МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Электронное голосование

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Пензина Александра Сергевича

Научный руководитель
ассистент _____ А. А. Лобов
22.01.2022 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Голосование – это способ принятия решений группой людей, в котором наиболее общее мнение выбирается путем подсчета волеизъявлений каждого из избирателей.

С развитием информационных технологий, стало возможно проводить электронное голосование.

Электронное голосование – процедура принятия решения с использованием специального электронного оборудования для голосования, подсчета голосов и оглашения результата. Одной из разновидностей такой процедуры голосования являются выборы через сеть интернет.

Технология электронных голосований позволяет сократить время, которые требуется для подсчета голосов, а также облегчить процедуру голосования людям, которым не нужно приходить на избирательный участок.

В некоторых странах, например, Эстонии, Бельгии, Франции, Норвегии и других, возможность электронного голосования предусмотрена законодательством.

Цель данной работы: реализовать систему электронного голосования.

Решаемые задачи:

- рассмотреть протокол тайного голосования;
- разработать систему электронного голосования, реализующую протокол тайного голосования;
- продумать механизм развертывания системы и выбрать инструменты для его реализации;
- реализовать развертывание разработанной системы электронного голосования.

Дипломная работа состоит из введения, 3 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 57 страниц, из них 40 страниц – основное содержание, включая 33 рисунка, 1 таблицу, список использованных источников из 14 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Протокол тайного голосования

Введем обозначения: A – агентство, проводящее электронное голосование; E – избиратель, легитимный участник голосования, B – цифровой бюллетень. B содержит выбор избирателя E и необходимые для безопасности протокола данные.

Рассмотрим протокол тайного голосования, в основе которого лежит использование ассиметричного шифрования и электронная подпись.

Протокол*:*

Шаг 1. A создаёт открытый и закрытый ключи (a_{public} и $a_{private}$) и выкладывает в общий доступ a_{public}. Кто угодно может зашифровать сообщение при помощи a_{public}, но расшифровать его сможет только А.

Шаг 2. E создает собственные ключи ЭЦП (e_{public} и $e_{private}$) и публикует e_{public}. Кто угодно может проверить подписанное сообщение E, но подписать сообщение – только сам E.

Шаг 3. E формирует сообщение B, где тем или иным образом выражает свою волю.

Шаг 4. E подписывает сообщение личным закрытым ключом $e_{private}$.

Шаг 5. E шифрует подписанное сообщение открытым ключом a_{public}.

Шаг 6. E отправляет шифрованное сообщение A.

Шаг 7. A собирает сообщения.

Шаг 8. A расшифровывает полученные сообщения при помощи закрытого ключа $a_{private}$ и проверяет подпись при помощи лежащего в открытом доступе e_{public}.

Шаг 9. A производит подсчет голосов и выкладывает результаты.

2 Реализация системы электронного голосования

В ходе работы было разработано веб-приложение, реализующее протокол тайного голосования, предоставляющий возможности пользователям зарегистрироваться, войти в систему, проголосовать по

выбранному голосованию и посмотреть результаты голосования. Также предусмотрена возможность создавать голосования для заранее зарегистрированного пользователя-администратора.

2.1 Описание приложения

При переходе на главную страницу сайта, пользователю предлагается зарегистрироваться или войти в систему.

При успешной регистрации пользователя на стороне клиента генерируется пара ключей (e_{public} и $e_{private}$) пользователя, а на стороне сервера email, хэш от пароля и открытый ключ e_{public} сохраняются в базе данных.

При успешной аутентификации, пользователь переходит на страницу со списком голосований. На рисунке 1 представлена страница со списком голосований. На этой странице отображается список голосований и переходы к действиям по этим голосованиям: цифрой 1 отмечен переход к странице просмотра результатов голосования, цифрой 2 – странице отправки голоса, цифрой 3 – странице с полученными сервером бюллетенями по данному голосованию. Цифрой 4 отмечена кнопка «Создать голосование», которая отображается только, если авторизованный пользователь – администратор.

