УДК 004 **Информационные технологии**

Гаранин Никита Андреевич, *студент магистр, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*
Белов Юрий Сергеевич, *к.ф.-м.н., доцент, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*

ПРОТОКОЛ КОНТРАКТНОЙ МАРШРУТИЗАЦИИ ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ, ОСНОВАННЫЙ НА ТЕХНОЛОГИИ БЛОКЧЕЙН

Аннотация: Интернет вещей включает все сферы жизнедеятельности человека. Но существует ряд недостатков во взаимодействии, совместимости и конфиденциальности данной системы. В статье предлагается новый протокол контрактной маршрутизации на основе блокчейна для обеспечения безопасности и анонимности устройств Интернета вещей.

Ключевые слова: Интернет вещей, протокол, блокчейн, маршрутизация, гетерогенная сеть.

Abstract: The Internet of Things includes all spheres of human activity. But there are a number of shortcomings in the interaction, compatibility and confidentiality of this system. The article proposes a new blockchain-based contract routing protocol to ensure the security and anonymity of Internet of Things devices.

Keywords: Internet of Things, protocol, blockchain, routing, heterogeneous network.

Введение. Недавний прогресс в области беспроводной связи и мобильных вычислений позволил большому разнообразию устройств подключаться к Интернету, формируя Интернет вещей (IoT). IoT - это

гетерогенная сеть различных типов устройств от разных поставщиков, которые собирают, передают, обрабатывают и анализируют данные и предпринимают соответствующие действия [1]. Данная технология сталкивается с многочисленными проблемами из-за необходимости интеграции большого количества разнородных объектов. Но в данной технологии присутствует ряд проблем, основная из которых: заключается в том, что большое количество поставщиков IoT не могут просто договориться о централизованной системе управления. Это связано с проблемой доверия между поставщиками Интернета вещей и высокой стоимостью внедрения инфраструктур управления доверием, таких как инфраструктура открытых ключей (PKI). Еще одна проблема это - безопасность. Обеспечение безопасной связи и предотвращение вмешательства злоумышленников в процесс маршрутизации являются основными проблемами в этой сети.

Для решения данных проблем организуем децентрализованный протокол контрактной маршрутизации на основе блокчейна BCR (Blockchain Contract Routing) [2]. BCR позволяет устройствам Интернета вещей разных производителей доверять друг другу и сотрудничать во время передачи данных, сохраняя целостность и безопасность данных.

Протокол контрактной маршрутизации

В предлагаемом протоколе BCR каждое исходное устройство интернета вещей создает смарт-контракт для запроса маршрута к месту назначения или шлюзу данных на определенный период вместо создания управляющих сообщений RREQ. Каждый смарт-контракт, созданный устройством Интернета вещей, имеет отдельный адрес в блокчейне, который генерируется производителем блоков при размещении смарт-контракта в блоке. Устройство Интернета вещей может передавать этот адрес своим соседям, чтобы сообщить им о новом запросе маршрутизации. Протокол BCR реализован с использованием смарт-контрактов в блокчейне [3]. Таким образом, передача управляющих сообщений в существующих протоколах маршрутизации

заменяется вызовами функций смарт-контрактов в протоколе BCR. Далее подробно объясняется протокол BCR.

Состояния смарт-контракта протокола BCR описаны ниже:

1) Запрошенный маршрут: Когда исходному устройству интернета вещей необходимо подключиться к шлюзу, оно создает смарт-контракт в блокчейне и отправляет адрес смарт-контракта своим соседям. Он также устанавливает поле состояния в смарт-контракте для запрашиваемого маршрута. Этот смарт-контракт называется первоначальным контрактом.

2) Предлагаемый маршрут: Каждое соседнее устройство Интернета вещей, которое имеет действительный вход маршрута к шлюзу и хотело бы участвовать в ретрансляции пакетов данных, может реагировать на оригинальный смарт-контракт. Промежуточное устройство интернета вещей предлагает свои услуги исходному устройству, вызывая функцию в рамках исходного контракта и передавая некоторые из своих собственных токенов на адрес смарт-контракта. Промежуточный контракт хранит адрес первоначально выпущенного смарт-контракта или другого промежуточного контракта в $Parent\ Contract$ параметр.

