Информационные технологии

Концептуальная архитектура системы отслеживания контактов на основе блокчейн

И.А. Тарханов[I,II], И.А. Шмелев[III]

[I] Федеральное государственное бюджетное учреждение высшего образования «Государственный академический университет гуманитарных наук», г. Москва, Россия
[II] Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук», г. Москва, Россия
[III] Национальный Исследовательский Технологический Университет (НИТУ) «МИСиС», г. Москва, Россия

Аннотация. Статья посвящена разработке концепции системы, которая отслеживает потенциальные контакты между субъектами и уведомляет их в случае вероятности заражения COVID-19. На основе проведенного анализа похожих проектов предлагается и обосновывается выбор использования эксклюзивного блокчейн в качестве слоя для хранения и обмена данными. Детально описан алгоритм фиксации контакта, проведение риск-анализа заражения и распространения информации. Особое внимание уделено вопросу сохранения приватности субъектов. Описанная система может быть внедрена в крупных городах или даже в масштабах страны.

Ключевые слова: *социальные контакты, COVID-19, блокчейн, фиксация контакта.*
DOI: 10.14357/20790279210407

Введение

В эпоху пандемии становятся особо актуальны информационные технологии и построенные на них системы, которые направлены на борьбу с распространением тех или иных вирусов [1]. Отслеживание социальных контактов – важная часть этой проблемы. С одной стороны, такая система должна работать с большим объемом данных практически в режиме реального времени. С другой стороны, нужно обеспечить анонимность субъектов и безопасную обработку их персональных данных. Большинство изученных нами проектов основаны на личном подтверждении субъектом факта посещения того или иного места [3]. Проекты, которые подразумевают большую автоматизацию в вопросах подтверждения места посещения и факта заболевания активно используют Internet of Things (IOT) [6], Radio Frequency Identification (RFID) [1], Bluetooth Low Energy (BLE) [1,5], Distributed Ledger Technology [6] и др. Выделим основные функциональные подсистемы такого проекта:

- идентификация местоположения субъекта;
- хранилище информации о местоположении субъектов;
- подсистема определения социальных контактов между субъектами;
- подсистема публикации информации о заражении субъекта;
- уведомление субъектов о риске инфицирования;
- подсистема обеспечения безопасности персональных данных.

В качестве подсистемы хранения информации о местоположении субъектов был выбран эксклюзивный блокчейн, в котором обработка транзакций осуществляется определенным списком идентифицированных участников [10]. Использование блокчейн в качестве хранилища данных накладывает определенные ограничения на архи-

тектуру системы и основные ее элементы, такие как алгоритм определения социальных контактов, процесс распространения информации о заражении и проведение риск-анализа (определение вероятности заражения субъекта в конкретном месте). Рассмотрим эти вопросы подробнее.

1. Хранилище данных системы и ее общая архитектура

Для обеспечения неизменности и прозрачности статистических данных в качестве хранилища целесообразно использовать распределенный реестр (блокчейн, Distributed Ledger Technology, далее DLT) в том или ином виде. В работе [6] предложена концепция системы, использующей DLT, для построения саморегулирующейся открытой децентрализованной системы для отслеживания социальных контактов между людьми в целях контроля распространения заболеваний.

В роли хранилища данных для рассматриваемой системы может выступать эксклюзивный блокчейн [10], контролируемый консорциумом лиц, ответственными за платформу. У данного решения есть ряд преимуществ.

1. Эксклюзивный блокчейн проще поддерживать, вертикально масштабировать и в том числе существует возможность обеспечить SLA, что чрезвычайно важно для такого рода сервисов.
2. Неизменность данных, записанных в блокчейн, в сочетании с правильной балансировкой интересов сторон, поддерживающих сеть, позволят сделать прозрачную для всех пользователей и интересантов структуру, где можно строго идентифицировать субъектов, которые вносят информацию в блокчейн-сеть.
3. Контролируемая приватность данных – в зависимости от потребностей регуляторов, такой блокчейн можно сделать публичным для общества, а можно сделать приватным и предоставлять информацию через определенный интерфейс взаимодействия с сервисом.
4. Существуют прозрачные и верифицируемые правила для всех участников сети, по которым работает система.

