Итоги технического аудита электронного голосования 08.09.2019

Авторы:

ФИО/наименование	**контактная информация**	**род занятий**
Оксана Русакова	@zigulinka	Аналитик данных
ABDK Consulting	https://abdk.consulting/	компания, занимающаяся разработкой смарт-контрактов
Алексей Щербаков	falcon.mk2@gmail.com	Backend-разработчик

Оглавление

Введение

Отсутствие контроля за отданными голосами

О описании голосования, опубликованном на сайте mos.ru[1] сказано следующее:

Стать наблюдателем

Наблюдателем мог стать каждый. Получать официальный статус наблюдателя и приходить на избирательный участок было не обязательно — можно было следить за выборами дистанционно.

1. Следить через «ноду»[2] наблюдателя независимой организации

Такой вид наблюдения больше подходил для продвинутых пользователей, понимающих голосов в систему блокчейн в режиме реального времени. По окончании голосования владельцы независимых узлов блокчейн-сети могли проверить результаты голосования офлайн и сравнить их с официальными данными.

2. Наблюдать на портале mos.ru

Достаточно компьютера или мобильного устройства с выходом в Интернет.

На портале mos.ru все желающие могли в режиме реального времени наблюдать статистику появления в блокчейне новых голосов и число проголосовавших избирателей на текущий момент.

3. Наблюдать на электронном участке

Необходимо было получить статус наблюдателя и прийти на электронный избирательный участок.

В каждом из трёх избирательных округов проведения эксперимента было организовано помещение электронного избирательного участка. Официально зарегистрированные наблюдатели могли находиться в помещении электронного избирательного участка в единый день голосования и следить за статистикой хода голосования на экранах, вживую наблюдать работу принтеров, которые печатали обезличенную в блокчейне информацию о голосах избирателей в пользу кандидатов.

[1] https://www.mos.ru/city/projects/blockchain-vybory/

[2] **Нода (англ. Node)** — узел блокчейн-сети, состоящий из одного или нескольких компьютеров, являющийся минимальной и неделимой единицей блокчейн-сети.

Фактически это не так.

Всё, что наблюдатель может контролировать — это количество проголосовавших в каждом округе. Следующие виды нарушений, которые при традиционном голосовании наблюдатель может обнаружить, в случае электронного голосования обнаружить нельзя, хотя сами эти нарушения (или их близкие аналоги) при электронном голосовании возможны:

1. Избирателю выдали бюллетень, не идентифицировав его личность (не посмотрев паспорт или не сличив фотографию в паспорте с внешностью избирателя или не убедившись, что паспорт настоящий);

2. Избирателю выдали бюллетень, не убедившись, что он есть в списке избирателей;

3. Избирателю выдали бюллетень, но не отметили его в списке избирателей, как проголосовавшего;

4. Избирателю выдали бюллетень, хотя в списке проголосовавших он уже был отмечен, как проголосовавший;

5. Избиратель, получив бюллетень, передал его другому человеку, который этим бюллетенем проголосовал;

6. Избиратель вошёл в кабинку для голосования не один;

7. Избиратель бросил в урну сразу несколько бюллетеней;

На практике администратор смарт-контракта может зарегистрировать в смарт-контракте любой адрес в качестве адреса избирателя, имеющего право голосовать в каком-то из округов. При этом любой может легко генерировать сколько угодно адресов.

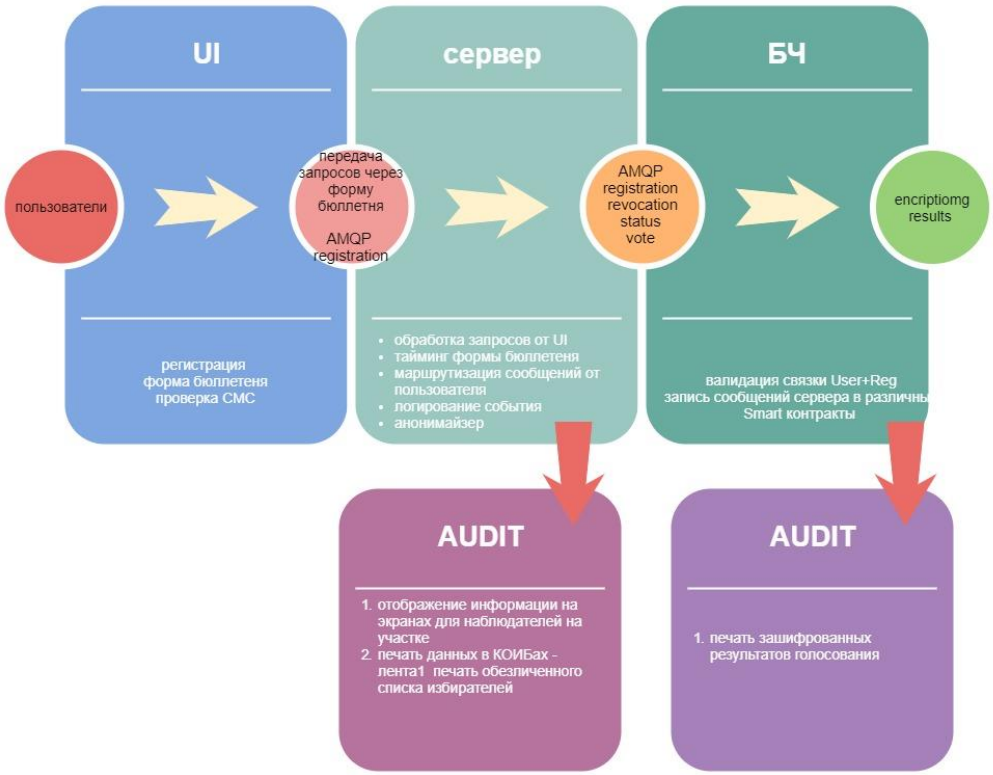
После окончания голосования у наблюдателей появляется возможность расшифровать бюллетени и узнать за кого был подан каждый голос. В отличие от традиционного голосования, при котором бюллетени перемешиваются в урне, в электронном голосовании при подсчёте голосов можно определить, когда был «опущен в урну» каждый конкретный бюллетень и с какого адреса он был отправлен.

Так например, медузе удалось расшифровать голос своего сотрудника.[3]

В ходе выборов в Москве 8 сентября 2019 и речи не шло о создании распределённой сети. Так, весь блокчейн и все внешние модули хранились на серверах Департамента Информационных Технологий города Москвы[4] (далее – ДИТ).

[3] https://meduza.io/slides/meriya-sluchayno-pozvolila-rasshifrovat-golosa-na-vyborah-v-mosgordumu-my-eto-sdelali-i-nashli-koe-chto-strannoe
[4] https://www.mos.ru/dit/

Архитектура электронного голосования



1. Пользовательский интерфейс и формы бюллетеня;

2. Анонимайзер (расположен на стороне ДИТ);

3. Генератор бюллетеня;

4. Передача данных для параллельного аудита (передача осуществляется с сервера ДИТ, до попадания данных в БЧ):

   a. Мониторы, установленные в УИК;

   b. Печать данных на ленте (на печатном носителе) [5] - кто проголосовал;

[5] Видеозапись первого заседания Рабочей Группы Департамента Электронного Голосования (далее – РГ ДЭГ): https://www.youtube.com/watch?v=2jd8pDbReaQ с 45 минуты

5. Запись данных в БЧ;

    a. Вторая печатная лента в зашифрованном виде печатается в процессе голосования

1. Проведение аудита голосования.

При проведении электронного голосования, важно организовать параллельное проведение аудита.

Для убеждения избирателей в правильности подсчёта голосов, предотвращения сбоев или мошенничества и проведения аудита, существует множество технологий. Некоторые системы используют криптографию[6], бумажное подтверждение, аудио контроль, а также технологию двойной записи (на электронный носитель и бумагу).

Профессор Ребекка Меркьюри, создатель концепции VVPAT[7] (аудит по заверенным избирателями бумажным бюллетеням), обосновывает эффективность распечатывания бумажного бюллетеня с целью дальнейшего его подтверждения избирателем перед окончательным учётом (впоследствии этот метод стали называть «Метод Меркьюри»). Чтобы быть окончательно подтверждённым, голос должен быть подтверждён избирателем без использования визуальных или аудиосредств. Если избиратель вынужден использовать, например, сканер штрих-кода для того, чтобы подтвердить свой выбор, то такой голос не может считаться действительно подтверждённым, потому что подтверждает не сам избиратель, а электронное устройство.

Системы голосования со сквозным аудитом выдают избирателям подписанные квитанции, которые можно забрать домой. Такие квитанции не позволяют узнать, как именно проголосовал избиратель, но позволяют проверить, что голос был учтён, узнать общее количество голосов и результаты голосования.

---

[6] Что такое криптография: https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F

[7] VVPAT аудит рекомендован к использованию ОБСЕ: https://www.osce.org/ru/odihr/elections/107771?download=true

Системы, позволяющие посторонним узнать, как именно проголосовал избиратель, никогда не использовались на государственных выборах, и были объявлены незаконными. Основная причина этого решения — возможность запугивать избирателей и покупать их голоса.

Также системы аудита могут использоваться для обнаружения неполадок оборудования и случаев мошенничества. В случае использования системы VVPAT, бумажный бюллетень является основным документом, а электронные голоса используются только для предварительного подсчёта. Для успешного аудита устройства для голосования необходима целая последовательность мероприятий.

На текущих выборах

1. Параллельный аудит происходил посредством печати данных из очереди на сервере до записи в блокчейн, то есть его инициировал централизованный орган и при этом у пользователя не было возможности проверить был ли учтён его голос;

2. Пользователи были лишены возможности просмотреть результат своего голосования и проконтролировать правильно ли учли его голос;

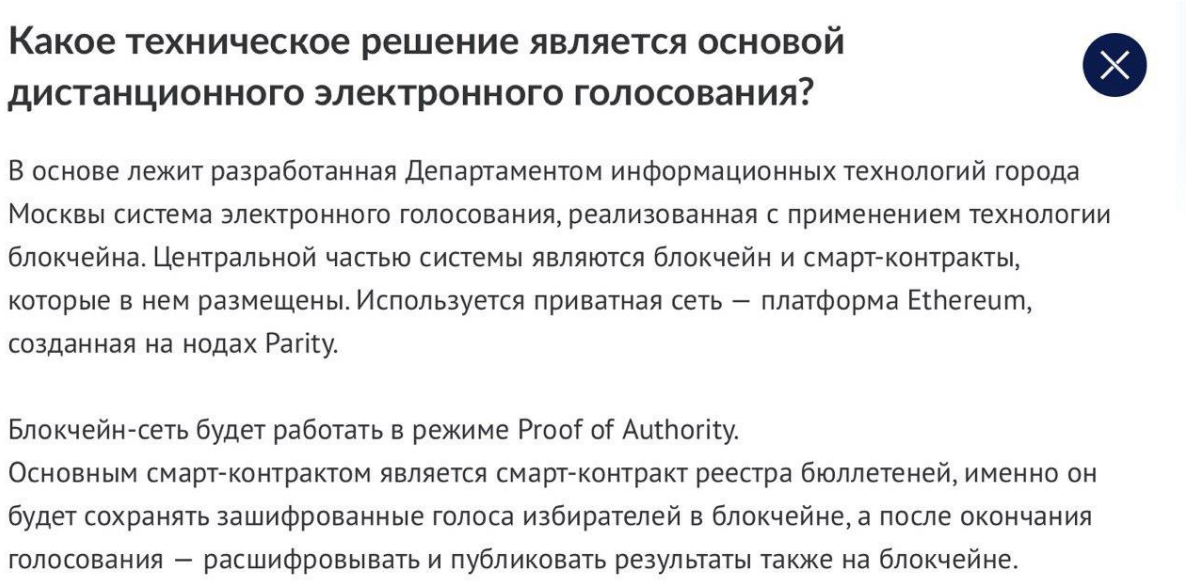
3. Избиратели не подтверждали свой голос без использования электронных средств.

Следовательно, сквозной аудит не проводился. Более того использование принтеров для дублирования расшифрованных данных из очереди на сервере ДИТ оставляет возможность для администраторов системы ЭГ сопоставить результат голосования и избирателя.

2. Использованные технологии

В данном разделе описаны использованные в разработке системы электронного голосования технологии.

2.1. Parity



Какое техническое решение является основой дистанционного электронного голосования?

В основе лежит разработанная Департаментом информационных технологий города Москвы система электронного голосования, реализованная с применением технологии блокчейна. Центральной частью системы являются блокчейн и смарт-контракты, которые в нем размещены. Используется приватная сеть — платформа Ethereum, созданная на нодах Parity.

Блокчейн-сеть будет работать в режиме Proof of Authority.
Основным смарт-контрактом является смарт-контракт реестра бюллетеней, именно он будет сохранять зашифрованные голоса избирателей в блокчейне, а после окончания голосования — расшифровывать и публиковать результаты также на блокчейне.

Работа с Ethereum возможна через огромное число клиентов, часть из которых «terminal-based», часть GUI и есть несколько гибридных решений. Своего рода стандартом является «Geth», который разрабатывается командой Ethereum.

В последнее время все чаще можно встретить другой клиент — Parity, написанный на Rust. Главным его отличием от Geth является встроенный web интерфейс.

Еще один плюс: Parity быстрее своих аналогов. Единственный нюанс — своей консоли в parity нет. Но можно без проблем использовать для этих целей Geth.

2.2. Proof of Autority

Proof of Authority (далее – PoA) является основанным на репутации алгоритмом консенсуса, который представляет практическое и эффективное решение для блокчейнов (особенно частных). Этот термин был предложен в 2017 году соучредителем Ethereum и бывшим техническим специалистом Гэвином Вудом.

Консенсус алгоритм PoA использует значение идентификаторов, которые означают, что валидаторы блока не создают стейки монеты, а вместо этого имеют собственную репутацию. Следовательно, блокчейны PoA защищён проверяющими узлами, которые произвольно выбирают заслуживающего доверия.

Модель Proof of Authority основана на ограниченном количестве валидаторов блока, и именно это делает её масштабируемой системой. Блоки и транзакции проверяются заранее утверждёнными участниками, которые выступают в качестве модераторов системы.

Согласованный алгоритм PoA может применяться в различных сценариях, и они считаются высокоценным вариантом для логистических приложений. К примеру, когда речь заходит о цепочках поставок, то PoA считается более эффективным и разумным решением.

Модель Proof of Authority позволяет компаниям сохранять свою конфиденциальность, пользуясь преимуществами технологии блокчейн. Microsoft Azure, это еще один пример реализации этого ПО.

Главная проблема механизма PoA состоит в отказе от децентрализации. Можно сказать, что это не модель алгоритма консенсуса, а просто попытка сделать централизованные системы более эффективными. Хотя это и делает PoA привлекательным решением для крупных корпораций с логистическими потребностями. Системы PoA имеют высокую пропускную способность, но этот плюс нивелируется лёгкостью, с которой может быть установлена цензура.

Таким образом такая система полностью подконтрольна валидаторам (администраторам). В реализации электронного голосования 2019 бsk один валидатор - ДИТ Москвы.

2.3. Шифрование 1024 бит

Незадолго до голосования разработчики увеличили длину ключа до 1024 бит, после того как 14 августа французский криптограф Пьеррик Годри выявил уязвимость ключа шифрования, позволявшую заранее узнать результаты голосования.

В Приложении № 1 к решению. Московской Городской Избирательной комиссии от 03.09.2019 № 112/4 «О дополнительных гарантиях обеспечения гласности при проведении эксперимента по дистанционному электронному голосованию»[8] на сайте мосгоризберкома описан следующий алгоритм работы с ключам шифрования:

2. Работа с ключом расшифрования

2.1. Не позднее 18.00 7 сентября 2019 года должна быть проведена процедура формирования закрытого (приватного) ключа, предназначенного для расшифрования волеизъявления избирателя.

2.2. Для повышения защищенности процедуры шифрования, а также для осуществления публичного контроля за процедурой расшифрования волеизъявления избирателя должно производиться разделение закрытого (приватного) ключа шифрования на не менее чем семь частей.

2.3. Процедура разделения закрытого (приватного) ключа производится при помощи автономного программного модуля СПО, на персональном компьютере, не подключенном к локальным вычислительным сетям, в присутствии лиц, указанных в пункте 2.1.

2.4. Каждая из частей записывается в файл на flash-носителе.

Каждый flash-носитель помещается в запечатываемый конверт,

[8] http://www.mosgorizbirkom.ru/documents/10279/19437521/rs_112_4.pdf/6569509f-3c35-4562-b337-e58988071d7c

2.5. После завершения процедуры разделения закрытого (приватного) ключа сам ключ, все его части должны быть безвозвратно удалены с персонального компьютера. Затем устройство выключается и опечатывается

2.6. 8 сентября не ранее 20.30 в помещении Мосгоризбиркома в присутствии лиц, указанных в пункте 2.4, производится сбор частей закрытого (приватного) ключа.

Для этой цели конверты, переданные на хранение в соответствии с пунктом 2.4, за исключением конвертов, переданных представителям участковых избирательных комиссий избирательных участков дистанционного электронного голосования № 5001, 5002 и 5003, не позднее 20.00 доставляются в Московскую городскую избирательную комиссию получившими их лицами или их представителями.

Конверты, доставленные в Московскую городскую избирательную комиссию, в присутствии лиц, указанных в пункте 2.4, вскрываются и из них извлекается flash-носитель. В ведомости (приложение № 2) в соответствующей графе делается отметка о приеме исходных flash-носителей.

Конверты, переданные представителям участковых избирательных комиссий избирательных участков дистанционного электронного голосования № 5001, 5002 и 5003, вскрываются в помещении соответствующего избирательного участка в присутствии лиц, указанных в статье 23 Избирательного кодекса города Москвы. Части ключа с извлеченных flash-носителей передаются при помощи СПО Портала в Московскую городскую избирательную комиссию, а при отсутствии такой возможности передача выполняется посредством электронной почты.

Таким образом этот компонент системы можно считать безопасным.

2.4. Логирование

В качестве системы хранения и записи логов использовалось програмное обеспечение «Graylog».

Graylog это open source программное обеспечение, предназначенное для сбора логов в гигантских сетях их огромного количества источников различными способами. В нем можно удобно организовать сбор событий, фильтрацию, поиск, автоматизацию (отображение различных уведомлений) и т. д. Аналогичных средств множество, но Graylog предлагает очень высокую производительность с использованием современных компонентов, удобную аналитику и красивый интерфейс.

Тем не менее по заявлению представителей Дит Graylog с какой-то момент отключился.

При этом в whitepaper[9] продукта указано следующее:

«Начиная с приёма данных, Graylog использует журнал сообщений для фиксации данных на диске, предотвращая потерю данных в случае сбоя в сети или при сбое индексов или поисков. Graylog без проблем обеспечивает репликацию данных и восстановление данных, не требуя дополнительных компонентов. Такой подход позволяет пользователю быть уверенным в том, что не будет потери данных даже в периоды пиковых нагрузок или необычного использования.»

Если в журналах Graylog и отсутствуют какие-то данные, то это говорит о том, что его преднамеренно отключали.

2.5. Передача сообщений

Для передачи сообщений между компонентами системы использовался AMQP протокол в 4 этапа:

4. registration (регистрация);
5. revocation(отзыв);

[9] https://www.graylog.org/resources/the-graylog-advantage

6. status (статус);
7. vote (голосование).

AMQP (Advanced Message Queueing Protocol[10]) это —широко поддерживаемый открытый протокол для передачи сообщений между компонентами системы с низкой задержкой и на высокой скорости. При этом семантика обмена сообщениями настраивается под нужды конкретного проекта.

Основная идея состоит в том, что отдельные подсистемы (или независимые приложения) могут обмениваться произвольным образом сообщениями через AMQP-брокер, который осуществляет маршрутизацию, возможно гарантирует доставку, распределение потоков данных, подписку на нужные типы сообщений.

AMQP основан на трех понятиях:

1. Сообщение (message) — единица передаваемых данных, основная его часть (содержание) никак не интерпретируется сервером, к сообщению могут быть прицеплены структурированные заголовки.

2. Точка обмена (exchange) — в нее отправляются сообщения. Точка обмена распределяет сообщения в одну или несколько очередей. При этом в точке обмена сообщения не хранятся. Точки обмена бывают трех типов: fanout — сообщение передается во все прицепленные к ней очереди; direct — сообщение передается в очередь с именем, совпадающим с ключом маршрутизации (routing key) (ключ маршрутизации указывается при отправке сообщения); topic — нечто среднее между fanout и exchange, сообщение передается в очереди, для которых совпадает маска на ключ маршрутизации, например, app.notification.sms.* — в очередь будут доставлены все сообщения, отправленные с ключами, начинающимися на app.notification.sms.

[10] http://www.amqp.org/

3. Очередь (queue) — здесь хранятся сообщения до тех пор, пока не будет забраны клиентом. Клиент всегда забирает сообщения из одной или нескольких очередей.

Данный протокол стандартный для многопоточного обмена сообщениями и широко применяется в финансовых и банковских приложениях. Пример - приложение “Tabtrader”

Аудит смарт контрактов

Исполнитель: ABDK Consulting[11]

1. Введение

8 сентября 2019 года в Москве проводились выборы в Московскую Городскую Думу. Выборы проводились по мажоритарной система, в 45 одномандатных округах избирали 45 депутатов. В трёх округах в качестве эксперимента было организовано электронное голосование.

Исходный код некоторых компонентов программного обеспечения, задействованного в электронном голосовании, был выложен в открытый доступ. В частности, был выложен исходный код смарт-контрактов, использовавшихся при голосовании.

ABDK Consulting провела аудит исходного кода вышеупомянутых смарт-контрактов. Результаты аудита изложены в настоящем документе.

