**3 Компоненты блокчейн-технологии**

 Технология блокчейн может показаться сложной. Однако её проще понять, если рассматривать каждый компонент по отдельности. В общих чертах технология блокчейн использует хорошо известные механизмы вычислительных систем и криптографические примитивы (криптографические хеш-функции, цифровые подписи, асимметричное шифрование) в сочетании с концепциями хранения записей (например, "только добавление" реестров). В этом разделе отдельно рассматриваются все основные компоненты: криптографические хеш-функции, транзакции, асимметричное шифрование, адреса, реестры, блоки и то, как блоки связываются в цепочку.

**3.1 Криптографические хеш-функции**

 Важной составляющей технологии блокчейн является использование криптографических хеш-функций во многих операциях. *Хеширование* (*hashing*) – это метод применения криптографической хеш-функции к данным, которая вычисляет уникальное выходное значение, называемое *дайджестом сообщения*, или просто *дайджестом*, для входных данных почти любого размера (например, файлов, текста или изображений). Это даёт отдельным лицам возможность самостоятельно брать входные данные, хешировать эти данные и получать одинаковый результат при каждом хешировании, доказывая тем самым, что никаких изменений в данные внесено не было. Даже самое незначительное изменение на входе (например, изменение одного бита) приведёт к совершенно иному выходному дайджесту. Это наглядно проиллюстрировано в таблице 1.

Криптографические хеш-функции отвечают следующим важным свойствам стойкости:

1. *Устойчивость к поиску прообраза*. Это означает, что хеш-функция – односторонняя: с вычислительной точки зрения невозможно найти правильное входное значение при заданном выходном значении (то есть при заданном дайджесте, найти такое *x*, что *hash (x) = дайджест*).

2. *Устойчивость к поиску второго прообраза*. Это означает, что невозможно найти какой-либо вход, который хешируется в определённый выход. Если говорить более конкретно, то криптографические хеш-функции устроены таким образом, что, имея некоторый заданный вход, невозможно с вычислительной точки зрения найти второй такой вход, который даёт тот же выход (то есть при заданном *x*, невозможно найти такой *y*, при котором *hash (x) = hash (y)*). Единственно возможный подход – это исчерпывающий поиск ключей, но с вычислительной точки зрения это невозможно сделать с какими-либо шансами на успех.

3. *Устойчивость к коллизии*. Это означает, что невозможно найти два входа, которые хешируются в один и тот же выход. Если более конкретно, то с вычислительной точки зрения невозможно найти два таких входа, которые дают один и тот же дайджест (то есть найти *x* и *y*, при которых *hash (x) = hash (y)*).

 Специальной криптографической хеш-функцией, используемой во многих реализациях блокчейна, является алгоритм безопасного хеширования (SHA) с размером выхода в 256 бит (SHA-256). Многие компьютеры поддерживают этот алгоритм аппаратно, повышая скорость вычисления. SHA-256 имеет выход размером в 32 байта (1 байт = 8 бит, 32 байта = 256 бит), обычно отображаемый как 64-символьная шестнадцатеричная строка (см. таблицу 1).

Это означает, что существует 2256 ≈ 1077, или 115 792 089 237 316 195 423 570 985 008 687 907 853 269 984 665 640 564 039 457 584 007 913 129 639 936 возможных значений дайджеста. Алгоритм SHA-256, как и другие алгоритмы, опубликован в Федеральном стандарте обработки информации FIPS 180-4. Сайт Стандарта безопасного хеширования Национального Института Стандартов и Технологий США (NIST) содержит спецификации FIPS для всех алгоритмов хеширования, утверждённых NIST.

 Поскольку существует бесконечное множество возможных входных значений и конечное множество возможных значений выходного дайджеста, коллизия, при которой *hash (x) = hash (y)* (то есть хеш двух разных входов даёт один и тот же дайджест), возможна, но крайне маловероятна. Считается, что SHA-256 устойчив к коллизиям, поскольку для того, чтобы найти коллизию в SHA-256, нужно выполнить алгоритм в среднем около 2128 раз (что составляет 340 ундециллионов, или, точнее, 340 282 366 920 938 463 463 374 607 431 768 211 456 раз, примерно 3,402 x 1038 раз).

Для сравнения: скорость хеширования всей сети Биткойн в 2015 году составляла 300 квадриллионов хешей в секунду (300 000 000 000 000 000 000/с). При такой скорости для всей сети Биткойн потребуется примерно 35 942 991 748 521 (примерно 3,6 x 1013) год2 для порождения коллизии (заметим, что возраст Вселенной оценивается в 1,37 x 1010 лет)3. Даже если найдутся такие входные значения *x* и *y*, которые производят один и тот же дайджест, будет также очень маловероятно, что оба этих входа будут допустимы в контексте сети блокчейн (то есть и *x* и *y* будут валидными транзакциями).