Данное решение реализуемо на базе блокчейн-фреймворка, который поддерживает смарт-контракты, алгоритмы консенсуса DPoS (Delegated Proof-of-Stake) или PoA (Proof-of-Authority) и настройку прав для эксклюзивного блокчейна. Например, ими могут быть Substrat, Exonum, Cosmos SDK и др.

Каковы сценарии использования эксклюзивного блокчейн в системе отслеживания контактов в качестве хранилища данных? Для этого рассмотрим подробнее описанные в научной литературе решения.

В работе [9] описаны различные виды решений, построенные как на централизованной, так и на децентрализованной архитектуре. Классификация таких решений предложена в [1], где авторы разделили существующие технологические решения на 3 основные класса: централизованные, децентрализованные и гибридные.

Централизованные решения полагаются на центральный сервис, который инкапсулирует в себе процедуры по созданию идентификаторов устройств пользователей, отслеживанию контактов и уведомлению всех сторон о рисках заражения. Примером такой архитектуры являются системы, построенные на базе BlueTrace протокола [2]. Основной их проблемой является тот факт, что необходимо полагаться на 3-ю сторону, которая может быть умышленно или неумышленно скомпрометирована, что может поставить под угрозу безопасность персональных данных или корректность функционирования всей системы в целом. Данный аспект является критически важным для социально-значимых сервисов, коим и является рассматриваемая система. Примеры централизованных систем – BlueTrace [2], ROBERT [8]. Мобильные приложения, запущенные на основе централизованных алгоритмов – TraceTogether, CovidSafe (BlueTrace), StopCovid (ROBERT).

Децентрализованные решения также полагаются на центральный сервис, но только как на посредника для передачи вспомогательных данных, необходимых для риск-анализа, проводимого на самих устройствах. Примером может служить алгоритм PACT [7], который подразумевает, что все устройства обмениваются периодически генерируемыми кодами и сохраняют их для последующего анализа рисков инфицирования. Для генерации случайных кодов используются seed-данные, как основа для вычислений псевдослучайных функций. После того, как соответствующий государственный орган подтвердит, что участник был инфицирован, участник получает право загрузить seed-данные на центральный сервис, откуда остальные участники могут их получить и проверить риск заражения вирусом. Примеры децентрализованных систем - PACT (EAST-COAST), PACT (WEST-COAST), TCN, DP-T3, TP-T3 unlinkable, Pronto-C2.

Гибридные решения, как понятно из названия, – это сочетание централизованных и децентрализованных решений, где, например, создание идентификаторов пользовательских устройств

и обмен ими происходит с помощью клиентских приложений, а уведомления о заражении и анализ взаимодействия людей на предмет риска инфицирования производится центральным сервисом. Примеры гибридных систем: Desire [4,8], EpiOne, ConTra Corona. Примеры приложений, использующих децентрализованные алгоритмы: SwissCovid (DP-3T), CovidWatch (TCN), CovidSafe (PACT WEST-COAST) [1,3].

Очевидным плюсом гибридных решений является то, что они позволяют обеспечить анонимность пользователей (за счет того, что те сами управляют процессом раскрытия своих данных) и одновременно агрегировать информацию о контактах, заболевших на общем для всех участников сервисе. Такой подход предоставляет возможность организовать реестр больших данных, который в дальнейшем можно использовать для анализа поведения общества во время пандемий, нахождение очагов распространения заболевания, прогнозирование их появления, позволяя принимать точечные меры для борьбы с заболеванием, уменьшая ущерб, наносимый экономике.

Благодаря тому, что гибридная архитектура позволяет гибко совместить в себе разные аспекты централизованных и децентрализованных систем, именно ее целесообразно взять за основу для разработки концептуальной модели. Несмотря на то, что анонимность пользователей обеспечивается в гибридных архитектурах, на центральный сервис налагается очень высокая ответственность по сохранению целостности информации и защите от вмешательства в функционирование. Также полезные для общества данные находятся под контролем одной стороны и нет никаких гарантий, что они будут открыты для исследователей в исходном виде и не подвергнуты тем или иным манипуляциям.