2. Объект аудита

Объектом аудита были 4 файла, размещённые в открытом репозитарии GitHub[12]:

1. BallotsRegistry.sol
2. LockableTransactionAuthorizer.sol
3. Ownable.sol
4. VotersRegistry.sol

[11] https://abdk.consulting/

[12] https://github.com/moscow-technologies/blockchain-voting/tree/51aa4300aceb22795b567daf910af25ecbb55634/smart-contracts/packages/smart-contracts/contracts:

Файлы представляют собой исходный код на языке программирования Solidity[13], предназначенном для разработки смарт-контрактов под платформу Ethereum[14].

Смарт-контракт — это программа, код которой публикуется в блокчейне, благодаря чему код доступен всем и после опубликования не может быть изменён. Опубликованный смарт-контракт получает свой собственный адрес и возможность распоряжаться активами. Поведение смарт-контракта полностью определяется его кодом, и никто, в том числе автор смарт-контракта, не может это поведение изменить. Поскольку код смарт контракта известен всем, а поведение полностью определяется кодом, любой желающий может изучить код смарт-контракта и удостовериться, что тот ведёт себя честно. Благодаря этому, люди могут доверять смарт-контракту, даже если они лично не знают, или не доверяют его автору.

3. Использованные материалы

Помимо самих аудируемых файлов, в процессе аудита ABDK Consulting использовала следующие материалы:

1. Файлы, лежащие в одном Git-репозитории с аудируемыми файлами[15];
2. Информационные материалы об электронном голосовании, размещённые на официальном сайте мэра Москвы[16].

4. Общее описание объекта аудита

В аудируемых файлах описано три самостоятельных смарт-контракта с именами LockableTransactionAuthorizer, VotersRegistry и BallotsRegistry, а также смарт-контракт с именем Ownable, который не может быть использован сам по себе, но предназначен быть основой для других смарт-контрактов.

[13] https://solidity.readthedocs.io/

[14] https://www.ethereum.org/

[15] https://github.com/moscow-technologies/blockchain-voting

[16] https://www.mos.ru/city/projects/blockchain-vybory/

4.1. LockableTransactionAuthorizer

Этот смарт-контракт не имеет прямого отношения к электронному голосованию. Он является частью конфигурации блокчейна, в котором выполняются остальные смарт-контракты, и определяет, кто и когда имеет права публиковать информацию в блокчейн. Контракт может находиться в двух состояниях: запертом и незапертом. В незапертом состоянии контракт разрешает всем желающим публиковать в блокчейне любую информацию, и, в то числе, смарт-контракты. В запертом состоянии контракт разрешает только простые переводы криптовалюты с одного адреса на другой, а также обращения к смарт-контрактам, но не публикацию новых смарт-контрактов.

4.2. VotersRegistry

Этот контракт представляет собой реестр избирателей: аналог списка избирателей, присутствующего на традиционном избирательном участке. Контракт ведёт учёт избирателей, зарегистрированных для участия в электронном голосовании, а также позволяет отозвать регистрацию уже зарегистрированного избирателя. Кроме того, этот контракт ведёт учёт избирателей, принявших участие в голосовании, и отвечает за то, чтобы один избиратель не смог проголосовать дважды.

4.3. BallotsRegistry

Этот смарт-контракт является аналогом избирательной урны и избирательного участка. Он ведёт учёт выданных и использованных бюллетеней. Во время голосования смарт-контракт сохраняет заполненные бюллетени в зашифрованном виде. После окончания голосования этот же контракт используется для опубликования ключа шифрования и расшифрованных бюллетеней.

4.4. Ownable

Этот смарт контракт не может быть использован сам по себе и предназначен быть основой для других смарт-контрактов. Он позволяет смарт-контракту иметь владельца и выполнять его команды. Смарт контракты VotersRegistry и BallotsRegistry используют Ownable в качестве основы. Смарт контракт LockableTransactionAuthorizer также может иметь владельца, однако он не использует Ownable, а реализует соответствующую функциональность самостоятельно.

5. Общие комментарии

Смарт-контракты, а также файлы, опубликованные в одном с ними Git-репозитории, являются лишь отдельными частями системы электронного голосования. Техническая документация и описание архитектуры системы в репозитории отсутствуют. Требования, на основе которых разрабатывались аудируемые смарт-контракты, во многом неизвестны.

В таких условиях проводить полноценный аудит кода весьма затруднительно. Сам аудируемый код не может служить надёжным источником информации о том, как по замыслу автором должна работать система, а только о том, как она на самом деле работает. Отличить ошибочное поведение от задуманного авторами бывает затруднительно.

Ситуация становится ещё сложнее в случаях, когда для анализа доступен код только отдельных частей системы. Даже если желаемое поведение системы в целом понятно, роль и связи отдельных компонентов, то, как поведение конкретного компонента сказывается на поведении системы в целом, может оставаться неясным.

Из-за ограниченности информации о системе, ABDK Consulting не может с уверенностью рассуждать о том, как выявленные в смарт-контрактах недостатки сказываются на работе системы в целом, и, в частности, могут ли эти недостатки быть использованы для вмешательства в ход проведения выборов, и могут ли они повлечь за собой нарушение тайны голосования.

6. Выявленные недостатки

В этом разделе описываются недостатки смарт-контрактов, выявленные в результат аудита.

6.1. Контракт VotersRegistry не привязывает номера избирателей к реальным людям

В контракте VotersRegistry избиратели хранятся в обезличенном виде: фактически хранятся только некие номера избирателей. Вот как выглядит код добавления избирателя в реестр:

```
147     function addVoter(uint256 voterId) public onlyOwner {
148         require(
149             isRegistrationStopped == false,
150             "Can not add Voter when registration is closed!");
151 
152         require(
153             voterId != 0,
154             "Voter Id can not be zero!");
155 
156         require(
157             voters[voterId].paricipationReceivedBlock == 0,
158             "Voter must not participate earlier!");
159 
160         voters[voterId].isParticipating = true;
161         voters[voterId].paricipationReceivedBlock = block.number;
162 
163         votersCount++;
164 
165         emit VoterParticipating(voterId);
166     }
```

Таким образом, информация, сохранённая в блокчейне, не позволяет проверить, что эти номера соответствуют реальным людям, и что каждому реальному избирателю присвоено не более одного номера.

Это проблему легко можно было бы исправить, сохраняя вместе с номером избирателя хэш от его персональных данных, например от серии и номера паспорта, дополненный секретной солью.

При таком подходе персональные данные избирателей не попадают в свободный доступ, но организаторы голосования, знающие секретную соль, могут при необходимости доказать, что номера избирателей соответствуют конкретным людям. Это похоже на список избирателей с избирательного участка, содержащий персональные данные избирателей и отметки о том, кому был выдан бюллетень. Список не находится в свободном доступе, но суд, при необходимости, может его изучить.

6.2. Контракт VotersRegistry не привязывает избирателей к округам

Контракт VotersRegistry ведёт учёт избирателей, зарегистрированных для участия в электронном голосовании, отслеживает проголосовавших избирателей и следит за тем, чтобы один избиратель не проголосовал дважды.

Электронное голосование на выборах в Мосгордуму было организовано в трёх округах. В каждом округе был свой список избирателей и свой набор кандидатов. Фактически, параллельно проводилось три независимых голосования. Однако вместо того, чтобы вести три отдельных списка избирателей, по одному на округ, смарт-контракт VotersRegistry регистрировал всех избирателей в едином списке, без привязки к округам. Вот как выглядит фрагмент кода смарт-контракта, отвечающий за регистрацию избирателя:

```
147     function addVoter(uint256 voterId) public onlyOwner {
148         require(
149             isRegistrationStopped == false,
150             "Can not add Voter when registration is closed!");
151 
152         require(
153             voterId != 0,
154             "Voter Id can not be zero!");
155 
156         require(
157             voters[voterId].paricipationReceivedBlock == 0,
158             "Voter must not participate earlier!");
159 
160         voters[voterId].isParticipating = true;
161         voters[voterId].paricipationReceivedBlock = block.number;
162 
163         votersCount++;
164 
165         emit VoterParticipating(voterId);
166     }
```

Единственная информация об избирателе, передаваемая смарт-контракту — это числовой идентификатор избирателя. Информация об избирательном округе, в котором имеет право голосовать избиратель, смарт-контракту недоступна.

В момент голосования смарт-контракт получает код избирателя и код округа, и проверяет, что данный избиратель в данном округе ещё не голосовал. Вот соответствующий фрагмент кода:

```
71      function issueBallotFor(uint256 voterId, uint256 votingId) external onlyOwner {
72          require(
73              voterId != 0,
74              "Voter Id can not be zero!");
75
76          require(
77              votingId != 0,
78              "Voting Id can not be zero!");
79
80          require(
81              voters[voterId].paricipationReceivedBlock != 0,
82              "Voter must participate!");
83
84          require(
85              voters[voterId].revocationReceivedBlock == 0,
86              "Voter must not revoke earlier!");
87
88          require(
89              voters[voterId].issuedBallots[votingId].ballotIssuedBlock == 0,
90              "Voter Issue Ballot twice!");
91
92          voters[voterId].firstBallotIssuedBlock = block.number;
93
94          voters[voterId].issuedBallots[votingId].isBallotIssued = true;
95          voters[voterId].issuedBallots[votingId].ballotIssuedBlock = block.number;
96
97          if (!votingsMap[votingId]) {
98              votingsMap[votingId] = true;
99              votingsList.push(votingId);
100         }
101
102         ballotsIssuedByVoting[votingId]++;
103
104         emit BallotIssued(voterId, votingId);
105     }
```

Здесь «voterId» — это код избирателя, а «votingId» — это код округа. Смарт-контракт не разрешает одному избирателю дважды голосовать в одном округе (строка 89), однако позволяет одному избирателю проголосовать в нескольких округах.

О том, что «votingId» это именно номер избирательного округа можно судить, например, вот по этой строчке[17]:

```
286         $buttons.on('click', function () {
287             var $button = $(this);
288             var votingId = parseInt( $('#district').val() );
289             var dataToEncrypt = parseInt($button.data('value'));
290             var entropy = userEntropy;
```

[17] https://github.com/moscow-technologies/blockchain-voting/blob/51aa4300aceb22795b567daf910af25ecbb55634/voting-form/src/ballot/static/js/forms/mgik/election.js#L288

Этот недостаток можно было бы легко устранить, записывая в смарт-контракт номера избирателей вместе с номерами округов, к которым эти избиратели относятся.

Другой способ исправить ситуацию: опубликовать не один, а три смарт-контракта VotersRegistry, по одному на каждый округ.

6.3. Контракт BallotsRegistry позволяет избирателям голосовать до начала и после окончания голосования

Контракт BallotsRegsitry, принимает от избирателей зашифрованные голоса и сохраняет их в своём хранилище. Каждый принятый голос контракт снабжает пометкой о времени, когда этот голос был принят. Однако, смарт контракт не проверяет, чтобы время подачи голоса укладывалось во временной интервал, отведённый для голосования. По текущим правилась голосование начинается в 8 утра и заканчивается в 8 вечера, но смарт-контракт может принимать голоса как до начала голосования, так и после его окончания. Вот как выглядит код, отвечающий за проверку поданного голоса:

```
233     function addBallot(uint256 votingId, bytes memory _A, bytes memory _B) public {
234         require(
235             isRegistryClosed == false,
236             "Registry must not be closed!");
237 
238         require(
239             votingId != 0,
240             "Voting Id can not be zero!");
241 
242         require(
243             allowedVotingsForVoters[msg.sender][votingId] == true,
244             "Voter must be allowed to pass Ballot for this Voting!");
245 
246         require(
247             votingsChecks[msg.sender][votingId] == false,
248             "Voter must not pass two Ballots for the Voting!");
249 
250         require(
251             privateKey.length == 0,
252             "Private Key must not present!");
```

Видно, что он нигде не учитывает текущее время.

Этот недостаток легко можно было бы исправить, добавив в смарт-контракт проверку, что текущее время находится во временном интервале, отведённом для голосования.

6.4. Контракт BallotsRegistry позволяет избирателям проголосовать позднее, чем через 15 минут после получения бюллетеня

В описании процедуры электронного голосования написано следующее[18]:

Обращаем внимание, что на голосование отводится 15 минут. Если по истечении этого времени бюллетень останется пустым, избирательное право все равно будет считаться реализованным, так как бюллетень избирателю был выдан.

Однако смарт контракт BallotsRegistry, принимающий и сохраняющий зашифрованные голоса избирателей, может принимать бюллетени даже если с момента их выдачи прошло более 15 минут.

С точки зрения смарт-контракта моментом выдачи бюллетеня можно считать момент, когда уникальный одноразовый адрес избирателя, сгенерированный на странице голосовании, регистрируется для участия в голосовании. Вот код, отвечающий за регистрацию адреса:

```
193     function addVoterToAllowedVoters(address voter, uint256 votingId) external onlyOwner {
194         require(
195             isRegistryClosed == false,
196             "Registry must not be closed!");
197 
198         require(
199             votingId != 0,
200             "Voting Id can not be zero!");
201 
202         require(
203             voter != msg.sender,
204             "Can not add Self to Allowed Voters!");
205 
206         require(
207             allowedVotingsForVoters[voter][votingId] == false,
208             "Can not add to Allowed Voters twice!");
209 
210         allowedVotingsForVoters[voter][votingId] = true;
211 
212         emit AllowedVoterAdded(voter, votingId);
213     }
```

Видно, что этот код не сохраняет в хранилище время регистрации адреса и, таким образом, не даёт возможность позднее проконтролировать, что время, отведённое на заполнение бюллетеня, не истекло.

[18] https://www.mos.ru/city/projects/blockchain-vybory/

Эту проблему можно было бы легко исправить, добавив в функцию регистрации адреса логику, которая сохраняла бы для данного адреса время его регистрации, а в функцию сохранения зашифрованного бюллетеня логику, проверяющую, что избиратель уложился в отведённое для заполнения бюллетеня время.

6.5. Контракт BallotsRegistry позволяет одному избирателя проголосовать в нескольких округах

Перед тем, как контракт BallotsRegistry будет готов принять заполненный бюллетень от избирателя, уникальный одноразовый адрес избирателя должен быть зарегистрирован в смарт-контракте, как имеющий право голоса в определённом округе. Смарт-контракт не позволяет зарегистрировать один и тот же адрес для голосования в одном и том же округе дважды, однако он позволяет зарегистрировать один адрес для голосования в нескольких округах, что противоречит идее мажоритарной системы выборов, при которой каждый избиратель участвует в выборах депутата только от одного округа. Код регистрации адреса приведён в предыдущем разделе.

6.6. Контракт BallotsRegistry позволяет администратору завершить голосование до того, как оно должно быть завершено по закону

После окончания голосования администратор смарт контракта должен закрыть реестр голосов, чтобы иметь возможность опубликовать закрытый ключ и начать процедуру расшифровки зашифрованных бюллетеней. Вот код, отвечающий за закрытие реестра:

```
93      function closeRegistry() external onlyOwner {
94          require(
95              isRegistryClosed == false,
96              "Registry must not be closed!");
97
98          isRegistryClosed = true;
99
100         emit RegistryClosed();
101     }
```

Видно, что этот код никак не учитывает текущее время, и, соответственно, позволяет администратору завершить голосование досрочно.

Эту проблему легко можно было бы решить, добавив в этого код логику, проверяющую, что время окончания голосования уже наступило.

Ещё лучше было бы изменить архитектуру смарт-контракта так, чтобы он самостоятельно завершал голосование в определённое время, не дожидаясь команды администратора.

6.7. Контракт BallotsRegistry позволяет администратору опубликовать закрытый ключ, не соответствующий ранее опубликованному открытому ключу

После окончания голосования администратор смарт-контракта BallotsRegistry должен опубликовать закрытый ключ, необходимый для расшифровки бюллетеней. Вот код, отвечающий за публикацию закрытого ключа:

```
215  function publishPrivateKey(bytes memory _privateKey) public onlyOwner {
216      require(
217          isRegistryClosed == true,
218          "Registry must be closed!");
219
220      require(
221          _privateKey.length != 0,
222          "Must pass proper Private Key!");
223
224      require(
225          privateKey.length == 0,
226          "Can not publish Private Key twice!");
227
228      privateKey = _privateKey;
229
230      emit PrivateKeyPublished(privateKey);
231  }
```

Видно, что этот код нигде не проверяет, что преданный ему закрытый ключ соответствует ранее опубликованному открытому ключу. Таким образом смарт-контракт позволяет администратору опубликовать неправильный закрытый ключ, непригодный для расшифровки бюллетеней.

Ситуация усугубляется тем, что смарт-контракт позволяет администратору опубликовать закрытый ключ только один раз. Если администратор по ошибке опубликует неправильный ключ, то исправить ошибку будет невозможно.

Эту проблему можно было бы исправить, добавив в код, отвечающий за публикацию закрытого ключа, код, проверяющий, что предлагаемый к публикации закрытый ключ на самом деле соответствует ранее опубликованному открытому ключу.

6.8. Контракт BallotsRegistry позволяет администратору опубликовать расшифрованный голос, не соответствующий ранее опубликованному зашифрованному голосу

После окончания голосования и публикации закрытого ключа, администратор смарт-контракта должен расшифровать бюллетени и опубликовать расшифрованные голоса избирателей. В момент публикации расшифрованного голоса, смарт-контракт не проверяет, что расшифровка соответствует ранее опубликованному зашифрованному голосу. Вот код, проверяющий расшифрованный голос, предлагаемый к публикации:

```
138   function storeDecryptedData(uint256 ballotIndex, uint256 data) external onlyOwner {
139       require(
140           isRegistryClosed == true,
141           "Registry must be closed!");
142
143       require(
144           ballotIndex < ballots.length,
145           "Must pass valid index!");
146
147       require(
148           privateKey.length != 0,
149           "Private Key must present!");
150
151       require(
152           ballots[ballotIndex].decryptedData == 0,
153           "Can not decrypt Ballot twice!");
```

Видно, что правильность расшифровки нигде не проверяется, то есть смарт-контракт позволяет администратору опубликовать под «сидом» расшифрованного голоса всё, что угодно, исказив таким образом волеизъявление избирателя.

Ситуация усугубляется тем, что для каждого конкретного бюллетеня, смарт контракт позволяет опубликовать расшифрованный голос не более одного раза. Если администратор по ошибке опубликует неправильную расшифровку, то исправить ошибку будет невозможно.

Заметим, что поскольку в блокчейне сохраняются и зашифрованные голоса, и расшифровки, и закрытый ключ, то кто угодно, имеющий доступ к блокчейну, может самостоятельно расшифровать голоса, сравнить результаты расшифровывания с тем, что опубликовать администратор и, если администратор опубликовал неправильную расшифровку, обнаружить подмену.

Проблему можно было бы легко решить, добавив в функцию, отвечающую за публикацию расшифрованных голосов, логику, проверяющую, что предлагаемая к публикации расшифровка соответствует ранее опубликованному зашифрованному голосу.

Ещё лучшим решением было бы реализовать расшифровку голосов в самом смарт-контракте, чтобы смарт-контракт расшифровывал голоса самостоятельно. Именно так были устроены ранние версии этого смарт-контракта. Однако 20 августа логика расшифровки голосов была удалена из смарт-контракта, одновременно с увеличение длины ключа. Скорее всего это было связано с тем, что старая логика не могла работать с ключами увеличенной длины, а времени на реализацию новой логики у авторов смарт-контрактов не оставалось.

6.9. Контракты VotersRegistry и BallotsRegistry не взаимодействуют между собой

Смарт-контракт VotersRegistry отвечает за регистрацию избирателей и учёт проголосовавших избирателей. Контракт BallotsRegistry отвечает за учёт голосов, поданных избирателями. Когда избиратель голосует, информация об этом попадает в оба контракта, и было бы разумно предполагать, что оба контракта при этом работают вместе по принципу «всё или ничего». То есть либо запись о голосе появляется в обоих контрактах, либо ни в одном. Сама технология смарт-контрактов позволяет связывать смарт-контракты между собой для выполнения подобных согласованных действий, однако в данном случае контракты VotersRegistry и BallotsRegistry не связаны. Вот код, отвечающий за регистрацию голоса в контракте VotersRegistry:

```
92          voters[voterId].firstBallotIssuedBlock = block.number;
93
94          voters[voterId].issuedBallots[votingId].isBallotIssued = true;
95          voters[voterId].issuedBallots[votingId].ballotIssuedBlock = block.number;
96
97          if (!votingsMap[votingId]) {
98              votingsMap[votingId] = true;
99              votingsList.push(votingId);
100         }
101
102         ballotsIssuedByVoting[votingId]++;
103
104         emit BallotIssued(voterId, votingId);
```

Видно, что он не содержит обращений к контракту BallotsRegistry. Вот код, отвечающий за регистрацию выданного бюллетеня в контракте BallotsRegistry:

```
210         allowedVotingsForVoters[voter][votingId] = true;
211
212         emit AllowedVoterAdded(voter, votingId);
```

Код, отвечающий за регистрацию голоса в контракте BallotsRegistry:

```
254         EncryptedData memory encData = EncryptedData({
255             A: _A,
256             B: _B
257         });
258
259         uint256 newBallotsAmount = ballots.push(Ballot({
260             voter: msg.sender,
261             votingId: votingId,
262             receivedBlock: block.number,
263             receivedBlockTimestamp: block.timestamp,
264             decryptedBlock: 0,
265             decryptedTimestamp: 0,
266             encryptedData: encData,
267             decryptedData: 0,
268             index: ballots.length
269         }));
270
271         bytes32 controlHash = keccak256(abi.encodePacked(msg.sender, votingId, encData.A, encData.B));
272
273         votingsChecks[msg.sender][votingId] = true;
274         ballotsByVoting[votingId].push(newBallotsAmount - 1);
275         ballotsByControlHash[controlHash] = BallotIndex({
276             index: newBallotsAmount - 1,
277             present: true
278         });
279
280         if (!votingsMap[votingId]) {
281             votingsMap[votingId] = true;
282             votingsList.push(votingId);
283         }
```

Видно, что и со стороны контракта BallotRegisty нет обращений к контракту VotersRegistry. Таким образом согласование работы двух контрактов отдано на откуп программному обеспечению, работающему вне блокчейна. Сами контракты позволяют

отметить избирателя как проголосовавшего, но при этом не сохранить его голос и даже не выдать ему бюллетень, и наоборот: сохранить голос, н не отметить, что избиратель проголосовал.

