Хэш Цепочки Технологии Hasq

Олег Мазонка и Влад Попов
Hasq Technology Pty Ltd, Австралия, 2014
om@hasq.org, vp@hasq.org

Аннотация. Эта статья описывает особую схему связи записей, основанную на хэш-функциях. Схема концептуально проста и легко реализуется в программном обеспечении, что позволяет просто и безопасно передавать право собственности на цифровые объекты между участниками обмена.

Ключевые слова: Blockchain, Bitcoin, Криптовалюты, Hasq

Содержание

1. Введение

Понятие права собственности предполагает наличие контроля над объектом собственности. Это утверждение верно как для реальных объектов, так и для абстрактных понятий, таких как банковский счет, копирайт или цифровые данные. Примеры действий владельцев, управляющих своими активами, включают добровольную передачу прав собственности другой стороне. В действительности, однако, возможность контроля ограничена фундаментальным свойством большинства вещей, которыми можно владеть - собственность может быть утрачена против воли ее владельца. Например, наличные деньги могут быть утеряны, как и другие физические объекты, банковский счет может быть заблокирован, копирайт может быть утрачен в результате судебного разбирательства, цифровые данные могут быть незаконно скопированы, фактически лишая владельца возможности контроля.

Однако, существует понятие, принципиально отличающееся от вышеописанных - авторское право. Авторское право не может быть отнято или передано другому лицу. *Таблица 1* систематизирует вышеприведенные рассуждения.

Несмотря на то, что в некоторых случаях потеря контроля может стать следствием законного судебного решения, в других случаях это может быть результатом преступных действий злоумышленника, что очень нежелательно. Значительные ресурсы на протяжении многих лет были затрачены на разработку решений для физической и электронной безопасности. Цифровые данные особенно уязвимы из-за простоты их копирования. В то же время, важность защиты таких данных постоянно увеличивается в силу того, что информационные технологии в современном обществе играют все более важную роль.

Очевидным решением проблемы права собственности на цифровые данные является электронный реестр, который хранит записи, связывающие владельцев с их цифровыми объектами. Однако, использование такого реестра подразумевает доверие к нему, что не всегда приемлемо. Кроме того, эксплуатационные расходы часто увеличивают стоимость использования таких реестров, что может ограничивать их клиентскую базу. Существует длительная история попыток ослабить требование доверия к реестру и уменьшения затрат на его поддержание [1-6]. Одним из последних прорывов в этой области является публичная книга реестра технологии *Blockchain*, первоначально воплощенная в криптовалюте *Bitcoin* [7].

Два свойства права собственности, указанные в *Таблице 1*, соответствуют шаблону *Да - Да* для обычных объектов и *Нет - Нет* для авторского права. Объект с шаблоном *Нет - Да*, однако, соответствовал бы чему-то с неотъемлемой защитой от утраты как у авторского права, но с возможностью быть переданным по желанию владельца другому лицу. У объекта такого рода существовало бы множество применений, так как он на практике воплощал бы идею максимальной защиты от незаконного отъема.

Свойства права собственности	Реальные объекты, деньги (наличные)	Деньги (банковский счет)	Копирайт	Авторское право	Токен Hasq
Может быть утеряно против желания	Да	Да	Да	Нет	Нет
Может быть передано по желанию	Да	Да	Да	Нет	Да

Таблица 1

В следующем разделе мы представляем схему, которая позволяет управлять цифровым объектом (токеном Hasq) настолько безопасно, как будто его текущий владелец является его же автором. В то же время, схема позволяет передавать право собственности на объект другой стороне, к которой, возможно, нет доверия.

2. Цепочка хэшей

Предположим, что существует общедоступная текстовая база данных. Для простоты можно представлять базу данных как эквивалент выпусков журнала или газеты. База данных состоит из списка записей, связанных между собой особым образом с использованием хэш-функции. У каждой записи есть следующие текстовые поля, разделенные пробелами:

$$N \quad S \quad K \quad \{ G_1 \quad G_2 \quad ... \} \quad O \quad [D]$$

где

N - порядковый номер определенного *S*;

S (токен) - результат вычисления хэш-функции (хэш) от цифровых данных произвольного характера, *S* представлен в виде строки шестнадцатеричных цифр[1];

K, *G* и *O* - хэши, значение которых объясняется далее;

D - опциональное текстовое поле данных.

