

Zero-Knowledge Proofs for Secure and Private Voting Systems

24-25J-136

Project Proposal Report

Rafeek A.M

BSc (Hons) in Information Technology specialized in Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology

August 2024

Zero-Knowledge Proofs for Secure and Private Voting Systems

24-25J-136

Project Proposal Report

Rafeek A.M

Supervised by – Mr. Kavinga Yapa Abeywaradana

BSc (Hons) in Information Technology specialized in Cyber Security


Department of Information Technology

Sri Lanka Institute of Information Technology

August 2024

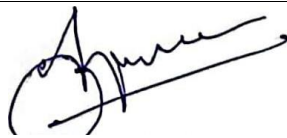
Declaration

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Rafeek A.M	IT21318252	

The supervisor/s should certify the proposal report with the following declaration.

The above candidate is/are carrying out research for undergraduate dissertation under my supervision.

Supervisor	Date	Signature
K.Y. Abeywardena	22 nd of August 2024	

Abstract

Maintaining the integrity of democratic processes depends on ensuring the security and privacy of electronic voting (i-Voting) systems. Advanced cyber risks are difficult to counter with traditional security techniques, particularly when those threats involve unauthorized data transfers from compromised voter PCs. One promising cybersecurity method that makes real-time attack identification and prevention possible is dynamic taint analysis. But there is still a lot of unanswered study regarding its use in i-Voting systems.

By creating a dynamic taint analysis methodology especially for i-Voting systems, this project aims to close this gap. The goal is to monitor and secure data flows during voting sessions by integrating Explicit Propagation Tracking and Control Flow Analysis. This will improve system security and stop unwanted access. The principal aim is to protect i-Voting systems from cyber risks by identifying and addressing unapproved data transfers. Analyzing current i-Voting architectures, creating the taint analysis system, and thoroughly testing it are some of the specific objectives.

As part of the method, a comprehensive system diagram is used to enable the creation of a system architecture that unifies dynamic taint analysis with i-Voting procedures. To guarantee thorough system validation, the project will specify both functional and non-functional requirements in addition to anticipated test cases. A suggested budget will encompass development, software tools, and testing infrastructure, accompanied by cost reasons to guarantee alignment with project goals. With the goal of creating new benchmarks for the security of electronic voting systems, this proposal closes a significant gap in i-Voting security and expands on previous studies.

Table of Contents

Declaration.....	3
Abstract.....	4
Table of Contents	5
Introduction.....	6
Background & Literature survey	7
Research gap.....	8
Research Problem	8
Objectives.....	8
Primary Objective	8
Sub objectives	9
Methodology	10
Tasks and sub tasks	10
Project Requirements	12
Functional Requirements	12
Non - functional Requirements	13
Expected Test Cases	13
Budget and Budget Justification.....	14
Budget Overview	14
Budget justification	14
Conclusion	15
References	16

Introduction

The legitimacy of modern political processes depends on the security and integrity of electronic voting (i-Voting) technologies. The necessity to protect these systems against advanced cyberattacks has grown as they become more widely used. Protecting sensitive voter data from unwanted access and manipulation is the main worry with i-Voting systems, especially in light of vulnerabilities brought about by malware on voter devices. Numerous studies have been conducted over the years to improve the security of i-Voting systems, and one potential method for detecting and preventing such attacks in real-time is dynamic taint analysis.

In the area of software security, dynamic taint analysis is a technique that monitors data flow through a system to identify unauthorized or malicious activity that has been thoroughly examined. TaintCheck [1], Flayer [2], and Dytan [3] are among the early research that established the basis for the use of dynamic taint analysis in software systems for data flow monitoring and control. These research showed how useful taint analysis is for spotting breaches of security, especially when it comes to stopping data leaks and finding attacks. However, there is a lot of undiscovered research potential regarding the application of dynamic taint analysis to i-Voting systems.