У данной проблемы нет простого решения в рамках выбранной архитектуры, поскольку если изменения в данные обоих контрактов будут вноситься одновременно, то появится возможность сопоставить избирателей и их голоса, и таким образом будет нарушена тайна голосования. Однако, в принципе решения данной задачи существуют. Смотри ссылку на соответствующую статью в разделе «Заключение».

7. Выводы

Блокчейн и смарт-контракты — это мощный инструмент, позволяющий множеству людей действовать сообща, без необходимости доверять организаторам. В частности, эта технология позволяет организовывать организовывать анонимные онлайн-голосования, гарантирующие правильность подсчёта голосов. При этом приватность голосования будет гарантироваться криптографическими протоколами с нулевым разглашением (zero-knowledge protocols), основанными на сложности дискретного логарифмирования и гомоморфных комитентах (homomorphic commitments). Примером такого протокола является Open Vote Network, а его реализация в блокчейне Ethereum представлена в статье «A Smart Contract for Boardroom Voting with Maximum Voter Privacy»[19] При таком подходе тайна голосования не будет нарушена даже если удастся связать зашифрованные голоса избирателей с конкретными людьми.

В московском электронном голосовании использовалась более простая схема, в которой отдельные голоса расшифровываются независимо друг от друга. При таком подходе невозможно одновременно обеспечить тайну голосование и оставить возможность проконтролировать, что все голоса поданы реальными людьми.

Даже тот упрощённый подход, который использовался в Москве, позволяет возложить на смарт-контракты контроль за соблюдением многих важных правил

[19] McCorry P., Shahandashti S. F., Hao F. A smart contract for boardroom voting with maximum voter privacy //International Conference on Financial Cryptography and Data Security. – Springer, Cham, 2017. – С. 357-375.

проведения выборов, таких как время начала и окончания голосования, невозможность для избирателя голосовать в чужом округе, время получения электронного бюллетеня, в течение которого избиратель имеет возможность проголосовать и т.п. Таким образом, контроль за соблюдением этих правил можно было бы сделать прозрачным и доступным для независимых наблюдателей. Однако, этого сделано не было. Контроль за соблюдением большинства правил проведения выборов отдан на откуп программному обеспечению, функционирующему вне блокчейна и недоступному для независимого контроля. Фактически, единственная функция, отданная смарт-контрактам — это подсчёт голосов.

Смарт-контракты и блокчейн гарантируют, что вся записанная в них информация соответствует правилам, прописанным в коде смарт-контрактов и правилам блокчейна, а будучи записанной, информация не может быть изменена или удалена. Однако, эти гарантии действуют только если доступ к блокчейну в реальном времени имеют независимые наблюдатели. Любая попытка записать информацию, нарушающую правила, или изменить ранее записанную информацию будет немедленно замечена.

В процессе проведения московского электронного голосования, у независимых наблюдателей доступа к блокчейну не было, а значит организаторы в принципе имели возможность в любой момент подменить весь блокчейн целиком, заменить несколько последних блоков, или запустить несколько блокчейнов параллельно с разными результатами голосования. Все эти манипуляции остались бы незамеченными.

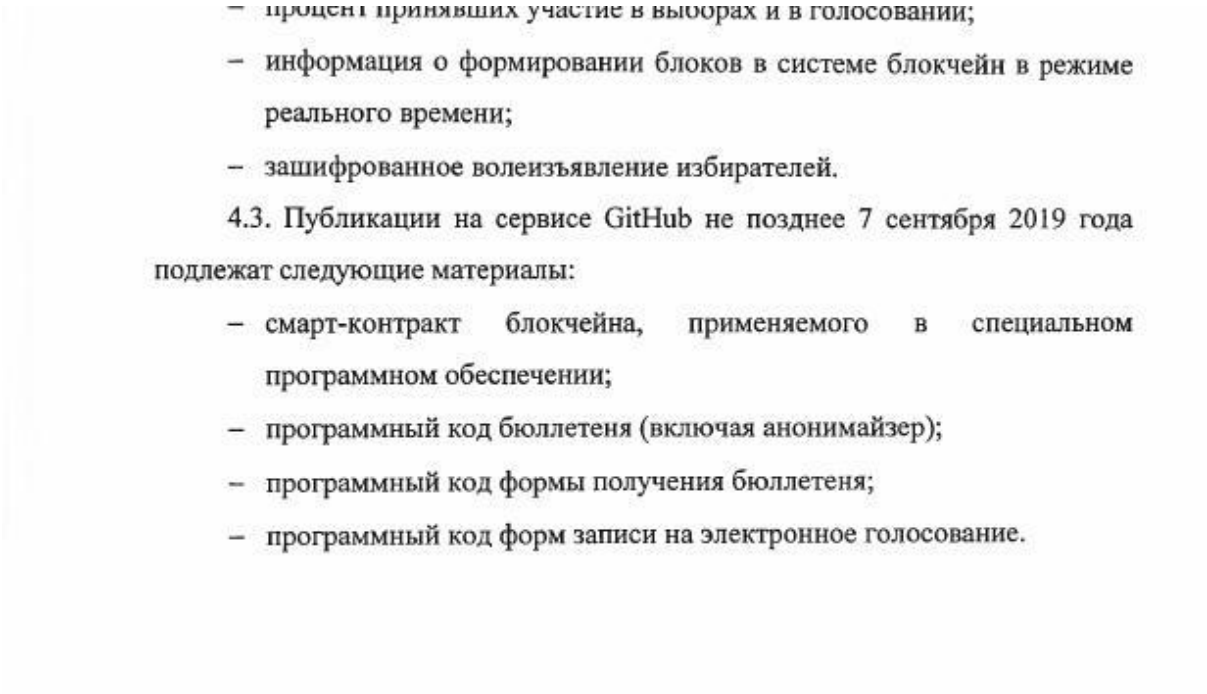
Обобщая всё вышесказанное, можно сказать, что использование блокчейна и смарт-контрактов на электронном голосовании по выборам депутатов в Московскую Городскую Думу носило скорее бутафорский характер и имело своей целью создать у общественности впечатление, что используемые технологии гарантируют соблюдение правил и честный подсчёт голосов. При этом на самом деле организаторы имели возможность вмешиваться в процесс голосования и искажать, таким образом, волю избирателей, а возможности для независимого контроля в случае электронного голосования были даже ниже, чем при других, более традиционных способах организации избирательного процесса.

Аудит PHP кода

1. Введение

Одним из важнейших элементов электронного голосования является php формы. Это формы бюллетеней, Форма регистрации, верификации СМС, анонимайзер.

Согласно Приложению № 1 к решению. Московской Городской Избирательной комиссии от 03.09.2019 № 112/4 «О дополнительных гарантиях обеспечения гласности при проведении эксперимента по дистанционному электронному голосованию»[20].



– процент принявших участие в выборах и в голосовании;
– информация о формировании блоков в системе блокчейн в режиме реального времени;
– зашифрованное волеизъявление избирателей.

4.3. Публикации на сервисе GitHub не позднее 7 сентября 2019 года подлежат следующие материалы:

– смарт-контракт блокчейна, применяемого в специальном программном обеспечении;
– программный код бюллетеня (включая анонимайзер);
– программный код формы получения бюллетеня;
– программный код форм записи на электронное голосование.

Эти формы были обновлены в репозитарии GitHub 7 сентября 2019 года пользователем mbaibakov[21] Mikhail Baibakov.

Всего на Github ветка обновлялась двумя пользователями:

1. mbaibakov Mikhail Baibakov
2. a-borodenkov[22]

[20] http://www.mosgorizbirkom.ru/documents/10279/19437521/rs_112_4.pdf/6569509f-3c35-4562-b337-e58988071d7c
[21] https://github.com/mbaibakov
[22] https://github.com/a-borodenkov

последние обновления были размещены 07.09.2019 и согласно решению МГИК именно этот код-форма и был использован в ходе голосования 08.09.2019

2. Объект аудита

Программный код размещённый в репозитории на GitHub[23]

Отдельно нас интересовали:

1. Программный код бюллетеня;
2. Форма получения бюллетеня;
3. Форма записи на электронное голосование;
4. Анонимайзер.

3. Использованные материалы

Помимо самих аудируемых файлов, в процессе аудита ABDK Consulting использовала следующие материалы:

1. Файлы, лежащие в одном Git-репозитории с аудируемыми файлами[24];
2. Информационные материалы об электронном голосовании, размещённые на официальном сайте мэра Москвы.[25]

3.1 Анонимайзер.

Отдельно стоит сказать об анонимайзере, который должен был быть размещён на GitHube. По факту по данному адресу[26] находится кэш пользователей. Также здесь находится обращение к сервису «Крипто-про» и скрипт рассылки уведомлений. Можно

[23] https://github.com/moscow-technologies/blockchain-voting/tree/master/voting-form/src
[24] https://github.com/moscow-technologies/blockchain-voting
[25] https://www.mos.ru/city/projects/blockchain-vybory/
[26] https://github.com/moscow-technologies/blockchain-voting/tree/master/voting-form/src/crypt/registr

сказать, что требование Решения Московской Городской Избирательной комиссии от 03.09.2019 № 112/4 «О дополнительных гарантиях обеспечения гласности при проведении эксперимента по дистанционному электронному голосованию»[27] не было выполнено и анонимайзер не был выложен на GitHub.

Поэтому, можно сомневаться, был ли анонимайзер задействован в процессе голосования, а значит стоит угроза раскрытия Тайны голосования администраторами ДЭГ или третьими лицами.

4. Выявленные недостатки

4.1. Смарт-контакты не проверяют код из СМС и авторизацию пользователя.

Данная операция не записывается в блокчейн и полностью находится в зоне ответственности ДИТ Москвы и портала mos.ru. Проверка авторизации пользователя находится в PHP формах, но открытом доступе она отсутствует - в коде выложенном на GitHub[28] нет информации о запросах к серверу генерирующему смс.

Нам удалось найти только описание в форме бюллетеня:

blockchain-voting/voting-form/src/forms/forms/mgik/mgd-golosovanie/show.tpl

{include file=«$base_template_path/std_blocks/std_text.tpl» vid=«phone» label=«Телефон» class=«needConfirm» required=true name=«field[declarent.telephone1]» mask=«(999) 999-99-99» autocomplete_from=«REG_DATA:PHONE_MP» container_class=«disabled» disabled=true }

[27] http://www.mosgorizbirkom.ru/documents/10279/19437521/rs_112_4.pdf/6569509f-3c35-4562-b337-e58988071d7c

[28] https://github.com/moscow-technologies/blockchain-voting

Часть в коде:

blockchain-voting/voting-form/src/forms/common/htdocs/forms/js_v3/mgik/mgd-golosovanie.js

$target.data({ oldValue: '', confirmText: 'Телефон успешно подтвержден.', confirmCallback: function () { $('.vote-block').fadeIn('fast'); } });

Мы нашли в репозитории только ссылки на отображение всплывающего окна с уведомлением и поел для ввода СМС.

Можно сделать выводы:

1. Код в репозитарии не полный;
2. Проверка авторизации пользователя может быть отключена администратором системы в момент сбоев, когда всем пользователям вводилась информация об ошибке.

О существовании этой проблемы говорит и представитель ДИТ Сарватдинов Александр Евгеньевич[29]: «Сейчас ребятами проводится такая работа - они смотрят тех пользователей, которые переходили по кнопке бюллетень, но не попадали на него. И им возвращается статус, что они до сих пор не проголосовали. То есть им присылают сообщение, что вы можете получить бюллетень. Они смогут заново проголосовать. Два часа назад это 544 человека так были восстановлены. Полчаса назад порядка 400.»

Таким образом, администратор ДЭГ мог произвольно, без перезапуска системы очистить реестр с кодами СМС и инициировать повторную рассылку СМС сообщений. Что является влиянием на ход голосования и так как можно сопоставить время формирования бюллетеня и контактную информацию избирателя - номер телефона, привязанного к учётной записи MOS.RU, что

[29] https://www.google.com/imgres?imgurl=https://runet-id.com/files/photo/108/1080685_200.jpg?t%3D1523644721&imgrefurl=https://runet-id.com/1080685&docid=vWOh9V1EBLtEaM&tbnid=Ixvy5Qf8LYj6iM:&vet=1&w=200&h=207&source=sh/x/im

нарушает тайну голосования и предоставляет возможности очищать реестры без перезагрузки системы для администратора

4.2. Смарт-контракты не проверяют тайминг работы бюллетеня.

В официальном заявлении указано следующее:

Обращаем внимание, что на голосование отводится 15 минут. Если по истечении этого времени бюллетень останется пустым, избирательное право все равно будет считаться реализованным, так как бюллетень избирателю был выдан.

При этом смарт-контракты отвечающие за запись информации в блокчейн нигде этот тайминг не проверяют.

4.3. Счётчик с таймингом бюллетеня никак не связан со временем активности бюллетеня на сервере

Есть два раздельных таймера, не связанных друг с другом: На стороне клиента и на стороне сервера.

Разберёмся с таймером на сервере.

В формах, находящихся в репозитории GitHub указано время 10 минут, при запуске эти настройки могут быть изменены.

На форме бюллетеня. указывается время жизни бюллетеня:[30]

```
$(function() {

    var timeLimit = 15;
    var $buttons = $('.bulletin__btn');
    var $radios = $('.bulletin__radio');
    var $wrapper = $('.wrapper');
    var $html = $('html');
    var guid = $('#guid').val();
    var userEntropy = '';

    var spaceForButton = 64; // Место, которое займёт кнопка в мобильной вёрстке.
    var nameMinHeight = 80; // Минимальноа высота заголовка пункта в мобильной вёрстке.
    var defaultSpaceForButton = 30;
```

Запускается таймер в момент загрузки бюллетеня у пользователя, загружена должна быть вся страница полностью:[31]

```
                }

                sendHit('Успешное получение бюллетеня','sendBallot',JSON.stringify(data));

                initTimer(timeLimit * 60000, true, 1000, function () {
                    redirectToUrl('/election/error/?code=1');
                });
            },
            error: function (data) {
                sendHit('Ошибка при получении бюллетеня','errorBallot',JSON.stringify(data));
```

Фактически, по истечению 15 минут клиента просто перекидывают на страницу с ошибкой. Больше ничего не происходит. Пользователь может нажать «back» и вернуться обратно.

[30] https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/ballot/static/js/forms/mgik/election.js#L3

[31] https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/ballot/static/js/forms/mgik/election.js#L150

При этом на сервере в момент создания бюллетеня вычисляется время его жизни:[32]

```
249         }
250         $cachePreLife = $this->conf->get('cachePreLife', 600);
251         $lifeTo = $cachePreLife + time();
252         MemoryCache::set('g|'.$guid, array(
253             'okrug' => $data['okrug'],
254             'lifeTo' => $lifeTo,
255             'hash' => $hguid,
256             'opened' => 0), $cachePreLife);
```

Как видно бюллетеню определено время жизни 600 секунд = 10 минут.

В момент первой отправки бюллетеня пользователя время жизни вычисляется заново:[33]

```
316
317         if (empty($data['opened'])) {
318             $cacheLife = $this->conf->get('cacheLife', 600);
319             $data['opened'] = 1;
320             $data['session'] = session_id();
321             $data['lifeTo'] = $time + $cacheLife;
322             MemoryCache::set('g|' . $guid, $data, $cacheLife);
323
324             return true;
325         }
```

Фактически, существует два таймаута: с момента создания бюллетеня до момент первой отправки его пользователю: 10 минут, и с момента первой отправки пользователю до момента отправки заполненного бюллетеня в блокчейн ещё 10 минут.

[32] https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/crypt/registr/v1/index.php#L251
[33] https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/crypt/registr/v1/index.php#L321

У пользователя есть всего 10 минут с момента получения бюллетеня, до момента, когда от должен отправить заполненный бюллетень.

Серверные параметры можно поменять в конфигурации при запуске, но мы не можем сказать были ли они изменены в день голосования.

Так как прямая связь между таймера на клиенте и на сервере отсутствует, то пользователь никогда не узнает был и принят его голос и успел ли он проголосовать в отведённое для этого времени. Наблюдатели также об этом не узнают. так как не имеют доступа к логам сервера и не видят с какими настройками сервер был запущен.

4.4. Проверка авторизации с помощью СМС не проводится, достаточно ввести смс по маске

Данная операция не записывается в блокчейн и полностью находится в зоне ответственности ДИТ Москвы и портала mos.ru. Проверка авторизации пользователя находится в PHP формах, но открытом доступе она отсутствует - в коде выложенном на GitHub[34] нет информации о запросах к серверу генерирующему смс.

Нам удалось найти только описание в форме бюллетеня:

blockchain-voting/voting-form/src/forms/forms/mgik/mgd-golosovanie/show.tpl

{include file=«$base_template_path/std_blocks/std_text.tpl» vid=«phone» label=«Телефон» class=«needConfirm» required=true name=«field[declarent.telephone1]» mask=«(999) 999-99-99» autocomplete_from=«REG_DATA:PHONE_MP» container_class=«disabled» disabled=true }

[34] https://github.com/moscow-technologies/blockchain-voting

Часть в коде:

blockchain-voting/voting-form/src/forms/common/htdocs/forms/js_v3/mgik/mgd-golosovanie.js

$target.data({ oldValue: '', confirmText: 'Телефон успешно подтвержден.', confirmCallback: function () { $('.vote-block').fadeIn('fast'); } });

Мы нашли в репозитации только ссылки на отображение всплывающего окна с уведомлением и поле для ввода СМС.

Можно сделать выводы:

1. код в репозитарии не полный
2. Проверка авторизации пользователя может быть отключена администратором системы в момент сбоев, когда всем пользователям вводилась информация об ошибке

В течение дня голосования, проверка загрузки голосов была отключена администратором, и бюллетени находящимся на сервере были расшифрованы и инициирована новая рассылка сообщений пользователям, чтобы они вошли и проголосовали. При этом проверка корректности ввода смс не использовалась.

СМС сообщение выглядело вот так:

Комментарий наблюдателя ПСГ участок 5003:

– Как вас зовут.

– Александр.

– На данный момент те голоса, которые сейчас не учитываются, что сейчас с ними происходит?

– Смотрите.

– Они подвисли?

– Они не подвисли, там заявитель, он шёл оформить бюллетень. На стороне пгу, московский портал городских услуг, это терминология техническая. Была совершена транзакция о том, что ему выдан бюллетень. И в базу он так и попал. Все. Соответственно он не может второй раз перейти на бюллетень, но бюллетень ему не был выдан. Сейчас ребятами проводится такая работа - они смотрят тех пользователей, которые переходили по кнопке бюллетень, но не попадали на него. И им возвращается статус, что они до сих пор не проголосовали. То есть им сбрасывают сообщение, что вы можете получить бюллетень. Они смогут заново проголосовать. Два часа назад это 544 человека так были восстановлены. Полчаса назад порядка 400.

– То есть это порядка 140 с лишним человек, они получили такое уведомление, перезашли и смогли проголосовать.

– 544 два часа назад, и около 400 полчаса назад.

– То есть это около 900 человек...

– Им всем посылается сообщение о том, что они могут проголосовать электронным голосованием. То есть когда произошел сбой через им всем было отослано сообщение. То есть это где-то более 900.

4.5. В коде PHP также как и в анализе смарт-контрактов отсутствует проверка на уникальность голосующего пользователя.

В ходе аудита смарт-контрактов были выявлены нарушения:

6.1. Контракт VotersRegistry не привязывает номера избирателей к реальным людям.

6.2. Контракт VotersRegistry не привязывает избирателей к округам.

6.5. Контракт BallotsRegistry позволяет одному избирателя проголосовать в нескольких округах.

6.9. Контракты VotersRegistry и BallotsRegistry не взаимодействуют между собой.

Их можно было бы решить нивелировать, добавив в бекэенд формы бюллетеня проверку на уникальность пользователя, но это сделано не было:[35]

4.6. В код бюллетеня добавлена строка, позволяющая трактовать голос избирателя несколькими способами

В последний момент в код бюллетеня был добавлена скрипт, позволяющий произвольно трактовать действия пользователя на форме электронного бюллетеня:[36]

```
10    <script type="text/javascript" src="{$CFG_JS_HOST}/js/forms/mgik/dit.bundle.js"></script>
11    <script type="text/javascript" src="{$CFG_JS_HOST}/js/forms/mgik/election.js?{$smarty.now|date_format:'%Y-%m-%dT%H'}"></script>
12    <script type="text/javascript" src="{$CFG_JS_HOST}/js/forms/mgik/LeavingPageChecker.js?{$smarty.now|date_format:'%Y-%m-%dT%H'}"></scrip
13    <script type="text/javascript" src="{$CFG_JS_HOST}/js/forms/mgik/LeavingPageCheckerInit.js?{$smarty.now|date_format:'%Y-%m-%dT%H'}"></s
14    {if !empty($security)}
15        <script type="text/javascript" src="{$security}?{$smarty.now|date_format:'%Y-%m-%dT%H'}"></script>
16    {/if}
17
18    <div class="wrapper">
19
```

Строка была добавлена в код 7 сентября, и специалисты не имели возможности оценить ее перед выборами.[37]

Эта строка загружает скрипт с сервера ДИТ и выполняет на компьютере у избирателя. Для того, чтобы ей воспользоваться, не нужен доступ к компьютеру избирателя, но нужен доступ к серверу ДИТ.

Изначально в коде бюллетеня была проблема с тем, что при наличии на компьютере пользователя установленного расширения, могла осуществляться подмена голоса.

