Nimses Blockchain: система цифрових активів

Nimses Inc

nimses.com

Анотація. Економічна система, що забезпечує учасникам отримання безумовного базового доходу за кожну прожиту хвилину їхнього життя, не може існувати винятково на довірі з боку користувачів і при цьому мати достатню пропускну здатність транзакцій. Розповсюджені рішення, засновані тільки на довірі чи криптографії, не відповідають поставленим вимогам.

Гібридна система Nimses розв'язує цю проблему. Централізований сервер ставить мітки часу на транзакції, з'єднуючи їх у ланцюжок, хеші блоків ланцюга зберігаються у загальновизнаному публічному незмінному реєстрі. Модель транзакцій допускає паралельну й незалежну обробку, а пропускна здатність обмежена виключно швидкістю досягнення внутрішнього консенсусу стосовно порядку транзакцій.

1. ВСТУП

Сучасні централізовані системи електронних активів, що вирішують проблему подвійної витрати за допомогою довіреного посередника, в своїй основі будуються на довірі з боку користувачів системи. На сьогоднішній день такі системи, базовані тільки на довірі, не можуть конкурувати з аналогами, що ґрунтуються на криптографії та, як правило, мають одноранговий устрій[1]. У подібних системах зазвичай міститься механізм заохочення, який мотивує користувачів підтримувати життєздатність системи та вимагає величезних обчислювальних ресурсів, а такі елементи, як довірений посередник та центральний емітент, вилучаються повністю. Важливий недолік подібних систем - низька швидкість обробки транзакцій, що робить їх непридатними для глобального використання.

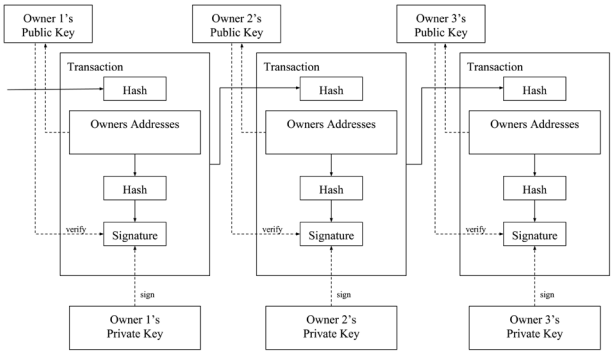
Необхідна система, що базується на криптографії, а не на довірі, і при цьому має прийнятну пропускну здатність для масового повсякденного використання.

Вирішення, знайдене Nimses, засноване на централізованому сервері міток часу і представлення електронного активу у вигляді ланцюжка цифрових підписів. Сервер виступає в ролі довіреного посередника для захисту від подвійної витрати, підтверджуючи хронологічний порядок транзакцій. Довіра гарантується закріпленням усієї історії операцій у публічному незмінному реєстрі.

2. ТРАНЗАКЦІЇ

Визначимо електронний актив як публічний ланцюжок цифрових підписів, а рахунок як упорядковану пару: унікальний ідентифікатор, який надалі називається адресою, а також певний актив.

Здійснюючи черговий переказ, власник рахунку створює та підписує транзакцію, до якої, в свою чергу, прикріплюється хеш[1] попередньої. Ця інформація приєднується до активу. Отримувач може перевірити кожний цифровий підпис, щоб переконатися в коректності цілого ланцюжка.



[1] Тут і далі під хешем розуміємо результат односторонньої функції.

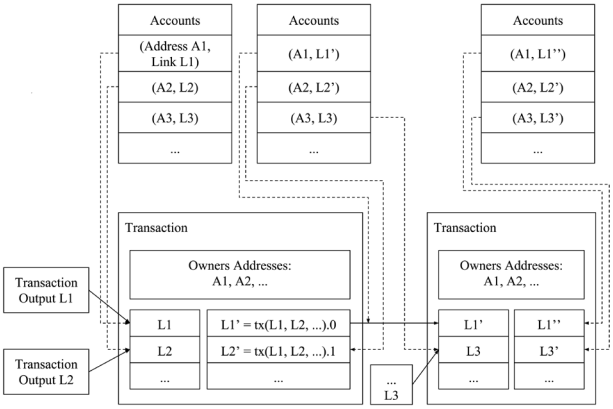
Розглянемо проблему неможливості для отримувача визначити, скільки разів поточний актив був витрачений попереднім власником. Найрозповсюдженішими є два підходи до її розв'язання. Перший полягає у перекладанні цих турбот на центральну довірену особу, але це породжує залежність від неї цілої грошової системи. Другий базується на створенні однорангової пірингової мережі, в якій учасникам необхідно відкрито публікувати транзакції, а також уміти досягати згоди відносно єдиного порядку їхнього слідування. Це тягне за собою низьку пропускну здатність мережі й величезні обчислювальні витрати.

