# Математическая модель и программная архитектура для поиска по зашифрованным данным при подборе персонала

*Д. А. Финджоян, С.Г. Синица, В. О. Осипян*

*Кубанский государственный университет, Краснодар*

**Аннотация:** Рассматривается задача поиска по зашифрованным данным при помощи гомоморфного шифрования в распределённых программных системах. В работе рассматривается применение разработанного авторами подхода для реализации прототипов распределенной системы выдачи цифровых дипломов и сертификатов о получении компетенций с записью в IPFS и смарт-контракт Ethereum и поисковой системы для подбора персонала. Представленный в работе прототип системы демонстрирует применение гомоморфного шифрования, позволяющего поисковой системе и пользователям взаимодействовать с прошедшими прямое преобразование данными в публичных сетях блокчейн без выполнения обратного преобразования, обеспечивая их безопасность.

**Ключевые слова:** блокчейн, распределенные программные системы, облачные вычисления, гомоморфное шифрование, математическая модель, транзакции, смарт-контракты, цифровой диплом, подбор персонала.

## Введение

Технология блокчейн стремительно набирает темпы развития и проникает в различные сферы деятельности человека [1]. Развитие облачных вычислений, распределенных программных систем и реализация децентрализованных приложений для построения сервисов, использующих преимущества подобных систем, является перспективной [2].

Одновременно с ростом интереса к блокчейн, облачным вычислениям и децентрализованным приложениям растёт нужда в использовании криптографии для защиты данных. Потому создание алгоритмов, реализующих безопасную работу с данными, находящимися в распределённых программных системах – актуальная тема на сегодняшний день [3].

Гомоморфное шифрование является особой формой прямого и обратного преобразования, позволяющей реализовать операции сложения и умножения над данными после их прямого преобразования и получить

преобразованный результат, который соответствует аналогичным операциям над исходными не преобразованными данными [4]. Таким образом появляется возможность не только хранения, но и обработки прошедших прямое преобразование данных в распределенных системах.

**Определение.** Пусть k – ключ прямого преобразования, m – подлежащий прямому преобразованию текст. Тогда функция:

$$Enc(k,m) = C_{km} \quad (1)$$

является функцией, выполняющей прямое преобразование, а $C_{km}$ – результатом работы функции прямого преобразования, т.е. шифротекстом.

Функция (1) называется гомоморфной относительно операции умножения подлежащих прямому преобразованию текстов $m_1$ и $m_2$, если существует работающий за полиномиальное время алгоритм $Comp$, получающий на вход два прошедших прямое преобразование фрагмента $C_{km1}$ и $C_{km2}$ и формирующий в качестве результата шифротекст:

$$S_{km_1m_2} = Comp\big(Enc(k,m_1), Enc(k,m_2)\big) \quad (2)$$

такой, что при обратном преобразовании шифротекста (2) функцией:

$$Dec\big(k, S_{km_1m_2}\big) = m' = m_1 * m_2 \quad (3)$$

будет получено соотношение (3) [5].

Как правило, рассматривается прямое или обратное преобразование целых или вещественных чисел, а операция умножения является умножением таких чисел. Аналогично определяется функция, гомоморфная относительно операции сложения чисел.

Реализация алгоритмов, позволяющих безопасно обрабатывать данные при помощи гомоморфного шифрования [4, 5], позволяет создавать приложения и сервисы, в которых основной держатель ключей обратного

преобразования – конечный пользователь или некоторый доверенный пользователь [6, 7]. Вдобавок к этому, пользователь сможет изменять данные, хранимые в преобразованном виде, иметь к ним полный доступ, а в конечном счёте – использовать ряд приложений, построенных на идее использования распределённых программных систем и гомоморфного шифрования для действительности скрытности работы с информацией. Действительность заключается в подтверждении этих данных за счёт механизмов подтверждения транзакций в блокчейн, в то время как скрытность обеспечивается средствами гомоморфного шифрования.

Если рассматривать стандартные криптосистемы, позволяющие проводить прямое и обратное преобразование данных, то в максимально общем виде каждая такая криптосистема может быть определена как набор из нескольких операций: генерация ключей прямого и обратного преобразования, прямое преобразование и обратное преобразование [8]. В случае с гомоморфной системой появляется дополнительная операция, осуществляющая вычисление ($Comp$ из определений выше).