На скорую руку разработчики в форму бюллетеня добавили этот скрипт. как раз для защиты от таких расширений.

---

[35]https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/crypt/registr/v1/index.php#L222

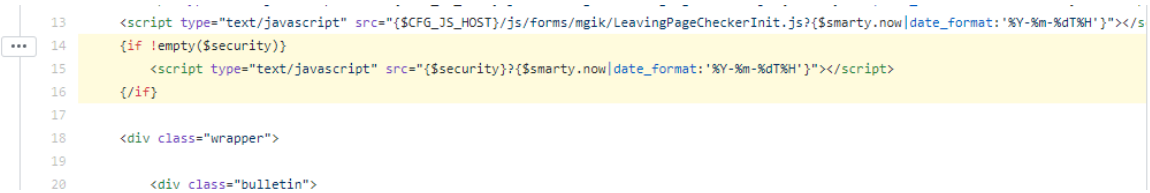
[36]https://github.com/moscow-technologies/blockchain-voting/blob/master/voting-form/src/ballot/common/module_tpl/election/default/show.tpl#L14-L16

[37] https://www.youtube.com/channel/UCpdJvbGI3AOQZ0UZ8LzUKMg
https://www.youtube.com/watch?v=GBD3HDMXhaY&feature=youtu.be

Но, новый скрипт, который задаётся через указание в конфигурации адреса, откуда берётся. Если они поместят туда то, что в расширении было, получат подмену голоса. Если в ДИТ есть злоумышленник, он мог путём модификации конфигурации делать так, что нужный им кандидат получал бы больше голосов. И ему не нужен доступ к компьютеру избирателя.

Без знания значения параметра конфигурации, в котором адрес скрипта записывается, в течение дня голосования (а параметр мог меняться, так что сохранённая каким-то избирателем страница бюллетеня не панацея) нельзя полагаться на корректную работу системы.

На GitHub выложена часть исходных кодов системы, и в этих исходных кодах был найден параметр конфигурации, путём изменения которого можно существенно менять поведение системы.



5. Выводы

PHP формы не устраняют ошибки, выявленные в смарт-контрактах.

Данные формы сами по себе содержат бекдоры, критично влияющие на работоспособность системы, вместо того чтобы перекрывать недостатки смарт-контарктов.

Оставлены широкие возможности для вмешательства администраторов в ход голосования.

Аудит результатов голосования

1. Описание доступных данных

Данные представляют собой два набора файлов:

Файлы *ballots_encrypted_2019-09-08T20_30_00.csv* (размер 6 223 941 байт) и *keys.txt* (1 295 байт).

Файл *ballots_decrypted_2019-09-08.csv* (размер 918 793 байт).

После окончания голосования и публикации итогового CSV-файла с 9810 голосами избирателей приватный ключ был восстановлен из нескольких частей, заранее переданных доверенным лицам. Этот приватный ключ был записан в блокчейн. «Медуза» сумела найти нужный блок и транзакцию с приватным ключом через веб-интерфейс и скопировать данные до тех пор, пока доступ к блокчейну не был закрыт.

Обновление. В мэрии утверждают[38], что специально опубликовали приватный ключ.

Файлы были получены в результате расследования Медузы[39]

1.1. Первый набор данных

файлы *ballots_encrypted_2019-09-08T20_30_00.csv* (размер 6 223 941 байт) и *keys.txt* (1 295 байт).

Их *SHA256* хеш-суммы соответственно:

945F389DE9E80C4127323AB359023D59E3BF8C51874DB351E2EDC2A816B6ED6D

56E8F517CD933C0D9F259F577B9E5A8B2177EE4EDA32FBE5115664020369647F

[38] https://tass.ru/moskva/6884718
[39] https://meduza.io/slides/meriya-sluchayno-pozvolila-rasshifrovat-golosa-na-vyborah-v-mosgordumu-my-eto-sdelali-i-nashli-koe-chto-strannoe?source=post_page-----9a028bf225ea----------

Файл *ballots_encrypted_2019-09-08T20_30_00.csv* внутри себя содержит экспорт голосов из блокчейна в следующем формате: данные записаны в формате *csv* (разделитель точка с запятой - ‘;’)

Позиция	**Тип данных**	**Название**
1	строка	№ ГОЛОСА
2	число	ИЗБИР. ОКРУГ
3	строка	АДРЕС ГОЛОСА В БЛОКЧЕЙНЕ
4	строка	БЛОК ГОЛОСА
5	строка	ВРЕМЯ ЗАПИСИ БЛОКА ГОЛОСА
6	строка	ЗАШИФРОВАННЫЙ ГОЛОС

– “№ ГОЛОСА” — это строка внутри которой записано число, являющееся порядковым номером голоса. Значения от 1 до 9809;

– “ИЗБИР. ОКРУГ” – избирательный округ. Число принимающее три значения 1,10,30;

– “АДРЕС ГОЛОСА В БЛОКЧЕЙНЕ” – строка;

– “БЛОК ГОЛОСА” – строка, содержащая номер блока, в котором сохранен голос;

– “ВРЕМЯ ЗАПИСИ БЛОКА ГОЛОСА” – строка содержащая время записи голоса;

– “ЗАШИФРОВАННЫЙ ГОЛОС” - строка содержащая *json*-значение пары (*a*,*b*) стандартного представления шифр текста, являющегося результатом шифрования номера кандидата по схеме Эль-Гамаля (El-Gamal) с некоторыми изменениями.

Файл *keys.txt* содержит в себе параметры схемы Эль-Гамаля: (*p*,*g*,*y*,*x*) названные соответственно (*modulo*,*generator*,*publicKey*,*privateKey*).

1.2. Второй набор данных

Файл *ballots_decrypted_2019-09-08.csv* (размер 918 793 байт).

Его *SHA256* хеш-сумма:

3E3AE1DDD4A1DB8165C2D9E2EC35A0D391DA0B74D72B099590042EE06E2C0922

Файл *ballots_decrypted_2019-09-08.csv* также внутри себя содержит экспорт голосов из блокчейна в но в другом формате: данные записаны в формате *csv* (но с другим разделителем - запятой ‘,’) и набор полей отличается

Позиция	**Тип данных**	**Название**
1	Число	№ ГОЛОСА
2	Число	ИЗБИР. ОКРУГ
3	Строка	АДРЕС ГОЛОСА В БЛОКЧЕЙНЕ
4	Строка	БЛОК ГОЛОСА
5	Строка	ВРЕМЯ ЗАПИСИ БЛОКА

		ГОЛОСА
6	Число	НОМЕР КАНДИДАТА
7	Строка	ИМЯ КАНДИДАТА

– “№ ГОЛОСА” — это строка, внутри которой записано число, являющееся порядковым номером голоса. Значения от 1 до 9809

– “ИЗБИР. ОКРУГ” — избирательный округ. Число, принимающее три значения 1,10,30;

– “АДРЕС ГОЛОСА В БЛОКЧЕЙНЕ” – строка;

– “БЛОК ГОЛОСА” — строка, содержащая номер блока, в котором сохранен голос;

– “ВРЕМЯ ЗАПИСИ БЛОКА ГОЛОСА” — строка, содержащая время записи голоса;

– “НОМЕР КАНДИДАТА” — идентификатор кандидата;

– “ИМЯ КАНДИДАТА” — фамилия кандидата.

Нам вначале необходимо проверить факт эквивалентности этих наборов данных

1.3. Проверка эквивалентности наборов данных

Для проверки эквивалентности наборов данных нам необходимо реализовать расшифровку аналогичную той, что применялась в голосовании.

При реализации этой проверки помимо ответа на основной вопрос “**Является ли файл *ballots_decrypted_2019-09-08.csv* корректно расшифрованным файлом голосов**” мы должны ответить еще на дополнительные по корректности зашифрованных данных:

Содержит ли файл *ballots_encrypted_2019-09-08T20_30_00.csv* дублирующиеся пары (*a,b*)

А также оценить порядок времени, затрачиваемый на операции шифрования и расшифрования.

Реализация механизма проверки выполнена в проекте *Voting2019.Core*, файл *DatasetsValidator.cs*

Ответы на эти вопросы:

– Количество голосов в обоих файлах равно между собой;

– Файл *ballots_decrypted_2019-09-08.csv* является корректно расшифрованным файлом голосов и соответствует файлу *ballots_encrypted_2019-09-08T20_30_00.csv;*

– Файл *ballots_encrypted_2019-09-08T20_30_00.csv* не содержит дублирующихся пар (*a*,*b*).

Время расшифровки и проверки в однопоточном режиме на ноутбуке с процессором *Intel Core i7-4500U* (2 ядра) за 1 минуту 23 секунды. В многопоточном за 54 секунды. Шифрование голосов реализовано в браузере пользователя. Отсюда следует важный вывод что шифрование и расшифрование голосов создает несущественный вклад в полную нагрузку на инфраструктуру (полное время голосования - около 12 часов).

После того, как мы убедились что наборы данных эквивалентны - мы можем приступать к непосредственному анализу данных.

2. Анализ данных

Для анализа в дальнейшем используем файл *ballots_decrypted_2019-09-08.csv*

2.1. Общие сведения

Номера голосов в записях идут от 0 до 9809. Всего голосов 9910. Каждый голос имеет свой номер и они упорядочены по возрастанию.

Избирательных округов всего три: 1,10,30.

Адрес блока представлен 10 байтовым числом (80бит) записанным в шестнадцатеричной системе счисления

Блок голоса записан в виде номера, к которому приставлен символ ‘#’

Время записи представлено в виде трёх чисел объединённых символами ‘:’ вида *hh:mm:ss* где *hh*-часы, *mm*-минуты, *ss*-секунды

Номер кандидата представлен целым числом умещающимся в 64-битовое представление.

В поле имя кандидата записана фамилия кандидата. В рамках этого документа термины «фамилия кандидата» и «имя кандидата» будут использоваться как синонимы, обозначающие это поле.

В данных присутствуют 16 кандидатов (упорядочены по первому появлению в данных)

606684814	Картавцева
3595230402	Ульянченко
4266376216	Титов
3675433019	Русецкая
4040463197	Дашкевич
1351664840	Суворов
2381501672	Юнеман
2544658436	Жуковский

2563724978	Викулин
339493321	Крюков
2245307987	Дашков
160697653	Жагина
15638149	Милованов
3304824488	Никитушкина
1354976059	Галибин
4191670586	Цыба

2.2. Анализ метрик, связанных с блокчейном

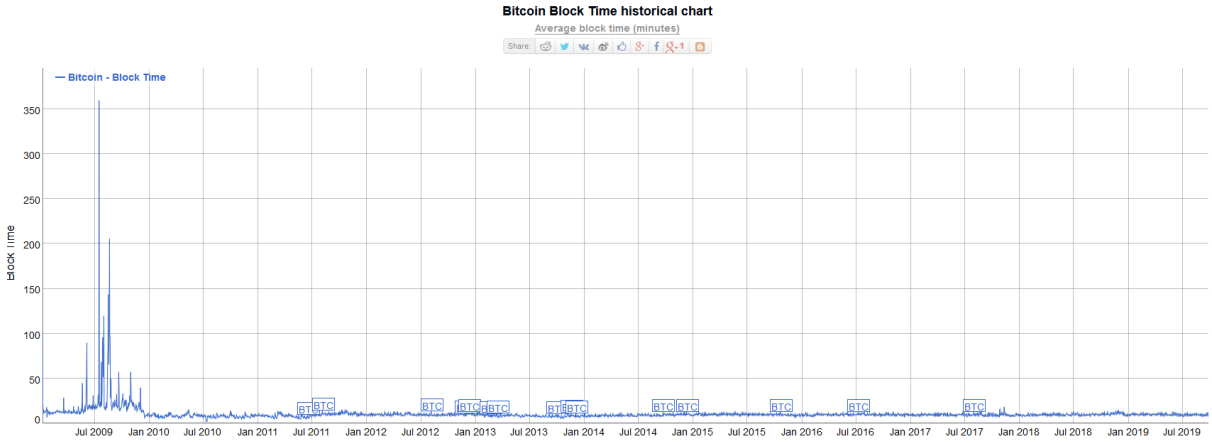
Отправной точкой в данном исследовании станет анализ процесса записи данных голосования в блокчейн.

Первое что следует отметить, что в имеющихся записях нет блоков с “пустыми бюллетенями”, поскольку это именно экспорт голосов, а не самих блоков.

Первая интересующая нас метрика — это «**время блока**» (**block time**). Это среднее время, за которое компьютеры объединённые в блокчейн-сеть выполняют запись одного блока в цепочку. Эта величина имеет важное для нас свойство:

При отсутствии проблем в работающей блокчейн-сети эта величина имеет приблизительно одинаковое значение.

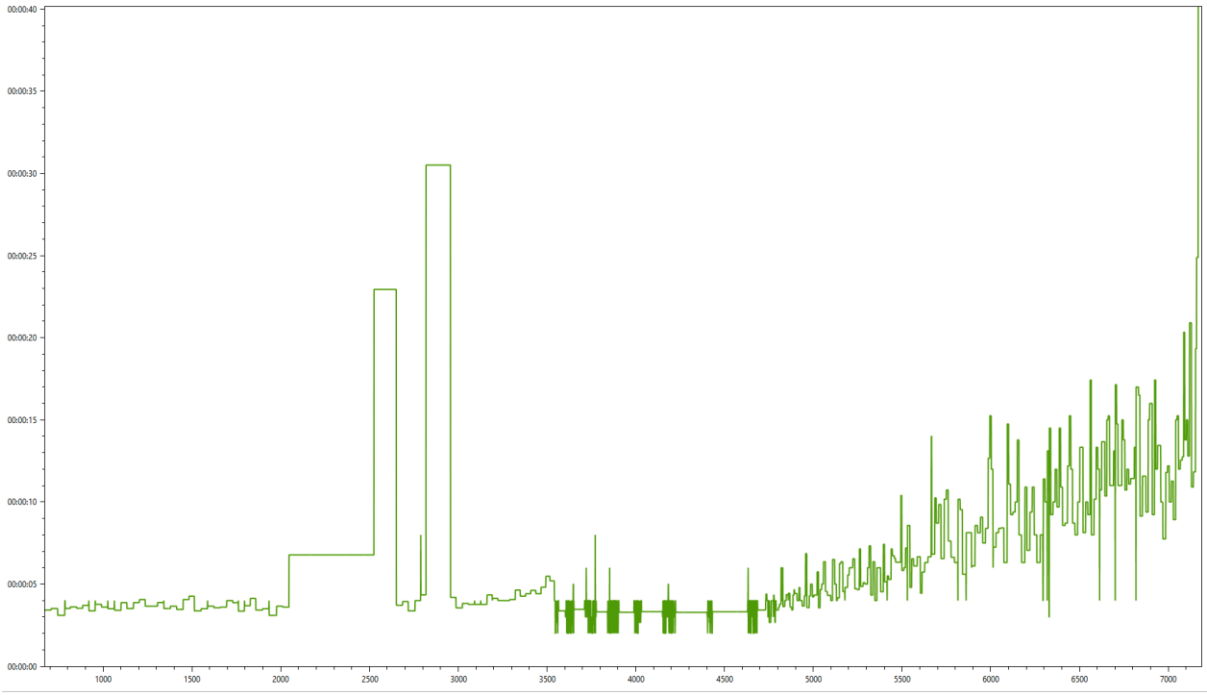
Это свойство демонстрируется, например на графике BitCoin:



За последние 9 лет блокчейн Bitcoin имеет слабо выраженные колебания времени блока при этом повышение более чем в разы было около 10 лет назад. Время блока для блокчейна BTC - около **10 минут**.

В случае Eth график ещё более интересный. Он демонстрирует возможность увидеть “бомбу сложности” и два хард-форка Byzantium (Византий) и Constantinople (Константинополь) после которых время блока стабилизировалось. Т. е., в случае если в блокчейн заложено усложнение вычислений - его среднее время блока будет расти. При этом рост в случае ETH был максимум двукратный в моменты предшествующие хардфорку Византий.

Построим такой же график для данных блокчейн голосования. Здесь важно еще учесть разницу в масштабах. Блок bitcoin вычисляется приблизительно в 10 минут, в то время как время eth приблизительно 15 секунд.

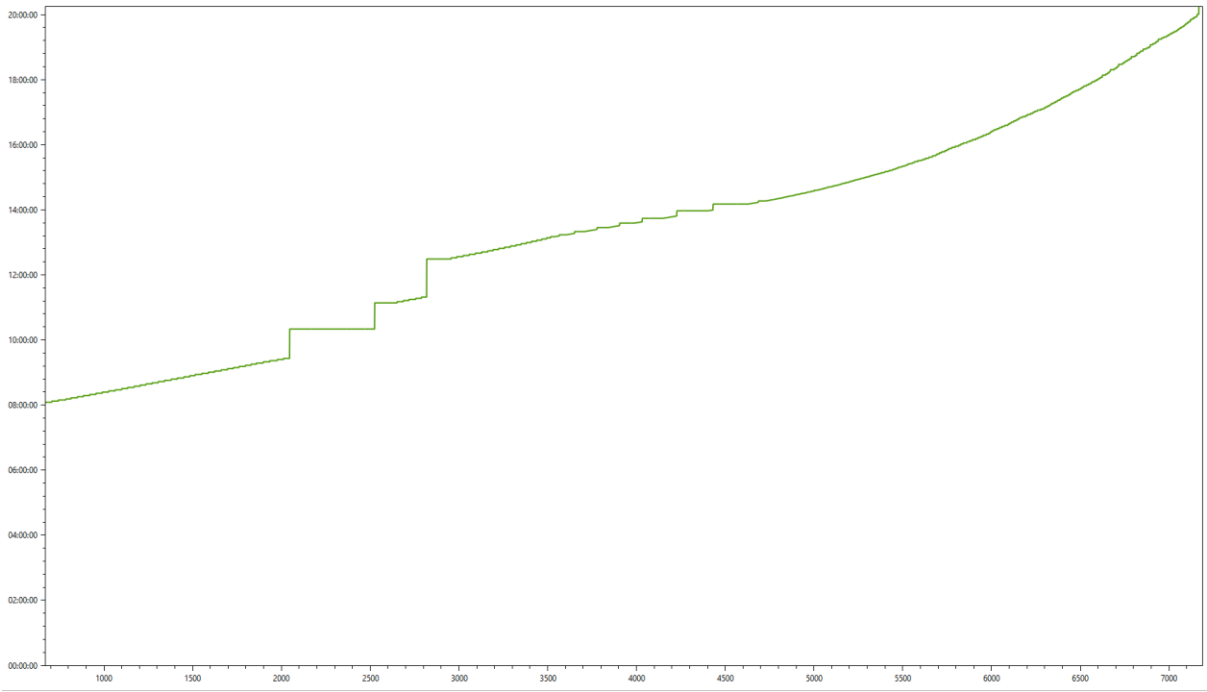


Зависимости block time от номера блока (если данных по блоку нет, то берём среднее время полученное из разницы block timestamp деленное на число блоков)

Исходя из графика мы видим, что блокчейн голосования был настроен изначально на время меньшее чем 5 секунд. Это очень важно для нас потому что в случае работы блокчейна в режиме Proof-of-Authority время блока фиксируется в начале его работы. Мы видим время нормального функционирования (до первого скачка), а также мы аномальные зоны, где либо время блока сильно отличалось от того что было на старте, либо общее поведение отличалось от стартового.

Вторая интересующая нас метрика - это ‘**время записи блока**’ (**block timestamp**). Эта величина описывает время в которое блок был сгенерирован.

Напрямую нам эти данные недоступны, но у нас есть время из ‘block timestamp’ без даты и без миллисекунд. Этого достаточно если block time у нас будет больше 1 секунды. В этом исследовании точность значений block time и block timestamp будет около 1 секунды.



Зависимость block timestamp от номера блока (если данных по конкретному блоку нет - берём предыдущее значение).

На основе графика времени формирования блока (block timestamp) мы отчётливо видим аномалии в в районах блоков: 2046,2525,2818

Соответствующие им аномальные записи выглядят так:

3307,30,0xA547CF0CA7055Bcc0Ac0d2Ff177DE92bC4B2F562,#2046,09:26:04,4191670586,Цыба

3308,30,0xF2E5F538FeAAB5ba45f03a8273E10F5dd5E72e46,#2525,10:20:12,2381501672,Юнеман

3342,10,0xd832bEa38B2FDeaAB8243883A75Ea4658848e524,#2525,10:20:12,606684814,Картавцева

3343,1,0x884fC8fDE90a32b249818db99260d7D366954Ef3,#2651,11:08:22,160697653,Жагина

3434,1,0x532413d7b71bE8846a37481b836097209AD5C874,#2818,11:19:08,3595230402,Ульянченко

3435,30,0x4C9f7eEd369e8989852f45332e7BC79C332DA1bD,#2956,12:29:18,3675433019,Русецкая

Здесь в наличии ~2 часовой интервал, за который был всего 1 блок с данными голосования. И ещё один интервал ~1 час, когда данных не было. Запомним эти зоны, чтобы потом рассмотреть их подробнее когда мы перейдём непосредственно к анализу данных голосования.

На основании графика block time мы можем помимо вышеуказанных зон найти еще 3541-4733 где данные пишутся странно и с разрывами. Это время с 13:10 по 14:16.

Также этот график позволяет нам оценить среднее время блока в 3-4 секунды для первых двух третей блокчейна, а также тот факт, что в последней трети это время начало стабильно расти, что не наблюдалось ранее.