Для усунення всіх вищезазначених недоліків обох підходів Nimses використовує гібридне рішення, засноване на застосуванні централізованого сервера міток часу, який забезпечує коректність та хронологічний порядок транзакцій. Рішення подібне до варіанту з центральною довіреною особою, але має механізи періодичного закріплення всієї історії операцій у публічному незмінному реєстрі.

3. СЕРВЕР МІТОК ЧАСУ

Правилами системи будемо називати певні загальновизнані вимоги, встановлені всередині неї. Наприклад: транзакційна комісія, податковий збір і т. ін.

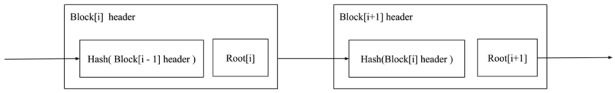
Отримавши транзакцію, сервер додає її в чергу обробки. Процес обробки налаштований таким чином: спочатку сервер перевіряє транзакцію на несуперечність поточному стану системи. За умови успішного результату приписує до неї останні ланки активів, що називаються вхідними й належать учасникам транзакції, а також певний результат, що іменується вихідними ланками та згенерований згідно з правилами системи. Вихідні ланки пов'язані зі вхідними через транзакції і є останніми для безпосередніх наступних транзакцій, тим самим гарантуючи хронологічний порядок виконання. Після успішної обробки сервер відкрито публікує транзакцію, попередньо прикріпивши до неї мітку часу. Мітка часу свідчить про те, що в даний момент конкретні дані існували і тому потрапили до ланцюжка.



Проблема, ясна річ, полягає в тому, що зберігання сервером правдивого ланцюжка будується на довірі. Користувач має знати, що попередні ланки будь-чиїх активів не були видалені. Для цього сервер періодично закріплює стан ланцюжка в публічному незмінному реєстрі. Так вибудовується ланцюг, де чергова ланка зміцнює всі попередні.

4. БЛОКИ

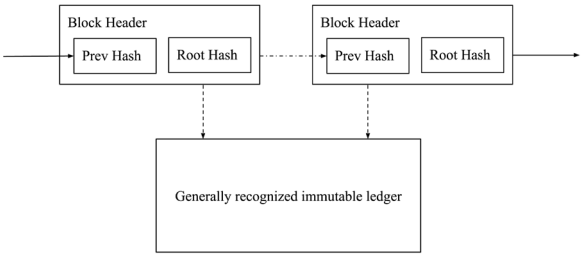
Для спрощення процесу закріплення історії даних транзакції організовуються в блоки, які складаються з дерев Меркла від хешів транзакцій та заголовка блоку. Дерева будуються на основі впорядкованої вибірки транзакцій ланцюжка за попередньо обумовлений період. Заголовком блоку назвемо впорядковану пару, що складається з хешу заголовка попереднього блоку та кореня дерева Меркла поточного. Таким чином, блоки пов'язані в упорядкований ланцюжок, де спроба змінити інформацію певного блоку вимагає перерахунку всіх наступних, що, в свою чергу, забезпечує достатність закріплення тільки хешу заголовка замість цілого ланцюжка транзакцій.



Використання ланцюжка блоків спрощує не тільки закріплення історії, але й перевірку існування транзакції. Для перевірки включеності транзакції до блоку користувачеві не потрібно скачувати цілу історію транзакцій, що є неприйнятним через розміри та швидкість зростання історії. Він може запитати посилання на блок, в якому міститься транзакція, і шлях Меркла до неї, аби переконатися в тому, що вона міститься в блоці, а всі наступні прийняті та підтверджені сервером.

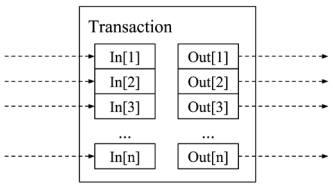
5. ЗАКРІПЛЕННЯ ЧАСОВИХ МІТОК

Під закріпленням історії мається на увазі відкрита публікація, як у газеті чи Usenet-постах, деяких даних, що гарантують однозначність її походження, в незмінному реєстрі. На момент 2017 року в якості реєстру для закріплення вважається доцільним використовувати відкриті однорангові децентралізовані системи, що мають достатній рівень надійності, як-то Bitcoin та Etherium.



6. ПОЄДНАННЯ ТА РОЗДІЛЕННЯ СУМ

Для підтримання переказів між більше ніж двома учасниками вхідні та вихідні ланки, які містяться в транзакціях, являють собою не пари, а впорядковані множини. Для прийняття такої транзакції системою необхідно, щоб вона була підписана кожним з відправників платежу.



