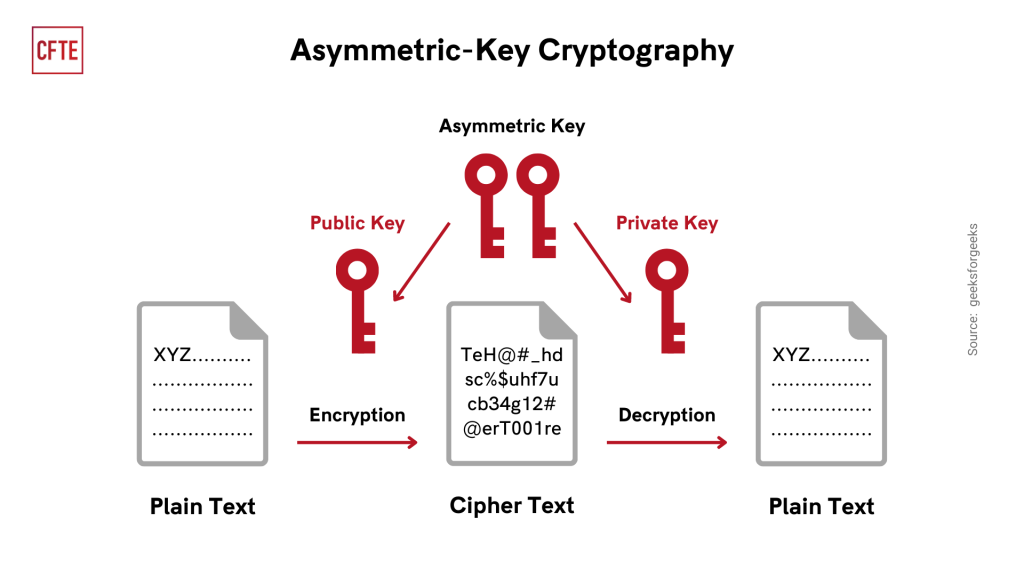


Figure 2.4: Asymmetric-Key Cryptography

## 2.8 Fully Homomorphic Encryption (FHE)

Fully Homomorphic Encryption (FHE) is a revolutionary cryptographic technique that enables computations to be performed on encrypted data without needing to decrypt it first. The concept of homomorphic encryption was first introduced by Rivest, Adleman, and Dertouzos in the late 1970s, but it wasn't until Craig Gentry's groundbreaking work in 2009 that the first fully homomorphic encryption scheme was proposed, making practical and secure computation on encrypted data feasible. FHE has the potential to transform fields where privacy-preserving computations are essential, including secure cloud computing, private data analysis, and electronic voting systems.

In homomorphic encryption, the term "homomorphism" refers to a property of the encryption scheme that allows certain operations to be carried out on encrypted data that, when decrypted, yield the same result as if the operations had been performed on the plaintext data. FHE, specifically, supports both addition and multiplication operations on ciphertexts, which is known as "both additively and multiplicatively homomorphic." This is significant because it allows for arbitrary computation on encrypted data, making it possible to process sensitive information without ever exposing it in plaintext form.

A key challenge of FHE is its computational complexity. Early FHE schemes, including Gentry's construction, were highly inefficient and impractical for many real-world applications due to their large overhead in terms of computational and memory resources. However, subsequent advancements in FHE schemes have led to significant improvements in efficiency and practicality. Modern implementations, such as those based on the Brakerski-Gentry-Vaikuntanathan (BGV) scheme and the Cheon-Kim-Kim-Song (CKKS) scheme, have made FHE more feasible for practical applications, although it still faces performance challenges when compared to traditional encryption methods.

In an FHE system, the encryption process is typically performed by the sender, who uses a public key to encrypt the data. This encrypted data is then sent to a computational server that can perform operations on the encrypted data without having access to the original plaintext. After computation, the server returns the encrypted results, which can only be decrypted by the intended recipient using the private key. This approach ensures that sensitive data remains confidential throughout the entire computation process, even in untrusted environments.