Рассмотрим каждую операцию детальнее:

1) Генерация ключей прямого / обратного преобразования: клиентская сторона при помощи данной функции получает пару ключей, использующихся для прямого (открытый ключ) и обратного (закрытый ключ) преобразования. В некоторых схемах гомоморфного шифрования для генерации ключа необходимо указывать набор параметров, которые могут накладывать некоторые ограничения на использование криптосистемы, если подобраны неверно.

2) Прямое преобразование: функция осуществляет прямое преобразование открытого текста при помощи открытого ключа.

3) Вычисление: функция, реализующая сложение / умножение двух шифротекстов, преобразованных одним и тем же открытым ключом. В

сценариях с участием распределённых программных систем и вычислений наличие открытого ключа позволяет делегировать редактирование данных другой стороне.

4) Обратное преобразование: функция получения исходного текста из шифротекста. Ключ для обратного преобразования не покидает клиентскую сторону, а значит и доступ к данным должен оставаться только у клиента.

Далее авторы приводят формальную модель криптосистемы с такими операциями, описывают модель для решения задачи публикации, индексирования и поиска с использованием гомоморфного шифрования и приводят пример практического применения для создания прототипа поисковой системы по подбору кадров.

## Определение математической модели криптосистемы с гомоморфным шифрованием

Дадим определение элементам математической модели исследуемой криптосистемы с применением гомоморфного шифрования.

Пусть даны параметры криптосистемы $\mu$ – данные параметры отличаются от схемы к схеме и могут накладывать некоторые ограничения $\mu$ в отношении отдельных элементов модели.

Пусть дано $M_\mu$ – множество открытых текстов $m$, для которых применима операция прямого преобразования. Под конкретным открытым текстом будем понимать последовательность символов $m_1 m_2 \dots m_n$.

Пусть $C_\mu = c_1 c_2 \dots c_j$ – множество всех возможных преобразованных прямым преобразованием текстов. Преобразованный текст может указываться с ключом $pk$ в нижнем индексе, например $c_{1pk}$, для обозначения того, каким конкретно ключом был преобразован соответствующий

открытый текст и определения возможности проведения соответствующих этому ключу операций.

Пусть даны функции:

$$GK(\mu) = [pk, sk] \tag{4}$$

$$E(pk, m) = c_{pk} \tag{5}$$

$$CMP_{\mu}(c_{1pk}, c_{2pk}) = c' \tag{6}$$

$$D(sk, c') = m' \tag{7}$$

Функция (4) является генератором ключей прямого/обратного преобразования (в некоторых случаях, $pk = sk$).

$E(pk, m)$ и $D(sk, c')$ – алгоритмы прямого и обратного преобразования соответственно. Функция (5) получает на вход публичный ключ и открытый текст $m$, формируя на выходе преобразованный прямым преобразованием текст $c_{pk} \in C$. Функция (7) производит обратное преобразование, получая открытый текст $m' \in M$. С учётом того, что гомоморфные системы позволяют редактировать преобразованные данные путём произведения математических операций $CMP$, будем в общем случае считать $m'$ и $c'$претерпевшими изменения на некотором шаге работы, хотя возможен случай, когда $m' = c'$.

Функция (6) $CMP_{\mu}(c_{1pk}, c_{2pk})$зависит от параметров криптосистемы и позволяет производить операции сложения и умножения над двумя преобразованными прямым преобразованием текстами. Основными требованиями является соблюдение ограничений $\mu$ (если таковые имеются) и

использование одного и того же ключа $pk$ для обоих аргументов данной функции.

Таким образом, всю математическую модель криптосистемы с применением гомоморфного шифрования можно определить в виде следующего кортежа:

$$HE = \langle M_{\mu}, C_{\mu}, CMP_{\mu}, E, D | R(E, CMP_{\mu}, D) \rangle \quad (8)$$

Элемент $R(E, CMP_{\mu}, D)$ в кортеже (8) определяет связь между операндами; в данном случае это означает, что любой открытый текст $m_i \in M_{\mu}$ может быть однозначно преобразован прямым и обратным преобразованием, а функция $CMP_{\mu}$, в свою очередь, не нарушает этой связи между операциями прямого и обратного преобразования, если будет применена между ними.