3) Принятый маршрут: Исходное устройство Интернета вещей определяет, следует ли принимать предложенный маршрут для отправки своих пакетов данных. Он выбирает следующего соседа для доступа к шлюзу на основе своих собственных внутренних политик.

4) Пройденный маршрут: Когда данные принимаются шлюзом данных, состояние смарт-контракта изменяется на данные, переданные шлюзом. Исходное устройство интернета вещей добавит свои текущие адреса в черный список любого вновь созданного смарт-контракта ($Blacklisted\ Addresses$).

5) Прервано: В любое время каждое устройство в сети Интернета вещей может прервать процесс маршрутизации, вызвав функцию прерывания внутри смарт-контракта.

6) Истек срок действия: Поскольку протокол BCR имеет различные таймеры, устройство Интернета вещей может запросить функцию истечения

срока действия внутри смарт-контракта для проверки таймеров и принятия соответствующих мер.

Переход BCR

Переход по протоколу определяет требуемые условия, которые запускают изменение состояния. Устройства интернета вещей выполняют триггер при вызове функции внутри смарт-контракта протокола BCR. Интернет вещей вызывает функции для запуска протокола BCR [4].

Параметры протокола BCR: параметры в смарт-контракте используются функциями смарт-контракта и могут быть видны публично. Параметры протокола BCR в устройстве интернета вещей устанавливаются устройством интернета вещей на основе его собственной внутренней политики. Каждое устройство интернета вещей может иметь свои собственные значения для этих внутренних параметров.

1) Адрес контракта хранит адрес смарт-контракта. Смарт-контракт может быть динамически создан внутри блокчейна исходным устройством интернета вещей или ранее создан владельцем устройства.

2) Состояние указывает текущее состояние смарт-контракта. Возможными состояниями являются Запрошенный маршрут, Предлагаемый маршрут, Принятый маршрут, Переданные данные, Истек срок действия и Прерван.

3) Источник, посредник и назначение хранят адреса устройств интернета вещей источника, посредника и назначения. Исходное устройство IoT запросило доступ к шлюзу передачи данных. Промежуточные устройства готовы ретранслировать пакеты данных с исходного устройства в пункт назначения или шлюз передачи данных.

4) Срок действия запроса маршрута (RRE) - это время истечения срока действия, до которого запрос маршрута действителен.

5) Связь с запросом маршрута (RRB) устанавливается исходным устройством интернета вещей и показывает количество токенов, которые исходное устройство IoT заплатит промежуточному устройству интернета

вещей, если маршрут к месту назначения работает должным образом и место назначения получает пакеты данных.

6) Срок действия предложения маршрута (ROV) показывает период, в течение которого действителен маршрут, предлагаемый промежуточным устройством Интернета вещей к исходному устройству IoT.

7) Облигация предложения маршрута (ROB) - это количество токенов, которые промежуточное устройство интернета вещей помещает в качестве облигации, чтобы гарантировать, что промежуточное устройство может успешно передавать пакеты данных на шлюз.

Описание алгоритмов BCR

Переход между состояниями смарт-контракта осуществляется путем вызова функций смарт-контракта. Каждый раз, когда узел IoT вызывает функцию, некоторые токены, указанные в Gas функции, будут перемещены из учетной записи блокчейна устройства интернета вещей в учетную запись производителя блоков [5].

1) Route Request(): Каждое устройство Интернета вещей, когда ему необходимо достичь пункта назначения или шлюза данных, может запросить, чтобы производители блокчейна создали смарт-контракт на блокчейне[6]. Исходное устройство IoT подписывает цифровую транзакцию для этой цели и устанавливает параметры смарт-контракта. Эта функция показана в Алгоритме 1.

