

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми забезпечення конфіденційності транзакцій
криптовалют»

Виконав студент 4-го курсу групи КБ-2
спеціальності 125 «Кібербезпека»

 Владислав Макарчук

Керівник, доцент кафедри КІ та КБ, к.т.н.
 Олександр Пірог

Рецензент, в.о. завідувача кафедри КН, к.т.н.

 Марина Граф

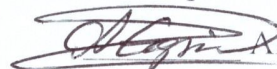
Житомир – 2022

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

1 березня 2022 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Макарчук Владислав Валерійович

Тема роботи: «Проект підсистеми забезпечення конфіденційності транзакцій криптовалют» затверджена Наказом університету від 1 березня 2022 р. № 128/с

Термін здачі студентом закінченої роботи 25 червня 2022 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

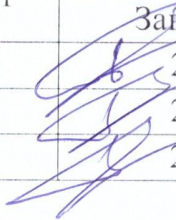
Об'єктом дослідження є підсистеми забезпечення конфіденційності транзакцій криптовалют з використання технологій Python, BlockChain, Monero, Telegram.

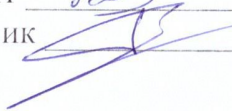
Предметом дослідження є технології проектування підсистеми забезпечення конфіденційності транзакцій криптовалют з використання технологій Python, BlockChain, Monero, Telegram.

Обладнання та програмне забезпечення:

1. Мова програмування Python;
2. Monero RPC Server;
3. Фреймворк PyTelegramBotApi;
4. Windows 10;

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пірог О.В.	 20.04.2022	 20.04.2022
2	Пірог О.В.	 20.04.2022	 20.04.2022
3	Пірог О.В.	 20.04.2022	 20.04.2022

Студент  Владислав Макарчук
Керівник  Олександр Пірог

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	01.03.2022	виконано
2	Розробка плану роботи	30.04.2022	виконано
3	Розробка розгорнутого плану	05.05.2022	виконано
4	Подача чернетки керівнику для перевірки	07.05.2022	виконано
5	Доопрацювання роботи та перевірка керівником	09.05.2022	виконано
6	Затвердження графічних матеріалів у керівника	12.05.2022	виконано
7	Проведення попереднього захисту	18.06.2022	виконано
8	Отримання відгуку керівника	21.06.2022	виконано
9	Допуск завідувача кафедри	21.06.2022	виконано
10	Подання роботи на рецензування	21.06.2022	виконано

Студент  Владислав Макарчук

Керівник  Олександр Пірог

АНОТАЦІЯ

В рамках випускної кваліфікаційної роботи розроблено сервіс на тему «Проект підсистеми захисту конфіденційності криптовалютних операцій».

Пояснювальна записка до випускної роботи містить 48 сторінок і 8 рисунків. Список використаних джерел містить 11 найменувань і займає 1 сторінку.

В даний час технологія блокчейн займає провідне місце серед технічних рішень, які використовуються в найрізноманітніших сферах нашого життя, наприклад, при захисті авторських прав або в електронному голосуванні. Цей протокол створено для вирішення прикладних проблем розподілених і децентралізованих надійних обчислень. Тому метою даної дипломної роботи є аналіз блокчейну та окремих його аспектів – підходів і методів забезпечення конфіденційності даних користувачів.

Метою роботи є розробка проекту підсистеми забезпечення конфіденційності операцій з криптовалютою.

У в кваліфікаційній роботі обґрунтовано актуальність захисту конфіденційності криптовалютних транзакцій. Проаналізовано можливі загрози та наведено можливі засоби та методи захисту від них.

Ключові слова: БЛОКЧЕЙН, КОНФІДЕНЦІЙНІСТЬ, БІТКОЙН, MONERO, HYPERLEDGER.

					КІК.КР.Б – 125 – 11 – ПЗ			
Ім'я	Арк.	№ докум.	Підпис	Дата				
Розроб.		Макаричук В.В.		28.06	ПРОЕКТ ПІДСИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ ТРАНЗАКЦІЙ КРИПТОВАЛЮТ	Літ.	Арк.	Аркушів
Керівник		Пірог О.В.		28.06			5	48
Реценз.		Граф М.С.		28.06		ФІКТ, Гр. КБ-2		
Н. контр.								
Затверд.		Сфіменко А.А.		28.06				

ANNOTATION

As part of the final qualification work, a service was designed on the topic "Project of a subsystem for the protection of confidentiality of cryptocurrency transactions".

The explanatory note to the graduation thesis contains 48 pages and 8 figures. The list of used sources contains 11 names and occupies 1 page.

Currently, blockchain technology occupies a leading place among technical solutions used in the most diverse areas of our lives, for example, in the protection of copyrights or in electronic voting. This protocol is built to solve applied problems on distributed and decentralized trusted computing. Therefore, the purpose of this thesis is to analyze the blockchain and its individual aspects - approaches and methods of ensuring user data privacy.

The purpose of the work is to develop a project of a subsystem for ensuring the confidentiality of cryptocurrency transactions.

In the qualification work, the relevance of protecting the confidentiality of cryptocurrency transactions is substantiated. Possible threats are analyzed and possible means and methods of protection against them are given.

Keywords: BLOCKCHAIN, PRIVACY, BITCOIN, MONERO, HYPERLEDGER.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. РОЛЬ БЕЗПЕКИ ДАНИХ КОРИСТУВАЧА.....	10
1.1 РОЛЬ БЕЗПЕКИ ТА КОНФІДЕНЦІЙНОСТІ.....	10
1.2 АНАЛІЗ МОЖЛИВИХ ЗАГРОЗ, ЯКІ ЗООСЕРЕДЖЕНІ НАВКРУГИ ОБ’ЄКТА ДОСЛІДЖЕННЯ	12
1.3. ЗАСОБИ ТА МЕТОДИ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ ДАНИХ	18
РОЗДІЛ 2. ПОНЯТТЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ В ПРОТОКОЛІ БЛОКЧЕЙН	23
2.1 РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ	23
2.2. ЗАГАЛЬНА ПРОБЛЕМАТИКА ТА ІСНУЮЧІ РІШЕННЯ.....	23
2.3 HYPERLEDGER SAWTOOTH.....	26
2.4. АРХІТЕКТУРА ПРИВАТНОГО БЛОКЧЕЙНУ.....	28
2.5. ПОРІВНЯННЯ ВЛАСТИВОСТЕЙ ПУБЛІЧНИХ ТА ПРИВАТНИХ БЛОКЧЕЙНІВ	29
2.6. ДОСЛІДЖЕННЯ РОБОТИ БЛОКЧЕЙНУ В ПОЄДНАНІ З МЕРЕЖОЮ LIGHTNING	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ПІДСИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ ТРАНЗАКЦІЙ КРИПТОВАЛЮТ	40
3.1. СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ	40
3.2. ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ	45
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ.....	48

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

I2P - Invisible Internet Project

ECDHM - Elliptic-Curve Diffie–Hellman-Merkle

BTC - Bitcoin

GUI - Graphical User Interface

TEE - Trusted Execution Environment

SGX - Software Guard Extensions

DKSAP - Dual-Key Stealth Address

ProtocolSDK - Software Development Kit

P2P - Peer-to-Peer network

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

В наш час інформаційні технології розвиваються з надзвичайною швидкістю і здебільшого люди навіть не замислюються яку цінність несе в собі інформація, яка циркулює в кібернетичному просторі. Саме через людську необізнаність ця інформація стає основною ціллю кіберзлочинців. За допомогою інформації, яку було отримано у разі нападу, злочинці можуть заволодіти вашими логінами, паролями, документами, коштами та інше.

Кожного дня, люди використовують різні види техніки (комп'ютери, ноутбуки, планшети, телефони, тощо), які мають доступ до інтернет середовища та інтенсивно переглядають різні види інформації, тим самим вже надаючи певні дані про себе. Адже, веб-сайти збирають конфіденційну інформацію так філігранно, що 98% користувачів, навіть, не знає, що саме про вас знають власники того чи іншого веб-ресурсу. Переходячи за посиланням до будь-якого інтернет ресурсу, ми даємо згоду на обробку даних та даних нашого комп'ютера або іншої техніки. Реєструючись на сайті користувач надає про себе ще більше інформації, яка в подальшому може бути використана проти нього самого. Зі сторони ресурсу на якому проходить реєстрація, це вид безпеки самого ресурсу та інформації на ньому. Проте, все ж слід бути уважним переходячи за посиланням, заповнюючи форми з даними про себе та звернути увагу на те, як в подальшому ці дані можуть оброблятися.

Сучасна технологія блокчейн розвивається дуже швидко та має чимало модифікацій і версій. Проте в ній наявні деякі суттєві недоліки, що так і не знайшли однозначного рішення, наприклад, стосовно приватності даних в мережі. Саме тому дана тема є дуже актуальною для вивчення.

Метою дослідження є аналіз блокчейну та окремих його аспектів та методів забезпечення приватності даних користувачів системи.

Практичне значення результатів полягає у покращенні забезпечення приватності даних користувачів та їх анонімності в технологіях на основі блокчейну, підвищення рівня децентралізації та удосконалення інших

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

характеристик даних систем. Як наслідок, збільшиться коло застосування блокчейну та знайдуться нові форми і сфери для його реалізації.

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. РОЛЬ БЕЗПЕКИ ДАНИХ КОРИСТУВАЧА

1.1 Роль безпеки та конфіденційності

Захист конфіденційності інформації та збереження анонімності в сучасному інформаційному просторі є одним з найбільш актуальних питань кібербезпеки сьогодення. Протягом історії людства способи розв'язання цієї проблеми визначались саме рівнем розвитку технологій. У сучасному інформаційному суспільстві технологія відіграє роль як активатора так і деактиватора цієї проблеми, в залежності від рівня обізнаності користувача.

Комп'ютерними називають злочини, пов'язані з будь-яким несанкціонованим втручанням у роботу інформаційно-комп'ютерного середовища людини, і злочини, в яких комп'ютери використовуються як необхідні технічні засоби.

Серед причин комп'ютерних злочинів і пов'язаних з ними викрадень інформації головними є такі:

- швидкий перехід від традиційної паперової технології зберігання та передавання інформації до електронної за одночасного відставання технологій захисту інформації, зафіксованої на машинних носіях;
- широке використання локальних обчислювальних мереж, створення глобальних мереж і розширення доступу до інформаційних ресурсів;
- постійне ускладнення програмних засобів, що викликає зменшення їх надійності та збільшення кількості уразливих місць.

Сьогодні ніхто не може назвати точну цифру загальних збитків від комп'ютерних злочинів, але експерти погоджуються, що відповідні суми вимірюються мільярдами доларів. Серед основних статей варто виокремити такі:

- збитки, до яких призводить ситуація, коли співробітники організації не можуть виконувати свої обов'язки через непрацездатність системи (мережі);
- вартість викрадених і скомпрометованих даних;
- витрати на відновлення роботи системи, на перевірку її цілісності, на доробку уразливих місць тощо
- викрадені фіатні та цифрові активи користувачів

Варто також враховувати й морально-психологічні наслідки для користувачів, персоналу і власників ІС та інформації. Що ж до порушення безпеки

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

так званих «критичних» додатків у державному і військовому управлінні, атомній енергетиці, медицині, ракетно-космічній галузі та у фінансовій сфері, то воно може призвести до тяжких наслідків для навколишнього середовища, економіки і безпеки держави, здоров'я і навіть для життя людей.

Економічні та юридичні питання, приватна та комерційна таємниця, національна безпека — усе це зумовлює необхідність захисту інформації та ІС.

Останнім часом спеціалісти використовують кілька понять для визначення безпеки інформації в інформаційних системах. Такими поняттями є *захист інформації в комп'ютерних системах, захист інформації в комп'ютерних системах і мережах, захист інформації в автоматизованих системах, захист інформації в інформаційно-комунікаційних системах*. Згадані поняття часто використовують як синоніми, але слід зазначити, що в останні роки вони зазнали певної еволюції.

Ще кілька років тому більш поширеним було поняття *захист інформації в комп'ютерних системах і мережах*, а в офіційних документах перевагу надавали поняттю *захист інформації в автоматизованих системах*.

Зараз загальноприйнятим в Україні є поняття *захист інформації в інформаційно-комунікаційних системах*.

Інформаційно-комунікаційною системою (ІКС) називають організаційно-технічну систему, яка виконує функції інформаційної системи, тобто такої організаційно-технічної системи, що реалізує певну технологію (або сукупність технологій) оброблення інформації, та (або) комунікаційної системи — технічної системи, що реалізує певну технологію (або сукупність технологій) передавання даних шляхом їх кодування у формі фізичних сигналів.

Далі ми зосередимо увагу на термінах, безпосередньо пов'язаних із питаннями захисту інформації в ІКС яка зосередженна навкруги рядового користувача. Перше питання, яке постає, — що саме ми захищаємо, і яка мета цього захисту. Навіть побіжно розглядаючи цю тему, можна переконатися, що захист інформації як такої, без чіткого визначення завдань захисту не має сенсу.