7. РЕАЛІЗАЦІЯ NIMSES

7.0. Технічні вимоги

• Потенційна аудиторія: 10^9 користувачів
• Швидкість підтвердження транзакцій: ~1 секунда
• Пропускна здатність: до 10^6 транзакцій за секунду

7.1. Активи

У поточній реалізації одному рахункові відповідає один тип активу. Початково визначено два типи активів: нім та інфінім.

Німи з'являються завдяки внутрішній емісії системи, яка є функцією від астрономічного часу. Інфініми емітуються шляхом «спалювання» певної кількості німів, згідно з опублікованими правилами.

Тип активу кодується 32 бітами.

7.2. Рахунки

Унікальна адреса рахунку складається з 16 байт числової послідовності та 4-байтового типу активу.

Рахунок описується наступним чином: тип рахунку, тип активу, поточний баланс, кількість транзакцій у ланцюжкові цифрових підписів, набір асоційованих активних публічних ключів, який верифікує власників рахунку, тип емісії та інші допоміжні дані.

Стан рахунку, в свою чергу, визначається часовою міткою, номером транзакції у відповідному ланцюжкові цифрових підписів та кінцевим сальдо останньої ланки, розрахованим на цю мітку часу відносно попередньої з урахуванням емісії та результату транзакції.

Типи рахунків діляться на службові та користувацькі, які підпадають під емісію нім-токену.

Рахунки – це окремі сутності в системі, відображають поточний стан активів користувачів. Вони створюються спеціальною транзакцією, яка асоціює унікальну адресу, початковий стан активу та набір публічних ключів. Після чого за зареєстрованою адресою можна здійснювати операції.

Певна підмножина транзакцій підписується не асоційованими з рахунком актуальними ключами, а спеціальними сервісними ключами, авторизованими генезисними кореневими ключами. Наприклад, податковий інститут має можливість списувати певні суми податкових відрахувань з рахунків користувачів.

```
Address  = {16 byte} + {asset_id: 4 byte}

Account  =  (Address,  Asset,  Type,  Emission,
{Public Key}, State)

State    = (Time Point, Balance, Nonce, ...)
Account Type = {
    COMMON,
    HUMAN,
    GENESIS,
    TAX,
    REMOVED,
    ...
}
```

7.3. Емісія

З моменту реєстрації в системі кожному користувачеві емітується по одному німу за хвилину.

Для визначення кількості емітованих німів під час проведення транзакцій сервер часових міток додатково помічає записувані в ланцюжок транзакції мітками астрономічного часу та розраховує вихідні ланки транзакції, враховуючи емісію німів.

Емісія інфінімів є особливим випадком, для якого виділено спеціальний рахунок, який накопичує німи до певної межі і з часом генерує один інфінім.

7.4. Генезис

Ланцюг починається з певного стану, який зветься «генезисним» і задається спеціальним типом рахунку та реєстрованими кореневими ключами. Генезисний рахунок визначає вихідну емісію усіх німів на момент початку роботи мережі.

7.5. Транзакції

Звичайна транзакція складається з трьох частин: заголовка, тіла й свідчень відправників.

```
Tx = (Header, Body, Witnesses)
```

Заголовок транзакції являє собою кортеж із: версії протоколу, типу транзакції та часового вікна, впродовж якого транзакція може бути записана в ланцюжок.

```
TxTypes  = {
    CREATE_ACCOUNT,
    SPEND,
    TAX_SPEND,
    GENESIS_SPEND,
    REG_KEY,
    REVOKE_KEY,
    ...
}

TxHeader = (Version, Type, Time Window)
```

Адресу рахунку та кількість отримуваного чи відісланого активу визначимо як посилання на значення.

```
ValueRef = (Account Address, Value)
```

Тіло транзакції – це впорядкована пара множин, що містять посилання на значення відправників та отримувачів відповідно.

```
Tx.Body  = (From: {ValueRef}, To: {ValueRef})
```

Свідченнями відправників називається впорядкована множина свідчень кожного з відправників. Свідченням відправника назвемо впорядковану пару: підпис відправника та хеш публічного ключа з підписуючої пари. В якості хеш-функції використовується SHA3-256 [4] (FIPS-202).

```
Witnesses = ({Witness})

Witness  = (Signature, Public Key Hash)
```

Цифровий підпис, отриманий шляхом підписання заголовка й тіла транзакції будь-якою парою ключів з набору, прикріпленого до рахунку відправника, є підписом відправника. В якості механізму цифрових підписів використовується ECDSA [2] (NIST.FIPS.186-4) на кривій SECP256R1 [3] (RFC 5480).

```
Signature = ECDSA(Public Key, Tx.SigId[i], Private Key)
```

Під час запису транзакції в ланцюжок вона доповнюється хешами вхідних ланок та вихідними ланками. Ланки ідентифікуються 256-бітною хеш-функцією. Їхня унікальність гарантується системою та перевіряється під час запису транзакції в ланцюг. Так само гарантується й унікальність самих тразакцій.