Алгоритм 1 – Route Request():

```
(1) function RoUTe ReqUesT(desTInATIon, RRB, RRE, BLACKLIsT,
                              pARenTAddRess(opTIonAL), Hop(opTIonAL) )
(2)   Передать токены Gas от вызывающей функции производителю блока
(3)   Передача токенов RRB от вызывающей функции на текущий адрес
контракта
(4)   Установка RRE в Route_Request_Expiry
(5)   Установить Черный список в Blacklisted_Adresses
(6)         если это оригинальный смарт-контракт, то
(7)               Установить Hop = 0
(8)         если это промежуточный смарт-контракт, то
(9)               Установить Hop = Hop
```

```
(10)              Установить Parent_Contract = Parent_Adress
(11)        Установить Timestamp = Now
```

2) Route Offer(): Когда промежуточное устройство интернета вещей устанавливает маршрут к месту назначения или шлюзу данных в своей внутренней таблице маршрутизации и готово передавать ему пакеты данных для исходного устройства IoT. Каждый контракт принимает до трех предложений о маршруте от устройств-посредников [7]. Эта функция показана в Алгоритме 2.

Алгоритм 2 – Route Offer():

```
(1) function RoUTe oFFeR(ROB, ROV)
(2)   Передать токены Gas от вызывающей функции производителю блоков
(3)   если адрес вызывающего абонента функции не указан в
Blacklisted_Adresses, а количество предложений меньше трех, то
(4)         Перевести токены ROB от вызывающей функции на текущий адрес
контракта
(5)         Установить значение ROV для действительного предложения
```

3) Route Accept(): Всякий раз, когда исходное устройство Интернета вещей решает принять предложенный маршрут, оно вызывает функцию Route Accept в блокчейне[8]. Производитель блоков запускает эту функцию, если вызывающий функцию IoT адрес устройства идентичен адресу исходного устройства интернета вещей в смарт-контракте. Эта функция показана в алгоритме 3.

Алгоритм 3 – Route Accept():

```
(1) function RoUTe ACCepT(InTeRMedIARy)
(2)   Передача токенов Gas от вызывающей функции производителю блока
(3)   если вызывающий функцию является Source, то
(4)         Переместить посредника для продажи Selected_Route
(5)         Передать токены ROB других промежуточных устройств обратно
```

4) Data Pass(): Всякий раз, когда целевое устройство Интернета вещей получает пакеты данных, оно может вызвать функцию передачи данных в блокчейне. Производитель блоков запускает функцию, если адрес вызывающей

функции совпадает с адресом назначения в смарт-контракте. Эта функция показана в Алгоритме 4.

Алгоритм 4 – Data Pass():

```
(1) function dATA pAss()
(2)   Передача токенов Gas от вызывающей функции производителю блока
(3)   если вызывающий функцию Destination, то
(4)         Перевести токены RRB и ROV в SR
```

5) Expire(): Всякий раз, когда целевое устройство IoT получает пакеты данных, оно может вызвать функцию передачи данных внутри блокчейна. Производитель блоков запускает функцию, если адрес вызывающего функцию устройства интернета вещей идентичен адресу целевого устройства интернета вещей в смарт-контракте. Эта функция показана в Алгоритме 5.

Алгоритм 5 – Expire():

```
(1) function expIRe()
(2)   Передать токены Gas от вызывающей функции производителю блока
(3)   если state - запрошенный маршрут, то
(4)         если текущее время больше, чем RRE, то
(5)               Перевести токены RRB обратно в Sourse
(6)               Перевести токены ROB обратно в Intermediary
(7)   если state - предлагаемый маршрут, то
(8)         если текущее время больше, чем ROV, то
(9)               Перевести токены RRB обратно в Sourse
(10)              Перевести токены ROB обратно в Intermediary
(11)  если state - принятый маршрут, то
(12)        если вызывающий функция является Intermediary или Destination,
то
(13)              Перевести RRB и ROB, в State
(14)        если вызывающий функцию является State, то
(15)              Перевести токены RRB и ROB в Selected_Route
```

6) Abort(): Всякий раз, когда устройство Интернета вещей желает выйти из контракта, оно может вызвать функцию прерывания [9]. В зависимости от состояния контракта функция прерывания возвращает токены устройствам IoT. Эта функция показана в Алгоритме 6.