Метою захисту інформації має бути збереження конфіденційності і недорканості даних без наданих на те привілеїв власника, збереження фіатних та цифрових активів користувача, забезпечення права на анонімність, запобігання втручання сторонніх осіб в приватне життя власника. Виходячи з цього, безпосередні заходи захисту спрямовують не так на самі інформаційні ресурси, як

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

на збереження певних технологій їх створення, оброблення, зберігання, пошуку та надання користувачам. Ці технології мають ураховувати особливості інформації, які й роблять її цінною, а також давати змогу користувачам різних категорій працювати з інформаційними ресурсами (створювати, знаходити, копіювати, узагальнювати, порівнювати, модифікувати, перетворювати, знищувати тощо).

Головною метою дослідження є забезпечити безпеку приватної інформації людини та надати їй можливість без ризиків використовувати весь спектр можливостей, які надає сучасний світ ІКТ.

Безпека інформації — це стан інформації, в якому забезпечується збереження визначених політикою безпеки властивостей інформації. Багаторічний досвід захисту інформації в ІКС дозволив визначити головні властивості інформації, збереження яких дає змогу гарантувати збереження цінності інформаційних ресурсів. Це конфіденційність, цілісність і доступність інформації.

Конфіденційність — властивість інформації, Завдяки якій лише вповноважені користувачі мають змогу її отримувати (тобто ознайомлюватися з інформацією).

Цілісність — властивість інформації, яка дає можливість лише вповноваженим користувачам її модифікувати.

Доступність — властивість інформації, завдяки якій уповноважені користувачі можуть використовувати її згідно з правилами, встановленими політикою безпеки, не очікуючи довше заданого (невеликого) проміжку часу. Тобто інформаційний ресурс має необхідний користувачу вигляд, знаходиться в тому місці, де це потрібно користувачу, і тоді, коли це йому потрібно.

Термін *доступність* вживають не лише, коли йдеться про інформаційні ресурси, але й до ІКС у цілому, до її компонентів або окремих ресурсів. Наприклад, коректно говорити про доступність сервера, сегмента мережі, служби електронної пошти тощо.

1.2 Аналіз можливих загроз, які зоосереджені навкруги об'єкта дослідження

Жоден з наявних на ринку ІКТ продуктів, на даний момент, не є цілком безпечним і не може забезпечити в повному обсязі потребу користувача зберегти недоторканість своїх даних та приватного життя. Але проаналізувавши велику кількість доступних для кожного системних утиліт та методик захисту

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

конфідесійності даних та анонімності користувача, можна стверджувати що використовуючи наявні рішення можливо забезпечити необхідний рівень безпеки для кожного.

Розгляд інформаційної безпеки людини необхідно розпочинати з інтерфейсу, який зв'язує її з широкими можливостями онлайн та оффлайн всесвіту. На сьогодні одним із найполярніших є Windows який встановлений на більше ніж 1млд одиницях комп'ютерної техніки.

Операційна система Windows не є в повній мірі безпечною. Не зважаючи на свою популярність вона має ряд вразливостей, та недоліків. Одним із основних недоліків якраз і є її популярність. Мільйони людей використовують цю ОС і саме тому вона викликає особливу увагу зловмисників, що постійно намагаються знайти в ній вразливості і створюють для неї купу шкідливого програмного забезпечення.

Windows має закритий код, що не дає змоги виправити якусь проблему чи недолік самотушки і в тей же час користувачі мають повний доступ адміністратора до облікових записів. Таким чином, при виникненні якоїсь загрози, в зоні ризику опиняється вся система.

Розглянемо більш детально, що може спричинити порушення безпеки інформації в мережі та проти чого, власне, застосовують заходи захисту інформації.

Несприятливий вплив — вплив, що призводить до зменшення цінності інформаційних ресурсів.

Загроза — будь-які обставини чи події, що можуть спричинити порушення політики безпеки інформації та (або) нанесення збитку мережі. Тобто загроза - це будь-який потенційно можливий несприятливий вплив.

Загроз, що чекають користувачів при підключенні комп'ютера до мережі, досить багато. Це лише основні з них:

- *Взлом комп'ютера* зазвичай проводиться з метою захоплення контролю над операційною системою і отримання доступу до даних;

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

- *Пошкодження системи* найчастіше організовується, щоб порушити працездатність (викликати відмову в обслуговуванні – «Denial of Service») будь-яких сервісів або комп'ютера (найчастіше сервера) цілком, а іноді – навіть всієї мережевої інфраструктури організації;

- *Крадіжка даних* через неправильно встановлені права доступу, при передачі даних або зламі системи дозволяє отримати доступ до захищеної, часто – конфіденційної інформації з усіма впливаючими звідси неприємними для власника цих даних наслідками;

- *Знищення даних* має на меті порушити або навіть паралізувати роботу систем, комп'ютерів, серверів або всієї організації.

- Потрапляння в інформаційну систему *шкідливого програмного забезпечення*: вірусів, троянських програм, мережевих хробаків, клавіатурних шпигунів, рекламних систем, що є актуальною проблемою для ОС Windows.

- *Атаки порушників* (хакерів).

- *BotNet* — це комп'ютерна мережа, що складається з деякої кількості хостів, із запущеними ботами — автономним програмним забезпеченням.

- *DdoS* — атака на відмову в обслуговуванні, розподілена атака на відмову в обслуговуванні — напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена.

- *Фішинг* — вид шахрайства, метою якого є виманювання персональних даних у клієнтів онлайн-аукціонів, сервісів з переказу або обміну валюти, інтернет-магазинів тощо.

Більш детально про *шкідливе програмне забезпечення*:

- *Комп'ютерний вірус* — це вид шкідливого програмного забезпечення, яке здатне поширювати свої копії з метою інфікування та пошкодження даних на

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

пристрої жертви. Віруси можуть потрапити на комп'ютер з інших вже інфікованих пристроїв, через носії інформації (CD, DVD тощо) або через Інтернет-мережу.

- *Троянська програма.* Типова троянська програма зазвичай «маскується» під будь-яку корисну утиліту (або може бути захована в якій-небудь програмі), а якщо користувач через незнання запустить її на виконання, така програма починає контролювати комп'ютер, відкриваючи автору троянської програми доступ до даних (так званий «backdoor» — «чорний хід»), викрадаючи та пересилаючи йому набрані з клавіатури паролі та ін.

- *Мережевий хробак* — це програма зі шкідливим кодом, яка атакує комп'ютери в мережі та поширюється через неї. Активний мережевий черв'як може знижувати продуктивність пристрою жертви, видаляти файли або навіть вимикати певні програми. Якщо на комп'ютері було виявлено загрозу такого типу, інфіковані файли рекомендується негайно видалити, оскільки вони можуть містити шкідливий код.

- *Клавіатурні шпигуни* — це програми, що реєструють натиснення клавіш на комп'ютері. Принцип їх дії простий — все натискання на клавіші перехоплюються програмою, і отримані дані записуються в окремий файл, який далі може бути відісланий з мережі на комп'ютер зломщика.

- *Рекламні системи* — це вид шкідливого програмного забезпечення, яке відображає не бажану рекламу на пристрої користувача. Дана загроза, як правило, інсталується під час завантаження легітимних додатків. Хоча іноді рекламне програмне забезпечення може бути безпечним, деякі спливаючі вікна здатні не тільки відображати рекламу, але й збирати дані про конкретного користувача для відображення оголошень відповідно до його запитів у браузері. У цих випадках рекламне ПЗ може перенаправляти користувача на шкідливі веб-сайти та

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

небезпечні сторінки через різні посилання, піддаючи ризику інфікування комп'ютерними вірусами та іншими загрозами.

Для смартфонів характерні ті самі загрози, що і для стаціонарних комп'ютерів: віруси, троянські програми, мережеві хробаки, рекламні модулі тощо, орієнтовані на різні типи мобільних пристроїв.

Всі вище перераховані терміни можна узагальнити терміном *атака*.

Атака — це спроба реалізації загрози. Якщо атака є успішною (здійснено подолання засобів захисту), це називають *проникненням*. Наслідком успішної атаки є порушення безпеки інформації в системі, яке називають *компрометацією*.

Слід звернути увагу на те, що за комплексного підходу до захисту інформації ми маємо розглядати не лише впливи, спрямовані на інформаційні ресурси, але й будь-які впливи, що можуть завдати шкоди мережі. Це вже було узагальнено твердженням про необхідність захисту не самої інформації, а насамперед технології її оброблення.

Вади захисту — сукупність причин, умов і обставин, наявність яких може призвести до порушення нормального функціонування системи або політики безпеки інформації. Здебільшого під вадами захисту розуміють особливості побудови програмних (а іноді й апаратних) засобів захисту, що за певних обставин спричиняють їхню нездатність протистояти загрозам і виконувати свої функції. Тобто вади захисту є окремим випадком уразливості системи.

У літературі іноді використовують інше тлумачення цих термінів, що не є коректним. Наприклад, часто замість терміну *загроза* вживають термін *атака*. Однак потрібно розрізняти атаку, яка є дією, тобто спробою реалізувати певну загрозу, та загрозу, яка робить потенційно можливим здійснення несприятливого впливу. Атака — це здебільшого цілеспрямований вплив, як правило, умисний. Загрози можуть бути випадковими, хоча втрати від цього не стають меншими. Тому захищати інформацію потрібно також від загроз, а не лише від атак.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Порушник — фізична особа (необов'язково користувач системи), яка порушує політику безпеки системи. Іноді використовують термін *зловмисник*, чим наголошують умисність здійсненого ним порушення, тоді як порушник може здійснювати порушення ненавмисно (наприклад, через необережність або недостатню обізнаність).

Часто вживаний термін *хакер* є доволі неоднозначним, тому його не використовують як синонім терміну *порушник*.

Модель загроз — абстрактний формалізований чи неформалізований опис методів і засобів здійснення загроз.

Модель порушника — абстрактний формалізований чи неформалізований опис порушника. Моделі загроз і порушника є вихідною інформацією для розроблення політики безпеки і проектування будь-яких систем захисту.

Захищена комп'ютерна система — комп'ютерна система, що здатна забезпечувати захист оброблюваної інформації від визначених загроз. Цей термін частіше вживають до обчислювальних систем або їхніх складових (програмних продуктів, окремих програмно-апаратних пристроїв). Іноді його застосовують до мереж, але слід розуміти, що будь-яка сучасна мережа має бути захищеною (навіть домашній комп'ютер із одним користувачем). Інакше її використання дуже швидко призведе до втрат інформації.

Спостережність — властивість мережі, що дає змогу фіксувати діяльність користувачів і процесів, використання пасивних об'єктів, а також однозначно встановлювати ідентифікатори причетних до певних подій користувачів і процесів із метою запобігання порушенню політики безпеки і (або) забезпеченню відповідальності за певні дії. Це дуже важлива властивість обчислювальних систем та мереж, яка досягається реалізацією засобів *реєстрації*, або *аудита*.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3. Засоби та методи захисту конфіденційності даних

У часи, коли комп'ютери не були об'єднані в мережі або підключені до Інтернету, про безпеку даних можна було особливо не піклуватися. Досить було забезпечити *фізичний захист* комп'ютера і контролювати доступ сторонніх користувачів до пристроїв запису (наприклад, до дисководів).

Після об'єднання комп'ютерів в мережі все змінилося – без серйозного захисту тепер уже не обійтися, інакше і операційна система, і дані, що зберігаються на комп'ютері або передаються по мережі, можуть стати легкою здобиччю зломисників, причому так, що користувачі, які працюють на цьому комп'ютері нічого не помітять.

Під *захистом інформації в мережі* розуміють діяльність, спрямовану на забезпечення безпеки оброблюваної в мережі інформації й мережі у цілому, що дає змогу запобігти реалізації загроз або унеможливити її, та зменшити ймовірність завдання збитків від реалізації загроз.

Захист інформації в мережі полягає у створенні системи технічних (інженерних, програмно-апаратних) і нетехнічних (правових, організаційних) заходів та в підтримці її роботоздатного стану.

Систему таких заходів називають комплексною системою захисту інформації.

Відтак *комплексна система захисту інформації (КСЗІ)* — це сукупність організаційних, інженерних і програмно-апаратних засобів, що забезпечують захист інформації в мережі.

Отже, захист інформації в мережі формально зводиться до створення і супроводу КСЗІ. Слід зазначити, що домашній комп'ютер з одним користувачем також має мати якусь КСЗІ, щоправда, недокументовану. Операційна система обов'язково має засоби контролю цілісності компонентів самої ОС і файлової системи. Найчастіше користувачем встановлюється антивірусний засіб і хоча б зрідка виконуються дії з оновлення його бази даних (або просто видаляється

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

застаріла програма і встановлюється нова). Час від часу ним робляться резервні копії своїх найцінніших файлів.

