УДК 614.849

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ЭЛЕКТРОННОГО ГОЛОСОВАНИЯ

Палеха Е. В., Черкесова Л. В., Сафарьян О. А.
Донской государственный технический университет, Ростов-на-Дону, Российская Федерация
v.manuilov2010@spark-mail.ru
sergeeva_ga@mail.ru

Приведена практическая реализация системы электронного голосования. Проведен анализ разработанного алгоритма, который был взят за основу при практической реализации программного продукта. Разработанный программный продукт может применяться для проведения электронного голосования на выборах.

Ключевые слова: электронное голосование, программный продукт, веб-приложение, криптосистема Эль-Гамаля, электронно-цифровая подпись.

UDC 004.41

DESIGN AND DEVELOPMENT OF ELECTRONIC VOTING SYSTEM

E. V. Palekha, L. V. Cherkesova, O. A. Safaryan
Don State Technical University, Rostov-on-Don, Russian Federation

v.manuilov2010@spark-mail.ru
sergeeva_ga@mail.ru

The paper presents practical implementation of an electronic voting system. The analysis of the developed algorithm was carried out, which was taken as the basis in practical implementation of the software product. The developed software product can be used for electronic voting at the elections.

Keywords: electronic voting, software product, web application, El–Gamal encryption system, digital signature.

Введение. В настоящее время проведение голосования путем ручного подсчета бюллетеней является довольно трудоемким, ресурсозатратным и кропотливым процессом. Организация проведения выборов на избирательных участках отнимает много времени и не всегда может обеспечить прозрачность и честность голосования. К тому же некоторые граждане в силу веских обстоятельств не могут прибыть к месту голосования.

В данное время компьютеры используются повсеместно. Поэтому вполне целесообразно их применение и при проведении выборов. Использование компьютерных систем не только сделает процесс голосования более удобным, но также сможет защитить его от фальсификации результатов. Рассмотрим одну из систем электронного голосования.

В данной системе принимают участие три стороны: избиратели, модераторы и администратор. Ни один из модераторов не имеет возможности подменить голос некоторого избирателя. Предотвратить изменение результата голосования можно путем использования усиленного варианта криптосистемы Эль-Гамаля [1], а также путем запрета доступа администратора и модераторов к той или иной таблице базы данных. Тем самым можно снизить вероятность фальсификации голосования.

Для подмены результатов злоумышленникам не только необходимо будет получить доступ к базе данных, но и взломать используемую криптосистему.

Постановка задачи. В работе описана разработка приложения, которое организует и сопровождает процесс голосования. Для того, чтобы приложение функционировало корректно на всех популярных операционных системах было принято решение разработать веб-приложение, так

как оно обеспечит максимальную кроссплатформенность и избавит от сложностей при использовании разных операционных систем — Windows, Linux, MacOS, Android и др. Целью данного приложения является обеспечение процесса голосования и контроль за процессом голосования.

Теоретическая часть. Криптосистема, которая используется в данном приложении, основана на усиленном варианте схемы Эль-Гамаля. Обоснование сложности схемы было приведено в статье [1]. Часть параметров генерируется согласно описанной в данной статье схеме. Остальные параметры генерирует администратор, за исключением оговоренных случаев. Проверяющих разделяем на команды по t человек, учитывая, что $n = kt$, где $n \in \mathrm{N}$ есть некоторое натуральное число, $k \in Z_+$ — целое положительное число. Для дешифрования потребуется участие t проверяющих, где $2 \le t \le n$.

Происходит генерация секретного ключа x. Далее делим x на секретные доли, которые получит каждый из проверяющих результаты голосования. После этого администратор публикует вторую часть открытого ключа в виде $k_o = (p, g, y = g^x)$, который будет использоваться голосующими для шифрования результатов своего голосования.