In the context of electronic voting systems, FHE can be used to ensure vote confidentiality while allowing the computation of election results. For example, votes can be encrypted before being cast, and the tallying process can be performed on the encrypted votes without decrypting them, ensuring that individual votes remain private. FHE thus provides a robust solution for maintaining both the security and privacy of voters, while also enabling transparent and verifiable election results.

Despite the promise of FHE, its high computational cost remains a limiting factor, especially for large-scale applications like national elections. However, as advancements continue to be made in the efficiency of FHE schemes, it holds significant potential for revolutionizing secure voting, private cloud computing, and a wide range of other privacy-sensitive applications. [19]

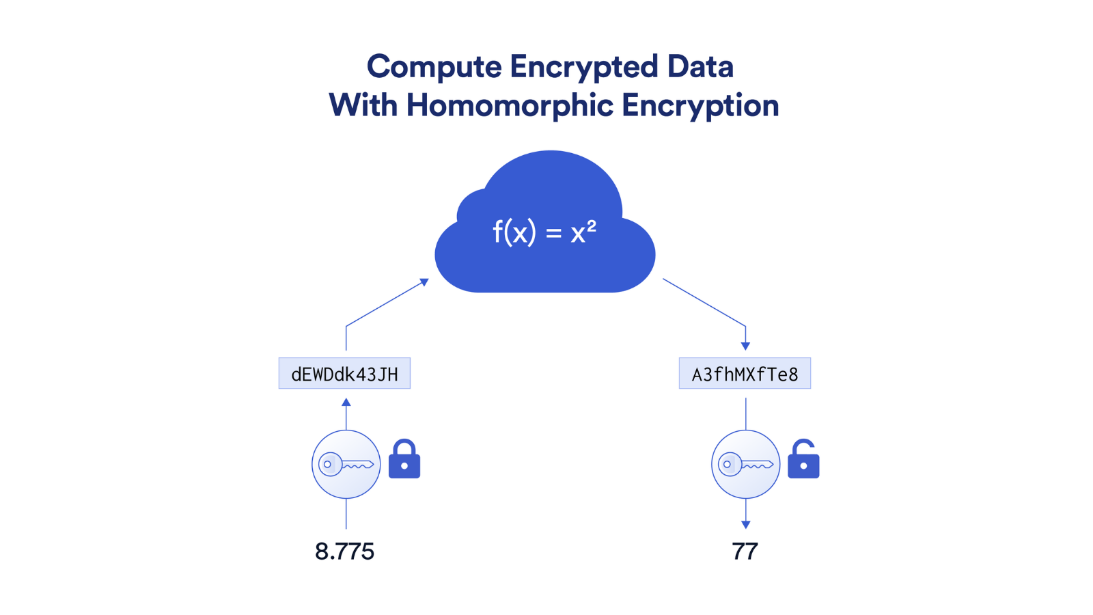


Figure 2.5: Work Flow of Fully Homomorphic Encryption (FHE)

## 2.9 Homomorphic Encryption in Voting

Homomorphic encryption (HE) is a cryptographic method that allows computations to be performed on encrypted data without revealing the underlying plaintext. This property makes it particularly valuable in electronic voting systems, where maintaining the confidentiality of individual votes while ensuring accurate and transparent tallying of election results is paramount. In voting systems, HE ensures that votes can be encrypted by voters, stored securely, and processed (e.g., aggregated) without being decrypted until the final results are calculated and verified.

The use of homomorphic encryption in voting systems provides several advantages. First, it ensures end-to-end vote confidentiality, as all computations (such as addition of votes) are carried out on ciphertexts, preserving voter privacy throughout the process. Second, it enables tamper-resistance, as encrypted votes cannot be altered without detection. Third, HE supports verifiable computation, allowing stakeholders to verify the correctness of the result without exposing individual votes. These properties make HE particularly suitable for addressing vulnerabilities associated with traditional and electronic voting systems, such as unauthorized access, vote tampering, and breaches of voter anonymity.