**Гомоморфное шифрование для поиска информации**

Поисковые системы в сети Интернет индексируют документы в открытом доступе. Однако некоторая информация, такая, как персональные данные пользователей, медицинские данные, данные об обучении, нуждается в защите и ее публикация в открытом виде не желательна. С использованием гомоморфного шифрования может быть создана поисковая система, осуществляющая поиск по данным, прошедшим прямое преобразование. При этом пользователи будут иметь возможность публиковать и менять данные децентрализовано. Для этого поисковая система генерирует пару ключей $pk, sk$ публикует открытый ключ $pk$. Для прямого преобразования данных $m$ перед публикацией пользователи используют открытый ключ $pk$ и публикуют в сети Интренет преобразованные данные $E(pk, m)$. Для того,

чтобы пользователи, заинтересованные в обновлении таких данных, например, медицинские или образовательные учреждения, могли изменять данные, они могут воспользоваться функцией $CMP$ и получить новую версию измененных данных пользователя, которая может быть опубликована в преобразованном виде вместо старой.

Для того, чтобы поисковая система могла обнаруживать вновь опубликованные пользователем преобразованные данные и обновления этих данных от других пользователей, уместно использовать открытые сети блокчейн с поддержкой смарт-контрактов, такие, как Ethereum [9]. Смарт-контракт поисковой системы $S$ содержит ее открытый ключ $pk$, список пользователей $I$, авторизованных поисковой системой для публикации данных и изменения данных, список текущих пользователей $L$ и преобразованные данные пользователей $E$. Объемные данные целесообразно вынести в распределенную файловую систему, такую, как IPFS [10], так как хранение больших объемов данных в смарт-контрактах Ethereum затруднено. На практике $E$ будет хранить ссылки и хеши данных в IPFS. Таким образом, преобразованные данные могут быть обнаружены и проиндексированы поисковой системой, описываемой смарт-контрактом со следующей структурой данных:

$$S = (pk, I, L, E) \tag{9}$$

Закрытый ключ $sk$ при этом используется только поисковой системой для обратного преобразования данных при реализации поиска и выдачи информации. Использование такого ключа поисковой системой и обратное преобразование данных в процессе поиска является недостатком предлагаемой модели. В [5] и [11] показана потенциальная возможность

выборки данных из базы данных преобразованных записей. Такая возможность в комбинации с иерархической генерацией ключей может устранить указанный недостаток. Однако на данный момент указанная возможность авторами не реализована, ее применение требует дополнительных исследований. Далее в работе рассматривается прототип поисковой системы, использующий гомоморфное шифрование только для организации распределенного процесса публикации, обновления и индексации данных.

## Прототип поисковой системы с применением гомоморфного шифрования

На основе представленных моделей авторами была разработана программная архитектура и прототип программной системы для публикации электронных документов об образовании в блокчейн Ethereum [9] и поиска персонала с использованием гомоморфного шифрования. Общая схема работы системы изображена на рис. 1. Такая система, с одной стороны, требует децентрализованной публикации информации о дипломах и компетенциях, в том числе, содержащей персональные данные, а с другой стороны, нуждается в поисковой системе, которая могла бы обрабатывать запросы к таким данным. Применение гомоморфного шифрования для построения такого рода системы позволит учебным заведениям самостоятельно публиковать сведения о выданных дипломах и степени освоения компетенций выпускниками в пригодном для проведения поиска и подбора сотрудников виде, при этом не публикуя в открытом виде персональные данные. Создание соответствующей инфраструктуры и поисковой системы уже ведется на национальном уровне в ряде стран [12] и будет способствовать развитию цифровой экономики [13 – 15].

Разработанный прототип поисковой системы индексирует профили пользователей по компетенциям и позволяет подбирать профили, удовлетворяющие запрашиваемому набору компетенций.

Указанную выше структуру данных $S$ хранит смарт-контракт Ministry. Адрес, с которого был размещён данный смарт-контракт, будем в дальнейшем называть Chairman. Данный смарт-контракт хранит в себе следующую информацию:

− список, содержащий образовательные учреждения Institution, зарегистрированные в смарт-контракте Ministry (процедуру регистрации может провести только Chairman);

− список зарегистрированных в Ministry учащихся Learner;

− список компетенций Competences, которые может получить Learner в рамках обучения в соответствующем Institution внутри данного Ministry.