В выборах участвуют $v \in \mathrm{N}$ кандидатов, избиратели могут отдать свой голос только за одного из кандидатов. Избиратель голосует, происходит шифрование его голоса, а полученный шифротекст отправляется людям, проверяющим результаты выборов. После проведения голосования проверяющие, получившие все шифротексты, восстанавливают секретный ключ x и расшифровывают векторы — голоса. Далее все векторы суммируются, полученный вектор объявляется результатом голосования.

Сложность взлома построенной криптосистемы является эквивалентной сложности решения общепризнанно трудной задачи принятий решения Диффи-Хеллмана в группе G [2].

Данная криптосистема является надежной, но ее практическая реализация является весьма трудоемкой. Исходя из этого при разработке веб-приложения была использована лишь часть этой схемы, а именно электронно-цифровая подпись Эль-Гамаля. Третьей стороной, участвующей в схеме, разработанной в [1] были проверяющие. При практической реализации они были заменены на модераторов. Электронно-цифровая подпись Эль-Гамаля применяется администратором к уникальному идентификатору избирателя и его голосу.

Практическая часть. Веб-приложение — клиент-серверное приложение, при котором клиент обращается к веб-серверу с запросом через браузер и получает ответ в виде HTML-страницы. Данные преимущественно хранятся на сервере, а обмен информации осуществляется по сети.

Разработанное приложение основано на концепции Model-View-Controller. Общая схема MVC представлена на рис. 1. Это схема разделения данных приложения, пользовательского интерфейса и логики его работы на три компонента:

– Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние;

– Представление (View) отображает полученные данные из модели пользователю, реагирует на изменение модели;

– Контроллер (Controller) является связующем звеном между моделью и представлением, оповещает модель о реагировании пользователя.

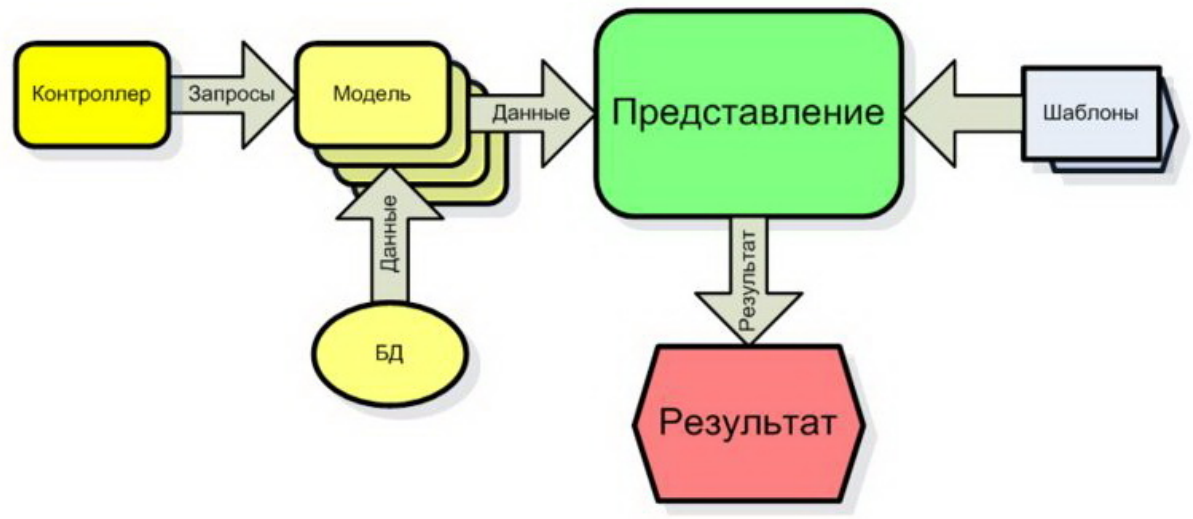

Рис. 1. Общая схема концепции MVC

В данном приложении используется одна из модификаций концепции — иерархическая, а именно HMVC. Она обычно используется с применением объектно-ориентированных средств языка. Суть ее заключается в том, что для каждой из концепций описываются три главных класса: общий класс модели, общий класс контроллера и общий класс представления. Далее при разработке описываются новые модели, которые наследуют свойства и методы общего класса модели. Аналогичные действия применяются к новым классам моделей и представлений.