Предложенная здесь концептуальная модель во многом базируется на [6], где рассматривается сеть с IOT-устройствами, использующими BLE-технологию. Эти устройства выступают в роли свидетелей физического присутствия пользователя в конкретной локации. Процесс состоит из пяти основных этапов (рис. 1):

1) регистрация социального контакта в определенной локации, в которой установлено IOT-устройство;
2) публикация информации этим устройством в DLT;

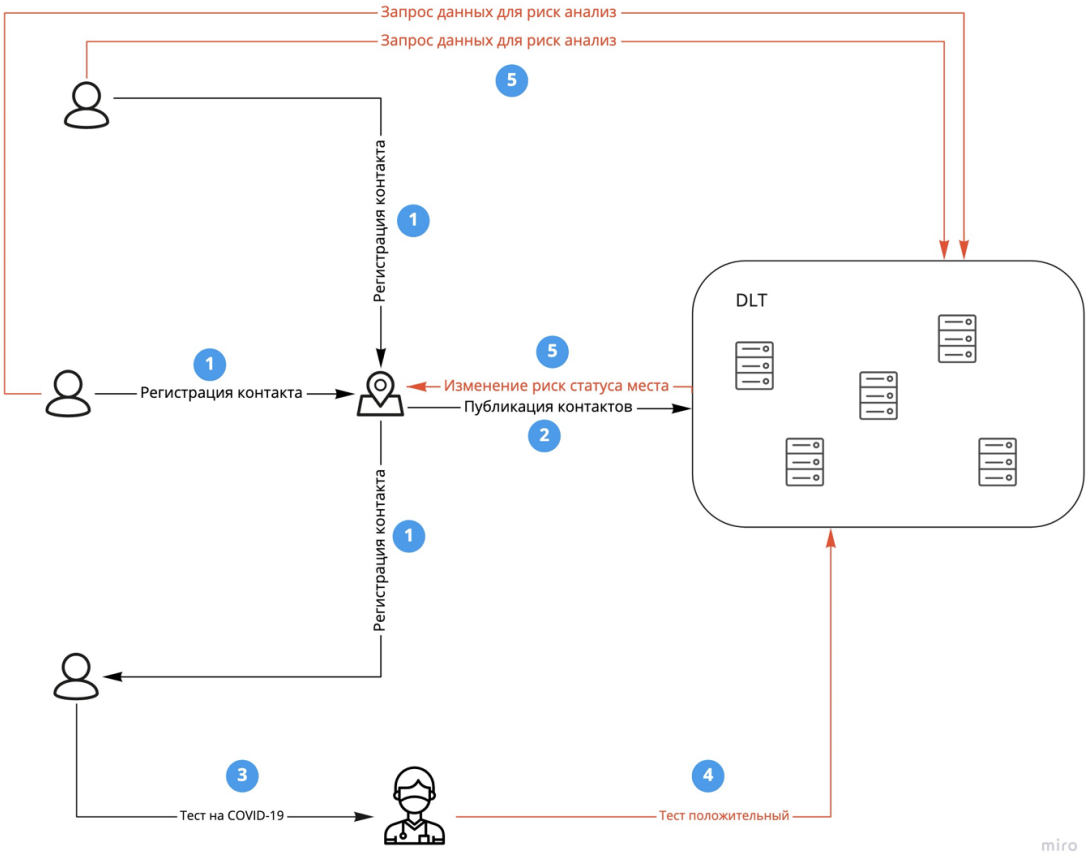

Рис. 1. Концептуальная схема системы отслеживания контактов

3. тестирование человека на COVID-19 и выявление факта инфицирования;
4. публикация информации об инфицировании в сеть;
5. уведомление людей, посещавших локацию и изменение риск-статуса данной локации.

В предложенной концептуальной схеме стоит обратить внимание на следующий ряд ключевых компонентов:

- алгоритм фиксации контакта;
- уведомления об инфицировании и анализ риска заражения;
- процесс публикации информации об инфицированном в сеть.

Компоненты из п.1 и п.2 выбраны на основе [1,9] и адаптированы под использование с выбранным хранилищем данных. Процесс из п.3 не является чем-то специфичным, но опирается на решения, принятые в предыдущих пунктах. Псевдокод смарт-контрактов, представляющих реализацию данной системы, представлен в Приложениях А, Б, В.

2. Идентификация местоположения и близости субъектов

Согласно обзору [9] инструментов идентификации местоположения субъекта, выделяют два основных аспекта – определение геолокации субъекта и близости между субъектами. Определение геолокации может выполняться с помощью:

- модуля глобальной навигационной системы (GPS, Глонасс и т.д.) [1,5];
- вышек сотовой связи [9];
- предварительно заданной информации о местоположении модуля фиксации: таких как BLE, RFID модулей, Wi-Fi или QR-кодов [2,9].

