Nimses Blockchain: система цифровых активов

Nimses Inc

nimses.com

Аннотация. Экономическая система, обеспечивающая участникам безусловный базовый доход за каждую прожитую минуту их жизни, не может основываться исключительно на доверии со стороны пользователей и при этом обладать достаточной пропускной способностью транзакций. Распространённые решения, основанные только на доверии или криптографии, не отвечают поставленным требованиям.

Гибридная система Nimses решает эту проблему. Централизованный сервер ставит метки времени на транзакции, соединяя их в цепочку, хеши блоков цепи сохраняются в общепризнанном публичном неизменяемом реестре. Модель транзакций допускает параллельную и независимую обработку, а пропускная способность ограничена исключительно скоростью достижения внутреннего консенсуса касательно порядка транзакций.

1. ВВЕДЕНИЕ

Современные централизованные системы электронных активов, решающие проблему двойной траты с помощью доверенного посредника, в своей основе строятся на доверии со стороны пользователей системы. На текущий момент такие системы не могут конкурировать, в контексте доверия, с аналогами, основанными на криптографии и, как правило, имеющими одноранговое устройство [1]. В мподобных системах обычно содержится механизм поощрения, мотивирующий пользователей поддерживать жизнеспособность системы и требующий огромных вычислительных ресурсов, а такие элементы, как доверенный посредник и центральный эмитент, полностью упраздняются. Важный недостаток подобных систем – низкая скорость обработки транзакций, что делает их непригодными для глобального использования.

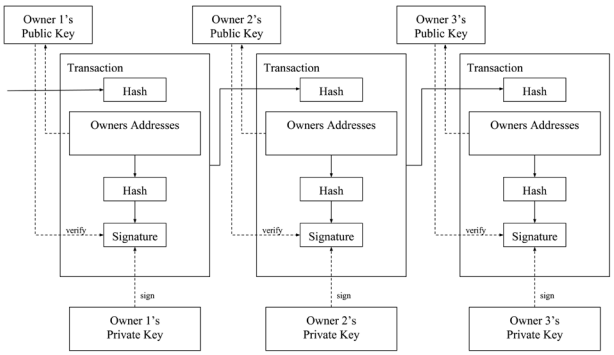
Необходима система, основанная на криптографии, а не на доверии, при этом обладающая приемлемой пропускной способностью для массового повседневного пользования.

Решение, найденное Nimses, основано на централизованном сервере меток времени и представлении электронного актива в виде цепочки цифровых подписей. Сервер выступает в роли доверенного посредника для защиты от двойной траты, подтверждая хронологический порядок транзакций. Доверие гарантируется закреплением всей истории операций в публичном неизменяемом реестре.

2. ТРАНЗАКЦИИ

Определим электронный актив как публичную цепочку цифровых подписей, а счёт как упорядоченную пару: уникальный идентификатор, далее именуемый адресом, и некоторый актив.

Совершая очередной перевод, владелец счёта создаёт и подписывает транзакцию, к которой, в свою очередь, прикрепляется хеш[1] предшествующей. Эта информация присоединяется к активу. Получатель может проверить каждую цифровую подпись, чтобы убедиться в корректности всей цепочки.



[1] Здесь и далее под хешем подразумевается результат односторонней функции.

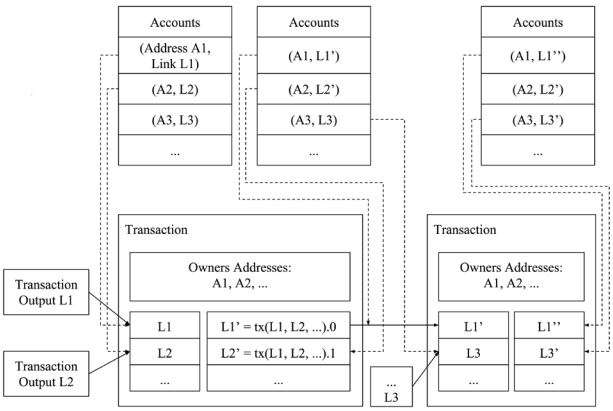
Рассмотрим проблему неспособности получателя определить, сколько раз текущий актив был потрачен предыдущим владельцем. Наиболее распространены два подхода к её решению. Первый заключается в перекладывании этих забот на центральное доверенное лицо, что порождает зависимость всей денежной системы от оного. Второй состоит в создании одноранговой пиринговой сети, в которой участникам необходимо открыто публиковать транзакции, а также уметь приходить к согласию относительно единого порядка их следования. Это влечёт за собой низкую пропускную способность сети и огромные вычислительные затраты.

