УДК 004.738.5 DOI: 10.26565/2519-2310-2019-1-05

ПОБУДОВА СИСТЕМИ ГОЛОСУВАННЯ З ВИКОРИСТАННЯМ БЛОКЧЕЙН ТЕХНОЛОГІЙ НА ПРИКЛАДІ HYPERLEDGER

Микита Гончаров, Євген Деменко, Микола Полуяненко, Володимир Шлокін

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
wdpgames@yandex.ru, demenjay@gmail.com, nlfsr01@gmail.com, vshlokin@ukr.net

Рецензент: Сергій Кавун, доктор економічних наук, к.т.н., проф., Харківський технологічний університет "Шаг", вул. Малом'ясницька, 9/11, Харків, 61010, Україна.
kavserg@gmail.com

Надійшло: Квітень 2019.

Анотація: *Обговорюється характеристики і особливості роботи в системі Hyperledger Fabric та характерні проблеми реалізації транзакцій. Проведено розгляд регістрів і принципів їх роботи в відповідних системах, зокрема підключення системи до голосування та перевірка блоків блокчейна. Визначено механізми захисту відповідних систем і характерні уразливості. Розглянуто інфраструктуру відкритого ключа в системі.*

Ключові слова: *комп'ютерні мережі; децентралізація; Hyperledger Fabric; блокчейн.*

1 Вступ

В представленій роботі розглядаються питання що пов'язані зі створенням системи голосування на основі блокчейн проекту - Hyperledger Fabric, яка, свого часу, була створена для побудови бізнес додатків корпоративного рівня [1], як проект в рамках IBM. Зі слів представників IBM, Hyperledger Fabric «розроблена для створення основи побудови мережевих блоків в корпоративному класі, які можуть швидко масштабуватися, оскільки нові учасники мережі приєднуються та здійснюють транзакції зі швидкістю більше 1000 транзакцій в секунду в великих екосистемах». Ця платформа (надалі Hyperledger) має відкритий вихідний код, надає деякі ключові можливості для диференціації порівняно з іншими популярними платформами.

Hyperledger є блокчейн системою з цифровим регістром, який виявляє спроби підробки та є захищеними від несанкціонованого втручання. Такі системи зазвичай реалізовані без центрального сховища та без централізованого управління. На своєму базовому рівні вони дозволяють співтовариству її користувачів реєструвати транзакції в загальному реєстрі в рамках цієї спільноти, таким чином, що при штатному функціонуванні блокчейн мережі, ніяка транзакція (теоретично) не може бути змінена після її опублікування. В загальному випадку блокчейн - це розподілені цифрові регістри транзакцій які криптографічно підписані, та згруповані в відповідні блоки. Кожен з них криптографічно пов'язаний з попереднім блоком після проведення відповідної перевірки та прийняття консенсусного рішення. Згодом, з додаванням нових блоків, старі блоки починають важко піддаватися змінам (*тобто створюється опір підробки*). Нові блоки записуються в копії реєстру блокчейн мережі, а будь-які конфлікти вирішуються автоматично, відповідно до встановлених правил [2].

Одним з ключових моментів диференціації є те, що Hyperledger був створений саме під Linux Foundation, який має успішну історію створення та просування проектів з відкритим кодом під управлінням яких, як показала історія, зростають сильні спільноти та процвітають екосистеми. Як стверджує спільнота розробників програмного забезпечення - відкритий вихідний код є основною «філософією» цього проекту: - «Тільки відкритий підхід до розробки програмного забезпечення може забезпечити прозорість, довговічність і підтримку, необхідних для просування технологій блокчейн в майбутньому» [3].

Одним з найважливіших диференціаторів платформи є підтримка підключення консенсусних протоколів, які дозволяють платформі більш ефективно налаштовуватися відповідно до конкретних випадків використання та моделей довіри.

Hyperledger може використовувати консенсус-протоколи, які не вимагають нативної крипто валюти для дорогого видобутку або для розробки сматр-контракту. Уникнення криптовалюти зменшує деякі значні вектори ризику/атаки, а відсутність криптографічних видобувних операцій означає, що платформа може бути розгорнута приблизно з тими ж операційними витратами, як і будь-яка інша розподілена система.

В основному цим проектом зацікавлені в таких галузях, як охорону здоров'я, послуги кредитних карток (фінансовий сектор), ланцюг постачання продукції та виробництво.

