ISSN 2223–5620 (Print), ISSN 2411–1554 (Online)
Електромагнітна сумісність та
безпека на залізничному транспорті, 2019, № 17

УДК 656.2+004.75

В. В. МАЛОВІЧКО – к.т.н., доцент, Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, vladimir_35@ukr.net
Р. В. РИБАЛКА – к.т.н., доцент, Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, r.v.rybalka@gmail.com
Н. В. МАЛОВІЧКО – асистент, Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, natali_mvv@i.ua

РОЗРОБКА ДОДАТКОВОГО ЗАХИСТУ АРХІВІВ ПОДІЙ ТА ПОРУШЕНЬ В СИСТЕМАХ МПЦ

Вступ

З розвитком залізничної автоматики все більшого використання набувають системи на сучасній комп'ютерній та мікропроцесорній елементній базі. Такі системи поступово замінюють релейні в станційній автоматиці, диспетчерській централізації, автоматиці на перегонах та переїздах.

Релейні системи, які є найбільш розповсюдженими на залізницях України, мають значні габарити, матеріалоємність, енергоспоживання та потребують істотних експлуатаційних витрат на їх обслуговування. Крім цього системами автоматики на релейній елементній базі не передбачено використання інформаційних технологій для реалізації обміну даними з автоматизованими системами верхніх рівнів, дистанційне телевимірювання параметрів об'єктів контролю, діагностування та прогнозування стану систем і т.д. В зв'язку з цим перехід на сучасну елементну базу всіх систем залізничної автоматики є лише питанням часу.

Однак масове впровадження мікропроцесорних систем на залізничному транспорті стримується крім економічної складової проблемою гарантування безпеки їх використання, яка випливає із неочевидності безпечної поведінки систем і програмного забезпечення та потребує використання нових шляхів підвищення безпеки їх функціонування [1]. На відміну від релейних систем, правильність функціонування яких в основному залежить від правильності проектування, підключення та обслуговування апаратури, в мікропроцесорних системах одною з основних складових безпечного функціонування є стабільність роботи програмного забезпечення та захист його від втручання сторонніх осіб.

В мікропроцесорних системах централізації (МПЦ) захист програмного забезпечення виконано на високому рівні і крім цього у більшості систем МПЦ, що експлуатуються в Україні на рівні автоматизованих робочих місць (АРМ), ведеться архів подій та порушень (ПП), в якому реєструються всі події, що відбуваються в системі, з їх обов'язковою архівацією [2]. В разі виникнення відмови або транспортної події, на базі записів у архіві ПП встановлюються причини події.

На жаль, ефективний захист від несанкціонованого втручання в архіви ПП переважно відсутній. Тому з метою уникнення відповідальності, обслуговуючим персоналом може бути змінено архів ПП. В зв'язку з цим, робота щодо створення додаткового захисту архівів ПП в системах МПЦ є актуальною. В даній роботі пропонується підсистема захисту інформації архівів ПП, яка забезпечує незмінюваність інформації та реєстрацію операцій із записами в архіві ПП.

Метою роботи є створення структури підсистеми захисту інформації в архівах ПП на базі технології блокчейн.

Постановка задачі

Перехід на нову елементну базу станційних систем автоматики призводить до підвищення надійності їх роботи, кращого контролю роботи як самої системи так і об-

слуговуючого персоналу, більш простого обігу документів та ведення електронних журналів. Розглянемо для прикладу зберігання архівів ПП в системі МПЦ-У, яка на даний момент активно впроваджується на залізничних станціях України [3]. Архіви в даній системі, як і в більшості існуючих систем МПЦ, зберігаються в АРМ чергового по станції та в АРМ чергового електромеханіка терміном до 365 днів. У разі несанкціонованої зміни даних в архіві одного з робочих місць виявити який з двох архівів змінено, а який містить достовірну інформацію проблематично. Для унеможливлення несанкціонованої зміни (чи видалення) даних архіву пропонується використати технологію блокчейн.

Стислий огляд технології блокчейн

Блокчейн (в оригіналі – block chain) – розподілена база даних (спільний та повторюваний реєстраційний журнал), що підтримує список записів (блоків – blocks), який постійно збільшується [4]. Блокчейн підтримується множиною вузлів мережі. Кожен вузол зберігає ідентичну копію реєстраційного журналу (ledger), який, як правило, представлено як ланцюг (chain) блоків [5]. Блок – впорядкована множина транзакцій [6]. Кожен блок містить геш-зв'язок з попереднім блоком, що гарантує незмінюваність реєстраційного журналу (РЖ).