There are two types of homomorphic encryption schemes relevant to voting systems:

1. Partially Homomorphic Encryption (PHE): Supports either addition or multiplication on encrypted data, but not both. For example, the Paillier encryption scheme is additive homomorphic and has been widely used in voting systems to enable the secure summation of encrypted votes.
2. Fully Homomorphic Encryption (FHE): Supports both addition and multiplication, allowing arbitrary computations on encrypted data. While computationally more demanding than PHE, FHE provides greater flexibility and security, making it an ideal candidate for complex voting scenarios.

A typical workflow for an HE-based voting system is as follows:

1. Voter Encryption: Voters encrypt their votes using a public key associated with the election.
2. Secure Transmission and Storage: The encrypted votes are transmitted to a central repository or distributed ledger for secure storage.
3. Homomorphic Tallying: The election authority or a computational server performs operations (e.g., addition) on the encrypted votes to compute the total tally without decrypting the individual votes.
4. Result Decryption: The final encrypted tally is decrypted by an authorized entity using the private key to reveal the election results.

Homomorphic encryption has been successfully implemented in various voting systems, such as the Helios voting system, which uses the Paillier encryption scheme to provide a secure and verifiable voting process. While FHE offers a more robust solution for secure voting, its high computational overhead remains a challenge for large-scale elections. Research continues to focus on optimizing the performance of HE schemes to make them more practical for widespread adoption in voting systems. [20]

## 2.10 Summary

This chapter provides an in-depth exploration of foundational concepts and technologies crucial to understanding modern web development. It begins with an examination of the fundamental components of a web page, elucidating the role of Hypertext Markup Language (HTML) in structuring content and Cascading Style Sheets (CSS) in styling and formatting. The Hypertext Transfer Protocol (HTTP) is then introduced as the underlying protocol governing communication between web servers and clients. The chapter proceeds to discuss server-side technologies such as Hypertext Preprocessor (PHP), which facilitates dynamic content generation, and XAMPP, a popular web server solution for local development environments. JavaScript (JS) is explored as a client-side scripting language that enhances interactivity and responsiveness in web applications. Furthermore, the chapter delves into more advanced concepts like Asynchronous JavaScript and XML (AJAX), which enables asynchronous data exchange between the client and server, enhancing user experience by allowing seamless updates without page reloads. The Model 1 and Model-2 (MVC) architectures are also elucidated, highlighting their significance in organizing and structuring web applications for scalability, maintainability, and code reusability.

# CHAPTER 3

# SYSTEM DESIGN AND IMPLEMENTATION

This chapter presents the system design, detail design, database design and implementations of the system.

## 3.1 Work Flow of the System

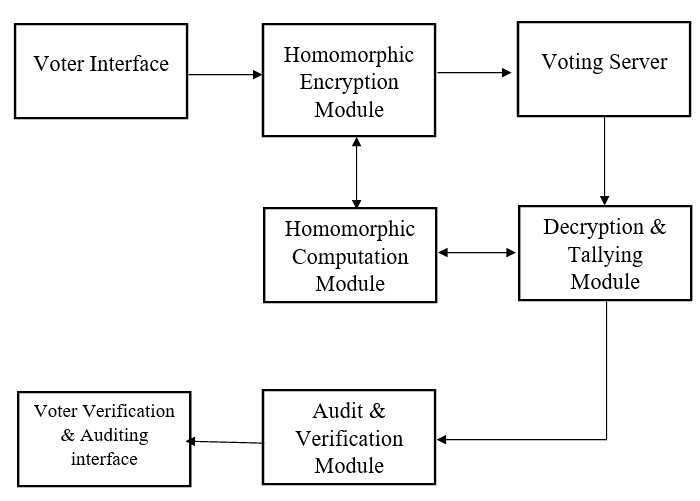
****

Figure 3.1 Work Flow of the system

Figure 3.1 shows a secure electronic voting system utilizing homomorphic encryption to ensure privacy, integrity, and verifiability of votes. Voters interact with the **Voter Interface** to cast their votes, which are encrypted in the **Homomorphic Encryption Module** using a public key for confidentiality. The encrypted votes are securely stored in the **Voting Server** and processed in the **Homomorphic Computation Module**, where operations such as vote tallying are performed directly on the encrypted data without decryption. The results are passed to the **Decryption and Tallying Module**, where authorized decryption occurs to reveal the final tally. Additionally, the system includes an **Audit & Verification Module** and a **Voter Verification Interface** to ensure transparency, accuracy, and verifiability of the election process. This workflow guarantees secure, private, and tamper-resistant voting.