Поля *K* (Key, Ключ), *G* (Generator, Генератор) и *O* (Owner, Владелец) используются для связывания записей. Количество полей *G* может быть произвольным, но фиксированным в пределах одной базы данных. Для простоты описания в дальнейшем используется только одно поле *G*. Однако, читатель с легкостью может предполагать любое другое их количество, включая ноль. База данных состоит из записей, включающих различные токены *S*. Если отобрать записи, содержащие одно значение *S*, то список этих записей будет выглядеть следующим образом (поле *D* опущено для простоты, так как оно не участвует в связывании записей):

…
$N_0\ S\ K_0\ G_0\ O_0$
$N_1\ S\ K_1\ G_1\ O_1$
$N_2\ S\ K_2\ G_2\ O_2$
$N_3\ S\ K_3\ G_3\ O_3$
$N_4\ S\ K_4\ G_4\ O_4$
…

Здесь N_0, N_1, N_2 ... - последовательные номера для выбранного *S,* т.е. $N_1 = N_0 + 1$, $N_2 = N_1 + 1$, и так далее.

---

[1] Здесь используются определенные формы для текстового представления полей записи и символа конкатенации. Эти формы выбраны специально для удобства описания схемы, хотя при этом утрачивается некоторая универсальность.

Записи связаны друг с другом в соответствии со следующими правилами:

G_0 = Hash(N_1, S, K_1)
O_0 = Hash(N_1, S, G_1)
G_1 = Hash(N_2, S, K_2)
O_1 = Hash(N_2, S, G_2)

и так далее

Хэш-функция Hash принимает аргументы в текстовом виде. Аргументы связаны между собой символом пробела. Результат вычисления представлен в виде строки шестнадцатеричных цифр.

Сервер, на котором размещена база данных, публикует новую запись *только* если она соответствует вышеуказанным правилам. Формирующаяся таким образом база данных называется целостной. Целостность всегда может быть проверена независимо.

На приведенном ниже рисунке показано графическое представление вышеуказанных правил.

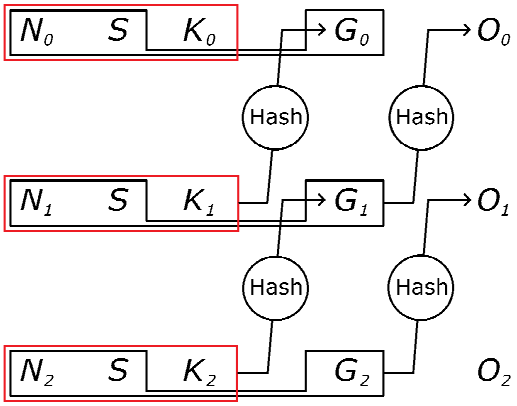

Рисунок 1

Право собственности на токен S с точки зрения пользователя реализуется следующим образом. Предположим, что последняя запись для S имеет вид: N_0 S K_0 G_0 O_0. Пользователь владеет токеном S когда он знает (секретные) ключи K_1 и K_2. Эти ключи, в свою очередь, могут быть сгенерированы из личного *пароля* пользователя по некоторому алгоритму. Например, простой метод генерации ключа может быть таким:

K_i = Hash(i, S , *пароль*)

Существует несколько способов передачи права владения токеном от одного лица другому. Наиболее интересным является сценарий, когда получатель (тот, кто получает токен, т.е. будущий владелец) желает остаться абсолютно анонимным[2]. В этом случае алгоритм передачи следующий. Получатель генерирует K_3, G_2 и O_1, затем отправляет O_1 текущему владельцу токена S (назовем его отправитель). Отправитель публикует запись N_1 S K_1 G_1 O_1, открывая свой первый секретный ключ K_1. После публикации получатель может убедиться, что O_1 появилось в базе данных. В этот момент времени ни отправитель, ни получатель не владеют токеном S, поскольку K_2 известен только отправителю, а K_3 известен только получателю. До тех пор, пока отправитель и получатель не достигли соглашения о продолжении передачи, токен заблокирован. Далее получатель генерирует O_2 (предварительно сгенерировав K_4 и G_3) и посылает G_2 и O_2 отправителю. Отправитель публикует запись N_2 S K_2 G_2 O_2. После публикации этой записи

[2] Анонимность в описываемой схеме отличается от псевдо-анонимности Bitcoin. Публикация записей основана на передаче на обслуживающий базу данных сервер сообщений, содержащих соответствующие записи. Как правило, сервер не требует TCP или других постоянных соединений с клиентом.