Переваги блокчейн полягають у впорядкованості блоків, їх незмінюваності та можливості криптографічної перевірки без єдиної точки «довіри» (яка виносить рішення) на відміну від централізованої бази даних (БД).

Перевірку коректності даних в РЖ може бути виконано в будь-який момент шляхом повторного обчислення геш-зв'язків блоків. Якщо виявлено відмінність між даними блоку та його відповідним гешем, то це свідчить про те, що транзакції некоректні [7].

У технології блокчейн є також ряд недоліків [8]:

– Обчислювальна складність. Блокчейн використовує відносно сильний криптографічний захист, який має досить високі вимоги до обчислювального ресурсу.

– Надмірність. Розподіленість БД обумовлює зберігання всіх транзакцій на всіх вузлах мережі. Розмір місця для зберігання зростає у часі лінійно для одного вузла та геометрично для всієї мережі.

У разі використання даної технології для захисту архівів ПП в системах МПЦ приведені недоліки не мають значного впливу, так як об'єми інформації в архівах порівняно незначні.

Застосування технології блокчейн в МПЦ дозволить унеможливити несанкціоновану зміну (чи видалення) даних з архіву ПП. Ідентичні локальні копії архіву можуть зберігатися в АРМ даної станції, АРМах сусідніх станцій обладнаних МПЦ та в АРМ змінного інженера дистанції. На АРМ змінного інженера дистанції можна покласти функції з адміністрування роботи запропонованої підсистеми і надати винятковий дозвіл на редагування налаштувань архівів, наприклад, на зміну тривалості зберігання даних.

Порівняння платформ блокчейн

Блокчейн має багато реалізацій (повністю функціонуючі та в процесі розроблення). Реалізовану блокчейн платформу часто називають «блокчейн структура» (Blockchain Fabric) [4]. Одна з найбільш відомих – Bitcoin.

В [9] подано гарне порівняння таких реалізацій блокчейн: Bitcoin, Litecoin, Dogecoin, Ethereum, MultiChain, Hyperledger Fabric, Hyperledger Sawtooth, Hyperledger Iroha, Steem, Elements Project, Lisk. Неповний перелік параметрів реалізації блокчейн:

– Призначення. Приклади: криптовалюта (Bitcoin, Litecoin тощо), виконання «розумних контрактів» (Ethereum), створення блокчейн для потреб промисловості

(Hyperledger Fabric), соціальні медіа (Steem).

– Наявність власної криптовалюти (Bitcoin, Litecoin, Ethereum тощо).

– Швидкість транзакцій. Приклади, транзакцій на секунду: Bitcoin – 7 (теоретичний максимум), Litecoin – 28 (теоретичний максимум), Ethereum – теоретично необмежено, Hyperledger Fabric – більше 10 тис.

Вимоги до блокчейн платформи, яку пропонується застосовувати для реєстрації подій та захисту інформації в електронних журналах систем автоматики на мікропроцесорній елементній базі:

– Зберігання записів, об'єм кожного з яких порівняно невеликий.

– Відсутність криптовалюти.

– Швидкість транзакцій повинна забезпечувати можливість оновлювати інформацію в журналі з інтервалом не більше 1 хв. Для порівняння, у Bitcoin завжди витрачається близько 10 хв.

Вказаним вимогам відповідає Hyperledger Fabric (HF), яку розглянуто далі. HF належить до множини бізнес блокчейн середовищ (frameworks), які підтримано Hyperledger. Hyperledger – результат зусиль, створений для просування міжіндустрійних блокчейн технологій, який має відкритий вхідний код [10]. Це глобальне співробітництво, яке підтримано Лінукс Фундацією (Linux Foundation).

Операції читання-запису в Hyperledger Fabric

Hyperledger Fabric – це платформа розподіленого РЖ, з регульованим доступом та відкритим вхідним кодом.

Високорівневу структуру типової взаємодії користувача з HF для двох вузлів (пірів) подано на рис. 1 [11]. Для встановлення, розроблення та функціонування HF обрано операційну систему Ubuntu.

Пір (peer) – елемент мережі, який підтримує РЖ (один або більше) та виконує розумні (смарт) контракти [6]. Мережа блокчейн в основному складається з множини однорангових вузлів (пірів). Пір є хостом (надає різні служби) для РЖ та розумних контрактів, тому додатки повинні взаємодіяти з піром.