## 3.2 Overall Design of the System

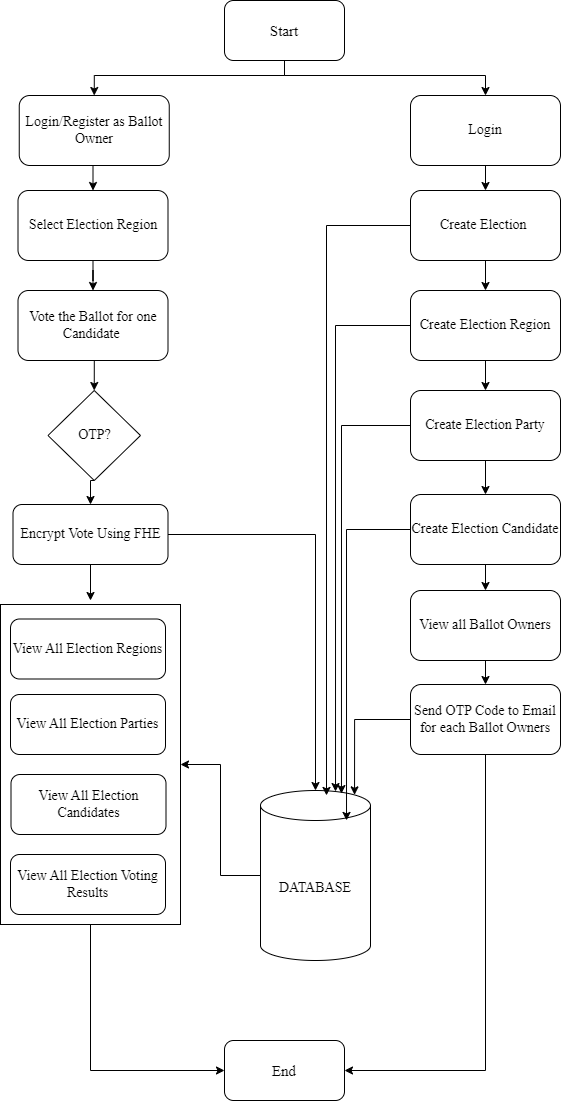


Figure 3.2 Overall System Design

Figure 3.2 shows a secure electronic voting workflow using Fully Homomorphic Encryption (FHE). The process begins with two primary roles: the **Ballot Owner** and the **Election Administrator**. Ballot owners log in, register, and select their election region to vote for a candidate. Before casting a vote, a One-Time Password (OTP) is verified for enhanced security, and the vote is encrypted using FHE. Election administrators, on the other hand, log in to create elections, regions, parties, and candidates, and manage ballot owners by sending OTP codes to their emails for verification. Both roles interact with a centralized database that stores encrypted votes, election configurations, and results. The system allows users to view election regions, parties, candidates, and voting results securely. This comprehensive workflow ensures data integrity, confidentiality, and transparency, leveraging encryption and OTP for secure authentication and data handling.