Зрештою, якщо користувач має вихід в Інтернет, він мусить вживати додаткових заходів безпеки.

Потрібно добре знати про те, що підключення до Інтернету є найкритичнішим із міркувань безпеки системи. Якщо без такого підключення користувач може працювати роками, не застосовуючи специфічних заходів безпеки, то з виходом в Інтернет йому, фактично, необхідно створити КСЗІ. Крім антивірусного ПЗ до складу цієї системи мають входити настроєні певним чином засоби міжмережної фільтрації (наприклад, персональний брандмауер), реєстрації подій, а, можливо, і виявлення вторгнень. Потрібно також періодично вживати заходів, на кшталт оновлення програмного забезпечення, встановлення виправлень, оновлення антивірусних баз тощо. Не менш необхідним фактором для успішного функціонування КСЗІ є обізнаність самого користувача. Йому необхідно знати хоча б базові принципи виявлення фішингу та спаму, навчитись приховувати свою особистість в мережі.

Що стосується мереж, які використовують у державних органах і установах, підприємствах різної форми власності, то для них створення КСЗІ є життєво необхідним. У багатьох випадках обов'язковість створення КСЗІ визначається чинним законодавством.

Основні заходи безпеки при роботі в мережі досить прості. Їх можна сформулювати у вигляді наступного набору правил:

- Вимикайте комп'ютер, коли ви ним не користуєтесь.
- Своєчасно оновлюйте операційну систему. У будь-якій ОС періодично виявляються так звані «уразливості», що знижують захищеність вашого комп'ютера. Наявність уразливостей потрібно уважно відстежувати (у тому числі, читаючи «комп'ютерну» пресу або інформацію в Інтернеті), щоб вчасно вживати заходів для їх усунення.

- Ретельно аналізуйте посилання за якими ви переходите
- Використовуйте лише офіційне ПЗ з підримкою від видавця

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Знати та дотримуватись основних принципів збереження(конфіденційності в мережі

- За можливості використовувати додаткове апаратне забезпечення, яке навіть при фізичному втручанні заблокує доступ зловмисну до інформації

- Розмежування обов'язків пристроїв(за можливості)

- Замінити використання стільниковий дзвінків на дзвінки за допомогою спеціально ПЗ, яке підтримує наскрізне шифрування

Для ОС Windows корпорацією Microsoft створено спеціальний веб-вузол Windows Update, звернувшись до якого (наприклад, за допомогою програми WUPDMGR.EXE або команди *Windows Update* в меню *Пуск*), неважко переглянути та завантажити список оновлень, рекомендованих для вашого комп'ютера;

- використовуйте обмежений набір добре перевірених додатків, не встановлюйте самі і не дозволяйте іншим встановлювати на ваш комп'ютер програми, взяті з неперевірених джерел (особливо з Інтернету). Якщо додаток більше не потрібен, видаліть його;

- без необхідності не надавайте ресурси свого комп'ютера у загальний доступ. Якщо ж це все-таки було потрібно, обов'язково налаштуйте мінімально необхідний рівень доступу до ресурсу тільки для зареєстрованих облікових записів;

- встановіть (або включіть) на комп'ютері *персональний міжмережевий екран (брандмауер)*. Якщо мова йде про корпоративні мережі, встановіть брандмауери як на маршрутизаторах, що з'єднують вашу локальну мережу з Інтернетом, так та на всіх комп'ютерах мережі;

- обов'язково встановіть на комп'ютер спеціалізоване антивірусне і «антишпигунське» програмне забезпечення. Налаштуйте його на автоматичне отримання оновлень як мінімум один раз на тиждень (краще – щодня або навіть кілька разів на день);

- навіть якщо ви єдиний власник комп'ютера, для звичайної роботи застосовуйте *користувацький обліковий запис*: у цьому випадку пошкодження системи, наприклад, при зараженні вірусом, буде незмірно менше, ніж якщо б ви

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

працювали з правами адміністратора. Для всіх облікових записів, особливо адміністративних, встановіть та запам'ятайте складні паролі.

Складним вважається пароль, що містить випадкову комбінацію літер, цифр та спеціальних символів, наприклад jxglrg\$N. Зрозуміло, пароль не повинен збігатися з ім'ям вашого профілю. В операційних системах Windows складний пароль можна згенерувати автоматично, використовуючи команду NET USER з ключем /RANDOM, наприклад:

- NET USER Ім'я_Користувача /RANDOM

Також необхідно пам'ятати, що будь-який пароль можливо взламати і для його ефективності пароль необхідно оновлювати як можна частіше.

Паролі для доступу у різні системи мають бути різними. Неприпустимо використовувати один і той самий пароль для адміністрування вашого комп'ютера і для входу, наприклад, на ігровий веб-сайт;

Для того щоб зберегти конфіденційність своїх паролів, використовується додаткове ПЗ таке як KeePass Password Safe, Password Safe та інші. Додаткове ПЗ рекомендовано використовувати для збереження паролів користувачів, оскільки випасані на папірці паролі, теж я вразливістю вашої системи, яка може призвести до втрати даних.

- при роботі з електронною поштою ніколи відразу не відкривайте вкладення, особливо отримані від невідомих відправників. Збережіть вкладення на диск, перевірте його антивірусною програмою і лише потім відкрийте. Якщо є така можливість, увімкніть у вашій поштовій програмі захист від потенційно небезпечного вмісту і відключіть підтримку HTML; Для особливого маскуванню при створенні поштової скриньки часто використовує браузер TOR, який мінімілізує деанонімізацію вашої особистості. Також корисною практикою, при роботі з поштовими скриньками, буде використання поштовий сервісів, які підтримують наскрізне шифрування. Таким чином користувач мінімізує ризики врати конфіденційних даних, не даючи зловмисникам можливості отримати доступ

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

по даних. Менш розповсюдженою практикою є самостійне шифрування листів, яке знижує ризик того, що ці дані може отримати 3-я особа, зводиться майже до нуля.

- при роботі з веб-сайтами дотримуйтесь заходів розумної обережності: намагайтеся уникати реєстрації, не передавайте нікому персональні відомості про себе і уважно працюйте з Інтернет-магазинами та іншими службами, де застосовуються онлайнові способи оплати за допомогою кредитних карт або систем типу WebMoney. При проведенні оплати переконайтеся, що з'єднання захищено шифруванням за допомогою технології *Secure Sockets Layer (SSL)* – у цьому випадку адресний рядок обов'язково повинен починатися з «https://»;

- перераховані вище заходи лише підвищують загальну захищеність системи і даних, але не дають ніякої гарантії від їх пошкодження або навіть повної втрати. Тому обов'язково слід створювати резервні копії системи і даних на знімному жорсткому диску або на DVD-RW – це дозволить вам легко відновити їх у разі втрати. При цьому одну копію має сенс зберігати не вдома, наприклад, в сейфі;

- під час резервного копіювання корисно використовувати утиліти для створення «образів» жорсткого диска (такі, як Norton Ghost). Резервну копію можна зняти з «системного» жорсткого диска після правильної установки на нього всіх необхідних програм і антивірусної перевірки та зберігати її на іншому жорсткому диску (мережевому або знімному), щоб у разі пошкодження системи швидко відновити її працездатність.

- виключно важливу роль грає навчання всіх користувачів основам безпечної роботи в мережах – як у домашніх, так і в корпоративних, - адже порушення правил одним користувачем ставить під загрозу всю систему захисту.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

РОЗДІЛ 2. ПОНЯТТЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ В ПРОТОКОЛІ БЛОКЧЕЙН

2.1 Розробка технічного завдання

Немає сумніву, що технологія блокчейн має величезний потенціал. Децентралізовані біржі, ринки прогнозування та платформи для управління активами – все це лише частина з існуючих цікавих додатків, які вивчаються блокчейн-розробниками.

Але попри загальновідомі переваги, блокчейн має кілька основних технічних бар'єрів, які роблять його непрактичними для застосування. Перерахуємо основні з них:

- обмежена масштабованість
- обмежена конфіденційність
- обмеження зберігання даних
- відсутність засобів управління та стандартів
- недостатнє оснащення

2.2. Загальна проблематика та існуючі рішення

Захист інформації є однією з серйозних та важливих проблем. Протягом усієї історії людства шляхи вирішення цієї проблеми повністю залежали від рівня розвитку технологій. У нашому інформаційному суспільстві технологія відіграє роль активатора цієї проблеми – комп'ютерні злочини стали візитівкою сьогодення.

Враховуючи, що транзакції в блокчейні безпосередньо не прив'язані до користувача, вони, на перший погляд, можуть здаватися достатньо приватними. Будь-хто має можливість анонімно створити новий електронний гаманець та з його допомогою здійснювати транзакції. Однак варто приділити цьому процесу та його реалізації трохи більше уваги, щоб мати ширше уявлення про нього.

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

З одного боку, безумовно вірно, що велика перевага технології блокчейн – анонімність. Транзакції записуються та зберігаються в публічній книзі обліку, а також пов'язані з адресою облікового запису, що складається виключно з цифр і букв. Якщо до цієї адреси не додано жодної ідентичності, визначити відправника транзакції здається неможливим. Проте подібне визначення повної безпеки може бути досить оманливим. Насправді, користувач зберігає свою приватність допоки його псевдонім не пов'язаний з його ідентичністю, але як тільки хтось виявить цей зв'язок – таємниця викривається. Один приклад такого явища був оприлюднений, коли правоохоронні органи визнали, що під час одного з своїх розслідувань були в змозі ідентифікувати конкретних користувачів біткойну, і таким чином "де-анонімізували" їх та порушили загальну передумову транзакційної невидимості. Веб-трекери та "cookie"-файли на веб-сайтах дозволяють витік інформації про транзакцію в інтернет мережі, де будь-хто, включаючи уряд, правоохоронні органи та зловмисників, з готовністю використовують ці дані.

Крім того, використовуючи такі блокчейн платформами, як Ethereum, користувачі взаємодіють зі смарт-контрактами, які обробляють більше даних, ніж у звичайних грошових переказах. Всі деталі зазначених смарт-контрактів загальнодоступні в блокчейні Ethereum, включаючи відправників, одержувачів, дані про транзакції, виконаний код і поточний стан мережі. Тому завантаження важливих бізнес-даних у блокчейн, де хакери, конкуренти або інші неавторизовані сторони можуть переглядати інформацію, просто не є можливим рішенням для більшості компаній. Розглянемо деякі приклади таких приватних даних:

1. *Електронні медичні записи*
2. *Дані перевірки ідентичності*
3. *Управління обліковими записами*
4. *Фінансові документи*

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Отож конфіденційність залишається основною перешкодою для окремих осіб, організацій і галузей, які піклуються про приватність і індивідуальний суверенітет

Інший спосіб забезпечення конфіденційності – це створення криптовалюти, яка є приватною за замовчуванням. Прикладом такої технології є Монеро ("Монеро"). На відміну від багатьох інших альтернатив, Монеро не є форком від біткойну. Натомість ця криптовалюта базується на альтернативному протоколі CryptoNote, що вимагає забезпечення двох властивостей, а саме неспроможності відстежити користувача та пов'язати його особу з певною транзакцією. Тому, щоб виконати ці вимоги, Монеро пропонує використання схеми "кільцевого підпису" та генерування одноразових "невидимих адрес" для кожної транзакції. Кільцеві підписи – це тип групового підпису, де кожен учасник має секретний і відкритий ключ. На відміну від традиційних криптографічних підписів, які доводять, що транзакція була "схвалена" одним підписувачем, використовуючи згаданий приватний ключ, підпис групи підтверджує, що один підписувач з фіксованої групи схвалив транзакцію, не розкриваючи його ідентичності.

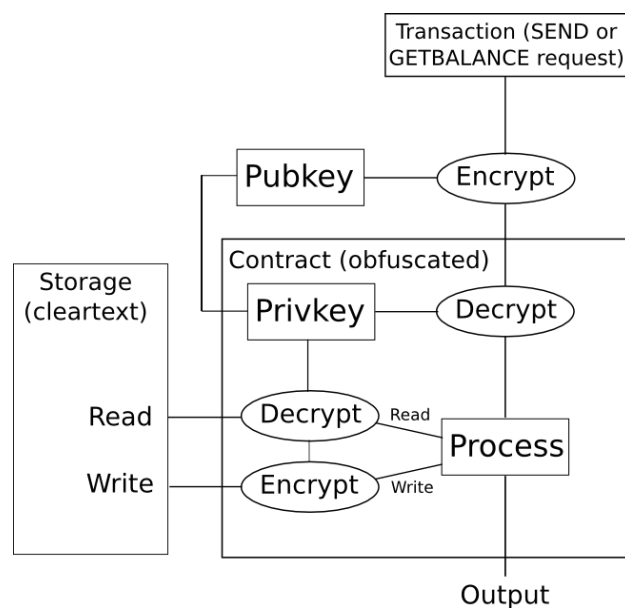


Рисунок 2.1 – Схема роботи смарт-контракту з заплутувачем коду

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

Ідентичності користувачів в блокчейн мережах приховані за допомогою використання різних криптографічних примітивів. Це робить, на перший погляд, неможливим визначити особу власника рахунку, що є однією з головних причин великої поширеності цієї технології. Однак слід зазначити, що криптовалюти на основі блокчейну не є повністю приватними. Оскільки всі дані транзакцій фіксуються, включаючи відправника, одержувача та суму обміну, й, крім того, загальнодоступні – їх легко зчитати та дослідити.