Рассмотрим корневой каталог приложения, содержащий проект. Он изображен на рис. 2.

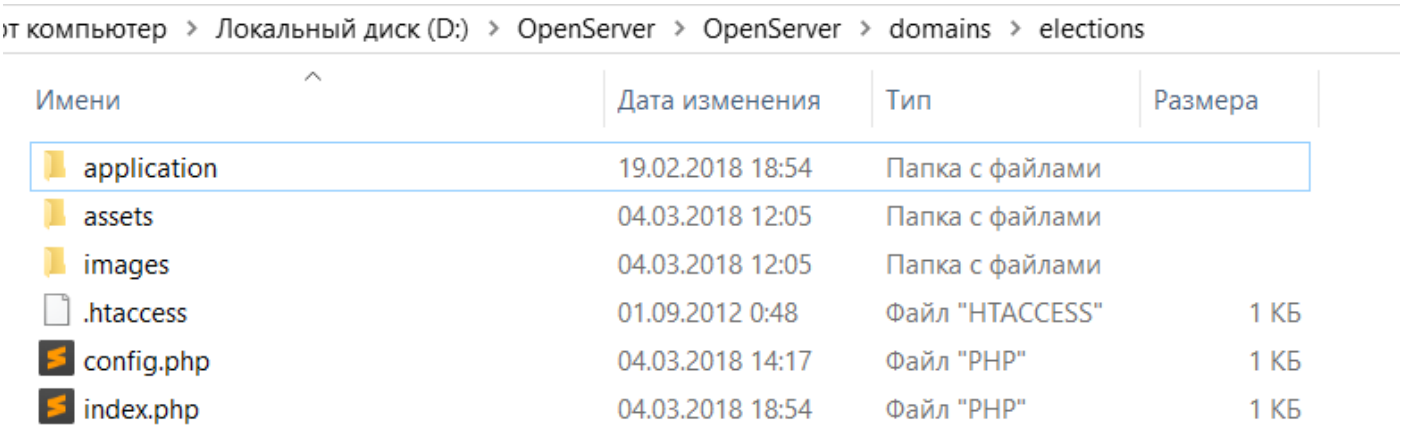

Рис. 2. Корневой каталог проекта

Концепция MVC подразумевает одну точку входа — файл index.php. Этот файл — хребет проекта. Через этот скрипт будут проходить все запросы к веб-приложению и вся логика проекта. Для того, чтобы реализовать такой подход, необходимо настроить работу сервера. Подразумевается, что сайт работает на сервере apache, именно для этого необходим файл .htaccess, который содержит правила маршрутизации URL. Также маршрутизация позволит создать человеко-понятные URL таким образом, что адрес страниц будет иметь вид: project.ru/view/action. В этом примере вызывается метод action класса контроллера view,

Файл config.php — файл настроек проекта, который содержит константы, такие как полный путь к проекту и константы подключения к базе данных (пользователь, пароль, хост, название базы данных).

В папке assets находятся файлы стилей и js-скрипты, а в папке images —изображения для проекта.

Ядром проекта является папка application. Она содержит всю структуру MVC концепции. Содержимое папки application показано на рис. 3.