Это позволяет сделать нам следующие предположения:

На промежутке времени с 09:26 до 12:29 генерация блоков останавливалась на всех нодах по крайней мере дважды

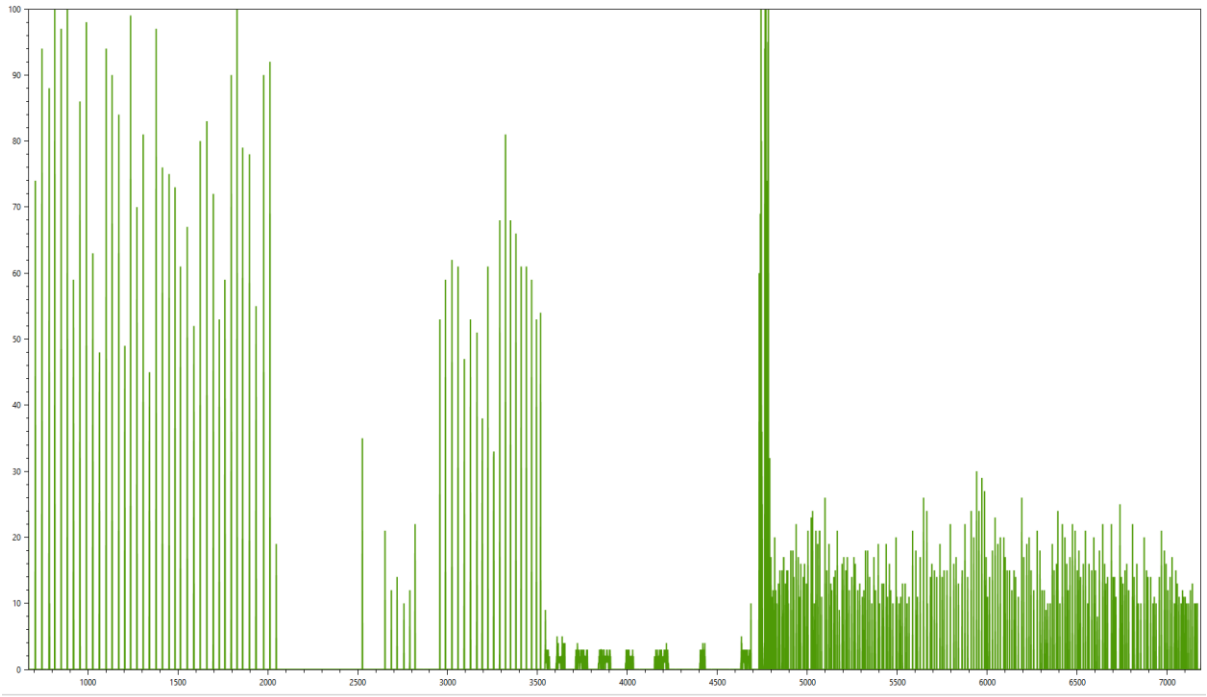
На промежутке времени с 13:10 по 14:16 данные по голосам частично не доходили блокчейна

После 14:16 произошло какое-то изменение в режиме работы блокчейна, которое сильно повлияло на общую стабильность системы.

2.3. Проверка результатов голосования

После того как мы разобрались с параметрами, которые непосредственно связаны с блокчейном, мы переходим к данным голосования.

Построим распределение голосов по блокам. Т. е. зависимость количества голосов в блоке от номера блока.



На основании этого распределения и двух распределений, полученных из метрик блокчейна, мы можем выделить следующие этапы работы системы. Эти зоны мы будем называть римскими цифрами.

Название	**N старт**	**N окончание**	**T старт**	**T окон**
Зона I	668	2046	08:02:58	09:26:04
Зона II	2046	2525	09:26:04	10:20:12
Зона III	2525	2651	10:20:12	11:08:22
Зона IV	2651	2818	11:08:22	11:19:08
Зона V	2818	2956	11:19:08	12:29:18

Зона VI	2956	3543	12:29:18	13:10:36
Зона VII	3543 (*)	4732 (*)	13:10:36	14:16:24
Зона VIII	4732 (*)	4804 (*)	14:16:24	14:20:44
Зона IX	4804 (*)	7168 (*)	14:20:44	20:00:44

Зоны отмеченные * имеют неточные границы.

Дадим характеристики этих зон.

– Зона I - зона устойчивой работы блокчейна. Максимальное время блока 4.2 секунды, минимальное - 3.1 секунды. Голоса сбрасываются в блокчейн каждые 32-36 блоков. Т. е. приблизительно каждые 2 минуты;

– Зона II - зона отсутствия записи голосов;

– Зона III - зона отсутствия записи голосов;

– Между зонами II и III находится единственный блок 2525;

– Зона IV - зона восстановления работы;

– Зона V - зона отсутствия записи голосов;

– Зона VI - зона восстановления работы;

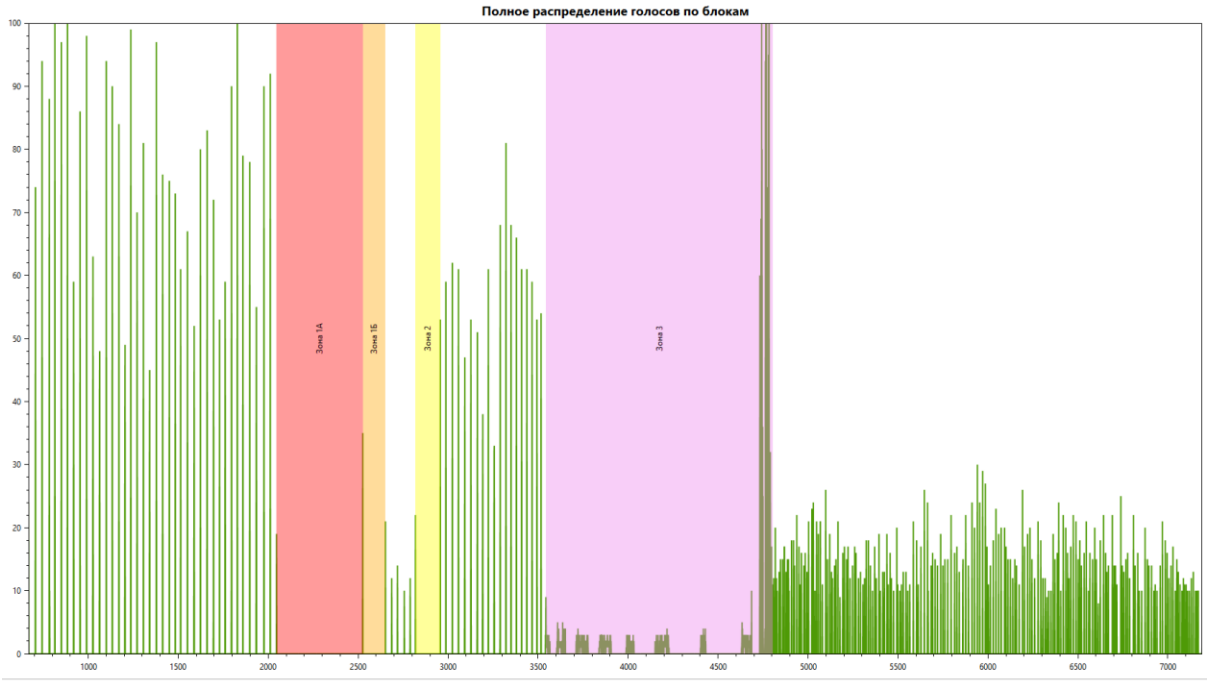
– Зона VII - зона с странной записью голосов. Во-первых внутри всех блоков максимум 5 голосов (а чаще 1,2,3 голоса). Во-вторых эта запись идет уже не блоками, а постоянно. Помимо этого есть участки, где нет голосов. Время блока 4 секунды там где есть запись данных и 3.3 секунды там где записи нет;

– Зона VIII - зона резкого всплеска записи голосов при этом время блока также оставалось около 4 секунд с пиками небольшими пиками по 4.4 секунды;

– Зона VIII - зона изменения характера записи данных в блокчейн, а также появление тенденции роста времени блока. Эта зона интересна тем, что содержит один блок после завершения времени голосования.

– Отметим, что максимальное количество голосов в блоке - 100. Такие блоки встречались в зонах I (нормальная работа) и VIII (аномальная запись голосов в блокчейн).

На основании этого мы выделим три зоны где система была явно неработоспособна.



Первая зона — это области II и III. Вторая зона - это область V. Третья зона — это области VII и VIII

Общее время неработоспособности системы на этих участках - 3:58:58 т.е. около 4 часов. Это оценка снизу не учитывает других аномалий

2.4. Расчет времени остановки блокчейна

Рассмотрим отдельно интервалы неработоспособности и оценим время, на которое останавливалось вычисление блоков на всех нодах. Это будет время, за которое могли быть сделаны действия, связанные с подменой голосов, ну или правки ошибок или в принципе любых действий связанных с **любым** изменением данных.

Среднее время блока у нас находится в интервале между 3-4 секундами. Оценим время остановки для среднего времени вычисления блока в 4 секунды.

Начало № блока	Конец № блока	Число блоков	Время начала	Время окончания	Время интервала	Оценка времени вычисления	Оценка времени остановки
2046	2525	479	09:26:04	10:20:12	0:54:08	0:31:56	0:22:12
2525	2651	126	10:20:12	11:08:22	0:48:10	0:08:24	0:39:46
2818	2956	138	11:19:08	12:29:18	1:10:10	0:09:12	1:00:58

Суммарно блокчейн был отключен чуть более 2-ух часов. Это оценочное время когда не работал ни один из узлов внутри блокчейн сети. Это происходило в зонах 1 и 2.

2.5. Анализ свидетельских показаний и фотоматериалов с участка 5003

Во время голосования на участке 5003 находились мониторы, которые давали нам представление о ходе голосования.

Основные поля на мониторах:

– всего зарегистрировано пользователей — это избиратели которые перешли на страницу авторизации. при обновлении страницы пользователем, счётчик +1, поэтому это число формально;

– ввели СМС - избиратели, которые прошли авторизация на сайте;

– получили бюллетень - на сервере произведена запись о том, что избирателю выдан бюллетень;

– проголосовали - в блокчейн записан голос избирателя (мы предполагаем, что данные записаны в блокчейн. так как расшифровывать бюллетень на сервере нельзя);

– число неиспользованных бюллетеней - расчётное поле, разница между выданными бюллетенями и записанных в блокчейн.

Фотографии экранов находятся в приложении (см ПРИЛОЖЕНИЕ 1. Фотографии).

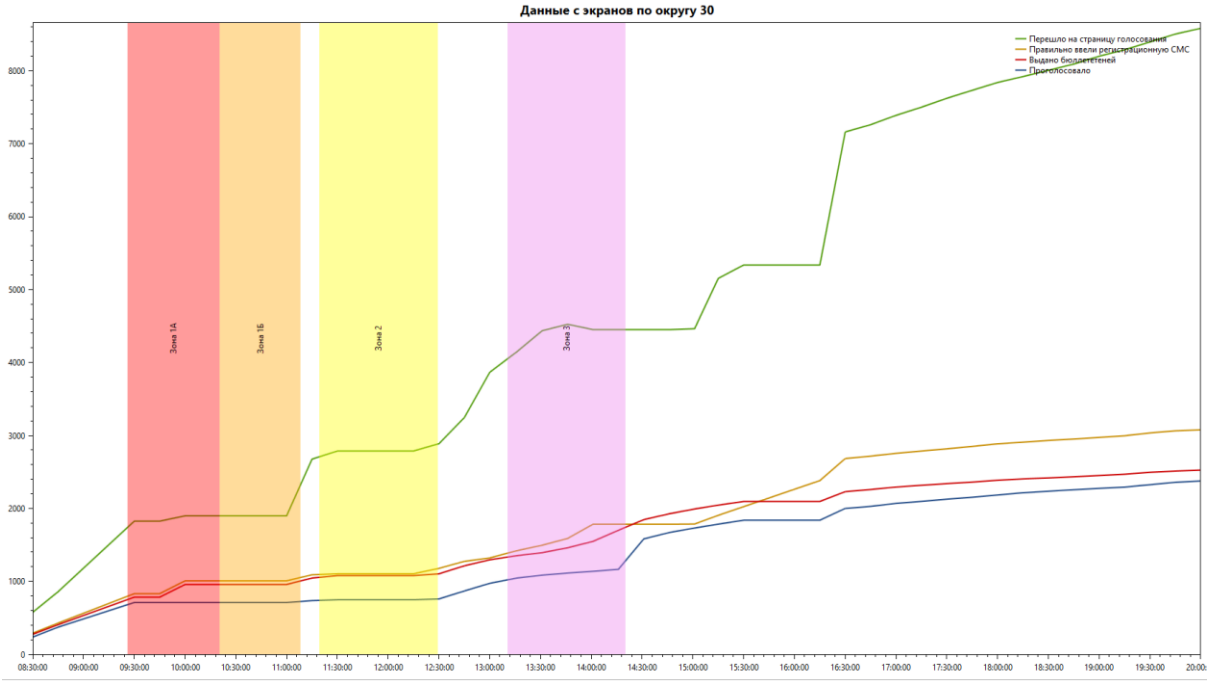
Время	**Зарегистр.**	**Ввели СМС**	**получили**	**Проголосовали**
08:30:00	575	289	274	236
08:45:00	858	428	406	372
09:30:00	1825	831	783	710
09:45:00	1825	831	783	710
10:00:00	1898	1007	957	710
10:15:00	1898	1007	957	710
10:30:00	1898	1007	957	710
10:45:00	1898	1007	957	710
11:00:00	1898	1007	957	710
11:15:00	2675	1090	1045	736

11:30:00	2787	1104	1079	748
11:45:00	2787	1104	1079	748
12:00:00	2787	1104	1079	748
12:15:00	2787	1104	1079	748
12:30:00	2887	1178	1103	759
12:45:00	3245	1274	1213	868
13:00:00	3866	1321	1295	973
13:16:00	4147	1420	1352	1045
13:31:00	4436	1495	1393	1085
13:46:00	4524	1588	1460	1114
14:01:00	4451	1783	1548	1138
14:16:00	4451	1783	1700	1165
14:31:00	4451	1783	1846	1582
14:46:00	4451	1783	1926	1669
15:01:00	4464	1786	1992	1731

15:15:00	5153	1905	2045	1784
15:30:00	5337	2024	2095	1838
15:45:00	5337	2143	2095	1838
16:00:00	5337	2262	2095	1838
16:15:00	5337	2381	2095	1838
16:30:00	7161	2683	2230	2000
16:45:00	7260	2717	2259	2028
17:00:00	7388	2755	2293	2069
17:15:00	7500	2787	2317	2096
17:30:00	7622	2817	2340	2127
17:45:00	7731	2850	2361	2153
18:00:00	7839	2885	2386	2184
18:15:00	7920	2908	2405	2214
18:30:00	8005	2932	2418	2235
18:45:00	8091	2951	2434	2256

19:00:00	8199	2973	2451	2275
19:15:00	8287	2996	2469	2292
19:30:00	8394	3035	2495	2325
19:45:00	8503	3064	2512	2358
20:00:00	8581	3077	2525	2376

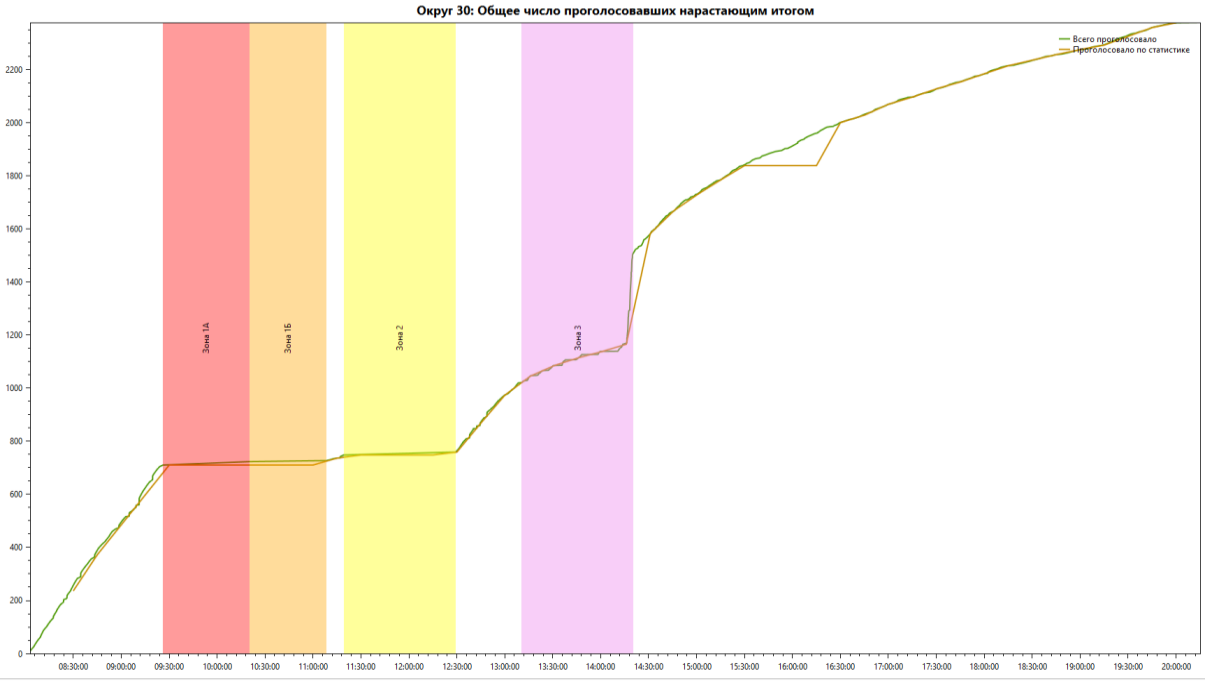
Визуализация этих данных показана ниже



С 14:31 по 15:30 количество выданных бюллетеней превышало количество людей, которые правильно ввели СМС, а в 14:01 зачем-то количество перешедших на страницу голосования было уменьшено на 73.

Любопытно то, что во время самого большого количества голосов, попавших в блокчейн, на страницу голосования по 30 округу никто практически не переходил.

Сравнив данные из блокчейна с данными по экранам, мы можем выявить аномальные данные статистики и ошибки фронтенда



Откуда видно, что данные статистики с 15:30 по 16:15 скорее всего содержат отставание, в то время как остальные данные довольно точно повторяют полученные нами из блокчейна

Ниже представлена таблица с блоками, и описанием сбоев.[40]

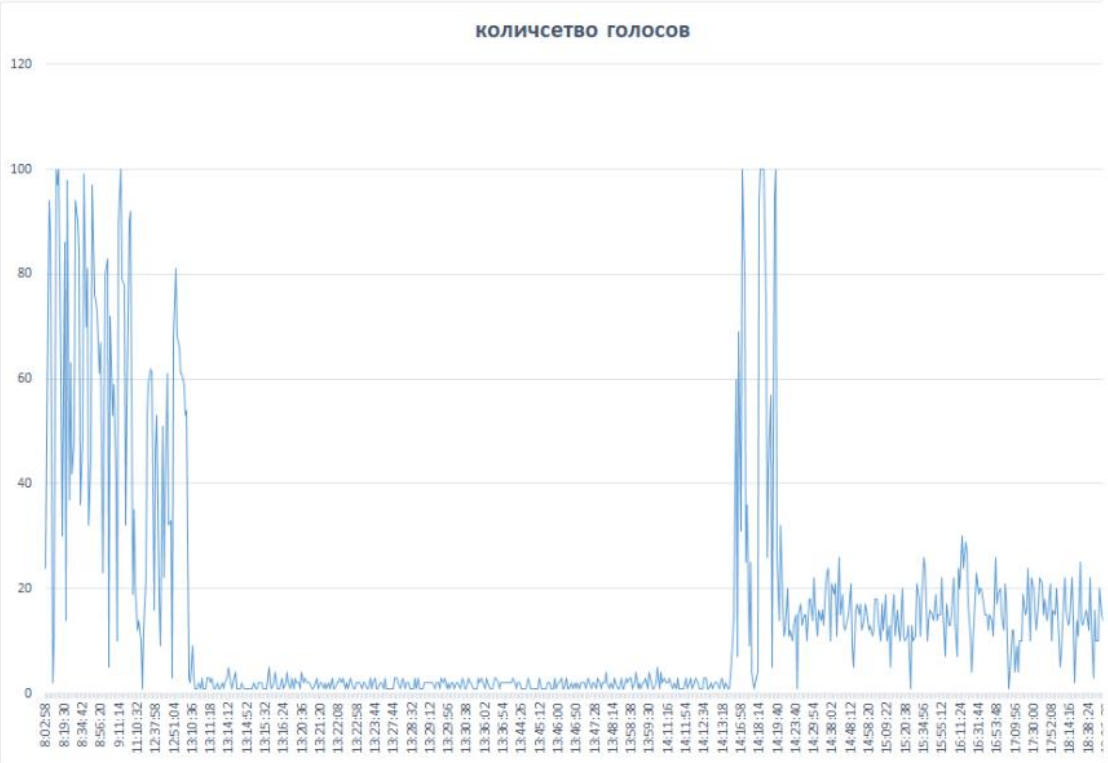
данные из блокчейна				данные с экранов мониторинга на УИК 5003					
номер блока	время записи блока	количество блоков без голосов, предшествующих данному блоку	среднее время формирования одного блока	всего зарегистрировано пользователей	ввели СМС	получили бюллетень	проголосовали	число неиспользованных бюллетеней	комментарий
	8:00:00 AM			0		0	0	0	Начало голосования в 8.00.
668	8:02:58 AM								записан первый блок с голосами
884	8:15:22 AM	34	0:00:04	291	144	129	102	27	Подошли к концу первые 15 минут для контроля. Появилась статистика для наблюдателей См фото в приложении.
1133	8:30:20 AM	33	0:00:04	575	284	274	236	38	Завершился очередной интервал для мониторинга, результаты отображены на экранах наблюдателей
1378	8:45:34 AM	37	0:00:04	858	428	406	372	34	Завершился очередной интервал для мониторинга, результаты отображены на экранах наблюдателей . Кол-во неиспользованных бюллетеней уменьшилось на 4 ед. это возмоно, так как бюллетени могли быть выданы в конце предыдущей 15-тиминутки
1623	9:00:38 AM	35	0:00:04	1129	540	513	482	31	Завершился очередной интервал для мониторинга, результаты отображены на экранах наблюдателей . Кол-во неиспользованных бюллетеней уменьшилось незначительно, это возмоно, так как бюллетени могли быть выданы в конце предыдущей 15-тиминутки
1859	9:15:20 AM	31	0:00:04	1461	690	649	627	22	Завершился очередной интервал для мониторинга, результаты отображены на экранах наблюдателей . Кол-во неиспользованных бюллетеней уменьшилось незначительно, это возмоно, так как бюллетени могли быть выданы в конце предыдущей 15-тиминутки
2046	9:26:04 AM	35	0:00:04						Последний блок, записанный перед началом технического сбоя.
	9:30:00 AM	нет блока с голосами		1825	831	783	710	73	
	9:45:00 AM	нет блока с голосами		1825	831	783	710	73	
	10:00:00 AM	нет блока с голосами		1898	1006	957	710	247	
	10:15:00 AM	нет блока с голосами		1898	1006	957	710	247	Запись данных в блокчейн не ведется. Пользовательский интерыейс, так же не доступен. Есть подтверждения от голосующих, о том, что система была недоступна, также представитель ДИТ в своем комментарии подтвердил этот сбой.
2525	10:20:12 AM	479	0:00:07						Система ненадолго заработала, в блоке загружено 35 голосов, в том числе 13 голосов по участку 5003, но на экране для наблюдателей результат не был отображен. UI по-прежнему остается недоступен
	10:30:00 AM	нет блока с голосами		1898	1006	957	710	247	
	10:45:00 AM	нет блока с голосами		1898	1006	957	710	247	
	11:00:00 AM	нет блока с голосами		1898	1006	957	710	247	Запись данных в блокчейн не ведется. Пользовательский интерыейс, так же не доступен. Есть подтверждения от голосующих, о том, что система была недоступна
2651	11:08:22 AM	126	0:00:23						Окончание 1 тех сбоя. Время которое система не работала: 1:42:18 (11:08:22 - 9:26:04)
	11:15:00 AM	нет блока с голосами		2675	1090	1045	736	309	Завершился очередной интервал для мониторинга, результаты отображены на экранах наблюдателей
2818	11:19:08 AM	29	0:00:04						последний блок перед 2 сбоем
	11:30:00 AM	нет блока с голосами		2787	1002	1079	736	343	
	11:45:00 AM	нет блока с голосами		2787	1002	1079	736	343	
	12:00:00 PM	нет блока с голосами		2787	1002	1079	736	343	
	12:15:00 PM	нет блока с голосами		2787	1002	1079	736	343	Запись данных в блокчейн не ведется. Пользовательский интерыейс, так же не доступен. Есть подтверждения от голосующих, о том, что система была недоступна, также представитель ДИТ в своем комментарии подтвердил этот сбой.
2956	12:29:18 PM	138	0:00:31						Окончание 2 сбоя (принудительного отключения ситемы) время отключения = 12:29:18 - 11:19:08 = 1:10:10