Рис. 1. – Схема взаимодействия учебных заведений и учеников для записи полученных компетенций в блокчейн Ethereum и IPFS

Запись о Learner в смарт-контракте содержит в себе информацию об учащемся, такую, как: имя, контактные данные, адрес в сети блокчейн, преобразованный прямым преобразованием вектор, элементы которого –

целые числа, обозначающие степень владения тем или иным навыком. Например, если считать первый элемент вектора оценкой некоторой компетенции (владение некоторой технологией), то соответственно индексом будет являться идентификатор компетенции, а значением по этому индексу будет являться оценка владения этой технологией. Соответственно, вектор из таких компетенций будет представлять общий профиль человека и позволит поисковой системе, владеющей ключами обратного преобразования, собирать данные о специалистах в той или иной сфере, а при поиске людей, владеющих определёнными навыками, будет выдавать список кандидатур в порядке убывания оценок по данной компетенции.

Профиль учащегося получает оценки по компетенциям от образовательных организаций Institution. По завершению курса обучения организация, которая проводила обучение, должна отредактировать данные об имеющихся у человека навыках. Организация, которая собирается выдать обучившемуся сертификат, должна составить вектор компетенций, в котором в соответствующем компетенции индексе в качестве значения будет стоять оценка. Вектор поэлементно преобразуется открытым ключом поисковой системы $pk$, складывается с вектором в смарт-контракте пользователя, а преобразованный вектор из смарт-контракта Learner замещается результатом сложения двух преобразованных векторов. В этот момент поисковая система должна произвести реиндексирование.

При реиндексировании поисковая система получает вектор из компетенций, выполняет его обратное преобразование, и производит анализ изменений. Изменения затем попадают в результаты поиска. При выполнении поискового запроса система подбирает кандидатов с наивысшими оценками по требуемой компетенции.

Взаимодействие всех компонентов системы и её архитектура представлена на рис. 2.

Выбор распределенной файловой системы IPFS [10] обусловлен тем, что для хранения преобразованной информации необходим существенный объем данных. С ростом объема записи в контракт растёт и стоимость транзакции, а хранение ссылки на файл в IPFS является более дешевым вариантом. В свою очередь, использование смарт-контракта оправдано тем, что таким образом достигается прозрачность взаимодействия и предсказуемое поведение распределенной системы.


Рис. 2. – Схема взаимодействия компонентов поисковой системы подбора персонала и её архитектура

Для создания прототипа, демонстрирующего работу системы, был выбран следующий стек технологий:

1) **Node.JS** – серверная платформа, предоставляющая возможность писать и исполнять код JavaScript без использования браузера [16].

2) **node-seal** – библиотека Node.JS [17] на базе библиотеки SEAL от компании Microsoft. Позволяет работать с алгоритмами гомоморфного шифрования на схемах BFV [18], CKKS [19], имеет поддержку WebAssembly, обладает большим функционалом в сравнении с альтернативами.

3) **web3.js** – библиотека [20] для работы с блокчейн Ethereum по протоколу Web3.

4) **Ganache-cli** – консольная утилита для запуска локального блокчейна в целях тестирования и разработки [21]. Не требует дополнительной конфигурации, пригодна для использования с web3.js.

5) **js-ipfs** – пакет для работы с IPFS – децентрализованным файловым хранилищем в JavaScript [22].

Основной код написан на TypeScript. Это язык программирования, представленный Microsoft в 2012 году [23]. Создатели позиционируют его как средство разработки различных программ, которое расширяет функционал JavaScript. Является обратно совместимым с JavaScript. Скомпилированный код можно использовать в современных браузерах и на платформе Node.JS. Основная особенность – статическая типизация, позволяющая следить за качеством кода, предупреждать об ошибках совместимости типов, задавать собственные интерфейсы и типы данных.

Фрагменты исходного кода разработанного прототипа опубликованы авторами под лицензией GPL [24].

## Заключение

Таким образом, авторами разработана математическая и функциональная модель, программная архитектура распределенной информационной системы и прототип отдельных алгоритмов поисковой системы с применением гомоморфного шифрования. В работе авторами

показано как данный вид шифрования может использоваться для защиты персональных данных таким образом, что они хранятся в публичном блокчейне Ethereum и распределенной файловой системе IPFS в зашифрованном виде и обновляются участниками системы без доступа к исходным расшифрованным данным.