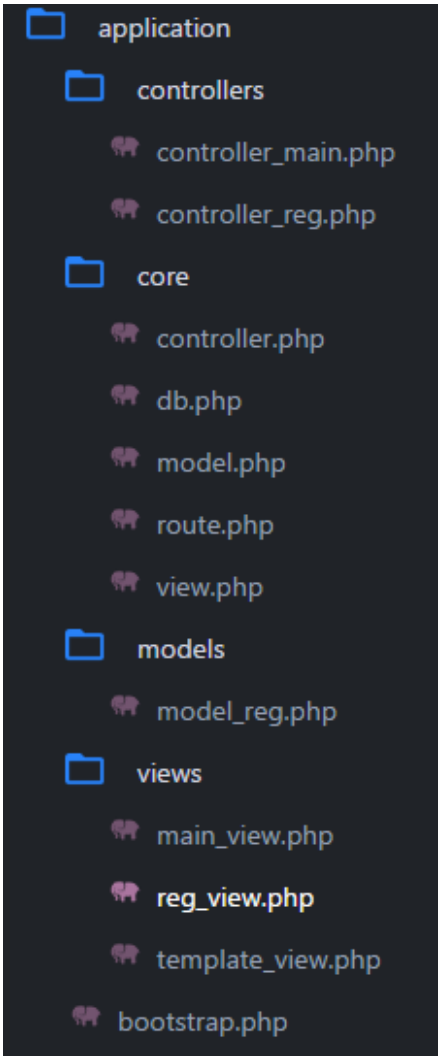

Рис. 3. Содержимое каталога application

В этой папке находится файл bootstrap.php. Он собирает в себя все файлы из ядра проекта. Все файлы папки core подключаются в файл bootstrap.php и вызывается маршрутизатор, который будет руководить приложением. Класс маршрутизатора описан в файле route.php и показан на рис. 4.

Задача маршрутизатора — выделить из адресной строки необходимый контроллер, вызвать указанный метод этого контроллера и предоставить пользователю выходные данные с помощью указанного вида. Для упрощения разработки и читаемости кода имена контроллеров, их моделей и представлений имеют одинаковые названия, за исключением префиксов. Также учитывается то, что адрес может быть намеренно изменен и искомого контроллера не существует. Для этого используется значение по умолчанию. Если не существует указанного метода, тогда выводится сообщение об ошибке. Стоит отметить, что файла модели может и не быть, а обязательным является только наличие контроллера.

```php
//  контроллер и действие по умолчанию
$controller_name = 'Main';
$action_name = 'index';
$routes = explode('/', $_SERVER['REQUEST_URI']);

// получаем имя контроллера
if(!empty($routes[1]))
{
    $controller_name = $routes[1];
}
if(!empty($routes[2]))
{
    $action_name = $routes[2];
}
//  добавляем префиксы
$model_name = 'Model_'.$controller_name;
$controller_name = 'Controller_'.$controller_name;
$action_name = 'action_'.$action_name;
//  подцепляем файл с классом модели (файла модели может и не быть)
$model_file = strtolower($model_name).'.php';
$model_path = 'application/models/'.$model_file;
if(file_exists($model_path)){
    include 'application/models/'.$model_file;
}
$controller_file = strtolower($controller_name).'.php';
$controller_path = 'application/controllers/'.$controller_file;
if(file_exists($controller_path))
{
    include 'application/controllers/'.$controller_file;
}
else
{
    exit('Файл контроллера не найден');
}
//  создаем контроллер
$controller = new $controller_name;
$action = $action_name;
if(method_exists($controller, $action))
{
    //  вызываем действия контроллера
    $controller->$action();
```

Рис. 4. Код класса Route

Общий класс контроллера Controller содержит в себе конструктор, в котором объявляется класс представления. Также описывается абстрактный класс action_index(), который необходим, чтобы при наследовании каждый отдельный контроллер мог выполнять свои задачи.

Описание общего класса представления содержит в себе лишь метод generate ($content_view, $template_view, $data = null). Этот метод подключает шаблон страницы $template_view, содержимое которой наполняется страницей $content_view, а параметр $data представляет собой массив, который содержит данные, полученные из модели. Стоит отметить, что шаблон страницы — значение по умолчанию — template_view.php, код которого описан на рис. 5.