В то же время определение близости между субъектами производится с помощью BLE, RFID [3]. Данная информация может храниться как на удаленном сервере, так и в локальной памяти устройств, в зависимости от подхода к обеспечению безопасности персональных данных субъектов.

Далее мы рассматриваем только решения, использующие пользовательские устройства для идентификации контактов между людьми, т. к. они наиболее безопасны с точки зрения возможностей идентификации, контактирующих лиц.

3. Алгоритм фиксации контакта

Аналогично [6] в предлагаемой модели не регистрируем контакты между устройствами пользователей напрямую, чтобы было проще обеспечить необходимый уровень приватности пользователей. Вместо взаимодействия двух анонимных устройств, в данном подходе коммуницируют:

- публично идентифицированное устройство – IOT-устройство с BLE-технологией, принадлежащее конкретной организации;
- устройство пользователя, не идентифицируемое никак, чтобы сохранить анонимность.

Идентифицируемость IOT-устройства обеспечивается за счет асимметричной криптографии:

- публичный ключ организации доступен в открытом доступе;
- приватный ключ доступен только организации, с помощью него IOT устройство создает подпись при рассылке своих сообщений, чтобы принимающее устройство могло убедиться в доступе устройства к системе.

Обозначения:

PK_u, SK_u – публичный и приватный ключ пользователя соответственно;

PK_o, SK_o – публичный и приватный ключ организации соответственно;

O – IOT-устройство организации;

U – устройство пользователя;

t_u – дата публикации информации о контакте;

RandInt – случайное число;

Adv(advertiser address) – уникальный идентификатор IOT-устройства;

BLE – Bluetooth Low Energy.

Полный цикл регистрации состоит из следующих этапов:

1. O через BLE рассылает всем окружающим устройствам сообщение:
   a) $Sgn_{hello} = Sign(Hash(Adv, RandInt), SK_o)$;
   b) $MSG_{hello} = \{Adv, RandInt, Sgn_{hello}\}$.
2. U выполняет следующие действия:
   c) извлекает из DLT открытый ключ устройства $PK_o = FindBy(Adv)$;
   d) проверяет, что полученное сообщение принадлежит O $Validate(Sgn_{hello}, Hash(Adv, RandInt), PK_o)$.
3. U генерирует уникальную пару PK_u, SK_u и фиксирует текущий временной отрезок в виде значения T_u.
4. U создает шифрованное сообщение:
   e) $Sgn_u = Sign(Hash(PK_u, PK_o, T_u), SK_u)$;
   f) $MSG_u = Encrypt(\{PK_u, PK_o, T_u, Sgn_u\}, PK_o)$;
   U отправляет MSG_u с помощью BLE устройству O.
5. В свою очередь, O при получении:
   g) декодирует $MSG_{dec} = Decrypt(MSG_u, SK_o) = \{PK_u, PK_o, T_u, Sgn_u\}$;
   h) проверяет $Validate(Sgn_u, Hash(PK_u, PK_o), PK_u)$;
   i) проверяет, что T_u корректно.

6. если 5.b выполняется, то O сохраняет локально сообщение вида:
   $MSG_o = \{MSG_{dec}, Enc(MSG_{dec}, SK_o)\}$.
7. Далее в течение T секунд устройство U не пытается зарегистрировать факт контакта.
8. Каждый промежуток времени T устройство O публикует набор $\{MSG_o1, MSG_o2, MSG_o3, \ldots, MSG_oN\}$ в DLT.

4. Уведомления об инфицировании и риск-анализ

Нужно отметить, что публикация информации об инфицировании субъекта в большинстве случаев производится должностным лицом, имеющим на это право после положительного теста на наличие инфекции. После этого непосредственно мобильное устройство пользователя или центральный сервис системы проводит риск-анализ на основе собранных данных и уведомляет контактировавших.

Каждое устройство периодически запрашивает информацию по риск-статусу локаций, в которых побывал пользователь и рассчитывает риск инфицирования локально на устройстве. При необходимости делает уведомление.

С помощью промежутка времени t можно сгруппировать всю информацию о посещениях инфицированными людьми и представить в виде временного ряда:

$$V = \{V_1, V_2, \ldots, V_N\},$$

где N = T / t, T – представляющее из себя unix timestamp.

Этот временной ряд описывает сколько инфицированных людей находились в заведении во временном слоте N.