Розробники Hyperledger на своєму сайті навели п'ять основних цілей свого проекту:

- Забезпечення нейтральної, відкритої та керованою громадою інфраструктури, що підтримується технічним та бізнес-управлінням;
- Створювання корпоративних класів, відкритих джерел;
- Створення технічних спільнот для розробки блокчейну;
- Ознайомлення громадськості з можливостями блокчейн технологій;
- Заохочування спільнот, використовуючи підхід з багатьма платформами.

Характеризуючи архітектуру Hyperledger, можна виділити такі аспекти, як:

- Підтримка підприємств, яка може забезпечити ступінь стабільності, мотивуючи тих, хто ще не впевнений в перспективах блокчейн;
- Модульна архітектура, в котрій саме користувач визначає, що йому використовувати, а що ні.
- Розподілена книга Hyperledger та платформа смарт-контрактів дозволяють використовувати приватні канали.

Наприклад: - *якщо у вас є велика блокчейн мережа і ви хочете поділитися окремими даними лише з певними сторонами, то ви можете створити приватний канал лише с цими учасниками.*

- Прозорий процес. Самі транзакції можуть бути і не прозорими, але сам процес розробки – може. «На цьому етапі головні команди Hyperledger були надзвичайно готові збалансувати потреби, щоб отримати важливі віхи з відкритим і прозорим процесом розвитку» - зазначив засновник системи «Skuchain» - Закі Манян.
- Смарт-контракти: - подібно до Ethereum, Hyperledger дозволяє використовувати смарт-контракти, які називаються "*ланцюговими кодами*".

Для написання смарт-контрактів (*чейнкод в контексті Hyperledger*) використовують Golang (хоча Hyperledger дозволяє використовувати і інші мови). А для розробки користувальницького додатка використовувався Node.js, Java та Go з відповідним Hyperledger Fabric SDK, а не обмежені доменні мови (DSL). Це означає, що більшість підприємств вже мають набір навичок, необхідних для розробки інтелектуальних контрактів, і тому не потрібно додаткового навчання для вивчення нової мови або DSL [4].

На відміну від Ethereum, Fabric не вимагає вбудованої криптовалюти. Можна розробити національну валюту або цифровий маркер з ланцюговим кодом, але це вимагатиме значного інвестування ресурсів розвитку. Тому відсутність криптовалют робить його більш практичним для створення бізнес–додатків корпоративного рівня.

Серед недоліків слід зазначити проблему масштабування [5]. Так як Hyperledger побудований на концепції каналів, то це значить, що для кожної зі сторін, які виконують певні операції, буде потрібно створити свій канал. Вочевидь, це може призвести до створення великої кількості каналів, та як наслідок, система вийде з під контролю. Нижче розглянемо питання щодо створення та керування транзакцій блокчейна та їх механізми захисту на основі реалізації системи голосування.

2 Регістр в системі Hyperledger

Регістр в системі Hyperledger являє собою збори транзакцій. Протягом всієї історії рукописні та паперові бухгалтерські книги використовувалися для відстеження обміну товарів і послуг. У наш час регістри зберігаються в цифровому вигляді, більшою частиною у великих

базах даних, що належать і управляються централізованою довіреною третьою стороною (*власником регістру*) від імені спільноти користувачів. Ці регістри з централізованою власністю можуть бути реалізовані *централізованим* або *розподіленим* способом (або єдиний сервер або координується кластер серверів).

В даний час все більше зростає інтерес до вивчення можливості розподілу власності на регістр. Технологія блокчейн дозволяє реалізувати саме такий підхід, використовуючи при цьому як розподілене володіння, так і розподілену фізичну архітектуру.

Розподілена фізична архітектура мереж блокчейн часто включає в себе значно більшу кількість комп'ютерів, ніж це характерно для розподіленої фізичної архітектури з централізованим управлінням. Зростаючий інтерес до розподіленої власності на регістри пояснюється можливими проблемами з довірою, безпекою і надійністю, що пов'язані з регістрами та централізованою власністю. Деякі важливі особливості цих двох систем наведені нижче:

1. Регістри з централізованим видом володіння можуть бути втрачені або знищені, а користувач, при цьому, повинен цілком покладатися на те, що власник підтримує систему належним чином.

• Мережа блокчейн є розподіленою за своєю суттю. Вона створює безліч резервних копій, кожна з яких постійно оновлюється і синхронізується з одними і тими ж обліковими даними між всіма одноранговими регістрами. При цьому ключовою перевагою технології блокчейн є те, що кожен користувач може зберегти свою власну копію реєстру, що значно ускладнює втрату або руйнування регістра. Слід звернути увагу на те, що певні реалізації мережі блокчейн забезпечують можливість проведення закритих транзакцій або закритих каналів. Принциповим є те, що закриті транзакції сприяють доставці інформації тільки тим сайтам, які беруть участь в транзакції, а не всієї мережі.