Для устранения вышеуказанных недостатков обоих подходов Nimses использует гибридное решение, основанное на централизованном сервере меток времени, обеспечивающем корректность и хронологический порядок транзакций. Решение подобно варианту с центральным доверенным лицом, но снабжено механизмом периодического закреплением всей истории операций в публичном неизменяемом реестре.

3. СЕРВЕР МЕТОК ВРЕМЕНИ

Правилами системы будем называть некоторые общепризнанные требования, установленные внутри неё. Например: транзакционная комиссия, налоговый сбор и так далее.

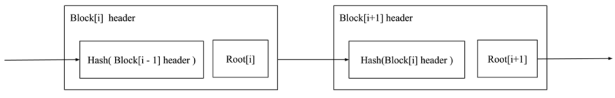
Получая транзакцию, сервер добавляет её в очередь обработки. Процесс обработки устроен следующим образом: сначала сервер проверяет транзакцию на непротиворечивость текущему состоянию системы. При успешном исходе, приписывает к ней последние звенья активов, именуемые входными и принадлежащие участникам транзакции, а также некоторый результат, именуемый выходными звеньями и сгенерированный согласно правилам системы. Выходные звенья связаны с входными посредством транзакции и являются последними для непосредственно следующих транзакций, тем самым гарантируя хронологический порядок выполнения. После успешной обработки сервер открыто публикует транзакцию, предварительно прикрепляя к ней метку времени. Метка времени свидетельствует, что в данный момент конкретные данные существовали и потому попали в цепочку.



Проблема, разумеется, состоит в том, что сохранение сервером подлинной цепочки строится на доверии. Пользователь должен знать, что предшествующие звенья чьих-либо активов не были удалены. Для этого сервер периодически закрепляет состояние цепочки в публичном неизменяемом реестре. Так выстраивается цепь, где очередное звено укрепляет все предыдущие.

4. БЛОКИ

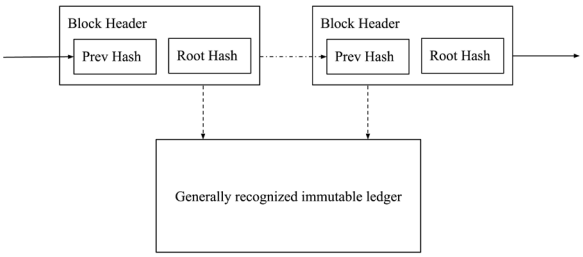
Для упрощения процесса закрепления истории данных, транзакции организуются в блоки, состоящие из деревьев Меркла от хешей транзакций и заголовка блока. Деревья строятся на основе упорядоченной выборки транзакций цепочки за предопределенный период. Заголовком блока назовём упорядоченную пару, состоящую из хеша заголовка предшествующего блока и корня дерева Меркла текущего. Таким образом, блоки связаны в упорядоченную цепочку, где попытка изменить информацию некоторого блока требует пересчёта всех последующих, что, в свою очередь, обеспечивает достаточность закрепления лишь хеша заголовка вместо всей цепочки транзакций.



Использование цепочки блоков упрощает не только закрепление истории, но и проверку существования транзакции. Для проверки включённости транзакции в блок, пользователю не требуется скачивать всю историю транзакций, что является неприемлемым, ввиду размеров и скорости роста истории. Он может запросить ссылку на блок, в котором находится транзакция, и путь Меркла к ней, чтобы убедиться в том, что она содержится в блоке, а все последующие приняты и подтверждены сервером.

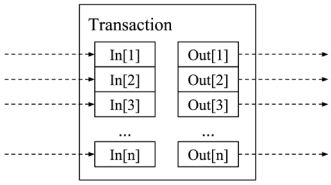
5. ЗАКРЕПЛЕНИЕ ВРЕМЕННЫХ МЕТОК

Под закреплением истории понимается открытая публикация, как в газете или Usenet-постах, некоторых данных, гарантирующих однозначность её происхождения, в неизменяемом реестре. На момент 2017 года полагается разумным в качестве реестра для закрепления использовать открытые одноранговые децентрализованные системы, обладающие достаточным уровнем надёжности, такие как Bitcoin и Etherium.