Алгоритм 6 – Abort():

```
(1) function ABoRT()
```

```
(2)   Передать токены Gas от вызывающей функции производителю блока
(3)   если state - запрошенный маршрут, а вызывающий функцию - Source, то
(4)         Передача токенов RRB обратно вызывающей функции
(5)   если s - предлагаемый маршрут, то
(6)         если вызывающий функцию является Source, то
(7)               Передача токенов RRB обратно вызывающей функции
(8)               Передать токены ROB всех промежуточных устройств обратно
(9)         если вызвана функция Intermediary, то
(10)              Передать токены ROB обратно вызывающему функцию
(11)  если state - принятый маршрут, то
(12)  если вызывающий функция является Intermediary или Destination, то
(13)              Перевести RRB и ROB, в State
(14)        если вызывающий функцию является Source, то
(15)              Перевести токены RRB и ROB в Intermediary
```

Заключение. Данный протокол решает поставленные проблемы и отлично подходит для IoT. Так же, протокол BCR может расширяться, поскольку любая организация может присоединиться к блокчейн-сообществу или покинуть его в любое время. Новым организациям не нужно изменять или развивать систему в соответствии с потребностями консорциума. Из-за простоты BCR, система интегрируется без особых дополнительных затрат на разработку, и при необходимости может быть отредактирована в зависимости от потребностей.

Библиографический список:

1. Гаранин Н.А. Белов Ю.С. Облачные сервисы в решении базовых проблем интернета вещей // Гуманитарный национальный исследовательский институт «НАЦРАЗВИТИЕ». Материалы международной научной конференции ГНИИ «НАЦРАЗВИТИЕ». В сборнике: «НАУКА. ИССЛЕДОВАНИЯ. ПРАКТИКА». 2021. С. 87-90.

2. Гаранин Н.А. Белов Ю.С. Защита устройств интернета вещей (IoT) с помощью блокчейн-фреймворка Hyperledger fabric // В сборнике: Научное обозрение. Технические науки. 2021. С. 17-21.

3. Bouzembrak Y., KlËuche M, Gavai A., and Marvin H. J. P. Internet of things in food safety: literature review and a bibliometric analysis // Trends in Food Science & Technology. 2019. vol. 94. pp. 54–64.

4. Muessigmann B., H. von der Gracht, and Hartmann E. Blockchain technology in logistics and supply chain management // IEEE Transactions on Engineering Management. 2020. vol. 67. no. 4. pp. 988–1007.

5. Raban D. R. and Gordon A. Evolution of data science and big data research: a bibliometric analysis // Scientometrics. 2020. vol. 122. no. 3. pp. 1563–1581.

6. Patil P., Sangeetha M., and Bhaskar V. Blockchain for IoT access control, security and privacy: a review, // Wireless Personal Communications. 2020. vol. 117. no. 1. pp. 1–20.

7. Boutaib S., Bechikh S., Palomba F., Elarbi M., Makhlouf M., and Said L. B. Code smell detection and identification in imbalanced environments // Expert Systems with Applications. 2021. vol. 166. article 114076.

8. Choo R., Yan Z., and Meng W. Editorial: blockchain in industrial IoT applications: security and privacy advances, challenges, and opportunities // IEEE Transactions on Industrial Informatics. 2020. vol. 16. no. 6. pp. 4119–4121.

9.Tandon, A. An empirical analysis of using blockchain technology with internet of things and its application // Int. J. Innov. Technol. Explor. Eng. 2019. vol. 8. pp. 1470–1475.