Зная продолжительность присутствия пользователя в заведении, разбитое на временные слоты количеством P, мы можем запросить из DLT подмножество посещений $K^P \subseteq V^N$ за тот период времени, в который пользователь в нем присутствовал.

Затем рассчитать риск инфицирования можно следующим образом:

$$Risk(P, V_{max\_o}) = min(1, \frac{\sum_{p=1}^{P} k_p}{V_{max\_o}}),$$

где V_{max_o} – максимальное количество посетителей в заведении O, которое хранится в блокчейне в связанной с заведением информации.

Поэтому область значений определяется следующим образом $E(Risk) \in [0,1]$. В зависимости от значения риск-функции пользователю будет отображаться уведомление о вероятности инфицировании.

5. Процесс публикации информации об инфицированном в сеть

Большинство анализируемых решений одним из основных приоритетов декларируют вопрос обеспечения безопасности персональных данных людей [1,2]. Как минимум, реализуют шифрование передаваемых персональных данных. Как максимум, производят генерацию уникальных анонимных ключей на устройствах пользователей, которые знают лишь контактирующие стороны, но не могут идентифицировать друг друга [2,7].

В процессе публикации секретного ключа аналогично Desire [4] придерживаемся установки, что медицинский центр поставивший диагноз ни в каком виде не раскрывает связи между секретным ключом и пациентом. Наш алгоритм предполагает следующие шаги:

1. Пациент выгружает все контакты с локациями – в виде $\{Sgn_u, T_u\}$ за последние две недели и SK_u для этих контактов.
2. Происходит дедупликация данных по значению $Hash(Sgn_u, T_u)$.
3. $\{Sgn_u, T_u, SK_u\}$ публикуется в DLT от имени медицинской организации, где смарт-контракт:
   a) находит связанные данные $\{PK_u, PK_o, T_u, Sgn_u\}$ по Sgn_u;
   b) проверяет, что предоставленные пациентом T_u совпадает с тем, что опубликовано в DLT;
   c) проверяет $Encrypt(\{PK_u, PK_o, T_u\}, SK_u) = Sgn_u$;
   d) если 3.b и 3.c выполняются, то создает пометку, что локацию посещал инфицированный человек.
4. С помощью алгоритмов, заложенных в смарт-контракт, рассчитывается и обновляется риск-статус локации.

Под риск-статусом локации подразумевается абстрактная оценка вероятности заразится, находясь в ней. Аналогично риск-анализу область ее значений $\in [0,1]$ и в данной статье подробно не рассматривается.

Заключение

В рассмотренной модели, в отличие от большинства других, уделено особое внимание вопросу безопасности персональных данных, участвующих в ней субъектов. Данный подход позволяет обеспечить приватность пользователей и одновременно оценивать риск-статус конкретных городских локаций. После публикации секретного ключа его невозможно связать с конкретным пользователем, потому что идентификатор пользователя не фигурирует в хранилище, а личность пользователя из-

вестна лишь медицинской организации. В то же время при взаимодействии с IOT-устройством организации, устройство пользователя себя не идентифицирует никоим образом, что также позволяет избежать выстраивания связей.

Предложенная концептуальная архитектура системы отслеживания контактов между субъектов является гибридной, наиболее гибкой и способна совмещать в себе технические аспекты, как централизованной, так и децентрализованной системы, а использование пользовательских устройства для идентификации контактов наиболее простое и безопасное решение.

Логично предположить, что системы такого рода предназначены для работы с большим количеством данных, а проблема масштабирования DLT является актуальной [11]. Вопросы масштабирования описанной здесь архитектуры отдельно рассматриваются в [12].