Розумний контракт (smart contract) – код встановлений на пірі, може бути викликаний клієнтським додатком, який є зовнішнім відносно мережі блокчейн [6]. В HF мають назву «chaincode».

РЖ у HF складається з «блокчейну» (рис. 2) та «бази даних стану» («світового стану») [11]. Якщо блок потрапив до блокчейн то його не може бути змінено. «Світовий стан» – база даних, яка містить поточне значення множини пар «ключ-значення» як результат всіх успішно зафіксованих (committed) транзакцій в блокчейн. У Hyperledger Fabric v1.4 підтримуються такі БД як levelDB та couchDB.

На рис. 2 подано структуру HF, яка використовує два піри. Основні етапи процесу запису в РЖ у HF (рис. 2) [11]:

1. Додаток отримує доступ до піру та викликає (рис. 2, крок 1) певний розумний контракт для оновлення РЖ.

2. Пір викликає розумний контракт, який створює пропозицію оновлення РЖ (рис. 2, крок 2).

3. Додаток отримує (рис. 2, крок 3) пропоноване оновлення (те яке буде застосовано за умови погодження з іншими пірами).

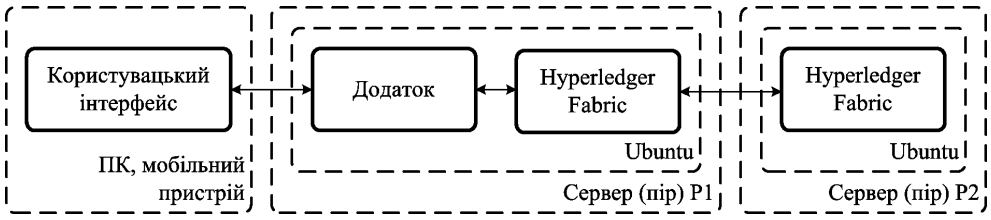

Рис. 1. Структура взаємодії користувача з Hyperledger Fabric

4. Додаток створює транзакцію з пропозицією, яка в загальному надсилається до кожного підтверджуючого піру (є підмножиною всіх пірів мережі). «Замовника» (Orderer) не задіяно (рис. 2, кроки 4, 5).

5. Підтверджуючий(і) пір(и) виконують розумний контракт (використовуючи транзакцією з пропозицією) та повертає підтверджений відгук на транзакцію до додатку (рис. 2, кроки 6, 7).

6. Якщо додаток отримує достатню кількість (критерії визначається політикою підтвердження) підтверджених відгуків від підтверджуючих пірів, то надсилає транзакцію до «замовника» (рис. 2, крок 8).

7. «Замовник» збирає транзакції з мережі у блоки (блок формується після досягнення заданого розміру чи вичерпання часового ліміту) та розподіляє до всіх пірів (рис. 2, крок 9), включно з поточним.

8. Кожен пір незалежно від інших обробляє отриманий блок: перевіряє, що всі транзакції в блоці є підтвердженими. Якщо це так, то транзакції доповнюються до РЖ (рис. 2, крок 10).

9. Після оновлення РЖ, пір генерує подію, яка отримується додатком.

Читання з РЖ не змінює стан РЖ, тому додаток, як правило (поведінку додатку можна змінити) не надає ці транзакції для замовлення, перевірки та фіксації [6]. Цим досягається ефективне читання з РЖ.

Продуктивність Hyperledger Fabric

У HF часова затримка для оновлення БД РЖ залежить від багатьох факторів. Наприклад, якщо швидкість надходження транзакцій 100 транзакцій/с і розмір блоку 100 транзакцій, то часова затримка складає 1 блок/с [5].

На жаль, продуктивність HF для версій, новіших за HF v1.0, не було достатньо досліджено (наприклад, [9, 12]). Очікується, що HF може обробляти транзакції на швидкості, більшій за 10 тис. транзакцій/с (залежить від налаштувань) використовуючи модель досягнення консенсусу BFT [9].

Вказана продуктивність HF є достатньою для застосування в МПЦ з метою унеможливлення несанкціонованої зміни (чи видалення) даних архіву, оскільки інформація з архівів не використовується для оперативного керування рухом поїздів.