[40] https://drive.google.com/file/d/15yXnvNn5Ci4WSFXvmIQyu31-qxSZwXmX/view?usp=sharing

2.6. Распределение голосов по блокам:

Максимальное количество голосов в одном блоке = 100.

Ограничение на максимальное количество голосов (бюллетеней) в блоке в коде мы не нашли



В период с 13:10:28 до 14:13:46 количество записанных голосов в БЧ резко снижается.

Блоки, в которые записано наибольшее количество голосов:

БЛОК ГОЛОСА	**ВРЕМЯ ЗАПИСИ БЛОКА ГОЛОСА**	**Кол-во голосов**	**Доля голосов в блоке от общего количества**	**% с накопистельным итогом**
814	8:11:12	100	1,02%	1,02%
884	8:15:22	100	1,02%	2,04%

1828	9:13:12	100	1,02%	3,06%
4742	14:17:06	100	1,02%	4,08%
4765	14:18:22	100	1,02%	5,10%
4767	14:18:30	100	1,02%	6,12%
4769	14:18:38	100	1,02%	7,14%
4783	14:19:32	100	1,02%	8,15%
1236	8:36:52	99	1,01%	9,16%
990	8:21:46	98	1,00%	10,16%
850	8:13:22	97	0,99%	11,15%
1378	8:45:34	97	0,99%	12,14%
4780	14:19:22	95	0,97%	13,11%
743	8:07:18	94	0,96%	14,07%
1100	8:28:12	94	0,96%	15,03%
4763	14:18:14	94	0,96%	15,98%
2011	9:23:58	92	0,94%	16,92%
1133	8:30:20	90	0,92%	17,84%
1796	9:11:14	90	0,92%	18,76%

1976	9:21:50	90	0,92%	19,67%
783	8:09:22	88	0,90%	20,57%
954	8:19:30	86	0,88%	21,45%
1170	8:32:30	84	0,86%	22,30%
1660	9:02:50	83	0,85%	23,15%
1306	8:41:08	81	0,83%	23,98%
3321	12:53:14	81	0,83%	24,80%
1623	9:00:38	80	0,82%	25,62%
4744	14:17:14	80	0,82%	26,43%
1859	9:15:20	79	0,81%	27,24%
1897	9:17:30	78	0,80%	28,03%
1413	8:47:42	76	0,77%	28,81%
1450	8:49:50	75	0,76%	29,57%
706	8:05:08	74	0,75%	30,33%
4772	14:18:50	74	0,75%	31,08%

Время записи одного блока в этой выборке 3-4 секунды. (тут я не нашла какой-то ошибки, но распределение по блокам и по времени выглядит довольно странно.

Время записи - с 8 утра до 1 сбоя. и с 14.17 до 14.19

В 34 блоках из 7190 блоков записаны 31,08 % голосов. все эти блоки получены в интервал времени = 1 час 20 минут от всего времени голосования. И можно было бы оправдать эти записи тем, что после сбоя накопленные данные записываются в блокчейн, но но момент записи этих блоков сбоев не было. До 9:26:04 система работала стабильно. и после 14:16:58 система также работала стабильно, информации о сбоях не поступало

См таблицу:

БЛОК ГОЛОСА	**ВРЕМЯ ЗАПИСИ БЛОКА ГОЛОСА**	**Кол-во голосов**	**Доля голосов в блоке от общего количества**
706	8:05:08	74	0,754%
743	8:07:18	94	0,958%
783	8:09:22	88	0,897%
814	8:11:12	100	1,019%
850	8:13:22	97	0,989%
884	8:15:22	100	1,019%
954	8:19:30	86	0,877%
990	8:21:46	98	0,999%
1100	8:28:12	94	0,958%
1133	8:30:20	90	0,917%
1170	8:32:30	84	0,856%
1236	8:36:52	99	1,009%
1306	8:41:08	81	0,826%

1378	8:45:34	97	0,989%
1413	8:47:42	76	0,775%
1450	8:49:50	75	0,765%
1623	9:00:38	80	0,815%
1660	9:02:50	83	0,846%
1796	9:11:14	90	0,917%
1828	9:13:12	100	1,019%
1859	9:15:20	79	0,805%
1897	9:17:30	78	0,795%
1976	9:21:50	90	0,917%
2011	9:23:58	92	0,938%
3321	12:53:14	81	0,826%
4742	14:17:06	100	1,019%
4744	14:17:14	80	0,815%
4763	14:18:14	94	0,958%
4765	14:18:22	100	1,019%
4767	14:18:30	100	1,019%
4769	14:18:38	100	1,019%
4772	14:18:50	74	0,754%

4780	14:19:22	95	0,968%
4783	14:19:32	100	1,019%

2.7. Анализ времени генерации блока

Также стоит обратить внимание на увеличение временного интервала для генерации каждого блока. Начиная с 2:13:46 PM начинает увеличиваться время генерации одного лока.

Так как в протоколе PoA 2.2. Proof of Autority отключение валидатора не влияет на скорость обработки транзакций - все валидаторы проверяют все транзакции, то вероятно мы видим плавную деградацию пропускной способности из-за накопления данных в очереди.

По словам экспертов, в конце «зоны 3» они решили дослать в блокчейн очень большое количество голосов по какой-то причине, не опубликованных ранее. Запустили, условно говоря, какой-то скрипт, который создал поток транзакций, в разы превышающий пропускную способность блокчейна. Очередь транзакций стала стремительно расти, но поначалу блокчейн справлялся, генерируя полные блоки каждые 4-5 секунд.

В какой-то момент размер очереди превысил некое пороговое значение, например переполнил кэш в оперативной памяти, и, условно говоря, очередь стала выгружаться на диск. В итоге производительность системы резко деградировала, количество транзакций в блоке упало, время блока начало расти.

Ну а дальше, поскольку транзакции продолжали поступать (от скрипта или из ЛК), очередь продолжала расти и производительность продолжала падать до конца дня. То есть система так и не смогла раскидать накопившуюся очередь и вернуться в нормальный режим.

У Parity есть вот следующие настройки:

--tx-queue-mem-limit=[MB]

Maximum amount of memory that can be used by the transaction queue. Setting this parameter to 0 disables limiting. (default: 4)

--tx-queue-size=[LIMIT]

Maximum amount of transactions in the queue (waiting to be included in next block). (default: 8192)

И вот такой параметр:

--no-persistent-txqueue

Don't save pending local transactions to disk to be restored whenever the node restarts.

Откуда следует, что для локальных транзакций, то есть отправленный самим узлом, у узна есть хранилище на диске, чтобы их не потерять.

Если предположить, что у них были настройки по-умолчанию, то в память влезло всего 4 МБ транзакций (или 8 тысяч, смотря во что упрёмся раньше). А дальше каждый узел стал писать свои транзакции на диск (свои — это те, которые были отправлены через этот узел), а чужие транзакции стал просто забывать.

Каждый голос — это две транзакции на самом деле

И голоса — это довольно тяжелые транзакции, так что скорее в 4 МБ упёрлись, чем в 8 тысяч

1. Чтобы человек мог проголосовать, сначала надо его адрес добавить в список разрешённых[41]:

```
192
193     function addVoterToAllowedVoters(address voter, uint256 votingId) external onlyOwner {
194         require(
195             isRegistryClosed == false,
196             "Registry must not be closed!");
197
```

2. А уже потом сам голос можно публиковать:[42]

```
233     function addBallot(uint256 votingId, bytes memory _A, bytes memory _B) public {
234         require(
235             isRegistryClosed == false,
236             "Registry must not be closed!");
```

[41] 1. https://github.com/moscow-technologies/blockchain-voting/blob/master/smart-contracts/packages/smart-contracts/contracts/BallotsRegistry.sol#L193

[42] 2. https://github.com/moscow-technologies/blockchain-voting/blob/master/smart-contracts/packages/smart-contracts/contracts/BallotsRegistry.sol#L233

Таким образом, мы предполагаем, что данная реализация электронного голосования не будет работать стабильно при масштабировании.

3. Выводы

Если собрать воедино данные об оценке записей в блокчейне, результаты анализа голосов и учесть распределение голосов по блокам. можно сформировать следующую таблицу:

номер сбоя	**блок и время начала**	**блок и время окончания**	**длительность сбоя**	**комментарий**
1	2046 9:26:04	2651 11:08:22	1:42:18	Запись данных в блокчейн не ведется. Пользовательский интерфейс, также не доступен. Есть подтверждения от голосующих, о том, что система была недоступна, также представитель ДИТ в своем комментарии подтвердил этот сбой.
2	2818 11:19:08	2956 12:29:18	1:10:10	Запись данных в блокчейн не ведется. Пользовательский интерфейс, также не доступен. Есть подтверждения от голосующих, о том, что система была недоступна, также представитель ДИТ в своем комментарии подтвердил этот сбой. ответ ДИТ: *“В 11 ч, 20 мин, предпринята* *попытка переключения на тестовый контур оборудования, однако на тестовом контуре наблюдалась нестабильная работа сервиса ГОСТ-шифрования, По* *итогам принято решение о переключении на другое оборудование и в 12 часов 29 минут голосование было возобновлено”*

3	4431 13:59:34	5155 14:47:02	0:47:28	весь интервал с 13:59 пользователи не переходят на страницу голосования и не могут ввести СМС. то есть система недоступна для пользователей, при этом продолжается учет голосов, выдача бюллетеней и запись голосов. по сообщению представителя дит, в указанный интервал пользователям начали рассылать смски с просьбой войти и переголосовать. так как их бюллетень не был учтен. Но сквозной мониторинг показывает отсутствие активности в UI и продолжении записи блоков с голосами. Мы рассматриваем это как вмешательство заинтересованных лиц в ход голосования и подтасовка результатов. Сотрудник Дит подтверждает на аудиозаписи факт вмешательства в систему, расшифровки «повисших» бюллетеней и идентификации избирателей. Им начинают отправлять СМСки с просьбой проголосовать. Избиратели готовы подтвердить этот факт
4	5603 15:30:50	6029 16:27:36	0:56:46	судя потому, что количество обновлений странице не изменяется, вероятно была нарушена связь между интерфейсом пользователя и сервером, как максимум, и как минимум, с сервера не передавалась информация для наблюдателей, т.е. сквозной аудит отсутствовал. При этом в БЧ продолжают записываться голоса. что это за голоса не понятно. если число авторизованных пользователей не изменяется.

5		4150 13:44:20	0:06:00	6 минут в блокчейн не записываются голоса.
6		4402 13:58:06	0:09:00	175 блоков и 9 минут в БЧ не записываются голоса
7*		4630 14:10:34	0:11:00	11 минут в БЧ не записываются голоса блок попадает в интервал №3

Во время 1 и 2 сбоев система была недоступна пользователям. и эту информацию подтверждает ДИТ, итого система была недоступна со слов ДИТ 2:52:22, что составляет 24% от общего времени проведение голосования.

По нашей оценке, пользовательский интерфейс не был доступен в интервалах: 1, 2, 3, 4, максимально это составляет 4:36:42 или 38% от общего времени проведение голосования.

Общее время сбоев записи данных в блокчейн, по нашей оценке, составило 4:15:14 около 35% от общего времени проведение голосования.

Суммарное время всех сбоев составило 4:51:42 или 41% от общего времени проведение голосования.

Есть подтверждённая информация от сотрудников ДИТ об их вмешательстве в систему голосования в 2 точках:

1. отключение крипто-про в 11:20 и переход на другое оборудование, наименование оборудования не указывается

2. Обработка списка голосующий на сервере с 14 до 17-00, удаление с сервера данных о более чем 944 бюллетенях, и повторная рассылка СМС пользователям с просьбой проголосовать (см ПРИЛОЖЕНИЕ 2. Расшифровка аудиозаписи)

Систему скорее можно признать нерабочей, чем рабочей

Примерно треть времени она не работала, треть времени блокчейн работал в аномальном режиме, а в тот недолгий период времени утром, пока система работала – были аномально завышенные результаты кандидатов от партии власти.

На стороне бэкэнда проводились никем не контролируемые операции как минимум с ~39% избирателей от числа проголосовавших в 30-ом округе.

Анализ рисков

1. Методология

Каждый из рисков, который возникает в ходе проведения ЭГ можно ранжировать в матрице рисков влияния на систему.

Матрица риска - это таблица или диаграмма, которая отображает значимость события на одной оси, и вероятность его возникновения на другой. Промежуточным итогом этой работы должен стать план реагирования. [43]

Риск – это вероятностное внешнее или внутреннее событие, влияющее на достижение целей.

Риски оценивают через две характеристики: Величина возможного ущерба; Вероятность реализации риска (возникновения события); Таким образом, риск - это комбинация вероятности и последствий. Соответственно матрица отражает различные варианты этой комбинации.

Степень влияния **Вероятность**	**Катастрофично (1)**	**Критично (2)**	**Важно (3)**	**Пренебрегаемо (4)**
Часто (A)	Высокий	Высокий	Серьёзный	Средний
Вероятно (B)	Высокий	Высокий	Серьезный	Средний

[43] Источник: https://www.fd.ru/articles/159502-matritsa-riskov-kak-sozdat

Единично (C)	Высокий	Серьезный	Средний	Низкий
Маловероятно (D)	Серьезный	Средний	Средний	Низкий
Крайне маловероятно (E)	Средний	Средний	Средний	Низкий
Устранено (F)	Устранено			

Для унификации оценки рисков в ходе отчета мы приняли следующие правила:

– если какой-то сбой произошел во время голосования, то вероятность его возникновения становится A;

– если риск не произошел и не возникал в ходе текущего голосования, но имелись прецеденты его возникновения, то вероятность возникновения минимум E;

– Если приняты меры предосторожности, которые подтвердили свою устойчивость к атакам похожего типа, то его вероятность F;

– Критичность;

– Если риск затрагивает более 50% системы то его критичность 1;

– Если риск затрагивает 15-49% системы то его критичность 2;

– Если риск затрагивает 3-15 % системы то его критичность 3;

- Если риск затрагивает менее 3% системы то его критичность 4;
- При оценке каждого риска определяем следующие показатели:
    - возникал ли он в ходе голосования. Если да, то нужны ссылки на каждый кейс;
    - сколько времени ушло на его устранения;
    - сколько пользователей пострадало от риска (с пруфами);
    - если риск не возникал в ходе голосования, имел ли он место вообще;
    - как решается эта проблема (ссылки на решения);
    - как она была решена в ЭГ (если не решена, то ссылка на пруф);
    - какие компоненты системы затрагивает этот кейс;
    - ила влияния - как долго в %, сколько пользователей в % (если нет предупреждающих атаку действий в ЭГ).

2. Распределение рисков для электронного голосования

Риски стоит поделить на группы:

1. Общие риски технологии блокчейн.

    - риски криптографии;
    - атака 51 - перезапись блоков;
    - попытки атаки 51.

2. Риски возникновения аппаратных проблем.

    - Отказ крипто-про;

- Отказ оборудования для аудита ЭГ (КОИБы);
- Отказ резервного контура;
- Обрыв связи.

3. Риски вмешательства 3х лиц – программные.

- Человеческий фактор – ошибки;
- Вмешательство администратора портала - насколько ограничено влияние администратора портала и ограничено ли оно вообще? чем обеспечивается недопустимость влияния 1 лица на результат и процесс ЭГ;
- Получение доступа третьими лицами к управлению проектом (есть ли защита и какая она).

4. Риски вмешательства 3х лиц не программные.

- Использование административного ресурса и социальной инженерии для получения нужного результата (описать примеры таких рисков).

5. Ошибки и бекдоры в коде продукта.

- Сбои работы пользователя с формами;
- Отказ работы сторонних сервисов;
- Программные нарушения в передаче данных;
- Потери общественного контроля над ходом голосования;
- Сбои в работе анонимайзера;
- Ошибки в работе смарт-контрактов.

2. Классификация рисков ЭГ

2.1. Ошибки и бекдоры в коде смарт-контарктов

2.1.1. Оставлена возможность голосовать одному пользователю в разных округах и информация об этом не учитывается в СК

Один пользователь может проголосовать в 3 округах, и смарт контракт запишет эту информацию в реестр избирателей и в реестр бюллетеней.

В зоне риска ⅔ оказывается ⅔ всех записанных бюллетеней.[44]

Время возникновения в день голосования	Не установлено
Дата возникновения риска в прошлом	
Доля избирателей/бюллетени, которую риск затрагивает	67 % каждый, максимально ⅔ голосов могут быть
Доля времени в день голосования, которое может затрагивает проблема	
Риск	ВЫСОКИЙ

[44] 6.2. Контракт VotersRegistry не привязывает избирателей к округам

6.5. Контракт BallotsRegistry позволяет одному избирателя проголосовать в нескольких округах

6.9. Контракты VotersRegistry и BallotsRegistry не взаимодействуют между собой

2.1.2. Риск потери данных из-за ввода неверного закрытого ключа

Нередко возникает проблема ошибки при вводе неверного закрытого ключа, умышленно или нет.[45]

Время возникновения в день голосования	Не установлено
Дата возникновения риска в прошлом	
Доля избирателей/бюллетненй, которую риск затрагивает	100% max
Доля времени в день голосования, которое может затрагивает проблема	
Риск	Серьезный

2.1.3. Риск получения бюллетеня третьим лицом

При проведении традиционного голосования возникают ситуации когда третье лицо пытается проголосовать, в данном случае сотрудник избирательной комиссии сверяет паспортные данные, и может отказать в выдаче бюллетеня пользователю.

В случае с электронным голосованием паспортные данные не проверяются на стороне смарт-контракта.[46]

[45] 6.7. Контракт BallotsRegistry позволяет администратору опубликовать закрытый ключ, не соответствующий ранее опубликованному открытому ключу

[46] 6.1. Контракт VotersRegistry не привязывает номера избирателей к реальным людям

Это проблему легко можно было бы исправить, сохраняя вместе с номером избирателя хэш от его персональных данных, например от серии и номера паспорта, дополненный секретной солью.

Оставленная такая возможность для фальсификации. ставит под сомнение прозрачность избирательного процесса, и мы оцениваем такой риск как высокий

Факт регистрации пользователя на электронное голосование и наличие у него полной записи на портале mos.ru зависит от действий администрации портала подконтрольного мерии.

В смарт-контрактах нет проверки факта, что может зарегистрироваться только пользователь. имеющий полную учетную запись на портале mos.ru. данная проверка остается на стороне mos.ru

100% заполненная запись означает сданную биометрию.

Примеры:

30 округ. участок 2061. в списках для электронного голосования была зарегистрирована женщина 1920х рождения

На УИК 2095 был избиратель, который записался на электронное голосование, а до этого умер. А когда его решили вычёркивать оказалось, что его нет вообще в основном списке избирателей

Время возникновения в день голосования	Выявлено 2 случая в 30 округе *
Дата возникновения риска в прошлом	-
Доля избирателей/бюллетеней, которую риск затрагивает	⅔

Доля времени в день голосования, которое может затрагивает проблема	
Риск	Высокий

2.1.4. Оставлена возможность проголосовать до начала времени голосования и после окончания голосования.