One must become proficient in a number of important approaches and procedures in order to work on this project. It is first necessary to have a thorough understanding of dynamic taint analysis, which includes control flow analysis and explicit propagation tracking. These methods, which are the foundation of the suggested framework, allow for the real-time observation of data flows inside the i-Voting system. Furthermore, it is essential to understand the architecture of i-Voting, particularly how voter inputs are handled and safely stored. It is vital to possess knowledge of cybersecurity standards and best practices in order to guarantee that the built system satisfies the stringent requirements for safeguarding electoral procedures.

Dynamic taint analysis has reached a substantial state of the art, with increasingly sophisticated and scalable methods being offered by popular tools such as Bitblaze [4] and libdft [4]. These methods, which have shown considerable promise in detecting and reducing cyber dangers, have mostly been used in general software security and malware detection. Their use in relation to i-Voting systems hasn't been thoroughly investigated, though. Prior studies [6] have mostly concentrated on conventional security mechanisms like encryption and access control to safeguard i-Voting systems. These techniques offer a foundational degree of protection, but they don't deal with the dynamic and ever-changing nature of cyber threats, especially those caused by malware that can change data flows covertly.

By including dynamic taint analysis and emphasizing real-time monitoring and the avoidance of unwanted data flows, this project improves the security of i-Voting. By merging Control Flow Analysis and Explicit Propagation Tracking, it closes a significant research gap and establishes new benchmarks for electronic voting security while providing a stronger defense against cyberattacks.

Background & Literature survey

Adopting i-Voting systems has many advantages, but it also poses major security risks, especially when it comes to malware and other online dangers. To keep the public's faith, voter data integrity and election confidentiality must be guaranteed. Access control and encryption are two examples of traditional security techniques that are ineffective against evolving threats. In software security, dynamic taint analysis follows sensitive data flows to identify unauthorized activity. Although this method works well in other contexts, there isn't much research on how well it works with i-Voting systems.

Studies have demonstrated the effectiveness of using dynamic taint analysis to stop software security vulnerabilities through the use of tools like TaintCheck [1] and Dytan [3]. But these developments haven't yet fully benefited i-Voting systems. Static methods are still used in i-Voting security today, and they are frequently insufficient to fend off changing dangers like advanced persistent threats (APTs).

The goal of this research is to create a dynamic taint analysis framework specifically designed for i-Voting systems by including Control Flow Analysis and Explicit Propagation Tracking. By providing a strong, real-time protection against complex cyberattacks, this strategy seeks to close the research gap and improve the security and integrity of electronic voting systems.

Research gap

There is still an enormous gap in the security of electronic voting (i-Voting) systems when it comes to efficiently detecting and stopping unwanted data flows. The majority of security features in current i-Voting systems, such as access restriction and encryption, are static and insufficient to combat dynamic and ever-evolving cyberthreats like malware. These systems are susceptible to sophisticated assaults that could jeopardize the integrity of the voting process since they lack robust procedures for real-time monitoring and mitigation of data flow irregularities. A key vulnerability in current i-Voting infrastructures is highlighted by the lack of dynamic security measures, specifically those that are able to detect and respond to unwanted data flows as they occur. In order to close this gap, this project will incorporate dynamic taint analysis into i-Voting systems, offering a fresh method for preventing and detecting threats in real time.

Research Problem

Unauthorized data flows pose a significant threat to the security and integrity of i-Voting systems, especially those that originate from voter PCs that have been compromised. Because current security measures do not provide real-time detection and prevention of such malicious activities, the systems are left open to cyber-attacks. This research aims to address the issue by investigating how dynamic taint analysis can be effectively implemented within i-Voting systems to detect and mitigate unauthorized data flows, thereby improving the overall security and reliability of the electoral process.

Objectives

Primary Objective

Creating and implementing a dynamic taint analysis framework especially for i-Voting systems is the main goal of this project. By enabling real-time detection and prevention of unwanted data flows particularly those resulting from compromised voter PCs. This framework seeks to improve the security and integrity of these systems. Throughout the voting process, the framework will monitor data exchanges using cutting-edge techniques like Control Flow Analysis and Explicit Propagation Tracking to make sure that any irregularities or unauthorized activity are quickly detected and addressed. This research aims to provide a more durable and dependable technique for safeguarding the integrity of electronic voting, thereby boosting public confidence in the electoral process, by tackling the present shortcomings in i-Voting security.

