

УДК: 004.043

ПРИМЕНЕНИЕ БЛОКЧЕЙН-ТЕХНОЛОГИИ В ИНТЕРНЕТЕ ВЕЩЕЙ

Тимофеев А.А.

к.т.н., доцент кафедры ИС Болтенков В.А.

Одесский Национальный Политехнический Университет, УКРАИНА

АННОТАЦИЯ. В работе рассмотрен пример практической интеграции блокчейн-технологии в интернет вещей. Показана последовательность действий на платформе Ethereum по созданию блокчейн-контракта для устройства Arduino, подключенного к IoT.

Введение. Интернет вещей (Internet of Things, IoT) является очередным этапом эволюции Интернета на пути к всеобъемлющему интернету (Internet of Everything, IoE). IoT содержит широкий спектр вещей, таких как сенсоры, исполнительные механизмы и услуги, развернутые различными организациями и частными лицами для поддержки различных приложений. В последнее время количество IoT устройств многократно увеличилось и продолжает стремительно расти. Только на конец прошлого года объем IoT-технологий достиг 4,5 млрд, а к 2020 г. в мире прогнозируется 20,8 млрд. подключенных устройств IoT [1,2]. Увеличение устройств сбора и обработки данных, подключенных к сети Интернет, приводит к возникновению проблем, связанных с безопасностью данных. Технология Blockchain предлагает новое решение проблемы безопасности и конфиденциальности в среде IoT, обеспечивая новый вычислительный слой, где данные могут быть безопасно обработаны и проанализированы, оставаясь частными [3,4]. Blockchain – распределенная структура данных, состоящая из последовательности блоков, в которой каждый блок содержит хэш предыдущего блока, образуя, как следствие, цепь блоков. Blockchain, работая как распределенная база данных, осуществляет учет всех операций в сети. Операции имеют метку времени и сохраняются в блоках, где каждый блок идентифицируется своим криптографическим хэшем. Для работы Blockchain не нужно обеспечивать доверия между узлами сети, поскольку любой узел может самостоятельно проверить, совпадает ли его копия базы с копиями, которые хранятся в других узлах.

Цель работы. Целью работы стала экспериментальная проверка возможности интеграции IoT и блокчейн-технологии, в частности, в бытовой сфере.

Основная часть работы. Для исследования выбрано условное устройство, созданное на основе платы Arduino и WiFi-модуля ESP8266. К общей схеме было добавлено несколько кнопок – условных триггеров. Устройство настраивается в WiFi-модуль, ему прописывается порт и адрес, по которому будет происходить связь с модемом. Далее на Arduino устанавливается скорость передачи данных в 115200 бит/с – это требуется для правильного взаимодействия микроконтроллера и модуля. Для условных триггеров пишутся условия, которые по каналу передачи будут выводить строки. В нашем условном эксперименте – это просто номер кнопки. Затем WiFi-модуль считывает данные с TX-порта и отправляет эти данные по сети. На этом настройка условного IoT-устройства закончена.

Принимающей стороной выступает компьютер, подключенный к тому же модему, что и устройство. Был разработан сервер на языке Python. Для подключения к модулю использована библиотека Serial, в которую после создания записана та же самая скорость передачи данных – 115200 бит/с. Затем указан порт, который использован при настройке WiFi-модуля и Arduino. Последний шаг – открытие последовательного порта с установленными настройками. Этого достаточно, чтобы слушать устройство и получать весь поток передачи данных.

Организована система реакции на нажатия кнопок. Вначале создается контракт, который позволит работать с системой Blockchain. Для этого была выбрана платформа Ethereum – она является открытой и значительно упрощает внедрение данной технологии [5]. Контракты Ethereum – это классы, написанные на каком-то из языков программирования, нами

использован язык Solidity (хотя есть возможность использовать также Serpent, LLL, Mutan). Перед отправкой в цепь контракт преобразуется в виртуальной машине в байт-код.