2. Регістри з централізованим володінням можуть перебувати в однорідної мережі, де все програмне та апаратне забезпечення і мережева інфраструктура може бути однаковими. В цих умовах, атака на одну частину мережі буде працювати на всіх її елементах, тому загальна стійкість системи може бути значно знижена.

• Блокчейн - це гетерогенна структура, де у силу великої кількості всіляких відмінностей між її вузлами, атака на один них не завжди буде працювати на інших її частинах.

3. Регістри з централізованим володінням, частіше за все, розташовані повністю в певній географічній локації (наприклад, все в одній країні). Тому в разі виникнення будь яких перебоїв в роботі мережі, регістр і служби, скоріш за все будуть недоступні.

• Мережа блокчейн, зазвичай, складається з географічно розподілених вузлів, що знаходяться в будь-якій точці світу. Внаслідок цього, а також рівноправній моделі організації роботи мережі блокчейн, вона більш стійка до втрати одного, або навіть цілої ділянки вузлів.

4. Транзакції в регістрі з централізованим володінням не проводяться прозоро і можуть бути недійсними. В цих умовах користувач може тільки вірити, що власник перевіряє всі отримані транзакції.

• Мережа блокчейн зобов'язана перевіряти дійсність всіх транзакцій. Якщо шкідливий вузол передає недійсні транзакції, то інші будуть їх виявляти і ігнорувати, таким чином запобігаючи поширення недійсних транзакцій по всій блокчейн мережі.

5. Список транзакцій в регістрі системи з централізованим управлінням може бути не повним, а її користувач повинен вірити, що власник записує всі отримані транзакції.

• Мережа блокчейн утримує всі прийняті транзакції всередині розподіленого регістру. При цьому, для того щоб побудувати новий блок, необхідно послатися на попередній. Отже, надбудувати нового над старим блоком. Таким чином, якщо вузол не містить посилання на останній блок, то інші вузли повинні його відкинути.

6. Інформація об операції в регістрах з централізованим управлінням може бути піддана змінам. Користувач повинен вірити, що власник не змінює минулі транзакції.

• Мережа блокчейн з метою забезпечення виявлення підробок та захисту від зовнішнього втручання в регістри, використовує відповідні криптографічні механізми (*цифровий підпис і криптографічна хеш-функція*).

7. Системи з централізованим володінням можуть бути небезпечними (*з точки зору можливостей парирування актуальних загроз безпеки*). Користувач повинен був упевнений, що пов'язані комп'ютерні системи та мережі своєчасно отримують критичні оновлення для операційних систем і систем інформаційної безпеки та впроваджують передові методи захисту. Така система може бути зруйнована, а чутлива інформація може бути вкрадена або скомпрометована.

• У мережі блокчейн, в силу її розподіленої структури, не може бути централізованої точки атаки. В цілому, інформація в блокчейн мережі загальнодоступна і, відповідно, нічого красти. В цих умовах щоб зробити атаку на користувачів блокчейн мережі, атакуючий повинен особисто визначити кожну з цілей. Вибір в якості мети всієї мережі блокчейн, зустріне опір справжніх вузлів, представлених у системі. Якщо один вузол був змінений, то це вплине тільки на нього, але не на систему в цілому.

Для прикладу розглянемо модель системи голосування, що була попередньо розроблена для ознайомлення з інфраструктурою Hyperledger та її тестування. Ця система має щонайменше одного адміністратора, готовий механізм взаємодії учасників та можливість породжувати і контролювати нові вузли. Це означає, що можна вільно додавати учасників, активи та проводити транзакції [6].

2.1 Підключення системи голосування до Hyperledger і перевірка блоків блокчейна

Для підключення моделі голосування треба ввести команду *composer network install --card PeerAdmin@hlfv1 --archiveFile voting-simple@0.0.2.bna* в термінал, щоб встановити зв'язок з системою голосування. Команда *./createPeerAdminCard.sh,* дозволяє створити карту доступу для отримання прав в участі для голосування.

Тест працездатності мережі забезпечується командою *composer network ping --card admin@voting-simple*. За допомогою команди *composer-rest-server*, згенеруємо REST-сервер.