6. СОЕДИНЕНИЕ И РАЗДЕЛЕНИЕ СУММ

Для поддержания переводов между более чем двумя участниками, входные и выходные звенья, содержащиеся в транзакциях, представляют собой не пары, а упорядоченные множества. Для принятия такой транзакции системой, необходимо чтобы она была подписана каждым из отправителей платежа.



7. РЕАЛИЗАЦИЯ NIMSES

7.0. Технические требования

- Потенциальная аудитория: 3 x 10^9 пользователей
- Скорость подтверждения транзакций: ~1 секунда
- Пропускная способность: до 10^6 транзакций в секунду

7.1. Активы

В текущей реализации одному счёту соответствует один тип актива. Изначально определено два типа активов: **ним** и **инфиним**.

Нимы появляются благодаря внутренней эмиссии системы, которая является функцией от астрономического времени. Инфинимы эмитируются путем “сжигания” определённого количества нимов, согласно опубликованным правилам.

Тип актива кодируется 32 битами.

7.2. Счета

Уникальный адрес счёта состоит из 16 байт числовой последовательности и 4-байтового типа актива.

Счёт описывается следующим образом: тип счёта, тип актива, текущий баланс, количество транзакций в цепочке цифровых подписей, набор ассоциированных активных публичных ключей, верифицирующий владельцев счёта, тип эмиссии и прочие вспомогательные данные.

Состояние счёта, в свою очередь, определяется временной меткой, номером транзакции в соответствующей цепочке цифровых подписей и конечным сальдо последнего звена, рассчитанным на эту метку времени относительно предыдущей с учётом эмиссии и результата транзакции.

Типы счетов подразделяются на служебные и пользовательские, подпадающие под эмиссию ним-токена.

Счета - это отдельные сущности в системе, отражают текущее состояние активов пользователей. Они создаются специальной транзакцией, ассоциирующей уникальный адрес, начальное состояние актива и набор публичных ключей. После чего по зарегистрированному адресу можно производить операции.

Некоторое подмножество транзакций подписывается не ассоциированными со счётом актуальными ключами, а специальными сервисными ключами, авторизованными генезисными корневыми ключами. Например, налоговый институт имеет возможность списывать определенные суммы налоговых отчислений со счетов пользователей.

```
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. Эмиссия

С момента регистрации в системе каждому пользователю эмитируется по одному ниму в минуту.

Для определения количества эмитируемых нимов при проведении транзакций, сервер временных меток дополнительно помечает записываемые в цепь транзакции метками астрономического времени и рассчитывает выходные звенья транзакции учитывая эмиссию нимов.

Эмиссия инфинимов является особым случаем, для которого выделен отдельный специальный счёт, накапливающий нимы до определённого предела, впоследствии генерирующий один инфиним.

7.4. Генезис

Цепь начинается с некоторого состояния, называемого «генезисным», задаваемого специальным типом счёта и регистрируемыми корневыми ключами. Генезисный счёт определяет изначальную эмиссию всех нимов на момент начала работы системы.

7.5. Транзакции

Обычная транзакция состоит из трех частей: заголовка, тела и свидетельств отправителей.

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

Заголовок транзакции представляет собой кортеж из: версии протокола, типа транзакции и временного окна, в течение которого транзакция может быть записана в цепочку.

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

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

Адрес счёта и количество получаемого или отправляемого актива определим как ссылку на значение.

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

Тело транзакции – это упорядоченная пара множеств, содержащих ссылки на значения отправителей и получателей соответственно.

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

Свидетельствами отправителей называется упорядоченное множество свидетельств каждого из отправителей. Свидетельством отправителя назовем упорядоченную пару: подпись отправителя и хеш публичного ключа из подписывающей пары. В качестве хеш-функции используется SHA3-256 [4] (FIPS-202).

```
Witnesses = ({Witness})

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

Цифровая подпись, полученная посредством подписания заголовка и тела транзакции любой парой ключей из набора, прикрепленного к счёту отправителя, является подписью отправителя. В качестве механизма цифровых подписей используется ECDSA [2] (NIST.FIPS.186-4) на кривой SECP256R1 [3] (RFC 5480).