Наразі існує велика кількість компаній, що аналізують різні аспекти діяльності клієнтів криптовалют, і на основі своїх результатів, в окремих випадках, можуть встановити особу користувача. Тож, як наслідок, з плином часу застосовується все більше й більше підходів та криптографічних методів для забезпечення приватності користувачів. У даному розділі було розглянуто основні з них. Особливу увагу приділено задачам, що вони виконують, та які недоліки мають у контексті технології блокчейн. Даний аналіз допоміг сформулювати головні вимоги до захисту анонімності користувачів блокчейну та збереження конфіденційності даних їх транзакцій.

2.3 Hyperledger Sawtooth

Hyperledger Sawtooth – це бізнес-проект з відкритим кодом, побудований на блокчейні. Він був створений організацією Intel Corporation в об'єднанні з R3cev (творцями Corda). Це платформа для розподілених рішень з модульною архітектурою, що забезпечує високу ступінь конфіденційності, надійності, гнучкості та масштабованості. Він був розроблений для підтримки різних реалізацій його компонент і враховує тонкощі та складності, які існують в економічній екосистемі.

Головні характеристики протоколу Hyperledger Sawtooth:

1. дозволяє додати різні типи консенсусу в одну й ту саму блокчейн мережу;
2. за умовчанням наявний такий механізм консенсусу як "Доказ вичерпаного часу" (PoET)

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

3. смарт-контракти можна писати практично на будь-якій мові програмування (Python, JavaScript, Go, C ++, Java і Rust);
4. включає підтримку смарт-контрактів Ethereum через інтеграцію з Hyperledger Burrow;
5. може реалізувати мережу як з обмеженим, так і необмеженим доступом з боку користувачів;
6. має модульну архітектуру;
7. високо масштабований;
8. транзакції можуть за необхідності виконуватися паралельно, даючи таким чином більшу загальну продуктивність.

Основна мета дизайну Sawtooth спрямована на збереження розподілених записів користувачів та безпечне використання смарт-контрактів, особливо для великих компаній. Sawtooth спрощує розробку додатків на основі блокчейну, відокремлюючи основну систему від домену проекту. Розробники додатків можуть вказувати бізнес-правила, які підходять саме у їх випадку, використовуючи мову програмування на їх розсуд, без необхідності знання базового дизайну основної системи.

Проект клієнта Sawtooth може базуватися на вбудованій бізнес-логіці на віртуальній машині зі смарт-контрактами. Насправді, обидва типи додатків можуть співіснувати в одному й тому ж блокчейні. Sawtooth дає змогу реалізувати ці архітектурні рішення на етапі обробки транзакцій. Отож, всі вимоги клієнта визначаються у спеціальному процесорі транзакцій для їх виконання.

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

2.4. Архітектура приватного блокчейну

Одна з цілей технології Sawtooth полягає в розподілі спільних записів про транзакції між учасниками системи. Можливість забезпечити консистентне копіювання даних на всіх вузлах у візантійському консенсусі є однією з основних переваг технології блокчейн.

Sawtooth представляє собою стан для всіх сімей транзакцій в одному екземплярі дерева Merkle Radix, вбудованого в кожний спеціальний "вузол-валідатор". Процес перевірки блоку у валідаторі гарантує, що однакові транзакції призводять до однакових переходів стану і що отримані дані однакові для всіх учасників мережі.

Стан розбивається на простори імен, що дозволяє творцям сімей транзакцій гнучко визначати, обмінюватися та повторно використовувати дані глобального стану між процесорами транзакцій. Зміни до стану додаються шляхом створення та виконання транзакцій. Клієнт створює транзакцію і передає її валідатору, а він, у свою чергу, застосовує транзакцію, яка призводить до зміни стану. Загальна структура пакетів і транзакцій, зображених на Рисунку 2.1 включає в себе поля *Batch* ("пакет"), *BatchHeader* ("заголовок пакету"), *Transaction* ("транзакція"), and *TransactionHeader* ("заголовок транзакції").

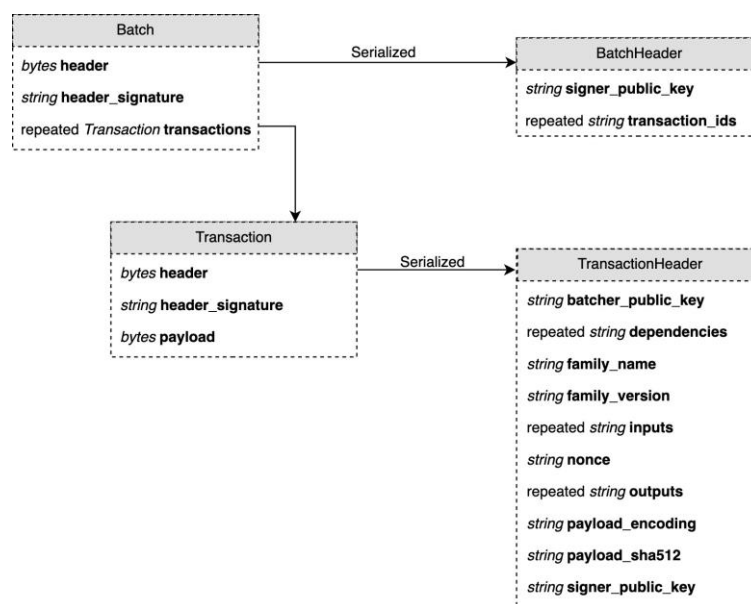


Рисунок 2.1 – Структура пакетів і транзакцій платформи Sawtooth

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5. Порівняння властивостей публічних та приватних блокчейнів

Ethereum – багатоманітний термін, адже він може означати клас технологій, що побудовані на базі віртуальної машини Ethereum ("Ethereum Virtual Machine", EVM), або мережу блокчейн, керовану проектом Ethereum. Дана мережа призначена бути відкритою та загальнодоступною. Це означає, що кожен вузол Ethereum відомий будь-якому іншому вузлу мережі. Таким чином, усі вузли мають копію однакових впорядкованих даних. Відносини між вузлами повністю децентралізовані, тобто немає головного учасника системи, який керує іншими – кожен вузол просто інформує спільноту мережі про типи своїх транзакцій та порядок їх виконання.

Кожен користувач мережі Ethereum взаємодіє з іншими через смарт-контракти, які повинні бути в нього "вбудовані". Процес інсталяції смарт-контракту подібний до того, як учасники мережі Ethereum ініціюють транзакції. Наприклад, автор контракту криптографічно підписує та передає його через свій вузол всім іншим учасникам. Контракт зберігається в деяких адресних сховищах даних мережі. Слід наголосити, що блоки в блокчейні не є сховищем даних – вони містять тільки опис функцій, які будуть викликатися в смарт-контрактах.

Hyperledger Fabric належить до класу блокчейнів з обмеженим доступом. У мережі Fabric вузли не спілкуються один з одним відкрито. Натомість, учасники системи, як правило, деякі бізнес-організації, щоб обмінюватися повідомленнями, повинні узгодити політики видачі дозволів своїм вузлам. Краще розглядати транзакції між сторонами як операції між бізнес-організаціями. Кожна організація може володіти одним або декількома вузлами. За допомогою Fabric вони налаштовуються для виконання конкретних завдань, таких як створення клієнтського вузла для взаємодії з кінцевими користувачами тощо.

Починаючи з версії Hyperledger Fabric 1.0, формування дозволів обміну повідомленнями між вузлами ґрунтується на використанні сертифікатів X.509.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

У цій моделі всі організації-учасники погоджуються використовувати загальний кореневий сертифікат, такий, який видається центром сертифікації Verisign, щоб перевірити, чи дозволено сторонам здійснювати операції один з одним.

Hyperledger Burrow EVM. Він дає змогу виконувати смарт-контракти Ethereum, написані на мові програмування Solidity. В Sawtooth є еквівалент майнера або вузла замовлення – "валідатор". Його завдання полягає в тому, щоб перевірити та переконатися, що транзакція (або повідомлення, що викликає виконання транзакції) має дійсний підпис. Потім він направляє її до відповідного процесора транзакцій, де реалізована бізнес-логіка, що відповідає за правильність обробки транзакції. Коли процесор транзакції завершує перевірку своєї бізнес-логіки, повідомлення надсилається назад валідатору, який потім відповідає за упаковку транзакції в блок і додавання його до блокчейну, а також за його передачу іншим валідаторам, щоб переконатися, що консенсус досягнутий. На додачу до групування транзакцій в блоки, він також відповідальний за оновлення глобального стану мережі, тобто сховища даних на основі дерева Merkle.

У мережі Sawtooth може існувати не один, а багато валідаторів. Так само, як і майнер в Ethereum, тільки один валідатор може оновлювати блок. Проте замість PoW, Sawtooth робить це за допомогою алгоритму PoET.

Важливою особливістю Sawtooth є його здатність групувати транзакції і формувати так звані пакети, які згодом можуть бути додані до блоку. Таким чином, Sawtooth спроможний обробляти складні угоди. Уявімо ситуацію, де Джон продає свою машину Паулу, що може включати ряд пов'язаних між собою операцій: обмін грошима, передача власності та страхового полісу тощо. Щоб спростити обробку цих даних, можна налаштувати процесори транзакцій спеціальним чином. Проте всі ці операції мають бути або успішними або невдалими у сукупності. Іншими словами, об'єднані транзакції повинні відповідати принципу атомарності, консистентності, ізоляції та тривалої дії (ACID). Також можна налаштувати операції, пов'язані з передачею права

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

власності та страхового полісу, щоб вони виконувалися паралельно з операцією, пов'язаною з обміном грошима, яка повинна виконуватися тільки після того, як перші дві були успішно завершені. А коли обмін грошей здійснюється, весь пакет транзакцій задовольняє принципам ACID.

Нарешті, щоб налаштувати Sawtooth як публічну мережу або чи мати обмеження в плані доступу, потрібно маніпулювати процесорами транзакцій, а саме налаштувати маршрутизацію трафіку від клієнтів відповідно їх вимог: приймати транзакції від будь-якого клієнта в мережі (випадок з загальнодоступним блокчейном) або приймати повідомлення лише від тих клієнтів, яким це дозволено (тобто блокчейн з обмеженим доступом).

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

2.6. Дослідження роботи блокчейну в поєднанні з мережою lightning

Блокчейн – прогресивна технологія, що набуває великої поширеності, звичайно, все більшого списку вимог. До найкритичніших на сьогоднішні недоліків даного протоколу можна віднести неспроможність до масштабованості та прогалини в збереженні конфіденційності даних всередині системи. Тож, була запропонована нова децентралізована розподілена система – ”Lightning Network” (далі позначимо це поняття як ”мережа Lightning”), яка має на меті мінімізувати перераховані недоліки та удосконалити стандартний протокол блокчейну в цілому. Дана технологія може змінити спосіб ведення бізнесу між людьми. Вона відкриває світ мікро-транзакцій, а також допомагає в реалізації додаткових традиційних бізнес-моделей, що раніше не мали змоги використовувати блокчейн через повільність його роботи, викриття деталей кожної транзакції загалом тощо. У цьому розділі детально досліджено механізми та підходи в мережі Lightning, як саме вона доповнює роботу протоколу блокчейн та які переваги при цьому отримують користувачі.

Багато хто вважає, що біткойн забезпечує своїм користувачам анонімність їх операцій, проте експерти криптовалют знають, що насправді це не так. Блокчейн біткойну це загальнодоступна прозора книга обліку, що ніколи не видаляє стару інформацію. Тим не менш, дотримання вимог та безпека стали все більш важливим питанням у галузі криптовалют. Закони про боротьбу з відмиванням грошей (AML) та регуляції щодо принципу ”знай про свого клієнта” (KYC) вимагають обміну даних для моніторингу депозитів та зняття коштів, перевірки походження та призначення капіталів клієнтів. Тож, було створено новий підхід до грошового обміну між користувачами блокчейну, що вперше був реалізований саме в біткойні – мережу Lightning.