получатель становится новым владельцем токена *S*, поскольку только он знает новые секретные ключи K_3 и K_4.

3. Технология и ее применение

Hasq Technology Pty Ltd разработала высокопроизводительную распределенную систему, которая реализует описанную выше схему и решила многие технические вопросы, сопровождавшие разработку соответствующего программного продукта.

Копии базы данных Hasq расположены на серверах сети Hasq, общающихся между собой по простому сетевому протоколу. Ядро протокола составляет набор команд для получения имеющихся и добавления новых записей. Для работы системы не требуется доверие серверов друг другу, каждый из серверов работает независимо. При публикации новых записей сервера отправляют уведомления другим серверам. В силу независимости работы отдельных серверов полная синхронизация копий базы данных не требуется. Однако, предпринимаются все возможные усилия по обеспечению синхронизации на уровне отдельных токенов.

Изолированные сбои серверов или проблемы со связью не влияют на стабильность сети Hasq в целом, поскольку сервера динамически перенастраивают свои связи с другими серверами для поддержания целостности сети как можно дольше.

Предлагается несколько типов серверов для удовлетворения различных требований к аппаратной части. Эффективный контроль размера базы данных сервера также возможен с помощью конфигурационных параметров.

В некоторых случаях может потребоваться поддерживать несколько несовместимых баз данных. Это достигается путем использования различных хэш-функций. Если используется одна и та же хэш-функция, то добавление фиксированной публично известной строки во все хэш-вычисления делает базу данных уникальной.

Схема связи записей Hasq обеспечивает следующие свойства токенов. Токен может:

- быть передан через Интернет или по телефону
- быть передан совершенно анонимно для отправителя или для получателя
- служить банкнотой
- служить облигацией
- служить проверяемым правом собственности
- выступать в роли посредника, дающего доступ к другим электронным данным

Если организация, выпускающая токены, определяет группу неанонимных пользователей, тогда часть токенов может использоваться для учета долга одних пользователей перед другими. Схема учета долга может быть организована следующим образом. Каждый пользователь получает уникальный идентификационный токен. Долг пользователя определяется некоторым текстом в поле данных его идентификационного токена. Например, пользователь *A* имеет перед кем-то долг. Если другой пользователь *B* готов принять этот долг от пользователя *A*, пользователь *B* публикует запись о задолженности в поле данных своего токена. Пользователь *A* публикует ссылку на запись, где пользователь *B* принял задолженность, освобождая таким образом себя от нее.

Описанные выше примеры применения технологии являются лишь частью уже известного значительно более длинного списка применений подобных токенов [8]. Учитывая, что похожие технологии все еще находятся на ранних стадиях своего развития, можно ожидать расширения этого списка с течением времени.

Литература

[1] D. Chaum, *Security without Identification: Transaction Systems to Make Big Brother Obsolete*, 1985
http://dl.acm.org/citation.cfm?id=4373

[2] D. Chaum. A. Fiat and M. Naor, *Untraceable Electronic Cash*, 1988
http://dl.acm.org/citation.cfm?id=88969

[3] B. Hayes, *Anonymous One-Time Signatures and Flexible Untraceable Electronic Cash*, 1990
http://dl.acm.org/citation.cfm?id=716012

[4] T. Okamoto and K. Ohta, *Universal electronic cash*, 1991
http://pdf.aminer.org/000/120/358/universal_electronic_cash.pdf

[5] T. Okamoto, *An efficient divisible electronic cash scheme*, 1995
http://pdf.aminer.org/000/120/348/an_efficient_divisible_electronic_cash_scheme.pdf

[6] T. Sander and A. Ta-Shma, *Auditable, anonymous electronic cash (extended abstract)*, 1999
http://www.cs.tau.ac.il/~amnon/Papers/ST.crypto99.pdf

[7] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2009
https://www.bitcoin.org/bitcoin.pdf

[8] The Mega-Master Blockchain List
http://ledracapital.com/blog/2014/3/11/bitcoin-series-24-the-mega-master-blockchain-list