## 3.3 Detail Design for Admin

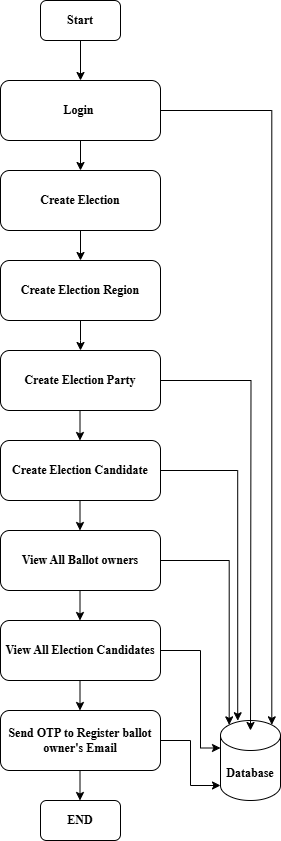


Figure 3.3 Detail Design for admin

The figure 3.3 illustrates the election setup process within the system, starting with an administrator logging in to manage the election framework. The administrator can sequentially perform tasks such as **creating an election,** defining **election regions,** specifying **election parties**, and adding **candidates** for the election. The system allows administrators to **view registered ballot owners** and the list of **candidates,** ensuring comprehensive management and transparency. To enhance security, the system integrates an **OTP (One-Time Password) mechanism** sent to the ballot owners' registered emails, ensuring their identity verification. All data, including elections, regions, parties, candidates, and ballot owner information, is securely stored in a centralized **database** for reliable access and management. This structured process streamlines election preparation, ensuring that all components are properly configured before voting begins.

## 3.4 Detail Design for User

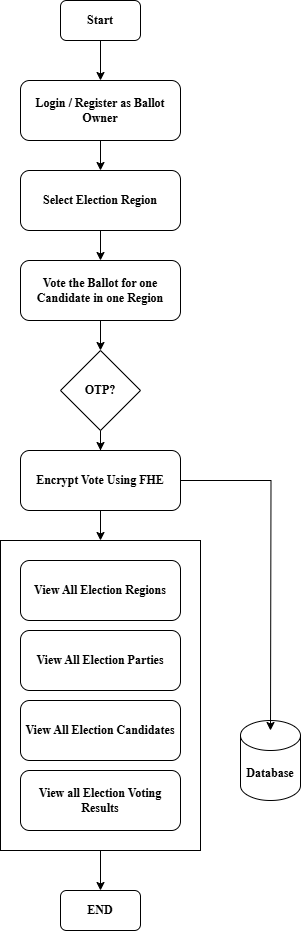


Figure 3.4 Detail Design for User

The figure 3.4 outlines the voter-side workflow in a secure electronic voting system. It begins with the ballot owner logging in or registering, after which they select the desired election region and cast their vote for a candidate within that region. To ensure security, the system verifies the voter’s identity through an **OTP (One-Time Password)** process. Once validated, the vote is encrypted using **Fully Homomorphic Encryption (FHE)** to maintain privacy and data security. The encrypted vote is stored in the database, and the system allows voters to access election-related information, including **regions, parties, candidates,** and **voting results**, ensuring transparency. This workflow ensures a secure, verifiable, and user-friendly voting process while safeguarding voter confidentiality

# CHAPTER 4

# CONCLUSION AND FUTURE WORK

## 4.1 Conclusion

This thesis explored the design and implementation of a secure electronic voting system using Fully Homomorphic Encryption (FHE) to address the critical challenges of privacy, security, and transparency in modern election processes. By leveraging FHE, the proposed system ensures that votes remain encrypted throughout the voting, storage, and computation phases, eliminating any risk of data breaches or unauthorized access. The integration of cryptographic protocols, secure voter authentication through OTP verification, and transparent auditing mechanisms further strengthens the system's reliability and usability.

The proposed solution demonstrates how FHE can enable secure computation on encrypted data, such as vote tallying, without compromising the confidentiality of individual votes. Additionally, the system is designed to scale efficiently with growing voter populations and diverse election scenarios, ensuring both performance and accessibility. This work highlights the potential of advanced cryptographic techniques to revolutionize electronic voting systems by offering end-to-end security while maintaining voter anonymity and trust in the electoral process. Future work can focus on optimizing computational efficiency and expanding usability features to make the system even more robust and accessible.

## 4.2 **Benefits of the System**

The benefits of the system are:

1. **Enhanced Security:** The system leverages Fully Homomorphic Encryption (FHE) to ensure that votes remain encrypted throughout the process, from casting to tallying, preventing unauthorized access or data breaches.
2. **Privacy Preservation:** Individual voter choices are encrypted, ensuring complete confidentiality and anonymity, even during vote processing and tallying.
3. **Verifiability and Transparency:** The system includes audit and verification mechanisms that allow voters and election administrators to verify the integrity of the election process and results without compromising security.
4. **Tamper-Resistant Design:** Using cryptographic protocols and secure storage in the database ensures that votes cannot be altered, deleted, or manipulated after being cast.
5. **Scalability:** The system is designed to handle large voter populations and complex elections with multiple regions, candidates, and parties, making it suitable for national and international use.
6. **Efficient Tallying:** With FHE, vote tallying can be performed directly on encrypted data, eliminating the need for decryption until the final results, ensuring both security and computational efficiency.
7. **Voter Authentication:** The integration of an OTP-based voter authentication system adds an additional layer of security, ensuring that only legitimate voters can cast their ballots.
8. **User-Friendly Interface:** The system provides an intuitive interface for both voters and administrators, simplifying the processes of voting, election management, and result viewing.
9. **Reduced Cost and Resources:** By transitioning to an electronic voting system, the need for physical ballot papers, transport, and manual tallying is reduced, leading to cost and resource efficiency.
10. **Increased Trust in Elections:** By ensuring secure, transparent, and accurate voting, the system enhances public confidence in the integrity of elections.

## 4.3 Limitation

The limitation of the system are as follows:

## 4.4 Further Extension

REFERENCES

[1] Avi Kak, “*Web Security: PHP Exploits and the SQL Injection Attack*”, Avinash Kak, Purdue University, May 2, 2015.

[2] Brad Bulger, Jay Greenspan and David Wall, “MySQL / PHP Database Application”, Second Edition, Wiley Publishing Inc, United State of America, 2004.

[3] Channu Kambalyal, “*3-Tier Architecture*”, 2002.

[4] Eckerson, Wayne W. "Three Tier Client/ Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." Open Information Systems 10,1 (January 1995): 3(20)

[5] Ei Maung, “*Professional Web Developer*”, Second Edition, Fairway Web Development, June 2015.

[6] Gerald C.Okereke, “Database Management System”, National Open University of Nigeria, Nigeria, 2009.

[7] Ian Jacobs, “*Architecture of the World Wide Web, Volume One*”, W3C Norman Walsh, Sun Microsystems, Inc., December 2004.

[8] IT SECURITY CENTER (ISEC), “*How to Secure Your Website*”, Fifth Edition, Information-Technology Promotion Agency, Japan, April 2011.

[9] Jin Jiawei, “*Web-Based Library Management System with PHP and MySQL*”, Turku University, May 2011.

[10] Luke Welling, Laura Thompson, “*PHP and MySQL Web Development*”, Second edition, Sams Publishing, 201 West 103rd Street, Indianapolis, February 2003.

[11] Robin Nixon, “*Learning PHP, MySQL, JavaScript, CSS & HTML5*”, Third Edition, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, June 2014.

[12] Taly Sharon and Ariel J. Frank, “*Digital Libraries on the Internet*”, Bar- Ilan University.

[13] Tony Johnson, “*Spinning the World Wide Web*”, Stanford University, 1994.

[14] “What is web server?”, Webdevelopersnotes. 2010-11-23. Retrieved 2010-11-23.

[15] "Website – definition of website by the Free Online Dictionary, Thesaurus and Encyclopedia". Thefreedictionary.com. Retrieved 2011-07-02.

[16] *“*The case for paper: A survey of electronic voting risks and failures”, IEEE Security & Privacy, 6(1), 27-35. Halderman, J. A., & Appel, A. W. (2008).

[17] “Electoral Engineering: Voting Rules and Political Behavior”. Cambridge University Press. Norris, P. (2004).

[18] “Handbook of Applied Cryptography”. CRC Press. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2001).

[19] “A Fully Homomorphic Encryption Scheme”. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 169-178. Gentry, C. (2009).

[20] “Receipt-Free Secret-Ballot Elections”. Proceedings of the 26th ACM Symposium on Theory of Computing, 544-553. Benaloh, J., & Tuinstra, D. (1994).

[21] <https://en.m.wikipedia.org/wiki/Web_page>

[22] <https://en.m.wikipedia.org/wiki/Static_web_page>

[23] <https://en.m.wikipedia.org/wiki/Dynamic_web_page>

[24] <https://www.javatpoint.com/model-1-and-model-2-mvc-architecture>