Ключова відмінність, яку приносить використання мережі Lightning, як доповнення до блокчейну, полягає в тому, що не вся інформація обміну коштами між користувачами доступна усім учасникам системи. Канали в мережі Lightning синхронізуються лише між її вузлами, тому робота цієї

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

технології швидша та ефективніша, ніж в традиційному блокчейні. Крім того, це дає більш високий ступінь конфіденційності інформації, що передається в системі. Дані каналу відомі тільки двом задіяним вузлам мережі. Якщо звичайна транзакція біткойну подібна до завантаження виписки з банку на загальнодоступний веб-сайт, то транзакція мережі Lightning більш схожа на показ конкретному продавцю лише тієї частини вашого гаманця, з якої йому сплачуються кошти. Таким чином, учасники все ще обмінюються інформацією, але її обсяги набагато менші.

Протокол біткойну базується на блокчейні, але саме через це має кілька ключових бар'єрів, які обмежують його функціональність:

Використання блокчейну для проведення транзакцій може бути досить дорогим. Оскільки в ньому обробляється все більше й більше транзакцій, він стає перевантаженим, і користувачі повинні платити вищі збори, щоб підтвердити свою транзакцію в наступному блоці.

Він відносно повільний. У випадку з біткойном підтвердження транзакції займає приблизно годину.

Дані транзакції в блокчейні відомі всій мережі. А тому, кожен користувач має уявлення про всі фінансові операції інших користувачів, що не задовольняє вимогам багатьох бізнес-моделей сьогодення.

Мережа Lightning – це нова технологія, що працює в кооперації з блокчейном, та є прикладом протоколу "другого шару". Перший шар – це блокчейн, другий – мережа Lightning. Слід зазначити, що крім біткойну дана технологія може працювати й на блокчейнах інших криптовалют. Тож, перерахуємо, які переваги приносить мережа Lightning в поєднанні з блокчейном:

Додає можливість миттєвих операцій. Оскільки перевірка більше не повинна відбуватися у блокчейні, немає необхідності чекати, поки угода буде підтверджена. Це означає, що платіж відбувається миттєво. Дана перевага дозволяє активізувати обмін коштами більш "спонтанно".

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Дозволяє мікро-ціноутворення. Оскільки миттєві мікро-платежі відбуваються поза основним блокчейном, користувачам більше не потрібно платити за їх підтвердження. Ціноутворення може бути збільшене в одиницях, наприклад, таких малих як 0,000000001 BTC. Це відкриває нову можливість раніше недоступних мікро-операцій.

Покращує збереження конфіденційності даних. На відміну від блокчейну, мережа Lightning надає більшу фінансову конфіденційність. Тепер користувачі не можуть зчитувати всі транзакції іншого користувача.

Розглянемо основні компоненти даної технології, а саме поняття вузлів, каналів, рахунків та багатьох інших, більш детально.

Вузол мережі Lightning має два обов'язки: моніторинг основного блокчейну та взаємодія з іншими вузлами мережі Lightning, щоб здійснювати транзакції. Кожен вузол мережі повинен контролювати один чи декілька блокчейнів, на яких він знаходиться. Окремо відзначимо, що мережа Lightning може працювати поверх кількох блокчейнів.

Протокол вимагає, щоб вузол мережі контролював основний блокчейн, в якому він тримає токени. Якщо вузол не робить це належним чином, кошти користувача можуть бути викрадені. Також кожен учасник взаємодії з іншими вузлами в одноранговій мережі (P2P), де проходить передача коштів від одного вузла іншому через так звані "канали". Кожен вузол відповідає за відстеження того, хто й що має в каналі. І якщо користувач хоче вийти з каналу, він виводить свої кошти назад в блокчейн.

Тепер розглянемо, як саме вузол мережі Lightning відрізняється від вузла у блокчейні біткойну. Найбільша відмінність полягає в тому, що вузол біткойну повинен перевіряти кожну транзакцію в мережі блокчейн, в той час як вузол Lightning верифікує лише дійсність тих транзакцій, з якими він безпосередньо взаємодіє. Тож другий підхід більш здатний до масштабування, а також надає більшу міру конфіденційності.

Рахунок в мережі Lightning – це спосіб отримання платежів у мережі. Він подібний до адреси в біткойні, але з деякими ключовими відмінностями.

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

Приклад рахунку Lightning: "lntblu1pvjluezpp5qqqsyqcyq5rqwzqfqqqsyqcyq5rqwzqfqqqsyqcyq5rqwzqfqpqddq5xysxxatsyp3k7enxv4jsxqzpuaztrnwngzn3kdzw5hydlzf03qdg2hdq27cq3agm2awhz5se903vruatfhq77w3ls4evs3ch9zw97j25emudupq63nyw24cg27h2rspfj9srp8". Він утворений з двох складових: частина доступна для читання користувачами та частина даних. Ці компоненти розділені останнім знайденим у рахунку символом "1".

У нашому прикладі "читабельною" частиною є "lntblu". Перша складова "lntb", вказує, в якій саме криптовалютній мережі валідний цей рахунок, у цьому випадку вона ідентифікує тестову мережу біткойну. Друга складова "lu" – це сума коштів, закодована в рахунку ("lu" рівний одному мікро-біткойну або 0,000001 біткойну).

Тепер перейдемо до частини мета-даних рахунку: "pwz5w78pp5e8w8cr5c30xzw92v36sk45znhjn098rtc4pea6ertnmvu25ng3sdpywd6hyetyvf5hgueqv3jk6meqd9h8vmmfvdjsxqrrsy29mzkzjf27u67evzu893heqex737dhcapvcuantkztg6pnk77nrm72y7z0rs47wzc09vcnugk2ve6sr2ewvcrtqnh3yttv847qqvqpvv398x".

Вона може містити таку інформацію, як:

Час, коли було створено рахунок, що дає можливість визначити, наскільки він старий.

Ідентифікатор вузла, який отримує платіж.

Терміну дії рахунку. На відміну від біткойн-адрес, рахунки мережі Lightning можуть стати не дійсними через деякий час.

Опис даного рахунку. Це може бути довільна строка з корисною інформацією про платіж.

Резервна адреса біткойну. Якщо платіж завершується з помилкою у мережі Lightning, то можна перейти до звичайної транзакції в блокчейні біткойну.

Маршрут для здійснення платежу. Потрібно пам'ятати, що мережа Lightning відрізняється від блокчейну в тому, що необхідно знайти маршрут для проведення платежу. Одержувач платежу може запропонувати такий маршрут.

Крім того, рахунки захищені цифровими підписами. Це означає, що якщо хтось змінить рахунок в мережі Lightning, підпис буде визнано недійсним. Даний

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

фактор досить критичний, оскільки в рахунку задована конфіденційна інформація, наприклад, ідентифікатор вузла, геш оплати та запасна біткойн-адреса. Якщо користувачам вдалося би змінити ці значення, вони мали б можливість вкрати кошти інших учасників.

Загалом, рахунки мережі Lightning є надзвичайно гнучкими та корисними. Вони надають набагато більше мета-інформації про оплату, ніж традиційна адреса біткойну. Це дозволяє користувачам мережі Lightning бути більш впевненими в тому, коли, де і як в мережі обробляється їх платіж.

Наступним поняттям в огляді архітектури технології є "канали" мережі Lightning. Саме вони дозволяють користувачам відправляти та отримувати гроші від інших вузлів мережі. Їх можна порівняти з підключеними один до одного грошовими потоками. Всі такі канали пов'язують лише два вузли мережі Lightning. Для наочності та зручності назовемо їх Алісою і Бобом. Канал має інформацію про біткойн-баланс як Аліси, так і Боба, та відстежує, скільки коштів є у кожного з них.

Один з користувачів – Аліса або Боб – можуть відкрити канал Lightning, створивши спеціальну біткойн-транзакцію, яка призначена для блокування коштів на блокчейні та подальшого їх розблокування у мережі Lightning. Цей підхід гарантує, що гроші не будуть витратитися у кількох місцях одночасно. Для користувачів блокчейну транзакція виглядає так само, як звичайна для мережі біткойн, тобто без реальних відмінностей, за винятком декількох технічних деталей. Слід зауважити, що все це робиться автоматично за допомогою гаманця біткойну, і звичайним користувачам не потрібно знати усі подробиці розробки. Коли створюється канал Lightning, учасник повинен визначити, скільки коштів він хоче в ньому мати. Наприклад, Аліса хоче зробити транзакцію у блокчейні, що дала б їй доступ до 100000 сатоші в мережі Lightning. Дана операція передбачає блокування такої ж суми у блокчейні біткойну та її розблокування в мережі Lightning.

Після створення каналу Lightning, його власники можуть відправляти та отримувати гроші через нього. Якщо Боб хоче надіслати Алісі 1000 сатоші (0.00001 біткойну), він потребує її рахунок. Використовуючи його, він може надіслати Алісі

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

платіж, збільшивши її баланс на 1000 сатоші і зменшивши свій на ту ж суму. Обидві сторони каналу відслідковують баланс один одного, переконавшись, що цифри сходяться після обміну коштами.

Як біткойн, так і мережа Lightning розроблені таким чином, щоб зловмисники не отримували ніякої вигоди. Якщо хтось намагається укласти недійсну угоду в мережі Lightning, наприклад, стверджувати, що деякі з грошей Аліси насправді належать Бобу, то є вбудований механізм покарання зловмисника, де жертва атаки може забрати у злочинця свої гроші. Більш детально цей механізм оглянемо трохи нижче.

Перейдемо до операції закриття каналу. Щоб це зробити, також необхідно створити звичайну транзакцію у блокчейні біткойну. Ця транзакція, у порівнянні зі створенням каналу, виконує зворотні дії, видаляючи кошти з мережі Lightning та розблокуючи їх в основному блокчейні. Канали мережі Lightning є джерелом її живлення, дозволяючи вузлам обмінюватися даними і вільно платити поміж собою. Без них вузли мережі знаходилися б окремо та не робили нічого корисного. Разом вони формують фінансову супермагістраль майбутнього, що дозволяє радикально поширювати нові дані та бізнес-моделі.

Також обговоримо, які існують засоби маршрутизації в контексті мережі Lightning. Як вже згадувалося, вона не була б такою практичною навіть потрібною, якщо учаснику треба було підключатися до кожного вузла в мережі, якому він хотів або відправити кошти, або отримати їх. Однією з основних особливостей та нововведень, що була закладена в специфікації мережі Lightning, є платіжні канали, які зв'язані разом, щоб дати змогу здійснювати платежі без необхідності безпосереднього з'єднання всіх учасників.

Відзначимо, що в каналі з двома вузлами обидва учасники підписують всі обміни коштами, до яких долучаються, тому що вони залучені в кожен змін стану мережі, враховуючи випадок, коли вихід з Lightning одного з них призводить до закриття спільного каналу. З цієї причини в даній системі працює схема мульти-підпису, при якому обидва учасники підписують одне й те ж повідомлення m , що представляє собою остаточний стан балансу кожного

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

користувача після виконання чергової операції. Протокол мережі Lightning повинен враховувати вірогідність викиду учасників з каналу на будь-якому етапі роботи системи. Більше того, учасники мережі підписують і ділять між собою частину транзакції, щоб кожен користувач за потреби міг згодом реконструювати її деталі. Це вимагає криптографічної схеми на основі агрегованих підписів. За визначенням Д. Бонеха [42], *схема агрегованого підпису* є цифровою схемою підпису з додатковою властивістю, яка полягає в тому, що послідовність підписів $@_1, \dots, @_n$ деяких повідомлень m_i з використанням відкритого ключа pk_i можна закодувати в єдину компактну сукупність підписів $@$, що перевіряє те, що m_i було підписано з pk_i для всіх $i = 1, \dots, n$. Під час процесу верифікації перевіряються всі входи $(pk_1, m_1), \dots, (pk_n, m_n)$ та приймаються чи відхиляються також в сукупності.

Для того, щоб у мережі Lightning маршрутизація працювала належним чином, необхідно знайти канали з достатньою ліквідністю. Розглянемо детальніше, що саме це означає. Повертаючись до нашого прикладу: якщо Керол хоче платити Алісі через Боба, то очевидно, що Керол має достатньо грошей у гаманці. Менш очевидним є факт того, що Боб також повинен мати певну суму грошей, в даному випадку у своєму каналі з Алісою. Без цих коштів він не зможе передати платіж Алісі. Тож, усі транзакції в мережі Lightning потребують побудови шляхів через об'єднання каналів з кількістю грошей, достатньою для завершення платежу. В якості компенсації за забезпечення ліквідності каналів вузли можуть вимагати плату за перенаправлення платежів. Ці збори, як правило, дуже малі, за порядком одноцифрового сатоші. Нагадаємо, що 1 біткойн дорівнює 100 мільйонам сатоші. Однак, якщо вузол добре зв'язаний і маршрутизує багато платежів, потенційно, оператор вузла може заробити невелику кількість грошей на його утриманні.

Маршрутизація дійсно є особливістю мережі Lightning. Концепція платіжних каналів існувала довгий час, але можливість децентралізованої і

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

безпечної маршрутизації через кілька сторін стала проривом порівняно з попередніми технологічними ітераціями.