Литература

1. *Ahmed N. et al.* A survey of COVID-19 contact tracing apps //IEEE access. 2020. T. 8.P. 134577-134601.
2. *Bay J. et al.* BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders //Government Technology Agency-Singapore, Tech. Rep.2020.
3. *Braithwaite I. et al.* Automated and partly automated contact tracing: a systematic review to inform the control of COVID-19 //The Lancet Digital Health. 2020.
4. *Castelluccia C. et al.* DESIRE: A Third Way for a European Exposure Notification System Leveraging the best of centralized and decentralized systems // arXiv preprint arXiv:2008.01621. 2020.
5. *Kleinman R.A., Merkel C.* Digital contact tracing for COVID-19 // CMAJ. 2020. T. 192. №. 24. P. 653-656.
6. *Lv W. et al.* Decentralized blockchain for privacy-preserving large-scale contact tracing //arXiv preprint arXiv:2007.00894. 2020.
7. *Rivest R.L. et al.* The PACT protocol specification //Private Automated Contact Tracing Team, MIT, Cambridge, MA, USA, Tech. Rep. 0.1. 2020.
8. *Roca V.* From ROBERT to DESIRE exposure notification: situation and lessons learned // Workshop on Security and Privacy in Contact Tracing. 2020.
9. *Shubina V. et al.* Survey of decentralized solutions with mobile devices for user location tracking, proximity detection, and contact tracing in the COVID-19 era //Data. 2020. T. 5. №. 4. P. 87.
10. *Sukhwani H. et al.* Performance modeling of hyperledger fabric (permissioned blockchain network) //2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). IEEE. 2018. P. 1-8.
11. *Zhou Q. et al.* Solutions to scalability of blockchain: A survey //IEEE Access. 2020. T. 8. P. 16440-16455.
12. *Тарханов И.А, Шмелев И.А.* Оценка масштабирования системы отслеживания социальных контактов на основе блокчейн // Искусственные общества. 2021. Т. 16. Вып. 3.

Тарханов Иван Александрович Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук», г. Москва, Россия. Старший научный сотрудник. Кандидат технических наук. Количество печатных работ: 41. Область научных интересов: электронный документооборот, блокчейн, моделирование бизнес процессов, информационная безопасность. E-mail: tarkhanov@isa.ru

Шмелев Илья Александрович. Национальный Исследовательский Технологический Университет (НИТУ) «МИСиС», г. Москва, Россия. Количество печатных работ: 2. Область научных интересов: блокчейн, информационная безопасность. Email: ishmelev23@gmail.com

Приложение А

```
struct Organization {
        String name
        PublicKey publicKey
        Integer maxPeopleCapacity
        String advertisementAddress
        Map<Integer, Integer> visits
}

struct MedicalEmployee {
        String name
        PublicKey publicKey
}

struct Admin {
        String name
        PublicKey publicKey
}

contract AccessController {
        Map<PublicKey, Organization> organizations
        Map<PublicKey, MedicalEmployee> medicalEmployees
        Map<PublicKey, Admin> admins

        addOrganization(name, publicKey, maxPeopleCapacity, advertisementAddress){
                if actorRole != 'admin' revert
                if organizations.contains(publicKey) revert

                organizations[publicKey] = new Organization(
                                name,
                                publicKey,
                                maxPeopleCapacity,
                                advertisementAddress
                        )
        }

        addMedicalEmployee(name, publicKey){
                if actorRole != 'admin' return
                if medicalEmployees.contains(publicKey) return

                medicalEmployees[publicKey] = new MedicalEmployee(
                                name,
                                publicKey,
                        )
        }

        addAdmin(name, publicKey){
                if actorRole != 'superadmin' revert
                if admins.contains(publicKey) revert

                admins[publicKey] = Admin(
                                name,
                                publicKey,
                        )
        }

    Organization getOrganizationInfo(publicKey) {
                return organizations[publicKey]
        }
}
```

Приложение Б

```
struct Contact {
        PublicKey organizationPublicKey
        PublicKey userPublicKey
        String userSignature
        Integer timeframe
        PrivateKey userPrivateKey = null
        PublicKey confirmatorPublicKey = null
}

contract SicknessRegistrar {
        Map<String, Contact> contacts

        registerContact(organizationPublicKey, userPublicKey, timeframe, userSignature) {
                if actorRole != 'organization' revert
                if actorPublicKey != organizationPublicKey revert
                if timeframe != currentTimeFrame revert

                contacts[userSignature] = new Contact(
                                organizationPublicKey,
                                userPublicKey,
                                timeframe,
                                userSignature
                        )
        }
    prune(){
      for(key, value in contacts){
        if value.timeframe < currentTimeframe - N {
          contacts.remove(key)
        }
      }
    }
}
```