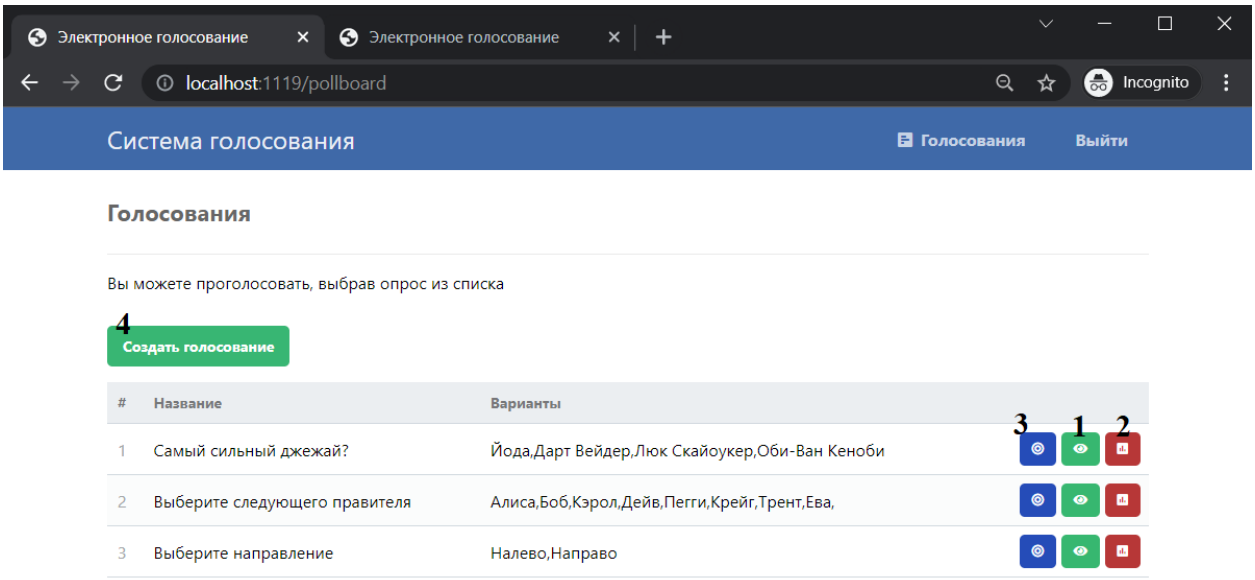

Рисунок 1 – Страница со списком голосований

На странице отправки голоса по выбранному голосованию, представленной на рисунке 2, отображены варианты ответа и кнопка «Отправить голос».

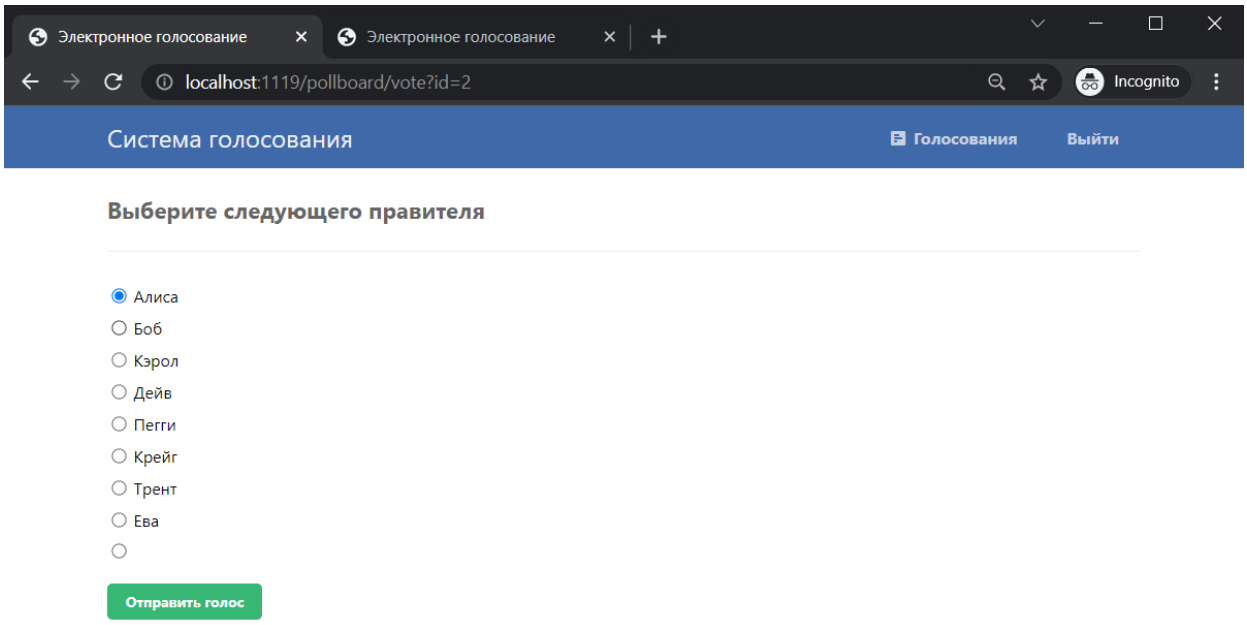

Рисунок 2 – Страница отправки голоса по выбранному голосованию

При нажатии кнопки «Отправить голос» на стороне клиента выбранный вариант ответа B подписывается закрытым ключом пользователя $e_{private}$, после чего зашифровывается открытым ключом голосования a_{public}. Далее сообщение $m = encrypt(a_{public}, sign(e_{private}, B))$ отправляется на сервер.