Далі йде серія питань, які будуть визначати параметри майбутнього сервера. Відповіді вводяться з клавіатури та підтверджуються натисканням клавіші «Enter». Для забезпечення коректної роботи моделі слід ввести параметри, що уточнюють як повинна працювати система, або за допомогою команди *composer-rest-server -c admin@voting-simple -n never -u true -w true* пропустимо етап для визначення параметрів сервера. Після цього у вікні терміналу з'явиться повідомлення:

Web server listening at: http://localhost:3000

Browse your REST API at http://localhost:3000/explorer.

Його відображення означає, що сервер працює.

Ознайомитися зі структурою системи можна за адресою *http://localhost:3000/explorer***.**

Після запуску сервера введемо в новому вікні терміналу наступну команду *docker ps –a*, що дозволить побачити інформацію про запущені контейнери вузли мережі.

Потрібно вибрати CONTAINER ID порту hyperledger / fabric-peer-1.2.1 і наступним кроком ввести в консоль команду *docker exec-it CONTAINER ID/bin /bash* (перейдемо в кореневе розташування цього контейнера).

Наступним шагом перейдемо в іншу директорію за допомогою команди *cd /var/hyperledger/production/ledgersData/chains/chains/composerchannel/* команда *ls* **(**виводить на консоль файли, що знаходяться у поточній директорії).

Для виведення блоку блокчейна використовується команда *peer chaincode query -C "composerchannel" -n qscc –c '{"Args":["GetBlockByNumber","composerchannel","Number"]}'*, замість Number термінал вводиться номер блоку (для виводу на екран). Блок складається з нуля і з кожним голосом регістр блоку перезаписує свою транзакцію, створюючи новий блок зі старими операціями залишаючи, при цьому, історію в попередньому блоці.

Відображення голосу кандидата в регістрі блокчейн виглядає наступним чином:

```
[
  {
        "$class": "org.voting.example.vote", - модель системи голосування;
```

"candidateVoteAsset": {Alice}, - ім'я кандидата;
"ifVotedAsset": {false/true}, - перевірка що кандидат проголосував;
"transactionId":" string", - транзакція записується в рядок;
"timestamp": "2019-05-06T22:24:06.804 Z" - час коли голос був доданий.
}
]

3 Захист та ключі в системі Hyperledger

Щоб зрозуміти систему захисту особистих даних, слід уявити ситуацію, в якої деяка особа відвідуєте супермаркет, щоб щось купити продукти, але на касі присутній знак, який оголошує, що приймаються тільки картки Visa, Mastercard і AMEX. В цьому разі, якщо особа буде намагатися розплатитися іншою карткою - назвемо її "ImagineCard" – буде зовсім неважливо, чи є ця карта справжньою та чи є на цьому рахунку достатньо коштів - в будь-якому випадку це буде неможливо виконати. В цих обставинах мати дійсну кредитну картку явно недостатньо – вона, також, повинна бути прийнята в конкретному магазині!

PKI та MSP (англ. *Membership Service Provider* - постачальник послуг членства) працюють разом однаково - PKI надає список ідентичностей, а MSP говорить, хто з них є членами даної організації, яка бере участь у мережі.

Органами сертифікації PKI та MSP надаються подібні комбінації функціональних можливостей. PKI подібний до постачальника карт - він видає велику кількість різних типів перевірених облікових записів. MSP, з іншого боку, подібний до списку постачальників карток, прийнятих магазином, визначаючи, які ідентичності є довіреними членами (*учасниками*) платіжної мережі магазину. MSP перетворюють перевірені ідентичності на членів мережі блокчейн.

Інфраструктура відкритого ключа (PKI – від англ. *Public Key Infrastructure*) - це сукупність інтернет-технологій, що забезпечують безпечні комунікації в мережі**.** Фактично це PKI, який додає S в HTTPS.

Відомі чотири ключові елементи для PKI: - цифрові сертифікати; - відкриті та приватні ключі; - органи сертифікації; - списки анулювання сертифікатів.

Цифровий сертифікат - це документ, який містить набір атрибутів, що стосуються власника сертифіката. Найбільш поширеним типом сертифікату є той, що відповідає стандарту X.509. Він дозволяє кодувати ідентифікаційні дані сторони в його структурі.

Сертифікати можуть бути широко розповсюджені, оскільки вони не включають в себе особистих ключів учасників та цифровий підпис. Вони можуть бути використані як інструмент довіри для автентифікації повідомлень, що надходять від різних учасників.

Центр сертифікації (ЦС), також має сертифікат, що є в широкому доступі. Це дозволяє споживачам ідентифікаційних даних, що видані даним ЦС, перевіряти їх, маючи на увазі що сертифікат може бути створений тільки власником відповідного закритого ключа.