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

При записи транзакции в цепочку она дополняется хешами входных звеньев и выходными звеньями. Звенья идентифицируются 256-битной хеш-функцией. Их уникальность гарантируется системой и проверяется при записи транзакции в цепь. Так же гарантируется уникальность и самих транзакций.

Транзакции могут быть подписаны либо секретными ключами счетов пользователей, либо специальными ключами.

7.6. Специальные ключи

В системе представлены специальные типы ключей, используемые для реализации необходимых функций экономической системы Nimses. Примерами таких случаев выступают налоговые списания или штрафы за нарушение правил. Эти особые типы ключей равносильны ключам счетов пользователей. Для удостоверения их полномочий открытые части регистрируются в системе за подписью корневых ключей, происходящих из генезисного состояния. Секретные части таких ключей хранятся на специальных устройствах HSM в защищённых сетях, и их невозможно оттуда извлечь.

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

В обычном случае правила валидации транзакций определяются их типом и типом участвующих счетов.

7.7. Консенсус

В одноранговых сетях участникам необходимо договариваться о единой версии истории транзакций для защиты от двойной траты, используя такие алгоритмы как “доказательство выполненной работы”, “доказательство полномочий”, “доказательство размера доли” и аналогичные, обеспечивающие единственность версии истории. Централизованные системы, лишённые этого недостатка, могут использовать более эффективные методы согласования порядка транзакций.

Отсутствие необходимости полной репликации всей истории между огромным количеством участников для принятия решений позволяет распараллелить обработку транзакций и обеспечить высокую пропускную способность.

Сервер меток времени представляет собой закрытую распределенную сеть, состоящую из независимых узлов, использующих глобально распределенное хранилище из класса NewSQL. Данное устройство обеспечивает линейную масштабируемость пространства и пропускной способности обработки транзакций. Консенсус между внутренними узлами достигается посредством алгоритма Paxos.

Готовые блоки выгружаются в общедоступное распределенное хранилище.

Обработка транзакции в Nimses занимает примерно одну секунду и представляется возможность получения упрощенных доказательств включенности транзакции в блок сразу же после его записи. При этом пропускная способность прямо пропорциональна скорости достижения внутреннего консенсуса.

7.8. Блоки

Блоки объединяют транзакции в упорядоченные наборы и служат в качестве инструментов, упрощающих обеспечение открытости, неизменяемости и хронологического порядка.

Блок состоит из набора транзакций и заголовка блока. Заголовок блока включает в себя хеш заголовка предыдущего блока и корни трёх деревьев Меркла.

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

После формирования и записи, хеши блоков записываются во внешний доверенный реестр. Также блоки могут быть дополнены подписями дополнительных валидаторов, подтверждающих верность цепочки и преобразований.

7.9. Дерево Меркла

Расчёт дерева Меркла для идентификаторов транзакций аналогичен алгоритму описанному в RFC6962 [5] секция 2.1.2. Применяется хеширующая функция SHA3-256 [4] (FIPS-202). На уровнях с нечетным количеством вершин последняя вершина дублируется.

8. ЗАКЛЮЧЕНИЕ

Гибридная система Nimses отвечает заданным техническим требованиям. Доверие с пользовательской стороны обусловлено неприемлемой трудоемкостью изменения данных, закрепленных во внешнем реестре. Специальная транзакционная модель, допускающая параллельную обработку, динамический размер и период блока, позволяет достичь пропускной способности, необходимой для общепланетарной системы обмена активами.

9. ССЫЛКИ

[1] Сатоши Накамото. «Биткойн: электронная денежная система одноранговой сети» - https://bitcoin.org/bitcoin.pdf

[2] Публикация федеральных стандартов обработки информации, стандарт цифровой подписи, июль 2013 года. Https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

[3] S. Turner, D. Brown, K. Yiu, R. Housley, T. Polk. Криптография криптографии с эллиптическими кривыми Тематический ключ, RFC 5480, март 2009 г. - https://tools.ietf.org/html/rfc5480

[4] Публикация федеральных стандартов обработки информации, стандарт SHA-3: функции хеширования на основе перестановок и расширенные выходные функции, август 2015 г. - https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf

[5] B. Laurie, A. Langley, E. Kasper. Прозрачность сертификата, RFC 6962, июнь 2013 г. - https://tools.ietf.org/html/rfc696