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>Выборы лучшего кандидата</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <!--[if lte IE 8]><script src="<?=SITE_PATH?>assets/js/ie/html5shiv.js"></script><![endif]-->
        <link rel="stylesheet" href="<?=SITE_PATH?>assets/css/main.css" />
        <!--[if lte IE 8]><link rel="stylesheet" href="<?=SITE_PATH?>assets/css/ie8.css" /><![endif]-->
    </head>
    <body>

        <?php include 'application/views/'.$content_view; ?>

    <!-- Scripts -->
            <script src="<?=SITE_PATH?>assets/js/jquery.min.js"></script>
            <script src="<?=SITE_PATH?>assets/js/jquery.scrolly.min.js"></script>
            <script src="<?=SITE_PATH?>assets/js/jquery.poptrox.min.js"></script>
            <script src="<?=SITE_PATH?>assets/js/skel.min.js"></script>
            <script src="<?=SITE_PATH?>assets/js/util.js"></script>
            <!--[if lte IE 8]><script src="<?=SITE_PATH?>assets/js/ie/respond.min.js"></script><![endif]-->
            <script src="<?=SITE_PATH?>assets/js/main.js"></script>

    </body>
</html>
```

Рис. 5. Код шаблона по умолчанию

Как видно из рис. 5, шаблон страницы содержит в себе лишь общую структура HTML-документа, а также подключение файла стилей и js-скриптов. Он также подключает содержимое необходимой страницы из $content_view.

Общий класс модели состоит только из метода get_data, который является абстрактным методом. Помимо этого, описан один отдельный общий метод, который можно причислить к модели. В нем идет обращение к базе данных. Файл db.php содержит описание этого класса. Класс DB является абстрактным, однако он содержит в себе много методов, которые могут пригодиться все классам, наследующим этим методы. С помощью этих методов можно проводить различные манипуляции с базой данных не используя язык SQL в чистом виде. Достаточно лишь описать новый класс, название которого будет говорить общему классу DB с какой таблицей из базы данных необходимо работать. Также необходимо указать, какие условия нужны для извлечения данных из базы. Например, по какому из столбцов провести поиск, либо указать лимит по выдаче результатов и т.д.

Итак, на главной странице приложения содержится список кандидатов, которых избиратели могут выбрать. Эта страница является страницей по умолчанию, контроллер которой описан в файле controller_main.php (данный контроллер не имеет свою модель и работает самостоятельно). В этом файле описан класс Controller_main, код которого показан на рис. 6.