Приложение В

```
struct Contact {
        PublicKey organizationPublicKey
        PublicKey userPublicKey
        String userSignature
        Integer timeframe
        PrivateKey userPrivateKey = null
        PublicKey confirmatorPublicKey = null
}

contract SicknessController {
        Map<String, Contact> contacts

        confirmContact(userSignature, timeframe, privateKey) {
                if actorRole != 'medicalEmployee' revert
          if not SicknessRegistrar.contacts.contains(userSignature) revert

                contact = SicknessRegistrar.contacts[userSignature]
                if contact.userPrivateKey != null revert
                if contact.timeframe != timeframe revert
                if encrypt(hash(contact.organizationPublicKey, contact.userPublicKey, timeframe),
privateKey) != userSignature revert

                contact.userPrivateKey = privateKey
                contact.confirmatorPublicKey = actorPublicKey

                organization = accessContract.organizations[contact.organizationPublicKey]
                if organization == null revert

                if organization.visits[timeframe] == null {
                        organization.visits[timeframe] = 0
                }
                organization.visits[timeframe] += 1
        }

        List<Integer> getSicknessRates(organizationPublicKey, timeframes){
                organization = accessContract.organizations[organizationPublicKey]
                if organization == null revert

                rates = []
                for timeframe in timeframes {
                        rates.add(organization.visits[timeframe])
                }
                return rates
        }
}
```

Conceptual architecture of a social contact tracking system based on blockchain

I.A. Tarkhanov[I,II], I.A. Shmelev[III]

[I] Laboratory for the Study of Blockchain in Education and Science (LIBON), State Academic University for the Humanities (GAUGN), Moscow, Russia

[II] Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

[III] National University of Science and Technology "MISiS", Moscow, Russia

Abstract. The article is devoted to the development of the concept of a system that monitors potential contacts between subjects and notifies them in case of the likelihood of infection. Based on the analysis of similar projects, it is proposed and justified the choice of an exclusive blockchain as a layer for storing and exchanging data, the algorithm for fixing a contact is described in detail, and the risk of analyzing infection and information dissemination is described. Particular attention is paid to the issue of maintaining the privacy of the subjects. The described system can be used in large cities or at the federal level.

Keywords: *social contacts, pandemic, blockchain, contact-tracing.*

DOI: 10.14357/20790279210407

References

1. *Ahmed N. et al.* A survey of COVID-19 contact tracing apps //IEEE access. – 2020. – T. 8. – C. 134577-134601.
2. *Bay J. et al.* BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders //Government Technology Agency-Singapore, Tech. Rep. – 2020.
3. *Braithwaite I. et al.* Automated and partly automated contact tracing: a systematic review to inform the control of COVID-19 //The Lancet Digital Health. – 2020.
4. *Castelluccia C. et al.* DESIRE: A Third Way for a European Exposure Notification System Leveraging the best of centralized and decentralized systems // arXiv preprint arXiv:2008.01621. – 2020.
5. *Kleinman R.A., Merkel C.* Digital contact tracing for COVID-19 //CMAJ. – 2020. – T. 192. – №. 24. – C. E653-E656.
6. *Lv W. et al.* Decentralized blockchain for privacy-preserving large-scale contact tracing //arXiv preprint arXiv:2007.00894. – 2020
7. *Rivest R.L. et al.* The PACT protocol specification //Private Automated Contact Tracing Team, MIT, Cambridge, MA, USA, Tech. Rep. 0.1. – 2020.
8. *Roca V.* From ROBERT to DESIRE exposure notification: situation and lessons learned // Workshop on Security and Privacy in Contact Tracing. – 2020.
9. *Shubina V. et al.* Survey of decentralized solutions with mobile devices for user location tracking, proximity detection, and contact tracing in the COVID-19 era //Data. – 2020. – T. 5. – №. 4. – C. 87.
10. *Sukhwani H. et al.* Performance modeling of hyperledger fabric (permissioned blockchain network) //2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). – IEEE, 2018. – C. 1-8.
11. *Zhou Q. et al.* Solutions to scalability of blockchain: A survey //IEEE Access. – 2020. – T. 8. – C. 16440-16455.
12. *Tarkhanov I.A, Shmelev I.A.* Assessing the scaling of a blockchain-based social contact tracking system // Artificial Societies – 2021. – V. 16. – Issue 3.

Tarkhanov I.A. PhD, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, 44/2 Vavilova str., Moscow, 119333, Russia, e-mail: tarkhanov@isa.ru

Shmelev I.A. PhD Student, National University of Science and Technology "MISiS", 4 Leninsky Prospect, Moscow, Russia e-mail: ishmelev23@gmail.com