Наш контракт имеет два поля: первое – это карта, которая хранит пары значений «строка - число» и второе массив строк, т.е. карта хранит полученные сигналы, а массив – список зарегистрированных устройств. Предусмотрены четыре метода. Первый – это конструктор, в который при создании заносится список устройств. Второй метод позволяет получить сохранить данные – в нашем случае это просто увеличивающийся счётчик. Третий метод позволяет просмотреть этот счётчик для определённого устройства, которое мы передали в качестве аргумента. Четвертый метод – проверка устройства на действительность – проверяется, есть ли переданное устройство в списке уже зарегистрированных.

Вернемся к построению сервера. Взаимодействие с Blockchain происходит с помощью библиотеки Web3Py. Для начала создается объект класса Web3, в который передаётся HTTP провайдер с адресом и портом нашей сети. Для работы с контрактом используется библиотека solc – с ее помощью создаётся отдельная переменная, которая приравнивается к методу compile_source с переданным в аргумент файлом на языке Solidity.

Для создания самого контракта используется фабричный метод класса Contract из той же библиотеки Web3Py. В нём нужно переопределить переменные байт-кода и его время исполнения, указав переменную с заранее откомпилированным кодом. Также передается бинарный интерфейс (ABI). Далее остаётся только развернуть наш контракт на сервере – это делается одноименным методом. Затем необходимо получить адрес нашего контракта. На следующем шаге создается образец нашего контракта с помощью настроенной нами фабрики, передачей как аргумента выше полученного адреса. Теперь имеется возможность использовать методы контракта. Добавляются условия на проверку серийного порта WiFi-модуля и реакцию на соответствующие значения с использованием образца контракта.

В результате при нажатии на кнопку плата Arduino считывает повышение напряжения на контакте. В зависимости от номера контакта, она отправляет по потоку передачи какие-то данные, в нашем частном случае – это просто цифровое обозначение кнопки по порядку. Далее по проводам данные идут в WiFi-модуль на соответствующий контакт. Тот в свою очередь по вписанному в него адресу и порту, передаёт сигнал далее на роутер. Принимающее устройство слушает тот же порт и при появлении сигнала на нём по серийному порту принимает начальные данные. Далее в соответствии с полученными данными выполняются какие-то определенные действия. Ими может быть разбиение данных, их обработка, сравнения и другие операции. Затем вызывается нужный метод в контракте и вносится информация в Blockchain. В случае необходимости можно вернуть данные, обработать их, и на их основе внести изменения в работу устройства.

Выводы. Практически была выполнена интеграция IoT и Blockchain-технологии на условном устройстве и продемонстрирована работоспособность интегрированной системы. Применение блокчейн-технологии в IoT обеспечивает безопасность и конфиденциальность в среде IoT за счет создания нового вычислительного слоя, где данные могут быть безопасно обработаны и проанализированы, оставаясь частными.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Risteska Stojkoska B.L. A Review of Internet of Things for Smart Home: Challenges and Solutions / B.L. Risteska Stojkoska, K. V. Trivodaliev // Journal of Cleaner Production. – 2017. – №140. – P.1454-1464.
2. McEwen A. Designing the Internet of Things / A. McEwen, H. Cassimally . – John Wiley and Sons. – 2016. – 324 p.
3. Перспективи використання технології блокчейн у мережі інтернет речей / Н.Г. Яцків , С.В. Яцків // Науковий вісник НЛТУ України. – 2016. – Вип. 26.8. – С.381-387.
4. Данильчук Р. К. Аналіз основних принципів технології blockchain / Р. К. Данильчук, О. С. Жураковська // Науковий огляд. – 2017. – № 10(42). – С. 1-11.
5. Prusty N. Building Blockchain Projects. Develop Real-time Practical DApps using Ethereum and JavaScript. — Birmingham — Mumbai: Packt Publ. , 2017. — 245 p.