Hyperledger Composer

Серед інструментів, які підтримано Hyperledger, є Hyperledger Composer (HC). Hyperledger Composer – це набір інструментів для створення блокчейн мереж, який спрощує та прискорює для розробників створення розумних контрактів та блокчейн додатків [10]. HC підтримує існуючу інфраструктуру та середовище HF.

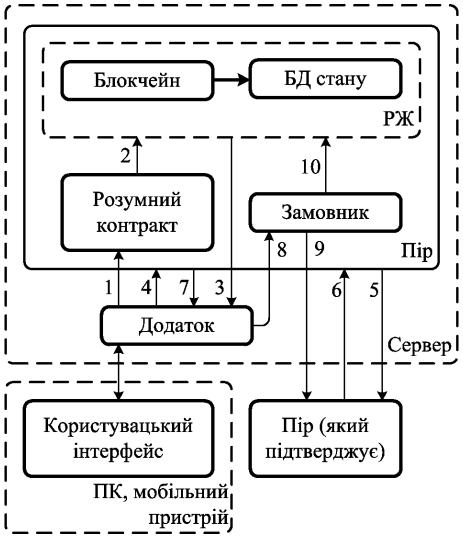

Рис. 2. Процес запису в Hyperledger Fabric

У [13] подано порівняння HF та HC (виконується на HF). Обидва можуть бути використані для реалізації блокчейн рішення. Відмінність полягає у рівнях абстракції, інструментах та мовах програмування.

Для прикладу рішення, розглянутого у [13], вказано, що кількість рядків коду у HF (мова програмування Go) більша за кількість рядків коду у HC (мови програмування Hyperledger Composer Modeling Language, Hyperledger Composer Access Control Language, Hyperledger Composer Query Language, JavaScript) приблизно у 10 разів. Тому використання HC для створення блокчейн рішення дозволить зменшити ризики та складність кодування, шляхом значного

зменшення кількості рядків коду, яке досягнуто використанням більш високого рівня абстракції.

HC містить в собі мову програмування Access Control Language, яка надає декларативний контроль доступу: права учасників щодо доступу до ресурсів (assets) [14]. Це значно зменшує кількість процедурних перевірок контролю доступу у бізнес логіці.

Серед інших переваг HC, які спрощують та прискорюють розроблення блокчейн рішення, є автоматичне створення: REST APIs (Representational State Transfer Application Programming Interface), які надають доступ до блокчейн логіки веб чи мобільним додаткам [14]; скелету Angular веб додатку. Логіку структури взаємодії Hyperledger Composer та Hyperledger Fabric подано на рис. 3.

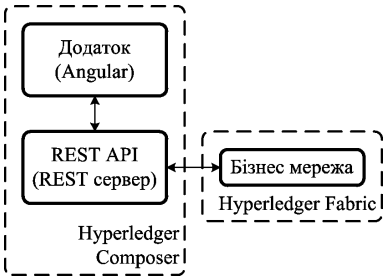

Рис. 3. Структура взаємодії Hyperledger Composer та Hyperledger Fabric

Вказане вище обґрунтовує доцільність використання HC для унеможливлення несанкціонованої зміни (чи видалення) даних архіву ПП.

Особливості функціонування Hyperledger Composer

Визначення мережі у HC описує всі ресурси, транзакції, операції та учасників для даного блокчейн рішення [4]. Воно може бути доступне шляхом використання інтерфейсу командного рядку або API.

Для опису принців функціонування HC необхідно визначити певні поняття [4]. Ресурс (asset) – матеріальна чи нематеріальна сутність. Учасник (participant) – представляє особу (чи організацію), яка бере участь в мережі. Транзакція (transaction) – подається (submit) учасником для впливу на ресурси (створення, оновлення, видалення ресурсів). Ресурси, учасники і транзакції зберігаються у відповідних реєстрах.

Для захисту даних у HC серед іншого використано:

– Регульований доступ: надання ідентичності учаснику та правила контролю доступу.

– Реєстрація всіх успішних транзакцій.

Доступ до HF є регульованим, тобто ідентичність (особа) кожного учасника відома та підтверджена. Це відрізняє регульований доступ від нерегульованого, за якого будь-хто може отримати доступ до системи (архівів ПП) та виконувати транзакції.

Учасник з відповідними правами (адміністратор) надає ідентичність (можливо, на обмежений інтервал часу) певному учаснику, яка використовується для взаємодії з мережею. HC в якості документів ідентичності використовує реєстраційні сертифікати HF. Ідентичність (identity) – унікальний ідентифікатор, асоційований з учасником [4].