```
<?php
class Controller_main extends Controller
{
    function action_index()
    {
        $model = new DB_Candidates(); // создаем объект модели
        $usersInfo = $model->getAllRows(); // получаем все строки
        $this->view->generate('main_view.php','template_view.php', $usersInfo);
    }
}
```

Рис. 6. Описание класса Controller_main

Здесь происходит выборка всех кандидатов при голосовании из базы данных и вызов представления main_view.php. В данный файл выводится информация о кандидатах и их краткое описание.

Для участия в голосовании избирателю необходимо пройти регистрацию. Для этого нужно перейти по ссылке «Зарегистрироваться». При переходе происходит смена контроллера на Controller_Reg, который вызывает форму регистрации. При прохождении регистрации необходимо указать ФИО, дату рождения, паспортные данные, дату выдачи и ИНН. После заполнения данных и отправки формы происходит проверка на пустоту, а также проверяется правильность введенных дат, серии и номера паспорта.

Серия паспорта состоит из четырех цифр, первые из которых содержат информацию о регионе, в котором выдали паспорт. Код региона проверяется по списку, например, Москва имеет код 45, а Ростовской области соответствует код 60. Это проверка необходима для того, чтобы избежать возможной попытки подделать результаты голосования путем введения несуществующего кода региона. Вторые две цифры серии паспорта означают год выдачи документа. Эти значения не могут быть больше 18 или меньше 97, так как именно 1 октября 1997 года паспорта СССР стали заменять на паспорта РФ.

Номер паспорта стоит из 6 цифр, в паре с серией они являются уникальным идентификатором избирателя. Совпадения по базе данных с точно такими же серией и номером паспорта невозможны. Поэтому происходит проверка проголосовавших избирателей на совпадения для того, чтобы избиратель не смог проголосовать больше одного раза. ИНН также должен быть заполнен по правилам и состоять из 12 цифр.

Помимо этого, при регистрации необходимо выбрать своего избирателя. Здесь используется отдельная таблица в базе данных, которая содержит два столбца: id избирателя, id кандидата. Они указывают на соответствие друг друга. Стоит отметить, что ни модераторы, ни администратор не могут изменить значения данной таблицы, так как у них есть права только на добавление данных, но не на модификацию.

Для дополнительной защиты данных используется специальная таблица, которая также содержит два столбца: id избирателя и результат ЭЦП Эль-Гамаля [1]. Именно администратор подписывает своим закрытым ключом конкатенацию из id кандидата серии и номера паспорта избирателя. У данной таблицы есть несколько назначений. Подпись служит для проверки результатов голосования и факта выбора избирателем данного кандидата. Эта проверка необходима лишь при попытке скомпрометировать систему электронного голосования. Для этого злоумышленнику необходимо получить доступ к суперпользователю root и подменить голоса в таблице.

После завершения регистрации избиратель переходит на страницу результатов голосования. Результаты голосования представляются в виде круговой диаграммы, которую можно увидеть на рис. 7.

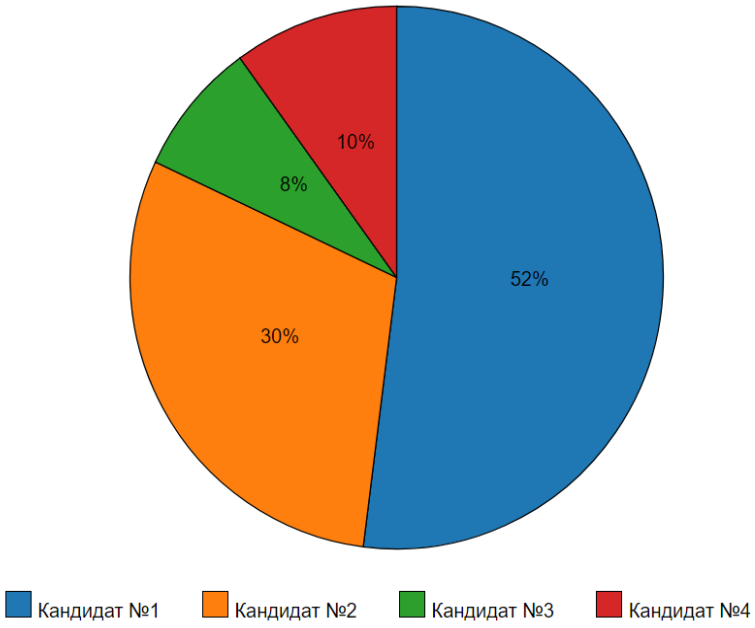

Рис. 7. Результаты голосования

Перед подсчетом голосов система проверяет на соответствие подписей. Только после этого выводятся результаты голосования.

Стоит отметить, что возможны случаи, когда избиратель случайно неправильно ввел свои паспортные данные, тем самым запретив проголосовать избирателю, чьи данные он ввел. Для этого нужны модераторы, которые могут изменить данные в таблице избирателей. После изменения данных о серии и номере паспорта, администратор снова подписывает конкатенацию из id выбранного кандидата серии и номера паспорта избирателя.

Заключение

Проанализирован усиленный вариант криптосистемы Эль-Гамаля для практической реализации системы электронного голосования. В результате было разработано веб-приложение для проведения электронного голосования, которое обеспечивает простоту, честность и прозрачность голосования. Разработанное приложение корректно функционирует на всех популярных операционных системах. Данное приложение содержит в себе защиту от подмены данных, тем самым значительно снижает вероятность того, что система будет скомпрометирована.

Библиографический список

1. Алгоритмическая оценка сложности системы кодирования и защиты информации, основанной на пороговом разделении секрета, на примере системы электронного голосования / А. В. Мазуренко [и др.] // Вестник Дон. гос. техн. ун-та. — 2017. — Т.17. — №3(90). — С.145–155.

2. Barbulescu, R., Gaudry, P., Joux, A., Thom´e, E.: A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 1 – 16. Springer, Heidelberg (May 2014).