Існують деякі ключові відмінності між тарифами в блокчейні і грошовими зборами за маршрутизацію платежів в мережі Lightning. Кожен раз, коли транзакція створюється в блокчейні, потрібно сплатити сервісний збір майнерам. Це компенсація за їх роботу, в результаті якої забезпечується безпека та валідність усієї мережі. Щоразу, коли майнер виявляє блок з набором транзакцій, він отримує винагороду за операційні витрати відповідно до характеристик блоку. Завдяки структурі блокчейну транзакції назавжди архівуються в системі. Це ще одна річ, за яку платять користувачі, коли транзакція перевірена та додана до блокчейну

база даних, що не підвладна редагуванню та доводить, що транзакція справді сталася. Ресурси обмеженої кількості, що забезпечують роботу блокчейну, – це електроенергія та файловий простір. Мережа Lightning поглинає інший дефіцитний ресурс – капітал. Для можливості здійснення транзакції в мережі Lightning, потрібно мати доступні кошти, щоб направити платежі до місця їх призначення. Якщо у користувача немає достатньої ліквідності в маршрутах, то надіслати платіж неможливо.

Мережа Lightning сконфігурована таким чином, що компенсує своїм учасникам можливість забезпечувати ліквідність мережі. Якщо ви є вузлом, який хотів би здійснювати маршрутні платежі, то можете встановити плату, яку інші вузли повинні враховувати, щоб використовувати свій капітал для маршрутизації.

		Макаруч В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ПІДСИСТЕМИ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ ТРАНЗАКЦІЙ КРИПТОВАЛЮТ

3.1. Створення програмного продукту

Для створення продукту було детально проаналізовано тему та створено перелік необхідних бібліотек та утиліт:

1. Бібліотеки:

- 1.1. Requests(для здійснення http запитів)
- 1.2. PyTelebotApi(для комунікації сервісу з API телеграму)
- 1.3. Selenium(для маніпуляцій з selenium webdriber)
- 1.4. Monero та Decimal(для обробки криптотранзакцій)
- 1.5. Pscorg2(для роботи з базою)
- 1.6. Hashlib(для хешування даних)

2. Утиліти:

- 1.1. Monero Walut GUI
- 1.2. Telegram
- 1.3. PyCharm
- 1.4. PgAdmin

Після створення переліку необхідних матеріалів їх було встановлено та протестовано їх працездатність.

Наступним етапом розробки підсистеми було створення її загальної логіки та створення діаграму яка відображає порядок роботи основних функцій.

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

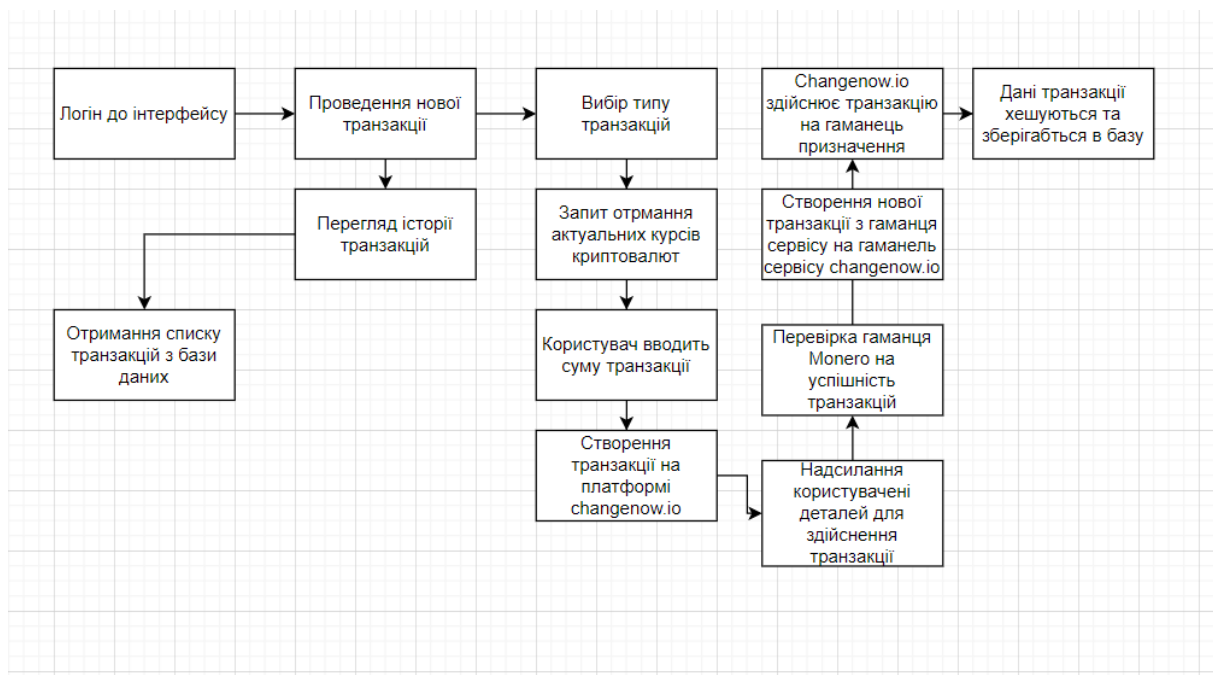


Рис. 3.1. Логіка роботи програмного продукту

На етапі логіну користувачеві доступні два варіанти ввійти в аккаунт або створити новий для використання підсистеми.

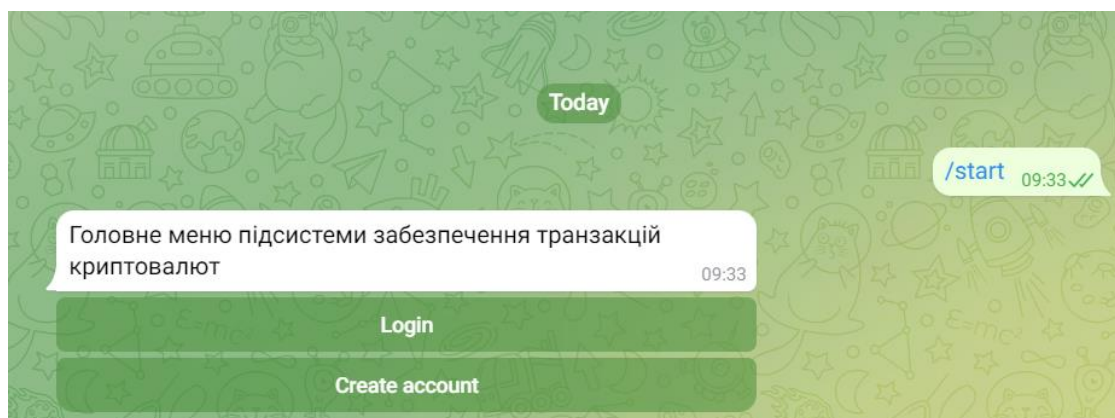


Рис. 3.2. Параметри входу в систему

Після успішного логіну в систему було створено дві основні функції:

1. Створити нову транзакцію.
2. Переглянути історію транзакцій.

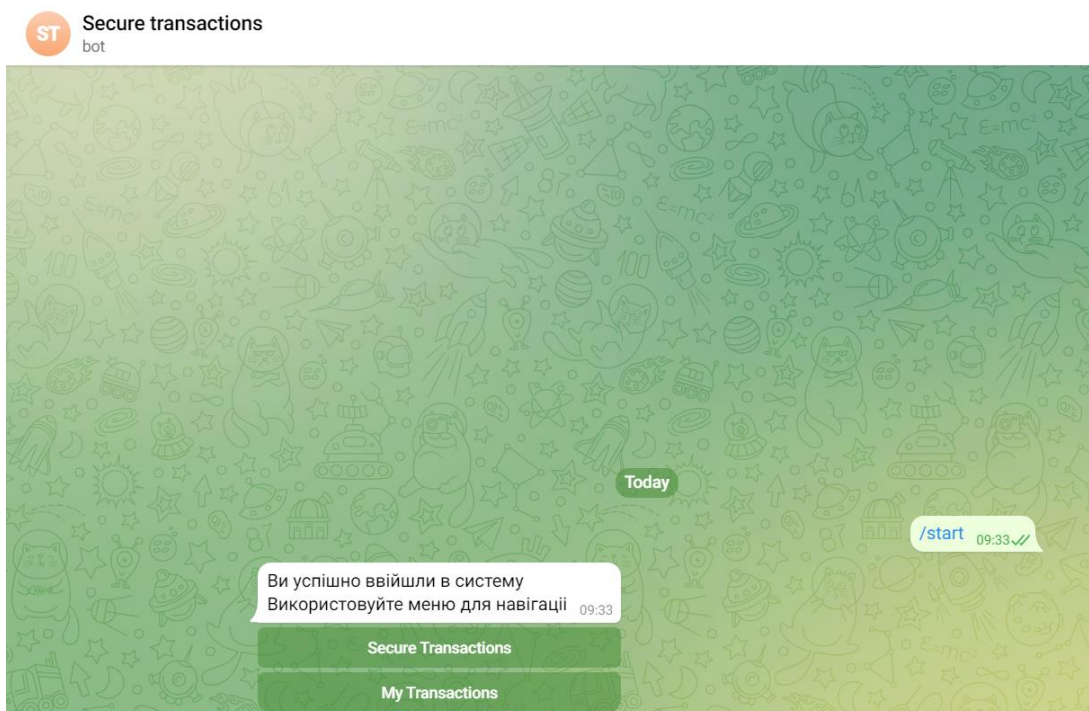


Рис.3.3. Головне меню

Якщо користувач хоче отримати інформацію про свої криптотранзакції інформація дістається з бази, дешифрується та надсилається користувачу, після того як бот надав користувачеві інформацію стосовно його транзакцій користувач автоматично повертається в головне меню для зрозумілості інтерфейсу та спрощення системи навігації.

Якщо киристувач хоче створити нову транзакцію йому надається можливість вибрати валютну пару з вже інтегрованах в систему валют.

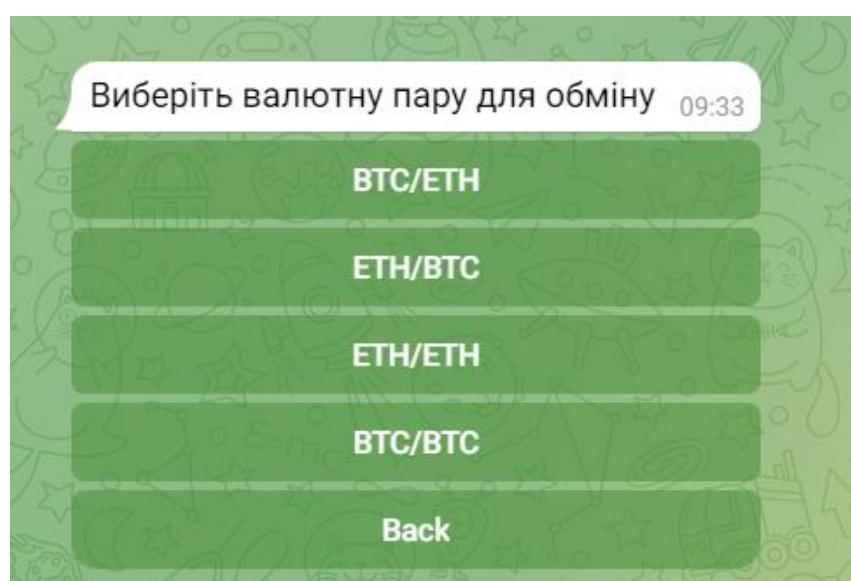


Рис.3.4. Вибір валютної пари

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Коли користувач вибрав валюту в якій він хоче здійснити переказ, підсистема робить запит до платформи coipari.io для отримання актуальних курсів валют для даного типу транзакції та надає можливість ввести необхідну суму транзакції.

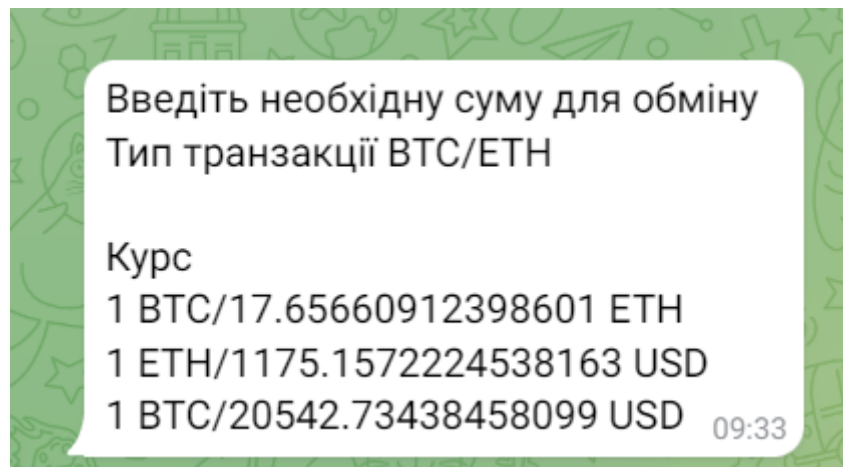


Рис.3.5. Відображення актуального курсу запит у користувача суми транзакції

Після того як система отримала від користувача інформацію стосовно суми транзакції користувач має можливість вибрати з якого гаманця він буде здійснювати транзакції.