Всі успішні транзакції, включно з інформацією щодо дати-часу створення, учасників та ідентичностей, які їх подали (ініціювали), записуються у спеціальний реєстр – Hyperledger Composer Historian. Він зберігає транзакції як ресурси типу HistorianRecord, причому всі учасники повинні мати дозвіл на створення цих ресурсів [14]. В іншому випадку транзакцію не буде виконано.

Правила контролю доступу дозволяють встановлювати права та обмеження для учасників [14]. Приклад: оператор АРМ ШН не може переглядати записи операторів з інших станцій при використанні технології для групи станцій на мікропроцесорній елементній базі. У HC використано мову HC Access Control Language для оголошення подібних правил доступу.

Функції оброблення транзакцій призначено для реалізації транзакцій, визначених в оголошенні мережі HC [15]. В цих функціях дозволено використання API більш низького рівня абстракції (рівня HF). Дані API обходять правила контролю доступу в HC,

що не допустимо при використанні НС для захисту архівів ПП в МПЦ. Тому розробнику програмних засобів потрібно використовувати HF API з обережністю та дотриманням відповідної політики розроблення.

Опис додаткового захисту архівів подій в системах МПЦ

Для унеможливлення несанкціонованої зміни (чи видалення) даних архіву МПЦ пропонується використати НС. Це вимагає внесення певних змін в існуючу систему МПЦ в частині реєстрації подій. На високому рівні абстракції вказані зміни можна подати так:

– Дані щодо доповнення (зміни) записів у архіві ПП повинні надходити у РЖ НС. Збереження даних в існуючій БД МПЦ не обов'язкове.

– Отримання даних з архіву ПП виконується з РЖ НС.

Перелік учасників та відповідних правил контролю доступу в НС залежить від виду системи автоматики та її структури (МПЦ для однієї станції, для групи станцій, МСДЦ тощо). Наприклад, «супер адміністратор» може створювати картки доступу для «адміністраторів» та «користувачів», переглядати записи в архіві тощо, але не має права видаляти записи з архіві.

Недоліком існуючих АРМ системи МПЦ є відсутність реалізації автоматичного оперативного інформування певних користувачів про події певного класу. Наприклад, інформування представників служби сигналізації і зв'язку регіональних філій АТ «Укрзалізниця» в разі виникнення певних відмов.

У HF для інформування певних додатків (певних користувачів) можна використати «події». «Події» (events) визначаються під час визначенні мережі, після чого можуть бути включені у функції оброблення транзакцій [4]. «Подія» утворюється лише після успішного завершення транзакції. НС надає можливість генерації «подій», на отримання яких через composer-client API підписано зовнішні додатки [14].

Висновки

Для унеможливлення несанкціонованої зміни (чи видалення) даних архіву МПЦ запропоновано використати технологію блокчейн. Обгрунтовано вибір реалізації цієї технології (HF) та використання НС як інструменту, що підтримує існуючу інфраструктуру та середовище HF. Це вимагає внесення певних змін в існуючу систему МПЦ в частині реєстрації подій.

В результаті зберігання архіву ПП з використання НС буде досягнуто:

– незмінюваність записів в РЖ: унеможливлення несанкціонованої зміни (чи видалення) даних архіву;

– реєстрація дій операторів АРМ: визначення ідентичності користувача, дати, часу та виконаної дії (транзакції);

– оперативне інформування зацікавлених користувачів системи МПЦ про події певного класу, наприклад, інформування представників служби сигналізації і зв'язку регіональних філій АТ «Укрзалізниця» в разі виникнення певних відмов.

В перспективі аналогічну технологію також можна запровадити для мікропроцесорних систем диспетчерської централізації (МСДЦ). Наприклад, в системі МСДЦ «КАСКАД» при використанні запропонованої технології архіви ПП зберігатимуться на сервері БД центрального поста [16], в АРМ поїзного диспетчера, енергодиспетчера та інженера СЦБ і зв'язку.

Це дасть змогу виявити та зафіксувати втручання в архів ПП і унеможливити його зміну. Крім цього через доступ до БД АСОУП, який є в системі «КАСКАД», зберігається можливість (як для систем МПЦ) автоматичного оперативного інформування обслуговуючого персоналу вищих рівнів ієрархії про виникнення відмов та подій певного класу із заданою періодичністю.