Проверка времени голосования, оставлена на стороне бэкенда на стороне ДИТ. для начала и завершения голосования требуется команда администратора смарт-контрактов. [47]

Я оценила риск как серьёзный, так как остаётся возможность посмотреть итоги голосования после расшифровки блоков.

Время возникновения в день голосования	Не выявлено
Дата возникновения риска в прошлом	Стандартный риск влияния человеческого фактора
Доля избирателей/бюллетеней, которую риск затрагивает	
Доля времени в день голосования, которое может затрагивает проблема	

[47] 6.3. Контракт BallotsRegistry позволяет избирателям голосовать до начала и после окончания голосования

Риск	Серьёзный

2.1.5. Оставлена возможность для администратора завершить голосование до окончания установленного законом времени голосования.

Проверка времени голосования, оставлена на стороне бэкенда на стороне ДИТ. Для начала и завершения голосования требуется команда администратора смарт-контрактов.[48]

Я оценила риск как высокий , так как это влияет на возможность свободного волеизъявления граждан и автоматизированная защита от этого риска отсутствует. Наблюдатели, в тм числе международные, не имели в день голосования доступа к блокчейну и не могли проконтролировать или повлять на время голосования. все то время свободное волеизъявление избирателей было невозможно.

Время возникновения в день голосования	2 раза голосование останавливали администраторы
Дата возникновения риска в прошлом	Стандартный риск влияния человеческого фактора
Доля избирателей/бюллетненй, которую риск затрагивает	
Доля времени в день голосования, которое может затрагивает проблема	
Риск	Высокий

[48] 6.6. Контракт BallotsRegistry позволяет администратору завершить голосование до того, как оно должно быть завершено по закону

2.1.6. Смарт-контракты не учитывают таймаут для заполнения бюллетеня

Согласно правилам электронного голосования. опубликованным на портале mos.ru, пользователь может заполнить бюллетень в течении 15 минут с момента. его получение, однако смарт контракт BallotsRegistry, принимающий и сохраняющий зашифрованные голоса избирателей, Может принимать бюллетени даже если с момента их выдачи прошло более 15 минут.[49]

По заявлению представителя ДИТ Сарватдинов Александр Евгеньевич: «Сейчас ребятами проводится такая работа - они смотрят тех пользователей, которые переходили по кнопке бюллетень, но не попадали на него. И им возвращается статус, что они до сих пор не проголосовали. То есть им сбрасывают сообщение, что вы можете получить бюллетень. Они смогут заново проголосовать. Два часа назад это 544 человека так были восстановлены. Полчаса назад порядка 400, 544 два часа назад, и около 400 полчаса назад.»

Им всем посылается сообщение о том, что они могут проголосовать электронным голосованием. То есть, когда произошёл сбой через им всем было отослано сообщение. То есть это где-то ...более 900”

Время возникновения в день голосования	944 голосас 14 до 17 -00 на УИК 5003
Дата возникновения риска в прошлом	
Доля избирателей/бюллетненй, которую риск затрагивает	Разделим 944 бюллетеня на общее число выданных бюллетеней в конце для голосования 2376 = 39,7%

[49] 6.4. Контракт BallotsRegistry позволяет избирателям проголосовать позднее, чем через 15 минут после получения бюллетеня

Доля времени в день голосования, которое может затрагивает проблема	
Риск	Высокий

Так как подобное вмешательство администраторов системы электронного голосования затронуло более 39% бюллетеней и нет автоматической системы контроля от подобных вмешательств смарт-контрактах. мы оцениваем данный риск как высокий.

1.7. Смарт-контракты оставляют возможность для подмены голосов [50]

Правильность расшифровки нигде не проверяется в коде смарт-контракта, то есть смарт-контракт позволяет администратору опубликовать под видом расшифрованного голоса всё, что угодно, исказив таким образом волеизъявление избирателя.

Ситуация усугубляется тем, что для каждого конкретного бюллетеня, смарт контракт позволяет опубликовать расшифрованный голос не более одного раза. Если администратор по ошибке опубликует неправильную расшифровку, то исправить ошибку будет невозможно.

Мы оцениваем риск как высокий, так как они может затронуть 100% голосов и блокчейн в данном проекте не обладает инструментами контроля и предотвращения подмены голосов.

[50] 6.8. Контракт BallotsRegistry позволяет администратору опубликовать расшифрованный голос, не соответствующий ранее опубликованному зашифрованному голосу

Время возникновения в день голосования	Не установлено
Дата возникновения риска в прошлом	Хардфорки в блокчейнах Etherium последний 13.09.2019
Доля избирателей/бюллетеней, которую риск затрагивает	100%
Доля времени в день голосования, которое может затрагивает проблема	
Риск	Высокий

2.2. Общие риски технологии блокчейн

2.2.1. Риск расшифровки подписи

В системе электронного голосования используется ключ 1024 бит. Это достаточно большой ключ, для обеспечения безопасности данных, но недостаточно хороший для использования в системах электронного голосования.

«Определить, какая длина ключа подходит для данного уровня безопасности, нелегко. Тем не менее консенсус в том, что ключи должны быть длиннее 1024 бит. Считается, что 1024-битные ключи можно взломать с помощью современных технологий и знаний, хотя такие публичные вычисления ещё не проводились. Текущий публичный рекорд — взлом ключа длиной 768 бит.» - криптограф из Франции Пьеррик Годри.[51]

[51] 2.3. Шифрование 1024 бит

Время возникновения в день голосования	Не возникало
Дата возникновения риска в прошлом	Рекорд по расшифровыванию ключа на сегодня (08.10.2019) 768 бит
Доля избирателей/бюллетеней, которую риск затрагивает	100%
Доля времени в день голосования, которое может затрагивает проблема	100%
Риск	Низкий

2.2.2. Атака 51 - перезапись блоков, замена информации в блокчейне

Традиционно риск Атаки 51 в децентрализованных блокчейн системах снижается вместе с ростом сети, чем шире сеть, тем быстрее снижается риск атаки 51. при прочих равных условиях.

В проекте Электронного голосования блокчейн находился на одном сервере, полностью подконтрольному узкому кругу лиц, возможно. заинтересованных в изменении блоков голосования и подмене данных. Помимо этого в смарт-контрактах отсутствовал инструмент проверки соответствия между приватным и публичными ключами, что многократно повышает риск подмены результатов голосования.[52]

[52] 6.7. Контракт BallotsRegistry позволяет администратору опубликовать закрытый ключ, не соответствующий ранее опубликованному открытому ключу

6.8. Контракт BallotsRegistry позволяет администратору опубликовать расшифрованный голос, не соответствующий ранее опубликованному зашифрованному голосу

Защита от подмены и атаки на уровне смарт-контрактов не предусмотрена.

Время возникновения в день голосования	Не установлено
Дата возникновения риска в прошлом	Хардфорки в блокчейнах etherium? последний 13.09.2019
Доля избирателей/бюллетненй, которую риск затрагивает	100%
Доля времени в день голосования, которое может затрагивает проблема	
Риск	Высокий

2.3. Бэкдоры в PHP коде

2.3.1 Форма бюллетеня для пользователя не связана с таймером жизни бюллетеня на сервере

Это может привести к тому, что «госа» пользователей не будут записаны. при этом пользователь не имеет возможности проверить записан ли его голос.

Время возникновения в день голосования	На протяжении всего дня голосования
Дата возникновения риска в прошлом	--

Доля избирателей/бюллетеней, которую риск затрагивает	100%
Доля времени в день голосования, которое может затрагивает проблема	100%
Риск	Высокий

2.3.2. Администратор может инициировать повторное голосование избирателей[53]

Время возникновения в день голосования	14:00 - 16:00
Дата возникновения риска в прошлом	--
Доля избирателей/бюллетеней, которую риск затрагивает	Разделим 944 бюллетеня на общее число выданных бюллетеней в конце для голосования 2376 = 39,7%
Доля времени в день голосования, которое может затрагивает проблема	Более 2 часов, с 14 до 16 часов приблизительно
Риск	Высокий

[53] 4.1. Смарт-контакты не проверяют код из СМС и авторизацию пользователя.

2.3.3. Возможность записывать результаты голосования избирателем независимо от его действий в форме бюллетеня [54]

Время возникновения в день голосования	Не установлено точно
Дата возникновения риска в прошлом	--
Доля избирателей/бюллетненй, которую риск затрагивает	100%
Доля времени в день голосования, которое может затрагивает проблема	100 %
Риск	Высокий

2.4. Итоговая матрица рисков

Высокий	1.1. Оставлена возможность голосовать одному пользователю в разных округах и информация об этом не учитывается в СК; 1.3. Риск получения бюллетеня третьим лицом; 1.5. Оставлена возможность для администратора завершить голосование до окончания установленного законом времени голосования; 1.6. Смарт-контракты не учитывают таймаут для заполнения бюллетеня;

[54] 4.6. В код бюллетеня добавлена строка, позволяющая трактовать голос избирателя несколькими способами

	1.7. Смарт-контракты оставляют возможность для подмены голосов; 2.2. Атака 51 - перезапись блоков, замена информации в блокчейне; 3.1 форма бюллетеня для пользователя не связана с таймером жизни бюллетеня на сервере; 3.2. Администратор может инициировать повторное голосование избирателей; 3.3. Возможность записывать результаты голосования избирателем независимо от его действий в форме бюллетеня.
Серьезный	1.2. Риск потери данных из-за ввода неверного закрытого ключа; 1.4. Оставлена возможность проголосовать до начала времени голосования и после окончания голосования.
Средний	
Низкий	2.1. Риск расшифровки подписи.
Устранено	

Выводы

В ходе исследования мы проанализировали

1. Схему взаимодействия компонентов системы электронного голосования;
2. Код смарт-контрактов;
3. php код форм бюллетеня;
4. Анонимайзер;
5. Провели аудит итоговых данных записанных в блокчейн.

И можем сделать следующие выводы:

1. Даже тот упрощённый подход, который использовался в Москве, позволяет возложить на смарт-контракты контроль за соблюдением многих важных правил проведения выборов, таких как время начала и окончания голосования, невозможность для избирателя голосовать в чужом округе, время получения электронного бюллетеня, в течение которого избиратель имеет возможность проголосовать и т.п. таким образом, контроль за соблюдением этих правил можно было бы сделать прозрачным и доступным для независимых наблюдателей. Однако, этого сделано не было. Контроль за соблюдением большинства правил проведения выборов отдан на откуп программному обеспечению, функционирующему вне блокчейна и недоступному для независимого контроля. **Фактически, единственная функция, отданная смарт-контрактам — это подсчёт голосов.**

2. В процессе проведения московского электронного голосования, у независимых наблюдателей доступа к блокчейну не было, а значит организаторы в принципе имели возможность в любой момент подменить весь блокчейн целиком, заменить несколько последних блоков, или запустить несколько блокчейнов параллельно с разными результатами голосования. Все эти манипуляции остались бы незамеченными.

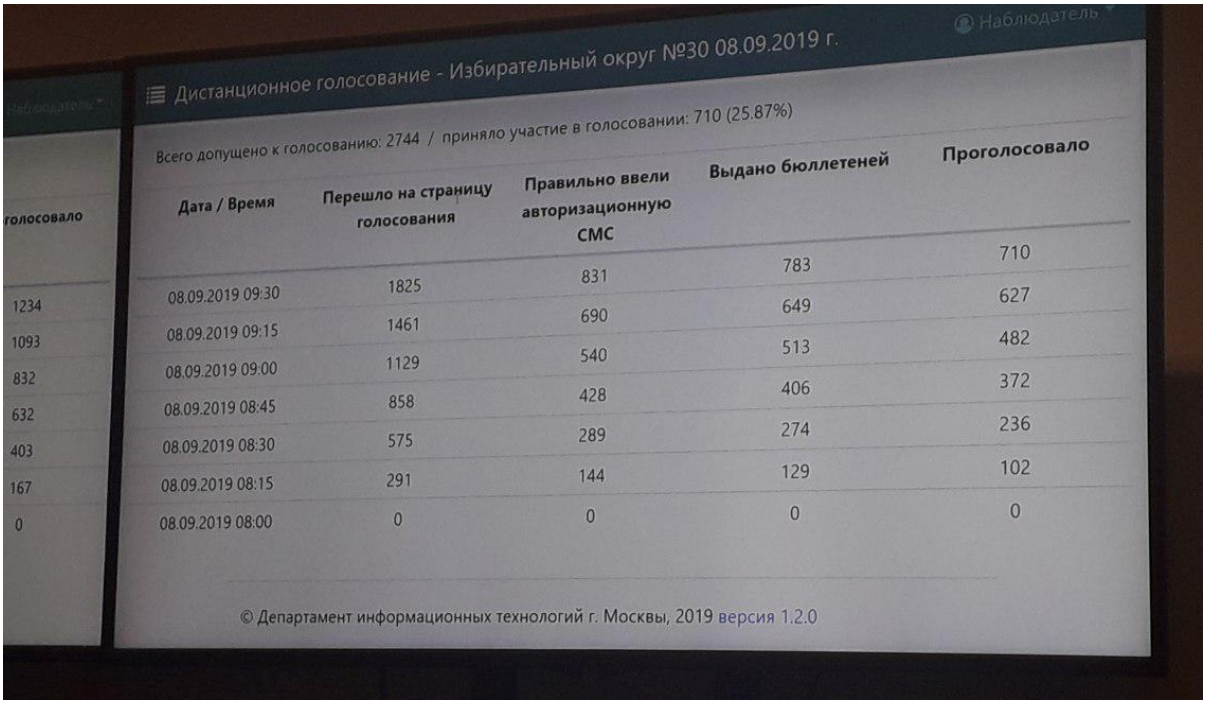
3. **Администратор ДЭГ мог произвольно, без перезапуска системы очистить реестр с кодами СМС и инициировать повторную рассылку СМС сообщений. Что является влиянием на ход голосования и так как можно сопоставить время формирования бюллетеня и контактную информацию избирателя - номер телефона, привязанного к учётной записи MOS.RU, что нарушает тайну голосования и предоставляет возможности очищать реестры без перезагрузки системы для администратора**

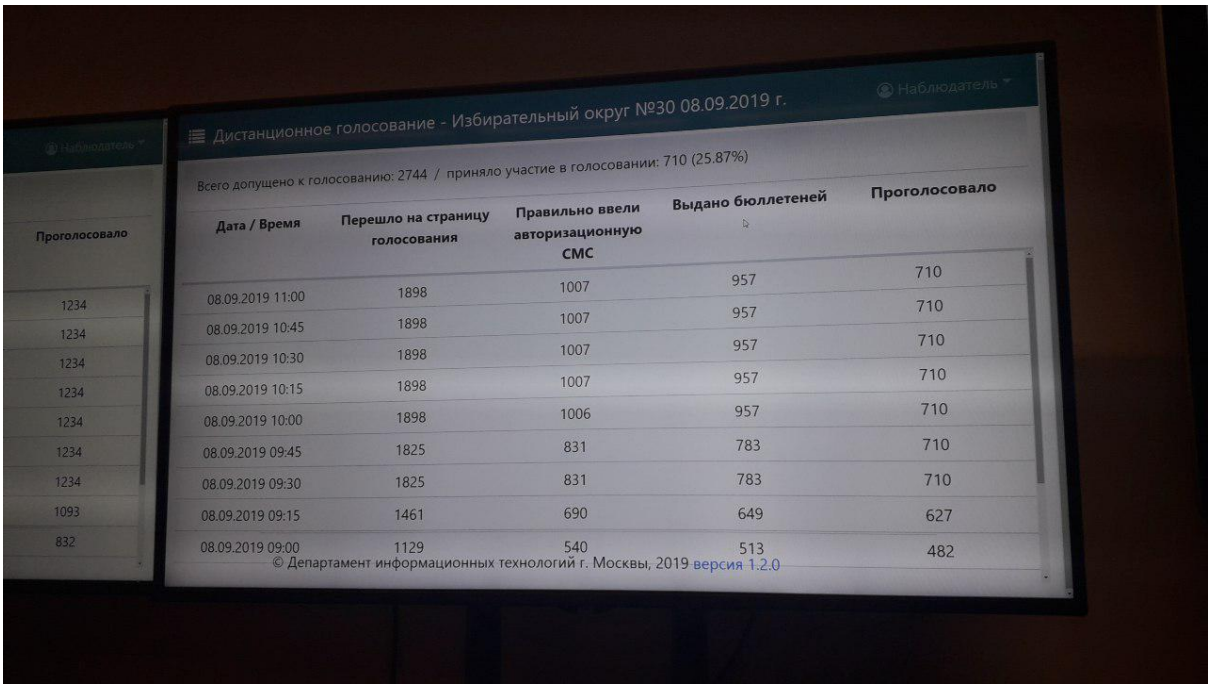
4. По нашей оценке, пользовательский интерфейс не был доступен в интервалах: 1, 2, 3, 4, максимально это составляет 4:36:42 (четыре часа 36 минут 42 секунды) или 38% от общего времени проведение голосования. Общее время сбоев записи данных в блокчейн, по нашей оценке, составило 4:15:14 около 35% от общего времени проведение голосования. Суммарное время всех сбоев составило 4:51:42 или 41% от общего времени проведение голосования.

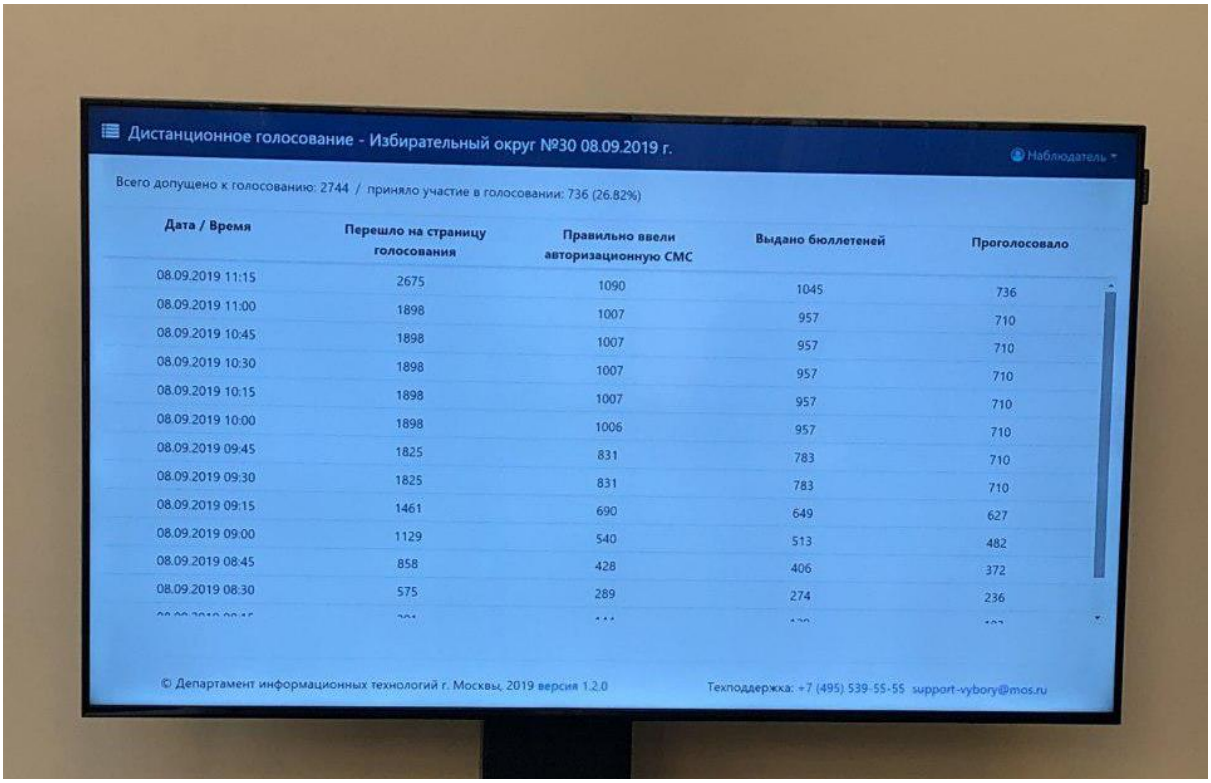
5. **На стороне бэкэнда проводились никем не контролируемые операции как минимум с ~39% избирателей от числа проголосовавших в 30-ом округе.**

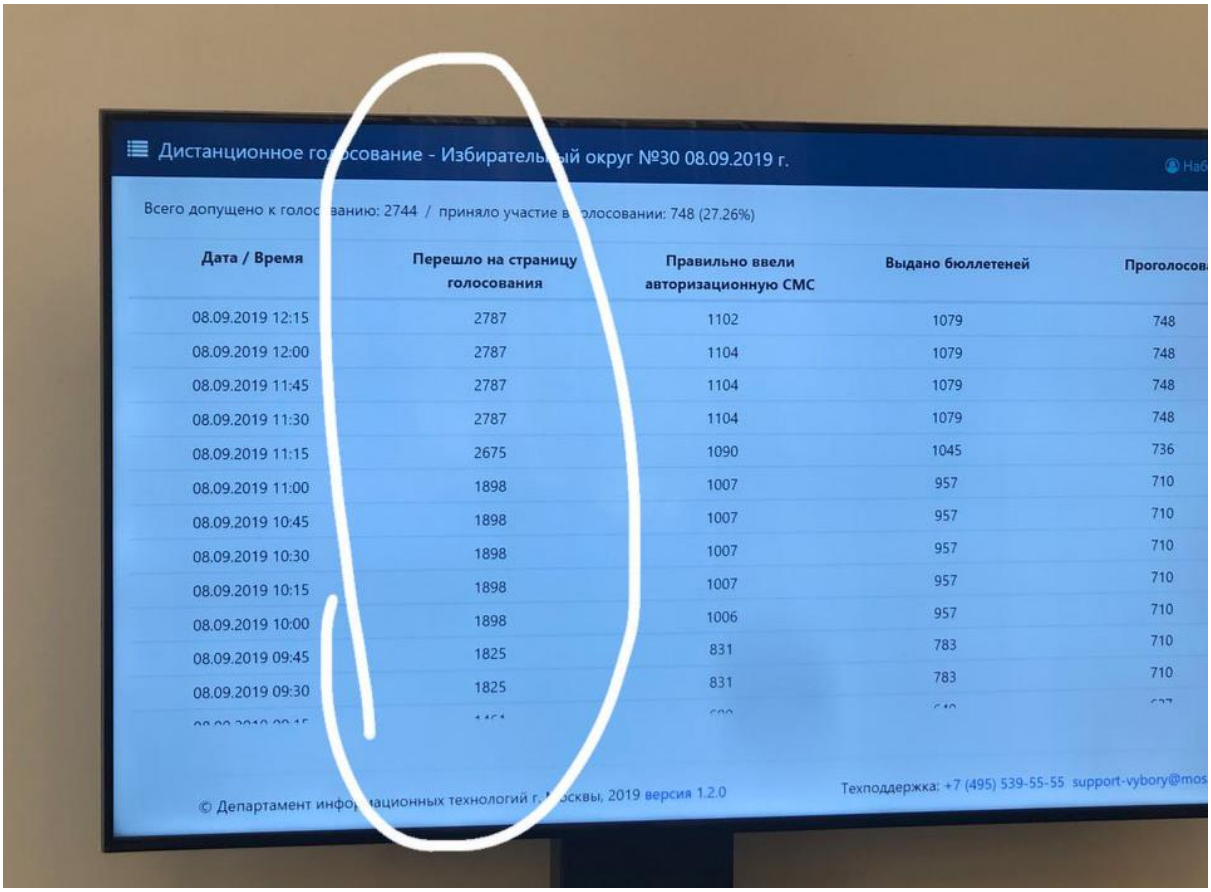
Обобщая всё вышесказанное, можно сказать, что использование блокчейна и смарт-контрактов на электронном голосовании по выборам депутатов в Московскую Городскую Думу носило скорее бутафорский характер и имело своей целью создать у общественности впечатление, что используемые технологии гарантируют соблюдение правил и честный подсчёт голосов. При этом на самом деле организаторы имели возможность вмешиваться в процесс голосования (что подтверждено фактами) и искажать, таким образом, волю избирателей, а возможности для независимого контроля в случае электронного голосования были даже ниже, чем при других, более традиционных способах организации избирательного процесса.