У налаштуваннях блокчейн, кожен користувач, що бажає взаємодіяти з мережею, потребує забезпечення його ідентичності. У цьому разі ЦС забезпечує основу того, щоб всі користувачі організації мали перевірену цифрову ідентичність.

Традиційними механізмами автентифікації є цифрові підписи, які забезпечують гарантії цілісності підписаного повідомлення. З технічної точку зору, механізми цифрового підпису вимагають, щоб сторона мала два ключа – відкритий та приватний (останній служить для створення цифрових підписів на повідомленнях). Взаємозв'язок між цими ключами робить можливим захищені комунікації. При цьому математична залежність між обома ключами така, що закритий ключ може використовуватися для створення підпису на повідомленні, яке може відповідати тільки відповідний відкритий ключ, і тільки на одне і те ж повідомлення.

Приклад використання приватного ключа [7] для підписання умовного повідомлення наведено на рис. 1. Згідно з ним Мері використовує свій приватний ключ для підписання повідомлення. При цьому підпис може перевірятися всіма, хто бачить підписане повідом-

лення за допомогою відкритого ключа. Цифрові підписи створюються за допомогою ECDSA. Слід підкреслити, що найважливішою перевагою ECDSA є можливість його роботи на значно менших полях . Як, загалом, з криптографією еліптичної кривої, передбачається, що бітовий розмір відкритого ключа, який буде необхідний для ECDSA, дорівнює подвійному розміру секретного ключа в бітах.

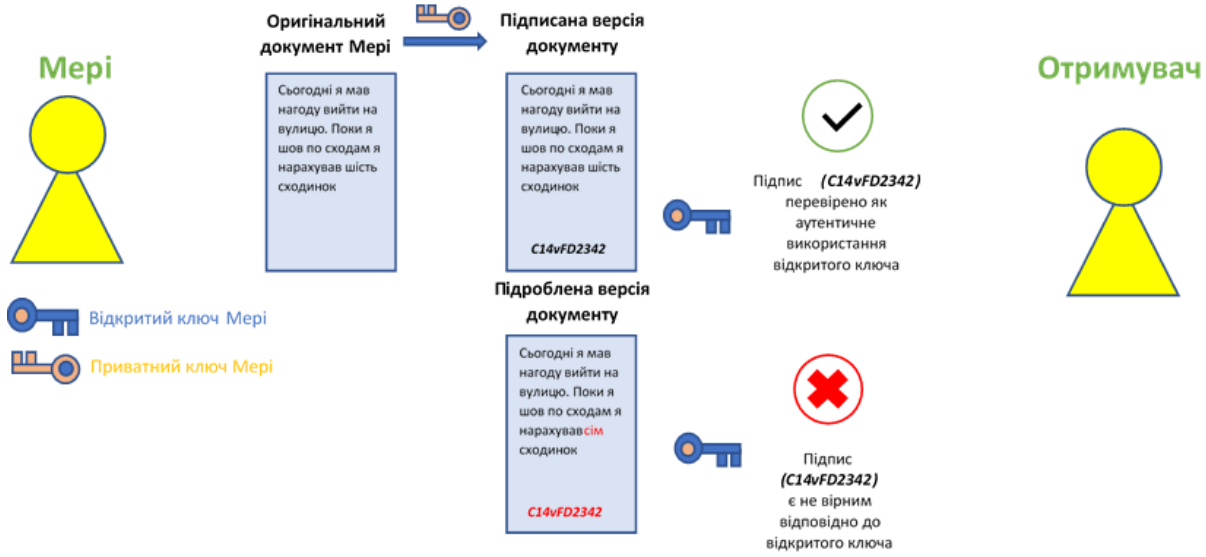

Рис. 1 – Використання приватного ключа для підпису повідомлення

У більшості випадків користувач зберігає облікові дані в себе, включаючи закритий ключ та він сам підписує транзакції. Проте варто зазначити, що деякі бізнес-сценарії можуть вимагати більш високого рівня конфіденційності. Однак слід зазначити, що в системі присутня проблема ключового зіткнення [8], але ця проблема не є специфічною для Hyperledger, це загальна проблема комп'ютерних наук і для існує багато способів її вирішення.

Повинно бути розуміння, що час коли SDK відправляє перший запит і час, коли ця транзакція зберігається в блокчейні, не є детермінованими. Це може зайняти кілька секунд, а більшість цих кроків виконуються паралельно. Отже, в чому тут проблема?