Бібліографічний список

1. Бойнік, А. Б. Шляхи підвищення безпеки функціонування станційних мікропроцесорних систем залізничної автоматики / А. Б. Бойнік, В. І. Мойсеєнко // Залізничний транспорт України. – 2010. – № 4. – С. 42–46.

2. Сапожников, Вл. В. Микропроцессорные системы централизации. / Вл. В. Сапожников и др. – М.: ГОУ «Учебно-методический центр по образованию на железнодорожном транспорте», 2008. – 398 с.
3. Басов, В. І. Мікропроцесорна система централізації МПЦ-У: навчальний посібник для студентів вузів залізничного транспорту. / В. І. Басов, В. В. Єлисєєв, О. В. Петренко, та ін. – К., 2014. – 430 с.
4. Hyperledger Composer. Glossary and definition of terms [Electronic resource] / Available at: https://hyperledger.github.io/composer/latest/reference/glossary.
5. Thakkar, P. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform [Virtual resource] / Parth Thakkar, Senthil Nathan N, Balaji Viswanathan. Available at: Thakkar.Performance Benchmarking and Optimizing_Hyperledger Fabric Blockchain Platform.pdf.
6. Hyperledger Fabric. Glossary [Electronic resource] / Available at: https://hyperledger-fabric.readthedocs.io/en/release-1.4/glossary.html.
7. Doubleday, K. Why Blockchain Immutability Matters / K. Doubleday. – 2018 Режим доступу: https://hackernoon.com/why-blockchain-immutability-matters-8ce86603914e.
8. Goeringer, S. A Simple Overview of Blockchains. Why They Are Important to the Cable Industry. A Technical Paper prepared for SCTE/ISBE [Virtual resource] / S. Goeringer / 2017 Fall Technical Forum. Available at: https://www.nctatechnicalpapers.com/Paper/2017/2017-a-simple-overview-of-blockchains-why-they-are-important-to-the-cable-industry/download.
9. Hintzman, Z. Comparing Blockchain Implementations. A Technical Paper prepared for SCTE/ISBE [Virtual resource] / Z. Hintzman / 2017 Fall Technical Forum. Available at: 2017-comparing-blockchain-implementations.pdf.
10. The Linux Foundation projects. About Hyperledger [Electronic resource] / Available at: https://www.hyperledger.org/about.
11. Hyperledger Fabric. Key Concepts. Peers [Electronic resource] / Available at: https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html.
12. Nasir, Q. Performance Analysis of Hyperledger Fabric Platforms [Virtual resource] / Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, and Ali Bou Nassif. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjqib3r_Z7gAhUKBywKHUmdDEkQFjACegQIBBAC&url=http%3A%2F%2Fdownloads.hindawi.com%2Fjournals%2Fscn%2F2018%2F3976093.pdf&usg=AOvVaw1vDeFBF_sgxm_N_nE-86TQ.
13. Getting Started with Blockchain Development. Buid Blockchain applications and business networks your way [Electronic resource] / Available at: https://blog.selman.org/2017/07/08/getting-started-with-blockchain-development/.
14. Key Concepts in Hyperledger Composer [Electronic resource] / Available at: https://hyperledger.github.io/composer/latest/introduction/key-concepts.
15. Transaction Processor Functions [Electronic resource] / Available at: https://hyperledger.github.io/composer/latest/reference/js_scripts.
16. Данько, М. І. Мікропроцесорна диспетчерська централізація «КАСКАД» / М. І. Данько, В. І. Мойсеєнко, В. З. Рахматов та ін: Навч. посібник. — Харків, 2005. – 176 с.

Ключові слова: мікропроцесорна централізація, автоматизовані робочі місця, архів подій та порушень, блокчейн, Hyperledger Fabric, Hyperledger Composer.

Ключевые слова: микропроцессорная централизация, автоматизированные рабочие места, архив событий и нарушений, блокчейн, Hyperledger Fabric, Hyperledger Composer.

Keywords: microcomputer interlocking, computer-aided workstations, archive of events and failures, blockchain, Hyperledger Fabric, Hyperledger Composer.

Рецензенти:
проф., д.т.н., А. Б. Бойник,
проф., д.ф.-м.н., В. І. Гарилюк.

Надійшла до редколегії 21.05.2019.
Прийнята до друку 27.05.2019.