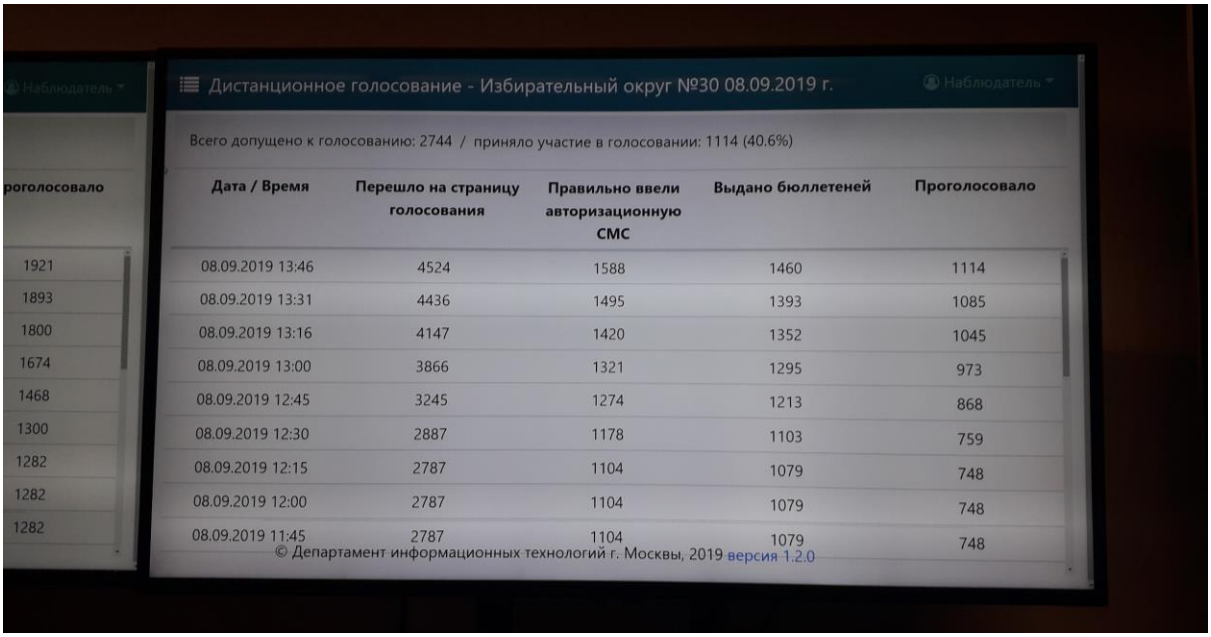
ПРИЛОЖЕНИЕ 1. Фотографии






Дистанционное голосование - Избирательный округ №30 08.09.2019 г.
Наблюдатель
Всего допущено к голосованию: 2744 / приняло участие в голосовании: 736 (26.82%)
Дата / Время
Перешло на страницу голосования
Правильно ввели авторизационную СМС
Выдано бюллетеней
Проголосовало
08.09.2019 11:15 2675 1090 1045 736
08.09.2019 11:00 1898 1007 957 710
08.09.2019 10:45 1898 1007 957 710
08.09.2019 10:30 1898 1007 957 710
08.09.2019 10:15 1898 1007 957 710
08.09.2019 10:00 1898 1006 957 710
08.09.2019 09:45 1825 831 783 710
08.09.2019 09:30 1825 831 783 710
08.09.2019 09:15 1461 690 649 627
08.09.2019 09:00 1129 540 513 482
08.09.2019 08:45 858 428 406 372
08.09.2019 08:30 575 289 274 236
© Департамент информационных технологий г. Москвы, 2019 версия 1.2.0
Техподдержка: +7 (495) 539-55-55 support-vybory@mos.ru


Дистанционное голосование - Избирательный округ №30 08.09.2019 г.
Всего допущено к голосованию: 2744 / приняло участие в голосовании: 748 (27.26%)
Дата / Время
Перешло на страницу голосования
Правильно ввели авторизационную СМС
Выдано бюллетеней
08.09.2019 12:15 2787 1102 1079 748
08.09.2019 12:00 2787 1104 1079 748
08.09.2019 11:45 2787 1104 1079 748
08.09.2019 11:30 2787 1104 1079 748
08.09.2019 11:15 2675 1090 1045 736
08.09.2019 11:00 1898 1007 957 710
08.09.2019 10:45 1898 1007 957 710
08.09.2019 10:30 1898 1007 957 710
08.09.2019 10:15 1898 1007 957 710
08.09.2019 10:00 1898 1006 957 710
08.09.2019 09:45 1825 831 783 710
08.09.2019 09:30 1825 831 783 710
Техподдержка: +7 (495) 539-55-55



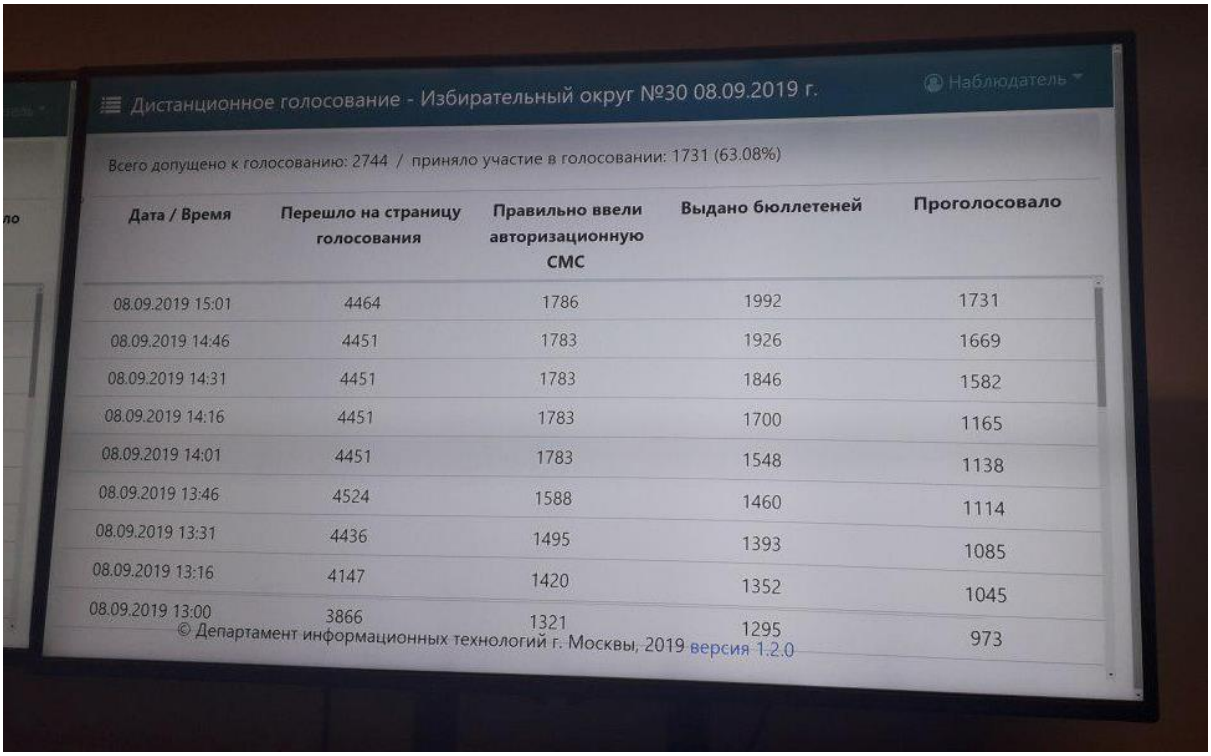
Дистанционное голосование - Избирательный округ №30 08.09.2019 г.

Наблюдатель

Всего допущено к голосованию: 2744 / приняло участие в голосовании: 1114 (40.6%)

Дата / Время	Перешло на страницу голосования	Правильно ввели авторизационную СМС	Выдано бюллетеней	Проголосовало
08.09.2019 13:46	4524	1588	1460	1114
08.09.2019 13:31	4436	1495	1393	1085
08.09.2019 13:16	4147	1420	1352	1045
08.09.2019 13:00	3866	1321	1295	973
08.09.2019 12:45	3245	1274	1213	868
08.09.2019 12:30	2887	1178	1103	759
08.09.2019 12:15	2787	1104	1079	748
08.09.2019 12:00	2787	1104	1079	748
08.09.2019 11:45	2787	1104	1079	748

© Департамент информационных технологий г. Москвы, 2019 версия 1.2.0



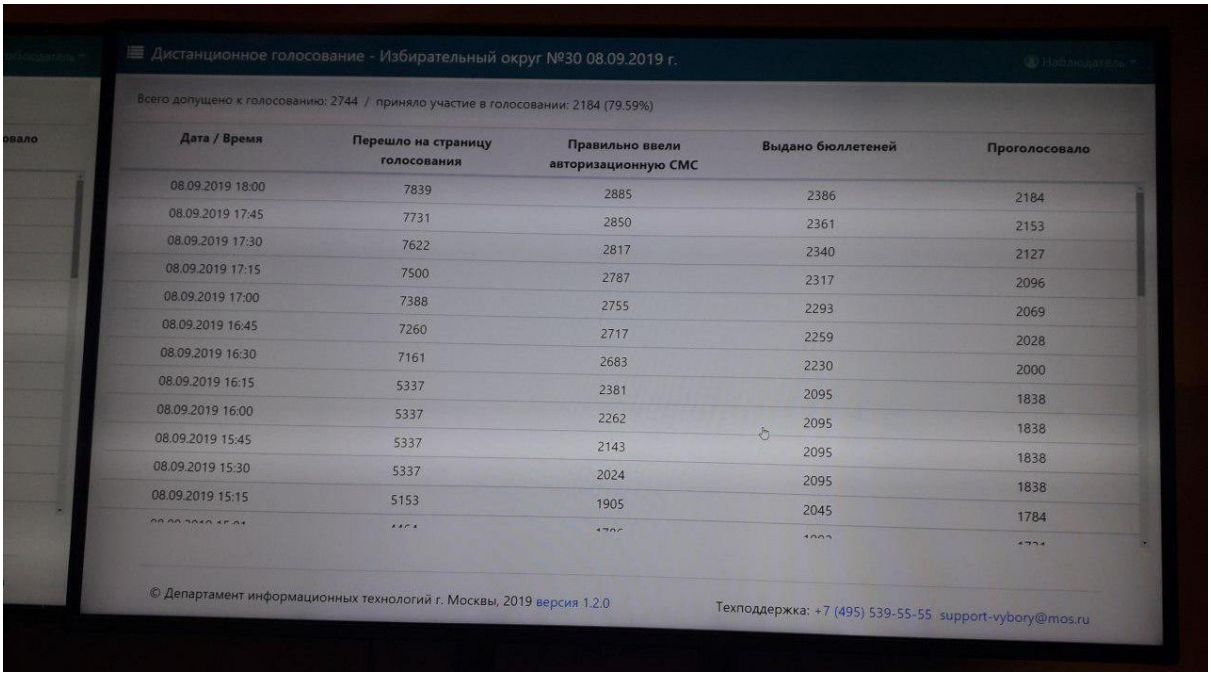
Дистанционное голосование - Избирательный округ №30 08.09.2019 г.

Наблюдатель

Всего допущено к голосованию: 2744 / приняло участие в голосовании: 1731 (63.08%)

Дата / Время	Перешло на страницу голосования	Правильно ввели авторизационную СМС	Выдано бюллетеней	Проголосовало
08.09.2019 15:01	4464	1786	1992	1731
08.09.2019 14:46	4451	1783	1926	1669
08.09.2019 14:31	4451	1783	1846	1582
08.09.2019 14:16	4451	1783	1700	1165
08.09.2019 14:01	4451	1783	1548	1138
08.09.2019 13:46	4524	1588	1460	1114
08.09.2019 13:31	4436	1495	1393	1085
08.09.2019 13:16	4147	1420	1352	1045
08.09.2019 13:00	3866	1321	1295	973

© Департамент информационных технологий г. Москвы, 2019 версия 1.2.0



Дистанционное голосование - Избирательный округ №30 08.09.2019 г.

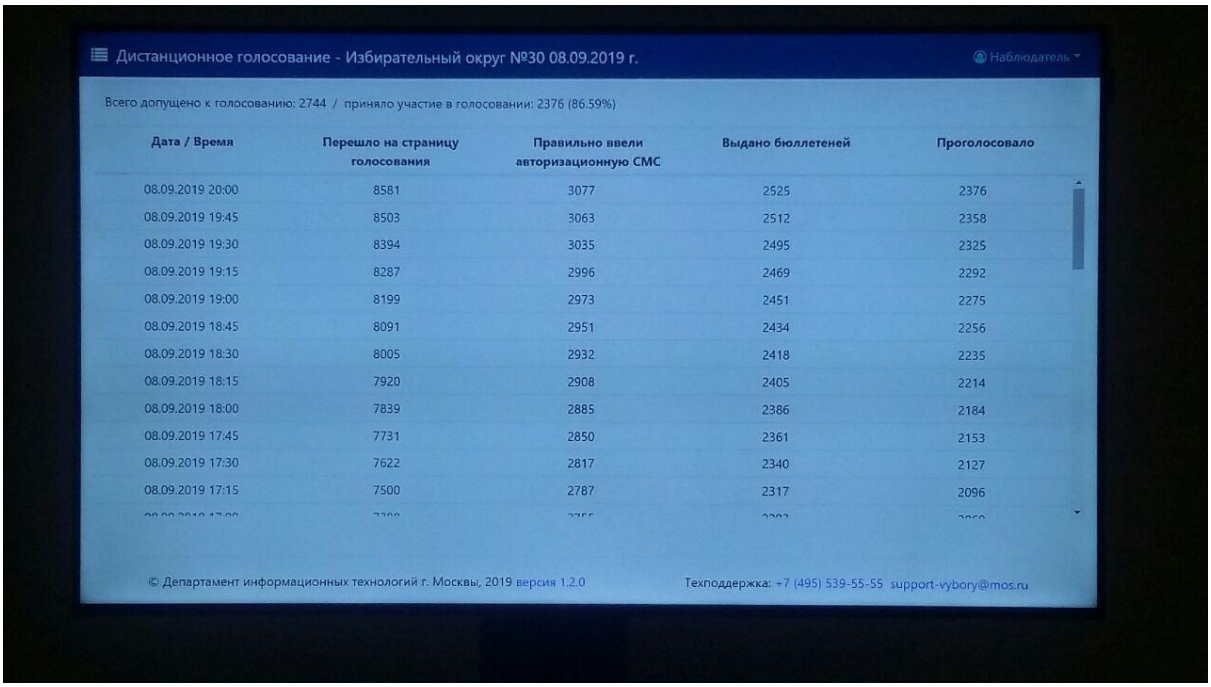
Наблюдатель

Всего допущено к голосованию: 2744 / приняло участие в голосовании: 2184 (79.59%)

Дата / Время	Перешло на страницу голосования	Правильно ввели авторизационную СМС	Выдано бюллетеней	Проголосовало
08.09.2019 18:00	7839	2885	2386	2184
08.09.2019 17:45	7731	2850	2361	2153
08.09.2019 17:30	7622	2817	2340	2127
08.09.2019 17:15	7500	2787	2317	2096
08.09.2019 17:00	7388	2755	2293	2069
08.09.2019 16:45	7260	2717	2259	2028
08.09.2019 16:30	7161	2683	2230	2000
08.09.2019 16:15	5337	2381	2095	1838
08.09.2019 16:00	5337	2262	2095	1838
08.09.2019 15:45	5337	2143	2095	1838
08.09.2019 15:30	5337	2024	2095	1838
08.09.2019 15:15	5153	1905	2045	1784

© Департамент информационных технологий г. Москвы, 2019 версия 1.2.0

Техподдержка: +7 (495) 539-55-55 support-vybory@mos.ru



Дистанционное голосование - Избирательный округ №30 08.09.2019 г.

Наблюдатель

Всего допущено к голосованию: 2744 / приняло участие в голосовании: 2376 (86.59%)

Дата / Время	Перешло на страницу голосования	Правильно ввели авторизационную СМС	Выдано бюллетеней	Проголосовало
08.09.2019 20:00	8581	3077	2525	2376
08.09.2019 19:45	8503	3063	2512	2358
08.09.2019 19:30	8394	3035	2495	2325
08.09.2019 19:15	8287	2996	2469	2292
08.09.2019 19:00	8199	2973	2451	2275
08.09.2019 18:45	8091	2951	2434	2256
08.09.2019 18:30	8005	2932	2418	2235
08.09.2019 18:15	7920	2908	2405	2214
08.09.2019 18:00	7839	2885	2386	2184
08.09.2019 17:45	7731	2850	2361	2153
08.09.2019 17:30	7622	2817	2340	2127
08.09.2019 17:15	7500	2787	2317	2096

© Департамент информационных технологий г. Москвы, 2019 версия 1.2.0

Техподдержка: +7 (495) 539-55-55 support-vybory@mos.ru

ПРИЛОЖЕНИЕ 2. Расшифровка аудиозаписи

Сарватдинов Александр Евгеньевич

Опрашивал Член ТИК с ПГС - Антон Журбенко.

Время записи примерно: 17:15-17:20 08.09.2019.

Длительность 5:39

По кнопке перейти к голосованию, видимо из-за того что бюллетень не шифровался, вываливалась ошибка.

– А во сколько примерно...

– В 9.39.

– Как вас зовут

– Александр

– На данный момент те голоса, которые сейчас не учитываются, что сейчас с ними происходит?

– Смотрите

– Они подвисли?

– Они не подвисли, там заявитель, он шел оформить бюллетень. На стороне пгу, московский портал городских услуг, это терминология техническая. Была совершена транзакция о том что ему выдан бюллетень. И в базу он так и попал. Все. Соответственно он не может второй раз перейти на бюллетень, но бюллетень ему не был выдан. Сейчас ребятами проводится такая работа - они смотрят тех пользователей, которые переходили по кнопке бюллетень, но не попадали на него. И им возвращается статус, что они до сих пор не проголосовали. То есть им сбрасывают сообщение, что вы можете получить бюллетень. Они смогут заново проголосовать. Два часа назад это 544 человека так были восстановлены. Полчаса назад порядка 400.

– То есть это порядка 140 с лишним человек, они получили такое уведомление, перезашли и смогли проголосовать.

– 544 два часа назад, и около 400 полчаса назад

– То есть это около 900 человек...

– Им всем посылается сообщение о том, что они могут проголосовать электронным голосованием. То есть когда произошёл сбой через им всем было отослано сообщение.

То есть это где-то ...более 900 И постепенно голоса восстанавливаются **спрашивает про время** Сейчас 5 часов, сбоев нет. Последнее 9.30 — 11. Потом включали на час, смотрели, что происходит. Потом начались проблемы и на час ещё отключили. Потом включили и она работает до сих пор.

– В течение какого времени работает система?

– Сейчас посмотрим... Правильно ли я понял, что в настоящий момент продолжается рассылка уведомлений?

– Да. С 12.30. с 12.15 не было голосов, с 12.30 они пошли.

– Это чисто технический сбой на стороне пгу, проблемы с …..... переходом на форму.

– Угу

– А, голоса они не обновлённые просто.

– Сколько проголосовали на данный момент?

– Проголосовали 2096.