Sub objectives

- **Identify critical components and data flows that require protection.**

The first goal of this research is to identify the crucial parts and data flows that are most susceptible to cyberattacks by performing a thorough architectural analysis of the current i-Voting systems. In order to detect potential exposure spots, this entails mapping out the full data flow within the system, from voter input to secure storage. Sensitive areas that could jeopardize the integrity and secrecy of the voting process, like voter verification, vote transmission, and data storage, will receive extra attention. By comprehending these weaknesses, the study will help to ensure that the dynamic taint analysis framework is designed with the most important components of the i-Voting system in mind.

- **Identify potential malware and cyber threat vectors that could exploit vulnerabilities in the voting system.**

Finding and analyzing the many kinds of malware and cyberthreat vectors that might be able to take advantage of the weaknesses in i-Voting systems is the second sub-objective. This entails developing a thorough threat model that describes the various ways in which malware could compromise voter computers, alter data transmissions, or interfere with the electoral process. Advanced persistent threats (APTs) and other sophisticated attack vectors that have been noted in comparable circumstances will also be taken into account by the threat model. Through comprehending these dangers, the study will lay the groundwork for creating mitigation plans inside the dynamic taint analysis framework, guaranteeing that it is capable of identifying and eliminating a variety of online risks.

- **Develop a dynamic taint analysis framework that combines Explicit Propagation Tracking and Control Flow Analysis for real-time monitoring.**

The creation of a dynamic framework for taint analysis that combines control flow analysis and explicit propagation tracking is the third sub-objective. The purpose of this combination is to improve the capacity to identify illegal alterations, stop any security breaches, and monitor data flows in real-time. As data moves through the system, explicit propagation tracking will make it possible to monitor it precisely, and control flow analysis will make sure that any deviations from the intended control flow are identified. The idea is to provide a platform that can identify illicit data transfers and offer the resources needed to track down and neutralize these threats as they arise.

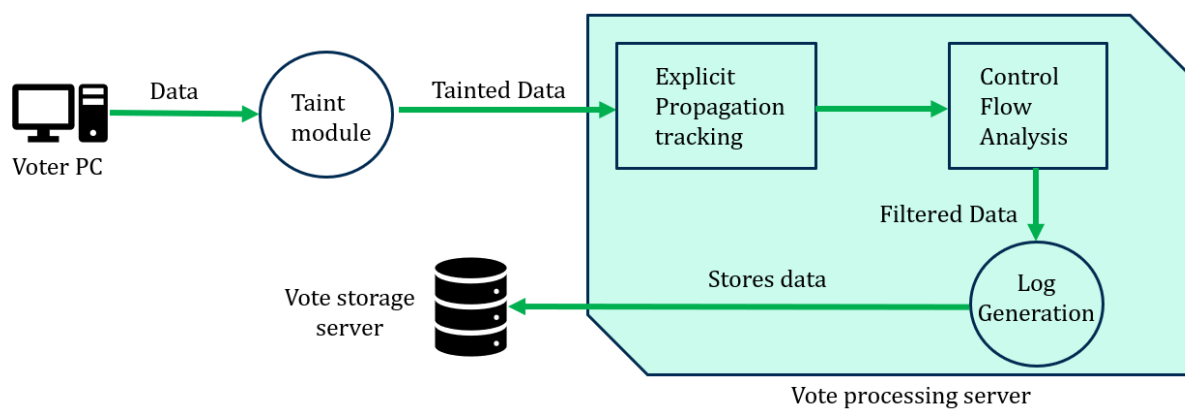
- **Ensure that the integrity of the voting process is maintained without compromising efficiency after implementation.**

The last sub-goal is to evaluate and guarantee that the i-Voting system's usability and efficiency are not jeopardized by the application of the dynamic taint analysis framework. For the purpose of determining how the framework affects system latency, resource usage, and overall user experience, extensive testing and performance assessments must

be carried out. The aim is to keep the system functional, scalable, and user-friendly while upholding the strict security and integrity criteria necessary for the voting process. The i-Voting system must strike a balance between security and performance in order to be deployed practically in actual electoral settings.