В работе рассматривается применение разработанного авторами подхода для реализации прототипов распределенной системы выдачи цифровых дипломов и сертификатов о получении компетенций с записью в IPFS и смарт-контракт Ethereum и поисковой системы для подбора персонала.

*Работа выполнена при поддержке гранта РФФИ № 19-01-00596*

**Литература**

1. Табернакулов А., Койфманн Я. Блокчейн на практике. Москва. Альпина Паблишер. 2019. 260 с.

2. Батура Т.В., Мурзин Ф.А., Семич Д.Ф. Облачные технологии: основные понятия, задачи и тенденции развития // Программные продукты и системы. 2014. № 3. С. 64–72.

3. Осипян В.О., Разработка математической модели дисимметричной биграммной криптосистемы на основе параметрического решения многостепенной системы диофантовых уравнений // Инженерный вестник Дона. 2020. №6. URL: ivdon.ru/ru/magazine/archive/N6y2020/6534.

4. Варновский Н.П., Шокуров А.В. Гомоморфное шифрование. Труды института системного программирования РАН. 2007. С. 27-36.

5. Gentry C. et al. A fully homomorphic encryption scheme. Stanford university. 2009. 209 p.

6. Трубей А.И. Гомоморфное шифрование: безопасность облачных вычислений и другие приложения (обзор) // Информатика. 2016. № 1. С. 90-101.

7. Ronald L. Rivest L. A., Dertouzos M. L. On Data Banks and Privacy Homomorphisms. Academic Press. 1978. pp. 160-179.

8. Parmar P.V. et al. Survey of various homomorphic encryption algorithms and schemes. Intern. J. of Computer Applications. 2014. Vol. 91, no. 8. pp. 26-32.

9. Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. URL: ethereum.org/en/whitepaper/.

10. Benet J. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561. 2014. URL: arxiv.org/pdf/1407.3561.pdf.

11. Kim P., Jo E., Lee Y. An Efficient Search Algorithm for Large Encrypted Data by Homomorphic Encryption. Electronics 10(4):484. 2021. DOI:10.3390/electronics10040484. URL: mdpi.com/2079-9292/10/4/484.

12. Kontzinos C., Kokkinakos P., Kapsalis P., Markaki O., Karakolis V., Psarras J. Leveraging Blockchain, Analytics and Decision Support to Facilitate Qualifications' Verification, Recruitment and Competency Management: The QualiChain Project and Initial Results. International Journal on Advances in Intelligent Systems. Volume 13, Number 3 & 4. 2020. pp. 177-191.

13. Абашева О.Ю., Амирова Э.Ф., Беляева С.В., и др. Цифровая экономика и сквозные цифровые технологии: современные вызовы и перспективы экономического, социального и культурного развития. Самара. ООО НИЦ «ПНК». 2020. С. 19-31.

14. Полетайкин А.Н., Синица С.Г., Кунц Е.Ю. Технология разработки и верификации профессиональных стандартов, их применения в системах управления обучением на основе онтологий // Экономика и управление: теория и практика. 2020. Т. 6. № 2. С. 37-46.

15. Полетайкин А.Н., Синица С.Г., Данилова Л.Ф., Черногорова И.В. Методика анализа соответствия образовательной программы состоянию рынка труда // Современное образование: повышение

конкурентоспособности университетов. Материалы международной научно-методической конференции, в 2 частях. Томск. 2021. С. 102-108.

16. Node.JS.URL: nodejs.org.

17. Node-seal – библиотека для работы с классом алгоритмов гомоморфного шифрования. URL: github.com/morfix-io/node-seal.

18. Fan J., Vercauteren F. Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch. 2012. URL: eprint.iacr.org/2012/144.pdf.

19. Cheon J.H., Kim A., Kim M., Song Y. Homomorphic encryption for arithmetic of approximate numbers. Advances in Cryptology, ASIACRYPT 2017. Springer. 2017. pp. 409–437. DOI:10.1007/978-3-319-70694-8_15. URL: researchgate.net/publication/321366831_Homomorphic_Encryption_for_Arithmetic_of_Approximate_Numbers.