Доступні опції:

1. Гаманець який був створений користувачем підсистемою(не реалізовано)
2. Сторонній гаманець користувача

Коли користувач зробив вибір гаманця йому необхідно вказати гаманець на який після завершення транзакції будуть переказані кошти.

		Макарчук В.В.			KIK.KP.B – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

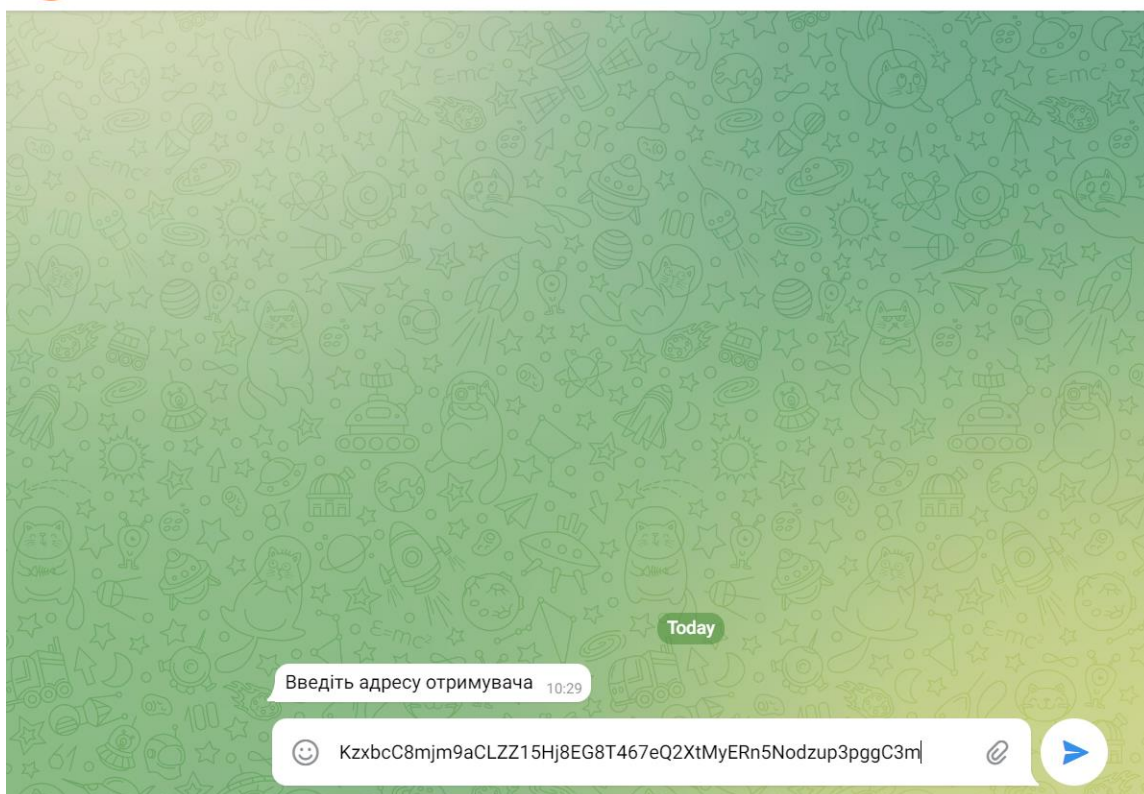


Рис.3.5. Запит у користувача адреси отримувача

Після успішного збору даних від користувача, за допомогою Selenium створюється транзакція на платформі changenow.io та надсилаються реквізити для оплати.

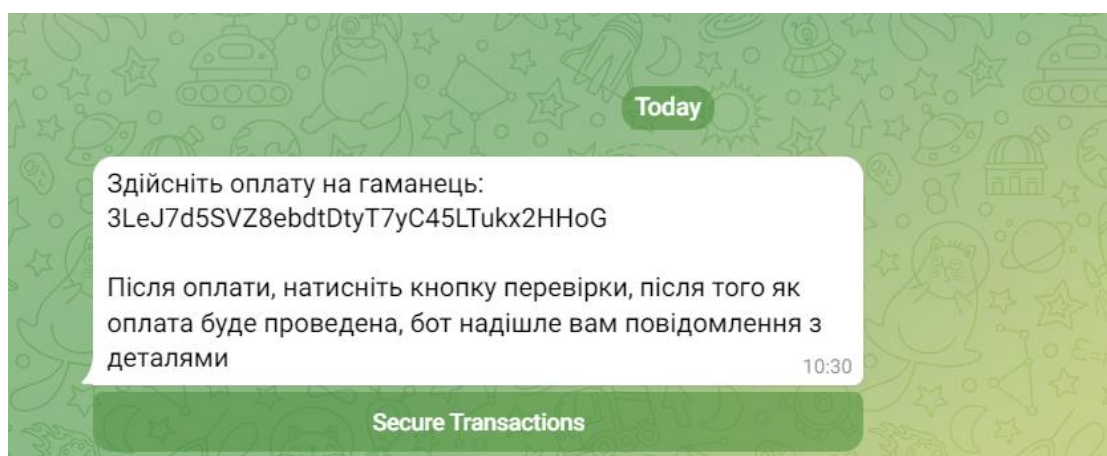


Рис.3.6. Нідіслані реквізити для оплати

Коли користувач здійснив оплату кошти надходять на XMR гаманець підсистеми, після отримання генерується нова транзакція, використовуючи реквізити цієї транзакції підсистема переказує кошти на рахунок обмінника звідки вони потрапляють на рахунок одержувача який попередньо був вказаний юзером.

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Таким чином система працює для всіх видів криптотранакцій перерахованих в меню.

Після того як кошти будуть відправлені на гаменець отримувача користувачеві надійде додаткове повідомлення про успішність транакції. Деталі транзакції будуть захешовані та збережені в базу для подальшого використання користувачем.

3.2. Висновок до третього розділу

Під час виконання третього розділу було розроблено структуру проекту, встановлено всі необхідні матеріали, налаштовано загальний інтерфейс підсистеми, проведено дослідження стосовно безпечності даного типу транзакцій. В результаті розробленого додатку було вирішено головну проблематику кваліфікаційної роботи, сервіс зроблений на базі телеграм, який зменшує кількість дій юзера є повністю безпечним та продуктивним в досягненні своєї мети.

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

ВИСНОВКИ

Технологія блокчейн і справді революційна та дає змогу не модифікувати старі підходи, а подивитися на проблему взаємовідносин користувачів та сторонніх інстанцій зовсім під іншим кутом. Раніше задача була в тому, як для своєї системи знайти найбільш надійних постачальників ресурсів, регуляторів та верифікаторів, зараз – яку форму технології блокчейн застосувати, адже його спільнота ефективно замінює всі перераховані ролі в системі. Саме тому написана ця робота: щоб розвивати та адаптувати блокчейн під теперішні виклики суспільства. Для цього не потрібно будувати модель довершеної технології, на практиці під час реалізації все одно прийдеться чимось жертвувати на перевагу пріоритетним характеристикам. Тож було обрано синтез адаптивних підходів, що передбачав:

1. Виокремлення бізнес-стратегій та сфер, в яких блокчейн не може використовуватися чи застосовуються тільки частково через певні ліміти технології;
2. Узагальнене та детальне оформлення вимог користувачів вищевказаних систем;
3. Побудову нових моделей блокчейну, які в тих чи інших умовах будуть забезпечувати максимум вимог користувачів.

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ethereum Project [Електронний ресурс]. Режим доступу: <https://www.ethereum.org/>
2. Bitcoin and Cryptocurrencies: Law Enforcement Investigative Guide [Електронний ресурс]. Режим доступу: <http://www.iacpsybercenter.org/wp-content/uploads/2018/03/Bitcoin.pdf>
3. Hoffman A. Web Application Security: Exploitation and Countermeasures for Modern Web Applications / Andrew Hoffman., 2020. – 330 с. – (1).
4. Інформаційна безпека [Електронний ресурс] – Режим доступу до ресурсу: <https://uareferat.com>
5. 6 Web Backend Security Risks to Consider in Development [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://geekflare.com/web-backend-security-risk/>.
6. Блокчейн [Електронний ресурс] - https://bankchart.com.ua/finansoviy_gid/investitsiyi/statti/scho_take_blokcheyn_#3
7. REST API [Електронний ресурс] - <https://webdevblog.ru/kak-obespechit-bezopasnost-rest-api/>
8. CryptoNote [Електронний ресурс]. Режим доступу: <https://cryptonote.org/>
9. Інформаційна безпека. Загрози при роботі в Інтернеті і їх уникнення [Електронний ресурс] – Режим доступу до ресурсу: <https://miyklas.com.ua/p/informatica/10-klas/informatciini-tekhnologiyi-v-suspilstvi-322205/informatciina-bezpeka-navchannia-v-interneti-321523/re-0cf3c5d6-6a11-458b-b39d-889f102e9e71>
10. Мистецтво бути невидим Кевін Митник 2019р
11. Хакінг – мистецтво експлойту 2 видання Джон Еріксон

		Макаруч В.В.			<i>КІК.КР.Б – 125 – 11 – ПЗ</i>	Арк.
		Пірог О.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Макарчук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import random
import time

import requests
import telebot
from telebot.types import InlineKeyboardMarkup, InlineKeyboardButton
from selenium import webdriver
from selenium.webdriver.common.by import By
from monero.wallet import Wallet
from monero.backends.jsonrpc import JSONRPCWallet
from decimal import Decimal

current_type = "
# w = Wallet(JSONRPCWallet(port=28088))
# address_xmr = w.address()
# incoming = w.incoming()
# print(address_xmr)
# print(incoming)

wallets = {}

TOKEN = '5412843710:AAH7jilDZl23wsq1GzIy4Ju1G5n1RcGHXzk'

bot = telebot.TeleBot(token=TOKEN)

main_mark = InlineKeyboardMarkup()
main_mark.add(InlineKeyboardButton(text='Login', callback_data='login'))
main_mark.add(InlineKeyboardButton(text='Create account', callback_data='create-account'))
headers = {"X-CoinAPI-Key": "A3AE1C67-2729-4DCB-82C4-A22F93FE5377"}
user_transaction_dict=dict = {}

@bot.message_handler(commands=['start'])
def send_welcom(message):
    bot.send_message(message.chat.id, "Головне меню підсистеми забезпечення транзакцій криптовалют\n\n",
    reply_markup=main_mark)

def handle_swap_btc_eth(message):
    i = int(message.message_id)
    while i > 0:
        try:
            bot.delete_message(chat_id=message.chat.id, message_id=i)
            i -= 1
            continue
        except:
            i -= 1
            continue
    amount = message.text
    try:
        amount = round(float(amount), 15)
        user_transaction_dict['amount'] = amount
        receiptWallet = bot.send_message(message.chat.id, text='Введіть адресу отримувача')
        bot.register_next_step_handler(receiptWallet, handle_wallet_trans)
    except Exception as e:
        print(e)
        bot.send_message(chat_id=message.chat.id, text='Вказана невірна сума')
        send_welcom(message)

def handle_wallet_trans(message):
    i = int(message.message_id)

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

while i > 0:
    try:
        bot.delete_message(chat_id=message.chat.id, message_id=i)
        i -= 1
        continue
    except:
        i -= 1
        continue
user_transaction_dict['reci'] = message.text
menu_mark = InlineKeyboardMarkup()
menu_mark.add(InlineKeyboardButton(text='BTC Secure wallet', callback_data='swap-btc-eth-my-wallet'))
menu_mark.add(InlineKeyboardButton(text='Сторонній гаманець', callback_data=f'swap-{current_type}-out-wallet-
{user_transaction_dict["amount"]}'))
menu_mark.add(InlineKeyboardButton(text='Back', callback_data='swap-btc-eth'))
bot.send_message(chat_id=message.chat.id,
                  text=f'Сума транзакції: {user_transaction_dict["amount"]}\nТип транзакції BTC/ETH\nВиберіть тип
гаманця прзначення',
                  reply_markup=menu_mark)

```