Methodology

Below shown is the system architecture of the implementation of dynamic taint analysis framework.



- The vote is entered via the interface.
- Upon entry the data is marked as tainted (i.e. entry from keyboard or mouse).
- An explicit propagation tool is used to filter data as it flows through the system.
- Furthermore, CFGs are produced to further analyze data, detect anomalies and identify unauthorized data flows.
- Once the data is ready to store a log is generated.
- And then the filtered data is stored.

Tasks and sub tasks

System Analysis and Design

- **Analyze Existing i-Voting Architectures**

Examine all of the present i-Voting systems in detail to find the crucial parts and data flows that are vulnerable to cyberattacks.

Draw a flowchart showing the data flow from voter input to safe storage, emphasizing any possible areas of compromise.

- **Modeling of Threats**

Determine which malware and cyberthreat vectors could take advantage of holes in the i-Voting system.

Create a threat model that describes the many kinds of threats and how they affect the system.

- **System Architecture**

Create the general system architecture for the dynamic taint analysis framework by combining the methodologies of Control Flow Analysis (CFG) and Explicit Propagation Tracking.

To see how data moves across the i-Voting infrastructure and how taint analysis tools are integrated, draw a system diagram.

Development of the Dynamic Taint Analysis Framework

- **Explicit Propagation Tracking implementation**

Create new explicit propagation tools or modify current ones to identify data as contaminated as soon as it enters the system (such as keyboard or mouse input).

Install a filtering mechanism that watches over data as it passes through the system, looking for irregularities or illegal changes.

- **Combining Control Flow Analysis**

For the i-Voting application, create Control Flow Graphs (CFGs) to track and examine the system's control flow.

In order to discover potential security breaches, integrate CFGs with the taint analysis framework to detect deviations from expected control flows.

- **Data Storage and Logging**

Establish a logging system to document all actions such as irregularities in data flow and security alerts that the taint analysis framework takes.

Provide safe data storage procedures that guarantee the voting process's integrity is maintained by storing only validated, filtered data.

Testing and Evaluation

- **Testing in a Controlled Environment**

To find out how well the dynamic taint analysis system finds and stops illegal data flows, run tests in a simulated environment.

Examine the system's performance, paying particular attention to anomaly detection accuracy, latency, and resource usage.

Documentation

- **Documentation of the Framework**

Document the design, implementation, and testing processes of the dynamic taint analysis framework.

Prepare technical documentation that outlines the system's architecture, functionalities, and security protocols.

Design and Gathering of Data

The information needed for this project consists of:

- System Architecture Data - Details on the data flows and current i-Voting systems.
- Threat Model Data - Information on known cyberthreats and malware that is pertinent to i-Voting systems.
- Performance metrics - Information gathered from system testing, such as accuracy of detection, latency, and resource utilization.

A combination of testing in a controlled environment, simulations, and literature research will be used to gather data. Because the main focus of the project is on technical implementation and system testing, neither surveys nor interviews are necessary.

Anticipated Conclusion

When the research is finished, a strong dynamic taint analysis framework that greatly improves the security of i-Voting systems is anticipated. The integrity and dependability of electronic voting procedures will be guaranteed by this framework's real-time identification and prevention of unwanted data flows. When this framework is successfully put into practice, it has the potential to raise the bar for i-Voting security by providing a scalable defense against changing cyberthreats.

Project Requirements

Functional Requirements

Libdft Tool for Analysis of Dynamic Taints

The i-Voting system's data flows will be tracked and monitored by Libdft, which will enable real-time detection of any illegal changes or anomalies.

Using CFG for Control Flow Analysis construction

The building of a Control Flow Graph (CFG) will be incorporated to monitor and analyze the control flow within the i-Voting application, looking for aberrations that might point to security breaches.

Python-Based Coding

Because of its many libraries and versatility, Python will be the main language utilized to construct the dynamic taint analysis framework.

Burp Suite for Examining Security

The dynamic taint analysis framework's efficacy will be verified and vulnerabilities will be found with the use of Burp Suite for security testing.

