**CSCI-6908 - Individual Research**

**CERTUS Voting**

**Statement of Work**

**Supervisor:** ProfessorMichael Clarkson

**Team:**

|  |  |  |
| --- | --- | --- |
| Ahmad Kouraiem | ahmadko@gmail.com | 703-477-8333 |
| Dmitriy Karmazin | dkarmazi@gwu.edu | 202-758-6826 |
| Hirosh Wickramasuriya | hirosh@gwmail.gwu.edu | 202-594-5448 |
| Steven Frink | sfrink1@gmail.com | 518-478-5880 |

**1. Introduction**

The purpose of this document is to define a final product we intend to build within the scope of this project, define what will be left out of scope of the project and all documentation and other essential deliverables to be completed with the project.

This document will also include a note on the grading and success evaluation.

CERTUS is an abbreviation that stands for Confidential Electronic Remote Trusted U? System.

**2. Goals**

CERTUS has been derived as a simplified basis for Du-Vote: Remote Electronic Voting with Untrusted Computers with key system components and architecture designed for easy compatibility and conversion to the latter in the future. We will implement a simple “double envelope” voting protocol rather than the full Du-Vote protocol.

This project is an implementation of CERTUS. The ultimate goal is to implement the system modules and graphical user interfaces described in the section 3 of this document compliant with section 5 and section 6 of this document. The system must be implemented in a secure manner following the principles of CIA.

The final system will be a fully functional voting system, although not all components of the system (detailed below) will be made fully secure in this project. The unfinished pieces will be designed as “black boxes”, so that another team could effectively complete the work in the future.

**3. Major System Components and their description**

1. Server - the server part of the system will include the following modules and functionalities:
   1. Authentication of users
   2. Authorization of users
   3. RMI backend for all platform functionality
      1. Users registration, profile management, forgotten password
      2. Users management by administrator
      3. Users key binding
      4. Elections management
      5. Voting (assigning the vote to a candidate) and vote verification
      6. Publishing election results
   4. Tallying of votes
2. Platform
   1. Voters registration
   2. Voters profile and keys management
   3. Voters forgotten password
   4. Voters management by administrator
   5. Voters authentication to the server
   6. Voters authorization to the server
   7. Election management (start a new election, stop election, candidates management, monitor election progress)
   8. Invitation of voters to register to the system
   9. Voting and encryption of votes
   10. Displaying results
3. Token
   1. Computation of digital signatures for encrypted votes
   2. Generation of key pairs protected by specified password

**5. Documentation**

Another important part that has to accompany the product is documentation which should contain the following deliverables:

1. Statement of work
2. Project requirements document
3. Design document - includes details on any decisions made along the development process regarding architecture of the system.
4. Installation instructions - instructions on how to install and run the system, change essential security and functional parameters.

**6. Development Requirements**

The defining timeframe for this project is one semester Spring 2014. The product must be delivered within the following time schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Phase** | **What to deliver** | **Deadline** |
| 1 | Requirements | User stories, Security Goals | Feb 15th (3 weeks) |
| 2 | Prototype | Established infrastructure of the project, basic user story that involves all components of the system | March 1st (2 weeks), March 15th (2 weeks) |
| 3 | Alpha | Simple voting functionality, all user stories directly related to voting | March 29th (2 weeks) |
| 4 | Beta | Complete system functionality with all user stories finished. All feedback from Alpha phase. | April 12th (2 weeks) |
| 5 | Final | All additional user stories which were not planned initially, all feedback from Beta phase. | April 26th (2 weeks) |
| 6 | Post Final | All documentation, junit testing and feedback from Final phase. | May 1st (1 week) |

The code must be well commented, clear and readable. Code must go through testing and assurance procedures before it can be considered trustworthy.

**7. Left out of scope of this project**

The following items will be left out of scope of this project:

* Secure Bulletin Board
* Back end and secure tallying protocol. However, we intend to build a simplified, not necessarily secure, tallying protocol which should be able to correctly determine and display the results of the election in order to enhance testing of the system.
* Manufacture of the token

**8. Technologies**

We propose a web-based application with the following main components:

1. Server S
   1. Standalone Java based service
   2. Database server (MySQL)
2. Platform P
   1. Web application Apache Tomcat, JSP which will communicate with S
   2. User interaction (encrypts votes)
3. Token H
   1. Java based standalone application which will simulate the operation of the hardware token
   2. Takes private key and encrypted vote as input
   3. Returns signed encrypted vote as output

HTML Zurb Foundation Framework

Jquery 2.1

Javascript

CSS

Java Version : 1.7

MySQL Version : 5.6.x

Tomcat : 7.0.x

IDE : Eclipse (Current Release is Kepler 4.3.1)

Source Control : GitHub

The communication between the main voting system components should be secured when applicable. The primary programming language for implementation of the system is Java.

**9. Measuring success**

The success of the project should be evaluated by the following criteria:

|  |  |
| --- | --- |
| Grade | Qualifying Criteria |
| B | Insecure functional voting system, including hardware token simulator, but lacking user interfaces for management and registration |
| B+ | Double envelope protocol |
| A | All user interfaces completed |
| A+ | Implementation of additional users stories:  - invitation of unregistered users to sign up for the system  - registration of invited users in the system  - restoring access to the system for invited users |