Розглянемо наступний приклад: є ланцюговий код, який переказує гроші від абонента Б до абонента А. Ланцюговий код повинен перевірити, що Б вистачає коштів до переказу, а якщо ні, то ця транзакція повинна бути скасована. Нехай на даний момент Б має 10 жетонів і поставив запит на передачу в бік А 10 з них. Чейнкод перевірить правильність цієї суми (*під час моделювання*), і ця транзакція буде надіслана замовнику для фіксації. Але, що станеться, якщо в той же час абонент Б послав би ще одну угоду, але вже до абонента В, і знову ж таки на 10 жетонів?

В цьому разі чейнкод перевірятиме, чи має він (абонент Б) необхідну суму, оскільки передача до абонента А буде перевіряється (*в симуляції*), але не здійснюється. Результатом буде те, що абонент Б відправляє 20 жетонів, у той час як у нього фактично є тільки 10. Це є класичний приклад "подвійної проблеми витрат".

З цієї причини Hyperledger реалізує MCVV - добре відомий і перевірений механізм для запобігання подібним ситуаціям. В цілому Hyperledger не дозволить користувачам створити такий тип ситуації - конфліктні транзакції будуть відхилені, а SDK буде проінформований про відповідну відмову.

В чому полягає проблема цієї архітектури? Якщо хтось намагаєтеся оновити один і той же ключ, всі транзакції, крім одного, будуть відхилені. Так наприклад, якщо, у вас є IoT датчик (від англ..*internet of things* - Інтернет речей), що передає дані з частотою 10 разів на секунду, а середній час створення нового блоку (для внесення даних) становить 1 сек, то 9 з 10 транзакцій будуть відхилені, оскільки вони конфліктують. Але уявимо

собі такий же сценарій з грошовим переказом: - це приклад дуже поганого досвіду так, як 9 з 10 фінансових транзакцій не вдасться.

В цьому контексті не варто забувати, що Hyperledger може працювати з частотою більш ніж 210 000 транзакцій за хвилину. Таким чином при проектуванні архітектури, структур даних і потоків необхідно враховувати можливості MVCC Hyperledger, а методи, які можуть бути використані, визначаються типами даних і потоками. Немає жодного «золотого правила» або «кожного з них».

Варто зазначити, що MVCC розшифровується, як керування паралельним доступом за допомогою багатоверсійності [9]. Без управління паралельного доступу, якщо хтось одночасно зчитує та записує інформацію бази даних, є можливість, що читач побачить тільки наполовину написану або непослідовну інформацію.

Ізоляція – це властивість, що гарантує паралельний доступ до даних. Ізоляція реалізується за допомогою протоколу управління паралельним доступом. Найпростішим методом буде змусити всіх читачів чекати поки записувач не закінчить свої дії, що відомо як *блокування читання-запису*. Я показав досвід такі блокування часто викликають суперечки, особливо між довгими транзакціями читання та обновлення. Задум MVCC полягає в тому, щоб зберігати декілька копій кожного елементу даних. Таким чином, кожен користувач, що під'єднаний до бази даних в конкретний період часу бачить лише так званий «знімок» бази даних. При цьому будь-які зміни, що здійснюються записувачем не будуть відображатися для інших користувачів до моменту безпосереднього завершення відповідних змін. Тому архітектура, структури і потік повинні бути адаптовані таким чином, щоб не відбувалося жодних конфліктуючих операцій або щонайменше їх виконувалось дуже мало.

Бази даних, які використовують MVCC, є центральними, вони виконують більшість операцій в пам'яті, можуть використовувати дуже низький рівень блокування і синхронізації процесу, а більшість операцій займають мілісекунди. Так як Hyperledger є децентралізованою, то багато її учасників можуть бути або в автономному режимі, або мати величезне мережеве відставання. Також в чейнкоді Hyperledger немає дубльованих ключів. Оскільки зіткнення відбувається тільки тоді, коли один і той же ключ оновлюється одночасно, а якщо ніколи не використовуємо той же ключ, то зіткнення ніколи не відбудеться. Це відносно просто реалізувати в ланцюговому коді, але пізніше викличе багато інших питань. Замість того, щоб мати один ключ для облікового запису, користувач буде мати, можливо, сотню ключів, і знати фактичну суму в цьому коді ланцюжка рахунку повинні приймати всі ці ключі, перебирати їх і визначати те, що є поточним значенням. Слід розуміти, що в цьому разі ви/користувач "обходите" MVCC, та відкриті для подвійної проблеми з витратами. Це можливо, тому що об'єктивне значення, яке перевіряє MVCC, не є одним ключем, а це є результат іншого процесу - ітерація всіх ключів і виконання певних математичних дій над значеннями. Цей підхід корисний, коли потрібно зберігати величезну кількість швидких даних, що зберігаються незмінно в блокчейні, але ці дані не будуть частиною бізнес-процесу.