```

@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    if call.data == 'login':
        menu_mark = InlineKeyboardMarkup()
        menu_mark.add(InlineKeyboardButton(text='Secure Transactions', callback_data='swap'))
        menu_mark.add(InlineKeyboardButton(text='My Transactions', callback_data='wallets'))
        bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
                              text='Ви успішно ввійшли в систему\nВикористовуйте меню для навігації',
                              reply_markup=menu_mark)
    if call.data == 'swap':
        menu_mark = InlineKeyboardMarkup()
        menu_mark.add(InlineKeyboardButton(text='BTC/ETH', callback_data='swap-btc-eth'))
        menu_mark.add(InlineKeyboardButton(text='ETH/BTC', callback_data='swap-eth-btc'))
        menu_mark.add(InlineKeyboardButton(text='ETH/ETH', callback_data='swap-eth-eth'))
        menu_mark.add(InlineKeyboardButton(text='BTC/BTC', callback_data='swap-eth-eth'))
        menu_mark.add(InlineKeyboardButton(text='Back', callback_data='login'))
        bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
                              text='Виберіть валютну пару для обміну',
                              reply_markup=menu_mark)
    if call.data == 'swap-btc-eth':
        current_type = 'btc-eth'
        rate_eth = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/ETH', headers=headers).json()['rate']) * 1.01
        rate_eth_usd = float(requests.get('https://rest.coinapi.io/v1/exchangerate/ETH/USD', headers=headers).json()['rate']) *
1.01
        rate_btc = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/USD', headers=headers).json()['rate']) * 1.01
        amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
                                        text=f'Введіть необхідну суму для обміну\nТип транзакції BTC/ETH\n\nКупц\n1 BTC/{rate_eth}
ETH\n1 ETH/{rate_eth_usd} USD'
                                        f'\n1 BTC/{rate_btc} USD',)
        bot.register_next_step_handler(amount, handle_swap_btc_eth)
    if call.data == 'swap-eth-btc':
        current_type = 'eth-btc'
        rate_eth = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/ETH', headers=headers).json()['rate']) * 1.01
        rate_eth_usd = float(requests.get('https://rest.coinapi.io/v1/exchangerate/ETH/USD', headers=headers).json()['rate']) *
1.01
        rate_btc = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/USD', headers=headers).json()['rate']) * 1.01
        rate_eth_btc = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/USD', headers=headers).json()['rate']) *
1.01
        amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
                                        text=f'Введіть необхідну суму для обміну\nТип транзакції BTC/ETH\n\nКупц\n1 BTC/{rate_eth}
ETH\n1 ETH/{rate_eth_usd} USD'

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        f'\n1 BTC/{rate_btc} USD\n1 ETH/{rate_eth_btc} BTC'),
    bot.register_next_step_handler(amount, handle_swap_btc_eth)
if call.data == 'swap-eth-eth':
    current_type = 'eth-eth'
    rate_btc = float(requests.get('https://rest.coinapi.io/v1/exchangerate/ETH/USD', headers=headers).json()['rate']) * 1.01
    amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
        text=f'Введіть необхідну суму для обміну\nТип транзакції ETH/ETH\n\nКупц\n1 ETH/{rate_btc}
USD'),
    bot.register_next_step_handler(amount, handle_swap_btc_eth)
if call.data == 'swap-btc-btc':
    current_type = 'btc-btc'
    rate_btc = float(requests.get('https://rest.coinapi.io/v1/exchangerate/BTC/USD', headers=headers).json()['rate']) * 1.01
    amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
        text=f'Введіть необхідну суму для обміну\nТип транзакції BTC/ETH\n\nКупц\n1 BTC/{rate_btc}
USD'),
    bot.register_next_step_handler(amount, handle_swap_btc_eth)
if 'swap-btc-eth-out-wallet' in call.data:
    menu_mark = InlineKeyboardMarkup()
    menu_mark.add(InlineKeyboardButton(text='Secure Transactions', callback_data='swap'))
    amout = call.data.split('swap-btc-eth-out-wallet-')[1]
    driver = webdriver.Chrome()
    driver.get('https://changenow.io/exchange?from=btc&to=xmr')
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="amount-field")
    element.clear()
    element.clear()
    element.send_keys(amout)
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="recipientWallet")

element.send_keys('45byuphfrccLDJTKaP9RPHGwog4vnBRnG2jthorToLms2fZHHJ4ZzP5apgVT39Sy3MbbZTM77Wfii
6oU5GixXvUyDGaRYZo')
time.sleep(1)
driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
time.sleep(5)
driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
time.sleep(2)
address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
text=f'Здійсніть оплату на гаманець: {address}\n\nПісля оплати, натисніть кнопку перевірки, після того як оплата
буде проведена, бот надішле вам повідомлення з деталями', reply_markup=menu_mark)
i = True
while i:
    global w
    incoming2 = []
    incoming = [{'name': '1'}]
    if incoming2 != incoming:
        i = False
        driver.get('https://changenow.io/exchange?from=eth&to=xmr')
        time.sleep(2)
        element = driver.find_element(by=By.ID, value="estimation-field")
        element.clear()
        element.send_keys(amout)
        time.sleep(3)
        element = driver.find_element(by=By.ID, value="recipientWallet")
        element.send_keys(user_transaction_dict['reci'])
        time.sleep(1)
        driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
        time.sleep(5)
        driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
        time.sleep(2)
        address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
        # txs = w.transfer(address, Decimal('2.54'))

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# time.sleep(30)
# if txs[0].height is None:
bot.send_message(chat_id=call.message.chat.id, text='Транзакція проведена успішно\nРоут транзакції:\n'
'-> Ваш гаманець\n'
'-> BTC гаманець обмінника\n'
'-> XMR гаманець сервісу\n'
'-> XMR гаманець обмінника\n'
'-> BTC гаманець призначення')

bot.register_next_step_handler(amount, handle_swap_btc_eth)
if 'swap-btc-btc-out-wallet' in call.data:
    menu_mark = InlineKeyboardMarkup()
    menu_mark.add(InlineKeyboardButton(text='Secure Transactions', callback_data='swap'))
    amout = call.data.split('swap-btc-btc-out-wallet-')[1]
    driver = webdriver.Chrome()
    driver.get('https://changenow.io/exchange?from=btc&to=xmr')
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="amount-field")
    element.clear()
    element.clear()
    element.send_keys(amout)
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="recipientWallet")

element.send_keys('45byuphfrccLDJTKaP9RPHGwog4vnBRnG2jthorToLms2fZHHJ4ZzP5apgVT39Sy3MbbZTM77Wfii
6oU5GixXvUyDGaRYZo')
time.sleep(1)
driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
time.sleep(5)
driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
time.sleep(2)
address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
text=f'Здійснить оплату на гаманець: {address}\n\nПісля оплати, натисніть кнопку перевірки, після того як оплата
буде проведена, бот надішле вам повідомлення з деталями', reply_markup=menu_mark)
i = True
while i:
    global w
    incoming2 = []
    incoming = [{'name': '1'}]
    if incoming2 != incoming:
        i = False
        driver.get('https://changenow.io/exchange?from=btc&to=xmr')
        time.sleep(2)
        element = driver.find_element(by=By.ID, value="estimation-field")
        element.clear()
        element.send_keys(amout)
        time.sleep(3)
        element = driver.find_element(by=By.ID, value="recipientWallet")
        element.send_keys(user_transaction_dict['reci'])
        time.sleep(1)
        driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
        time.sleep(5)
        driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
        time.sleep(2)
        address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
        # txs = w.transfer(address, Decimal('2.54'))
        # time.sleep(30)
        # if txs[0].height is None:
        bot.send_message(chat_id=call.message.chat.id, text='Транзакція проведена успішно\nРоут транзакції:\n'
        '-> Ваш гаманець\n'
        '-> BTC гаманець обмінника\n'

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

```
'-> XMR гаманець сервісу\n'
'-> XMR гаманець обмінника\n'
'-> BTC гаманець призначення')
```

```
bot.register_next_step_handler(amount, handle_swap_btc_eth)
if 'swap-eth-eth-out-wallet' in call.data:
    menu_mark = InlineKeyboardMarkup()
    menu_mark.add(InlineKeyboardButton(text='Secure Transactions', callback_data='swap'))
    amout = call.data.split('swap-eth-eth-out-wallet-')[1]
    driver = webdriver.Chrome()
    driver.get('https://changenow.io/exchange?from=eth&to=xmr')
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="amount-field")
    element.clear()
    element.clear()
    element.send_keys(amout)
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="recipientWallet")

element.send_keys('45byuphfrccLDJTKaP9RPHGwog4vnBRnG2jthorToLms2fZHHJ4ZzP5apgVT39Sy3MbbZTM77Wfii
6oU5GixXvUyDGaRYZo')
time.sleep(1)
driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
time.sleep(5)
driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
time.sleep(2)
address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
text=f'Здійсніть оплату на гаманець: {address}\n\nПісля оплати, натисніть кнопку перевірки, після того як оплата
буде проведена, бот надішле вам повідомлення з деталями', reply_markup=menu_mark)
i = True
while i:
    global w
    incoming2 = []
    incoming = [{'name': '1'}]
    if incoming2 != incoming:
        i = False
        driver.get('https://changenow.io/exchange?from=btc&to=xmr')
        time.sleep(2)
        element = driver.find_element(by=By.ID, value="estimation-field")
        element.clear()
        element.send_keys(amout)
        time.sleep(3)
        element = driver.find_element(by=By.ID, value="recipientWallet")
        element.send_keys(user_transaction_dict['reci'])
        time.sleep(1)
        driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
        time.sleep(5)
        driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
        time.sleep(2)
        address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
        # txs = w.transfer(address, Decimal('2.54'))
        # time.sleep(30)
        # if txs[0].height is None:
        bot.send_message(chat_id=call.message.chat.id, text='Транзвкція проведена успішно\nРоут транзакції:\n'
        '-> Ваш гаманець\n'
        '-> BTC гаманець обмінника\n'
        '-> XMR гаманець сервісу\n'
        '-> XMR гаманець обмінника\n'
        '-> BTC гаманець призначення')
```

		Макаrchук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

    bot.register_next_step_handler(amount, handle_swap_btc_eth)
if 'swap-eth-btc-out-wallet' in call.data:
    menu_mark = InlineKeyboardMarkup()
    menu_mark.add(InlineKeyboardButton(text='Secure Transactions', callback_data='swap'))
    amout = call.data.split('swap-btc-eth-out-wallet-')[1]
    driver = webdriver.Chrome()
    driver.get('https://changenow.io/exchange?from=eth&to=xmr')
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="amount-field")
    element.clear()
    element.clear()
    element.send_keys(amout)
    time.sleep(3)
    element = driver.find_element(by=By.ID, value="recipientWallet")

element.send_keys('45byuphfrccLDJTKaP9RPHGwog4vnBRnG2jthorToLms2fZHHJ4ZzP5apgVT39Sy3MbbZTM77Wfii
6oU5GixXvUyDGaRYZo')
    time.sleep(1)
    driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
    time.sleep(5)
    driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
    time.sleep(2)
    address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
    amount = bot.edit_message_text(chat_id=call.message.chat.id, message_id=call.message.message_id,
text=f'Здійснить оплату на гаманець: {address}\n\nПісля оплати, натисніть кнопку перевірки, після того як оплата
буде проведена, бот надішле вам повідомлення з деталями', reply_markup=menu_mark)
    i = True
    while i:
        global w
        incoming2 = []
        incoming = [{'name': '1'}]
        if incoming2 != incoming:
            i = False
            driver.get('https://changenow.io/exchange?from=btc&to=xmr')
            time.sleep(2)
            element = driver.find_element(by=By.ID, value="estimation-field")
            element.clear()
            element.send_keys(amout)
            time.sleep(3)
            element = driver.find_element(by=By.ID, value="recipientWallet")
            element.send_keys(user_transaction_dict['reci'])
            time.sleep(1)
            driver.find_element(by=By.XPATH, value="//*[text()='Next']").click()
            time.sleep(5)
            driver.find_element(by=By.XPATH, value="//*[text()='Confirm']").click()
            time.sleep(2)
            address = driver.find_element(by=By.CLASS_NAME, value="deposit-info--address").text
            # txs = w.transfer(address, Decimal('2.54'))
            # time.sleep(30)
            # if txs[0].height is None:
            bot.send_message(chat_id=call.message.chat.id, text='Транзвкція проведена успішно\nРоут транзакції:\n'
                '-> Ваш гаманець\n'
                '-> BTC гаманець обмінника\n'
                '-> XMR гаманець сервісу\n'
                '-> XMR гаманець обмінника\n'
                '-> BTC гаманець призначення')

    bot.register_next_step_handler(amount, handle_swap_btc_eth)
if 'check-btc-eth' in call.data:
    w = wallets[f'{call.message.chat.id}']
    w.scan()
    print(w.info())

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

@bot.message_handler(content_types=['text'])
def clear_chat(message):
    i = int(message.message_id)
    while i > 0:
        try:
            bot.delete_message(chat_id=message.chat.id, message_id=i)
            i -= 1
            continue
        except:
            i -= 1
            continue

```

```

def create_xmr_transaction(amount_eth):
    pass

```

```

bot.infinity_polling()

```

		Макаручук В.В.			КІК.КР.Б – 125 – 11 – ПЗ	Арк.
		Пірог О.В.				55
Змн.	Арк.	№ докум.	Підпис	Дата		