Транзакції можуть бути підписані або секретними ключами рахунків користувачів, або спеціальними ключами.

7.6. Спеціальні ключі

У системі представлені спеціальні типи ключів, використовувані для реалізації необхідних функцій економічної системи Nimses. Прикладами таких функцій виступають податкові списання або стягнення за порушення правил. Ці спеціальні ключі можуть бути використані замість користувацьких та реєструються в системі шляхом підписання кореневими ключами. Секретні частини таких ключів зберігаються на спеціальних пристроях, відомих як Апаратні Модулі Безпеки, у захищених мережах, і витягти їх звідти неможливо.

```
Key Types = {
    KEY_USER,
    KEY_ROOT,
    KEY_MASTER,
    KEY_IDENTITY,
    KEY_TAX,
    KEY_FAMILYPAYMENT,
    ...
}
```

У звичайному випадку правила валідації транзакцій визначаються їхнім типом та типом рахунків, що беруть участь.

7.7. Консенсус

В однорангових мережах учасники мусять домовлятися щодо єдиної версії історії транзакцій для захисту від подвійної витрати, використовуючи такі алгоритми як «доведення виконання роботи», «доведення повноважень», «доведення розміру частки» та аналогічні, які забезпечують єдиність обраної версії історії. Централізовані системи, позбавлені цього недоліку, можуть використовувати ефективніші методи узгодження порядку транзакцій.

Відсутність необхідності повної реплікації усієї історії між величезною кількістю учасників для прийняття рішень дозволяє розпаралелити обробку транзакцій та забезпечити високу пропускну здатність.

Сервер міток часу являє собою закриту розподілену мережу, що складається з незалежних вузлів, які використовують глобально розподілене сховище з класу NewSQL. Такий устрій забезпечує лінійну масштабованість простору та пропускної здатності обробки транзакцій. Консенсус між внутрішніми вузлами досягається через алгоритм Паксос.

Готові блоки вивантажуються в загальнодоступне розподілене сховище.

Обробка транзакції в Nimses займає приблизно одну секунду, і надається можливість отримання спрощених доказів включенсті транзакії в блок одразу ж після його запису. При цьому пропускна здатність прямо пропорційна швидкості досягнення внутрішнього консенсусу.

7.8. Блоки

Блоки об'єднують транзакції в упорядковані набори та слугують в якості інструментів, які спрощують забезпечення відкритості, незмінності та хронологічного порядку.

Блок складається з набору транзакцій та заголовка блоку. Заголовок блока включає в себе хеш заголовка попереднього блоку та корені трьох дерев Меркла.

```
Block Header = (Version, Height, CommitAt, Gen, Prev Block Hash, Tx Root, Witness Root, Receipt Root, Witness)
```

Після формування та запису хеші блоків записуються у зовнішній довірений реєстр. Також блоки можуть бути доповнені підписами додаткових валідаторів, які підтверджують вірність ланцюжка та перетворень.

7.9. Дерево Меркла

Розрахунок дерева Меркла для ідентифікаторів транзакцій аналогічний алгоритмові, описаному в RFC6962 [5] секція 2.1.2. Застосовується хешуюча функція SHA3-256 [4] (FIPS-202). На рівнях із непарною кількістю вершин остання вершина дублюється.

8. ВИСНОВОК

Гібридна система Nimses відповідає заданим технічним вимогам. Довіра зі сторони користувача обумовлена неприйнятною трудомісткістю змінювання даних, закріплених у зовнішньому реєстрі. Спеціальна транзакційна модель, яка припускає паралельну обробку, динамічний розмір та період блоку, дозволяє досягнути пропускної здатності, необхідної для загальнопланетарної системи обміну активами.

9. ПОСИЛАННЯ

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. – https://bitcoin.org/bitcoin.pdf

[2] Federal Information Processing Standards Publication, Digital Signature Standard, July 2013. – https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

[3] S. Turner, D. Brown, K. Yiu, R. Housley, T. Polk. Elliptic Curve Cryptography Subject Public Key Information, RFC 5480, March 2009. – https://tools.ietf.org/html/rfc5480

[4] Federal Information Processing Standards Publication, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015. – https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf

[5] B. Laurie, A. Langley, E. Kasper. Certificate Transparency, RFC 6962, June 2013. – https://tools.ietf.org/html/rfc6962