Приняв это сообщение, сервер расшифровывает его, используя закрытый ключ голосования $a_{private}$, который хранится только в базе данных сервера. Далее сервер проверяет подпись при помощи открытого ключа пользователя e_{public}. В случае, если подпись не прошла проверку, полученный ответ со стороны клиента игнорируется. Иначе производится проверка, был ли учтен голос от данного пользователя по текущему голосованию. Если не был – голос пользователя учитывается. Иначе – полученный ответ со стороны клиента игнорируется.

2.2 Используемые программные компоненты

В процессе работы над проектом были использованы следующие программные компоненты: язык программирования JavaScript, программная платформа Node.js, база данных MongoDB.

В таблице 1 представлены использованные в проекте сторонние пакеты Node.JS.

Таблица 1 – использованные сторонние пакеты Node.JS

Название пакета	Версия	Назначение
bcrypt	5.0.1	Генерация соли и вычисления хэша от введенного пользователем пароля на сайте
connect-flash	0.1.1	Хранение сообщений от сервера для клиента до момента их отправки
ejs	3.1.16	Генерация разметки HTML с использованием JavaScript
express	4.17.2	Каркас всего приложения
mongoose	5.13.13	Работа с базой данных MongoDB
passport	0.4.1	Аутентификация пользователей, ограничение доступа без авторизации
passport-local	1.0.0	
openpgp	5.0.1	Криптография с открытым ключом

2.3 Структура проекта

На рисунке 3 представлена структура проекта.

Рисунок 3 – Структура проекта

Файл в корне проекта app.js – основная точка входа приложения. В нём импортируются все сторонние зависимости Node.js и определяется соединение с базой данных.

Файлы в корне проекта package.json и package-lock.json содержит названия всех подключаемых сторонних библиотек Node.js

В каталоге models содержатся схемы моделей пользователя, голосования и зашифрованного бюллетеня.

Каталог public содержит JavaSctipt файлы с кодом, выполняемым на стороне клиента – генерация ключей, подпись и шифрование бюллетеня.

Каталог routes содержит 2 файла: users.js и index.js. В этих файлах описаны действия сервера по обработке HTTP запросов клиента к серверу. В файле users.js – обработка запросов регистрации пользователей и входа в систему. В файле index.js – обработка запросов, связанных с голосованиями, таких как получение списка голосований, отправка голоса и просмотр результатов.

Каталог views содержит ejs файлы, которые выполняют генерацию HTML страниц с использованием JavaScript.

3 Развертывание системы

3.1 Простое развертывание системы

Чтобы запустить веб-сервер на машине с установленными программной платформой Node.js и базой данных MongoDB, в первую очередь необходимо установить используемые приложением сторонние пакеты Node.js.

Далее, для запуска веб-сервера, достаточно открыть терминал, перейти в корень проекта и выполнить команду: 'node app localhost', где localhost – это адрес хоста, на котором установлена база данных MongoDB.

3.2 Docker

Docker — инструмент для разработки, доставки и эксплуатации приложений с использованием контейнеров. С помощью Docker можно отделить приложение от инфраструктуры и обращаться с инфраструктурой как с управляемым приложением. Используя контейнеры, разработчики могут собирать приложение со всеми зависимостями и разворачивать его как один пакет.

3.3 Объекты Docker

Docker образ — это шаблон, предназначенный только для чтения, с инструкцией по созданию контейнера Docker. Образ можно получить из реестра и использовать его, никак не изменяя. Или можно добавить к нему дополнительную инструкцию. Инструкция образа – это последовательность действий, каждое из которых выполняется при сборке образа.

Инструкция должна быть записана в файл Dockerfile. В процессе сборки образа Docker обращается к этому файлу, последовательно выполняет все содержащиеся в нем действия и на выходе выдает конечный образ.

Docker Контейнер – это запущенная модель образа, внутри которой работают все приложения с их окружениями. С контейнерами можно совершать такие действия как: создание, запуск, остановка, перемещение и удаление.

3.4 Развертывание приложения с использованием Docker.

В ходе работы было развернуто 2 Docker контейнера, общающихся между собой: контейнер базы данных MongoDB и контейнер веб-сервера. Раззвертывание приложения осуществлялось в операционной системе Windows 10 с использованием подсистемы Linux для Windows WSL.

Правильность работы веб-сервера была проверена тестированием сценария голосования. По итогам проверки можно сделать вывод об успешном развертывании разработанного приложения в Docker контейнеры и о том, что разработанное приложение можно запустить на любом компьютере, на котором установлен Docker.

ЗАКЛЮЧЕНИЕ

В работе был рассмотрен протокол тайного голосования.

В практической части работы была спроектирована и разработана система электронного голосования в виде веб-приложения на языке программирования JavaScript. Серверная часть приложения была реализована на платформе Node.js в сочетании с базой данных MongoDB. Также был изучен механизм развертывания системы Docker, и было реализовано развертывание разработанного веб-приложения с использованием Docker.

Поставленные задачи полностью решены.

Разработанное веб-приложение готово к развёртыванию на любом компьютере, на котором установлен Docker и может быть использовано для организации тайного электронного голосования.