Так, щоб отримати файл с ключами треба спочатку отримати відповідний CONTAINER ID за допомогою команди *docker ps*. Контейнер, котрий потрібний буде стояти під командою *"peer node start"* та буде мати образ *hyperledger/fabric-peer:1.2.1*.

Щоб отримати bash-оболонку контейнера Hyperledger використовують команду *docker exec /bin/bash*. Зазвичай, для того, щоб взнати, що міститься в команді використовують команду *docker exec -it <container name> <command>*, в нашому випадку це є доступ до корінної директорію контейнеру. *cd /var/hyperledger/production/* - команда, що веде в ще одну внутрішню директорію, а *cd /etc/hyperledger/peer/msp/keystore/* буде шляхом до нашого файлу, який знаходиться в цій папці.

ЦС є настільки важливими, що Fabric надає вбудований компонент ЦС, якій дозволяє створювати CA в мережах блокчейн, котрий ви формуєте. Fabric ЦС - приватний кореневий постачальник ЦС, здатний керувати цифровими ідентифікаціями учасників Fabric, які мають форму сертифікатів X.509. Оскільки Fabric ЦС є користувацьким ЦС, що орієнту-

ється на потреби кореневого ЦС у Fabric, то воно не здатне забезпечити SSL-сертифікати для загального/автоматичного використання в браузерах.

Крім того, для посилення рівня безпеки в системі використовуються відповідні списки анулювання сертифікатів (CRL – *Certificate Revocation List*). По суті це список посилань на сертифікати, про які ЦС знає що вони відкликанні. Так наприклад, відносно умов/сценарію магазину, CRL буде схожий на список викрадених кредитних карт.

В умовах коли третя сторона хоче перевірити особистість іншої сторони, вона спочатку перевіряє CRL відповідного ЦС, щоб переконатися, що сертифікат не був відкликаний. В цьому разі верифікатору не потрібно перевіряти CRL, але якщо вони цього не роблять, то ризикують прийняти порушений ідентифікатор. Слід звернути увагу на те, що відкликаний сертифікат сильно відрізняється від сертифіката, який закінчив свій термін дії. Термін дії анульованих сертифікатів не закінчився - вони, за будь-яким іншим показником, є повністю дійсним сертифікатом.

PKI може надавати перевірочні ідентичності через ланцюжок довіри. Наступним кроком є те, як ці ідентичності можна використовувати для представлення надійних членів мережі блокчейн. Саме там вступає в дію постачальник послуг членства (MSP) - він визначає сторони, які є членами даної організації в блокчейн мережі.

4 Висновки

Технологія блокчейн ґрунтується на розподіленому зберіганні частки інформації на кожному з вузлів, що підключені до системи, а кожен блок містить посилання на інші частини бази даних. Внесення змін відбувається послідовно, шляхом перезапису кожного блоку, а допуск до відповідних дій здійснюється за допомогою індивідуального логіна і пароля.

Основні переваги систем що впроваджують блокчейн полягають в наступному:

1. Безпека. Зламати систему складно - необхідно отримати доступ до механізму консенсусу який залежить від кількості вузлів в децентралізованій системі.
2. Відсутність комісійних зборів. Транзакції здійснюються безпосередньо між користувачами.
3. Висока швидкість роботи. Блоковий підхід використовує продуктивність всіх учасників/вузлів, розділяючи між ними практично всі навантаження.
4. Безперервність роботи. Блокчейн система діє 24 години на добу, та 365 днів на рік.

Блокчейн технологію вже сьогодні успішно застосовують не тільки для фінансових транзакцій, але і для зберігання інформація, а в перспективі можливе використання блокчейна в сферах освіти, комунальних службах та промисловості.

Однак блочне зберігання, не є ідеальними. До його мінусів слід віднести наступне:

1. Відсутність державного регулювання. На законодавчому рівні взаємовідносини в децентралізованих мережах поки ніяк не регламентуються.
2. Анонімність приваблює злочинні елементи, так як при здійсненні транзакцій не потрібно підтверджувати особу.
3. Відносна новизна технології викликає певну недовіру з боку великих компаній і корпорацій. Однак, стрімкий розвиток та збільшення популярності відповідних систем поступово переконує про перспективність проектів на основі блочного зберігання інформації.
4. Використання технології блокчейн все ще знаходиться на початковій фазі розвитку, але воно впроваджує зрозумілі та ґрунтовні криптографічні принципи.
5. Блокчейн використовує існуючі мережеві і криптографічні технології, однак, використання існуючої інфраструктури здійснюється на принципово нових принципах.