20. Ethereum JavaScript API. URL: github.com/ethereum/web3.js/.

21. Ganache-CLI: command line version of Ganache. URL: github.com/trufflesuite/ganache-cli.

22. The JavaScript implementation of the IPFS protocol. URL: github.com/ipfs/js-ipfs.

23. TypeScript – JavaScript that scales. URL: typescriptlang.org.

24. Примеры применения гомоморфного шифрования в связке с Ethereum blockchain, смарт-контрактами и IPFS. URL: github.com/starpl/he-eth-examples.

## References

1. Tabernakulov A., Kojfmann Y. Blokchejn na praktike [Blockchain on practice]. Moskva. Al'pina Pablisher. 2019. 260 p.

2. Batura T.V., Murzin F.A., Semich D.F. Programmnye produkty i sistemy. 2014. № 3. pp. 64–72.

3. Osipyan V.O. Inzhenernyj vestnik Dona. 2020. №6. URL: ivdon.ru/ru/magazine/archive/N6y2020/6534.

4. Varnovskij N.P., Shokurov A.V. Trudy instituta sistemnogo programmirovaniya RAN. 2007. [Homomorphic encryption. Proceedings of the Institute for System Programming of the Russian Academy of Sciences]. pp. 27-36.

5. Gentry C. et al. A fully homomorphic encryption scheme. Stanford university. 2009. 209 p.

6. Trubej A.I. Informatika. 2016. № 1. pp. 90-101.

7. Ronald L. Rivest L. A., Dertouzos M. L. Academic Press. 1978. pp. 160-179.

8. Parmar P.V. et al. Intern. J. of Computer Applications. 2014. Vol. 91, № 8. pp. 26-32.

9. Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. URL: ethereum.org/en/whitepaper/.

10. Benet J. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561. 2014. URL: arxiv.org/pdf/1407.3561.pdf.

11. Kim P., Jo E., Lee Y. Electronics 10(4):484. 2021. DOI:10.3390/electronics10040484. URL: mdpi.com/2079-9292/10/4/484.

12. Kontzinos C., Kokkinakos P., Kapsalis P., Markaki O., Karakolis V., Psarras J. International Journal on Advances in Intelligent Systems. Volume 13, Number 3 & 4. 2020. pp. 177-191.

13. Abasheva O.Y., Amirova E.F., Belyaeva S.V., i dr. Cifrovaya ekonomika i skvoznye cifrovye tekhnologii: sovremennye vyzovy i perspektivy ekonomicheskogo, social'nogo i kul'turnogo razvitiya. [Digital economy and end-to-end digital technologies: modern challenges and prospects for economic, social and cultural development]. Samara. OOO NIC «PNK». 2020. pp. 19-31.

14. Poletajkin A.N., Sinica S.G., Kunc E.YU. Ekonomika i upravlenie: teoriya i praktika. 2020. T. 6. № 2. pp. 37-46.

15. Poletajkin A.N., Sinica S.G., Danilova L.F., Chernogorova I.V. Materialy mezhdunarodnoj nauchno-metodicheskoj konferencii, v 2 chastyah. Tomsk. 2021. pp. 102-108.

16. Node.JS. URL: nodejs.org.

17. Node-seal – biblioteka dlya raboty s klassom algoritmov gomomorfnogo shifrovaniya [Node-seal – homomorphic encryption algorithm library]. URL: github.com/morfix-io/node-seal.

18. Fan J., Vercauteren F. Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch. 2012. URL: eprint.iacr.org/2012/144.pdf.

19. Cheon J.H., Kim A., Kim M., Song Y. ASIACRYPT 2017. Springer. 2017. pp.409–437.DOI:10.1007/978-3-319-70694-8_15. URL:researchgate.net/publication/321366831_Homomorphic_Encryption_for_Arithmetic_of_Approximate_Numbers.

20. Ethereum JavaScript API. URL: github.com/ethereum/web3.js/.

21. Ganache-CLI: command line version of Ganache. URL: github.com/trufflesuite/ganache-cli.

22. The JavaScript implementation of the IPFS protocol. URL: github.com/ipfs/js-ipfs.

23. TypeScript – JavaScript that scales. URL: typescriptlang.org.

24. Primery primeneniya gomomorfnogo shifrovaniya v svyazke s Ethereum blockchain, smart-kontraktami i IPFS [An implementation examples of homomorphic encryption with Ethereum blockchain, smart contracts and IPFS]. URL: github.com/starpl/he-eth-examples.