Visual Studio Code as IDE

The framework will be coded, debugged, and tested using Visual Studio Code (VS Code), which is flexible and supports a number of languages and extensions.

The ELK Stack for Monitoring and Logging

Real-time data will be gathered, stored, and visualized using the ELK Stack (Elasticsearch, Logstash, Kibana), which will enable efficient logging and security and system performance monitoring.

Non - functional Requirements

The Linux operating system

The operating system for creating and implementing the framework will be Linux, which was selected due to its stability, security, and compatibility with open-source technologies.

Sandbox Environment

Secure, isolated testing will be conducted in a sandbox environment, which allows safe simulation of real-world circumstances without jeopardizing the voting system itself.

High-Performance Servers

To meet the computational demands of the framework and provide effective data processing with low latency, high-performance servers would be needed.

Expected Test Cases

Testing for Taint Propagation

Test scenarios will assess the system's accuracy in tracking tainted data as it flows through the various i-Voting system components.

System Efficiency and Resource Utilization

Performance tests will be used to make that the framework does not impair the voting system's functionality by monitoring the system's response time, resource usage, and general efficiency.

Simulated Security Breach

To evaluate the framework's efficacy in identifying and thwarting actual threats, simulated security breaches will be carried out, which will aid in the improvement of its defensive capabilities.

Budget and Budget Justification

Budget Overview

Since the majority of the resources needed for development, testing, and implementation are open-sourced and publicly available, the project's overall budget is quite small. The following are the main parts of the budget:

Tools and Software

Libdft Tool: Open-source and free software.

CFG Tools for construction and analysis: open-source and free.

Python is open-source and free.

Burp Suite (Version Community): No cost.

Open-source and free is Visual Studio Code.

ELK Stack: Open-source and free.

Infrastructure

High-Performance Servers: The project can use high-performance servers at no expense because the institution grants access to Azure services.

Budget justification

The project's cost-effectiveness is maintained while its goals are met owing to the thoughtful application of open-source technologies and resources. The project may acquire the required infrastructure and processing power without going over budget by utilizing the university's Azure services. This strategy is an effective and long-lasting model for research and development since it not only maximizes the utilization of available resources but also conforms to budgetary restrictions.

Software & Tools

The project can move on without incurring any additional expenses or licensing since all required tools are freely available and well-supported in the development community.

High-Performance Servers

Although these are usually expensive, the university's use of Azure services makes it possible to obtain powerful computational resources for free, guaranteeing that the project can meet the demands of dynamic taint analysis without going over budget.

To summarize, the budget exhibits selective resource management, leveraging open-source and free software and the university's infrastructure to mitigate potential high-performance computing costs, thereby minimizing financial obstacles and enabling the project to concentrate on research and development.

Conclusion

In conclusion, the goal of this project proposal is to develop a dynamic taint analysis framework as a comprehensive means of improving the security and integrity of i-Voting systems. Through the application of sophisticated techniques like Control Flow Analysis and Explicit Propagation Tracking, the proposed framework addresses the serious vulnerabilities caused by unauthorized data flows, particularly those originating from compromised voter devices, and offers real-time monitoring and security breach prevention. The use of the university's Azure services in conjunction with a careful selection of open-source tools guarantees that the project will stay technically sound and economical. In the end, this framework's effective implementation could establish new benchmarks for i-Voting security, guaranteeing a more dependable and safe electoral procedure.

References

- [1] D. S. James Newsome, "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software.," NDSS, 2005.
- [2] D. D. D. E. Cristian Cadar, "Unassisted and automatic generation of high-coverage tests for complex systems programs.," OSDI.
- [3] J. L. W. & O. A. Clause, "Dytan: A generic dynamic taint analysis framework.," ISSTA, 2007.
- [4] D. e. a. Song, "Bitblaze: A new approach to computer security via binary analysis.," ICISS, 2008.
- [5] A. & B. D. Kemper, "Transparent operating system support for dynamic taint tracking.," RAID, 2012.
- [6] O. & K. C. Cetinkaya, "Survey on electronic voting systems.," IET Information Security., 2017.