Таким чином технологія блокчейн - це новий перспективний інструмент з потенційними додатками для організацій, що здатен забезпечити високій рівень безпеки транзакцій без необхідності існування вузла централізованого управління мережею.

Всі транзакції захищені криптографічними хешами, підписуються і перевіряються з використанням пар асиметричних ключів. Історія транзакцій ефективно і безпечно фіксує вісь ла-

нцюжок подій таким чином, що будь-яка спроба підміни або будь якої корекції минулої транзакції, вимагатиме перерахунку всіх наступних блоків транзакцій.

Посилання

[1] Hyperledger Fabric 2019. Instroduction URL: https://hyperledger-fabric.readthedocs.io/en/master/whatis.html.
[2] NISTIR 8202 Blockchain Technology Overview / Yaga D., Mell P., Roby N., Scarfone K. URL: https://csrc.nist.gov/ publications/detail/nistir/8202/final
[3] Нефедов Н. Hyperladger Fabric для чайников. URL: https://habr.com/ru/company/ibm/blog/444874.
[4] The plus and cons of hyperledgerfabric. URL: https://www.verypossible.com/blog/the-pros-and-cons-of-hyperledger-fabric.
[5] Установка інструментів Hyperledger Fabric для розробки та тестування блокчейн-мереж. URL:https://docs.google.com/document/d/1NWVvRCIHphirDHD169AYgT2VG5b4xhuFYC8N9WdF3Cw/edit#heading=h.f3a2riitnib6.
[6] Head_aefkz. Плюсы и минусы блокчейна. URL: https://aef.kz/blockchain/plyusy-i-minusy-blokchejna
[7] Hyperledger Fabric 2019. Identity. URL: https://hyperledger-fabric.readthedocs.io/en/master/identity/identity.html
[8] Refs and Transaction. URL: https://clojure.org/reference/refs
[9] Vankov I. How to prevent key collisions in Hyperledger Fabric chaincode. URL: https://medium.com/@gatakka/how-to-prevent-key-collisions-in-hyperledger-fabric-chaincode-303700716733.

Reviewer: Serhii Kavun, Doctor of Sciences (Economics), Ph.D. (Engineering), Full Prof., Kharkiv University of Technology "STEP", Malom'yasnitska St. 9/11, Kharkiv, 61010, Ukraine. E-mail: kavserg@gmail.com

Received on Aprilr 2019.

Authors:
Nikita Goncharov, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: wdpgames@yandex.ru

Eugene Demenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: demenjay@gmail.com

Nikolay Poluyanenko, Ph.D., Computer Science Department, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: nlfsr01@gmail.com

Vladimir Shlokin, Director of the Innovation Center, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: vshlokin@ukr.net

Building a voting system with using blockchain technologies in the example of Hyperledger.

Abstract. It discusses the characteristics and features of work in the Hyperledger Fabric system and the characteristic problems of the implementation of transactions. The registers and principles of their work in the respective systems are considered, in particular, the connection of the system to voting and the check of Blockchain blocks. Defined mechanisms for the protection of relevant systems and characteristic vulnerabilities. Considered public key infrastructure in the system.

Keywords: Computer networks; Decentralization; Hyperledger Fabric; Blockchain.

Рецензент: Сергей Кавун, доктор экономических наук, к.т.н., проф., Харьковский технологический университет "ШАГ", ул. Маломясницкая, 9/11, Харьков, 61010, Украина. E-mail: kavserg@gmail.com

Поступила: Апрель 2019.

Авторы:
Никита Гончаров, студент факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: wdpgames@yandex.ru

Евгений Деменко, студент факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: demenjay@gmail.com

Николай Полуяненко, к.т.н., викладач факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: nlfsr01@gmail.com

Владимир Шлокин, директор инновационного центра, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина. E-mail: vshlokin@ukr.net

Построение системы голосования с использованием блокчейн технологий на примере Hyperledger.

Аннотация. Приведена теоретическая информация о тестировании программного обеспечения методом фаззинга. Рассмотрены технологии обучения с подкреплением и интеллектуального фаззинга в процессе тестирования программного обеспечения. Описан алгоритм, с помощью которого реализуются указанные методы и технологии. Предложены статистические результаты исследований, которые были проведены во время тестирования некоторых программ и утилит, предназначенных для повседневного использования, а также программы разработанной студентами.

Ключевые слова: компьютерные сети; децентрализация; Hyperledger Fabric; блокчейн.