В сети блокчейн криптографические хеш-функции используются для решения множества задач, таких как:

* Получение адреса (рассматривается в главе 3.4);
* Создание уникальных идентификаторов;
* Защита данных блока – публикующий узел хеширует данные блока, создавая дайджест, который будет храниться в заголовке блока;
* Защита заголовка блока – публикующий узел хеширует заголовок блока. Если в сети блокчейн используется консенсусная модель "доказательства работы" (см. главу 4.1), публикующему узлу потребуется хешировать заголовок блока с разными значениями *nonce* (см. пункт 3.1.1), пока не будут выполнены условия головоломки. Хеш-дайджест заголовка текущего блока будет включён в заголовок следующего блока, где он обеспечит безопасность данных заголовка текущего блока.

Поскольку заголовок блока включает хеш-представление данных блока, сами данные блока также защищены, так как дайджест заголовка блока сохраняется в следующем блоке. Существует множество семейств криптографических хеш-функций, используемых в технологии блокчейн (SHA-256 – не единственная), например, алгоритм хеширования Keccak (выбранный NIST в качестве победителя конкурса на создание стандарта хеширования SHA-3) применяется также широко, как и RIPEMD-160.

**3.1.1 Криптографическое число *nonce***

Криптографическое число *nonce* – это произвольное число, использующееся только один раз. Криптографическое число *nonce* можно комбинировать с данными для создания различных хеш-дайджестов для каждого значения *nonce*:

*hash (данные + nonce) = дайджест*

Лишь изменение значения *nonce* обеспечивает механизм для получения разных значений дайджеста при сохранении одних и тех же данных. Эта методика используется в консенсусной модели "доказательства работы" (глава 4.1).

**3.2 Транзакции**

Транзакция представляет собой взаимодействие между участниками сети. В случае с криптовалютами, например, транзакция представляет собой пересылку криптовалюты между пользователями сети блокчейн. Для сценариев «бизнес для бизнеса» транзакция может быть способом регистрации операций, связанных с цифровыми или физическими активами. На рисунке 1 изображён условный пример транзакции с криптовалютой. Каждый блок в блокчейне может содержать ноль или более транзакций. Для некоторых реализаций блокчейна непрерывное добавление новых блоков (даже с нулевыми транзакциями) является критически важным для поддержания безопасности сети блокчейна. Благодаря непрерывному добавлению новых блоков, злоумышленники не могут «догнать» и создать более длинную изменённую цепочку блокчейна (глава 4.7).

**Рисунок 1 –** Пример криптовалютной транзакции

Данные, формирующие транзакцию, могут быть разными в каждой реализации блокчейна, однако механизм проведения транзакций в основном одинаков. Пользователь посылает информацию в сеть блокчейн. Посылаемая информация может включать: адрес отправителя (или другой релевантный идентификатор), открытый ключ отправителя, цифровую подпись, входы транзакции и выходы транзакции.

Одна криптовалютная транзакция обычно требует, как минимум, следующей информации, но может включать и больше:

* *Входы*. Входы, как правило, представляют собой список цифровых активов, подлежащих передаче. Транзакция будет ссылаться на источник цифрового актива (указание происхождения), в качестве которого может выступать предыдущая транзакция, в которой актив был передан отправителю, либо, в случае новых цифровых активов – событие "зарождения". Так как входом для транзакции является ссылка на прошлые события, цифровые активы не изменяются. В случае криптовалют это означает, что стоимость не может быть добавлена или вычтена из существующих цифровых активов. Вместо этого один цифровой актив можно разделить на несколько новых цифровых активов (каждый с меньшей стоимостью) или несколько цифровых активов можно объединить, чтобы сформировать меньшее количество новых цифровых активов (соответственно с большей стоимостью). Разделение или объединение активов будет указано в выходе транзакции. Отправитель также должен предоставить доказательство того, что у него есть доступ к указанным входам, как правило, цифровой подписью транзакции, подтверждая доступ к закрытому ключу.
* *Выходы*. Выходами обычно считаются аккаунты, выступающие в качестве получателей цифровых активов, а также количество цифровых активов, которые они получают. Каждый выход содержит количество цифровых активов, которые должны быть переданы новому владельцу (владельцам), идентификатор нового владельца (владельцев) и перечень условий, которым должны следовать новые владельцы, чтобы использовать эти активы. Если переданных цифровых активов больше, чем требуется, лишние средства должны быть явным образом отосланы обратно отправителю (так называемый механизм *«сдачи»*).

Хотя транзакции преимущественно используются для передачи цифровых активов, в более общем представлении они могут использоваться для передачи данных. Например, кто-то просто может захотеть перманентно и публично разместить данные в блокчейне. В случае систем смарт-контрактов транзакции могут использоваться для отправки данных, обработки этих данных и сохранения результата в блокчейне. Например, транзакция может быть использована для изменения атрибута оцифрованного актива, такого как местоположение груза в системе управления цепочками поставок на основе блокчейн-технологии (BCTeSCS).

 Независимо от того, как формируются и передаются данные, определение валидности и подлинности транзакции имеет важное значение. Валидность транзакции гарантирует, что транзакция соответствует требованиям протокола и любым формализованным форматам данных или требованиям смарт-контрактов, присущим реализации блокчейна. Подлинность транзакции также важна, поскольку она подтверждает, что отправитель цифровых активов имел доступ к этим цифровым активам. Транзакции, как правило, подписываются цифровой подписью с помощью закрытого ключа отправителя (криптография с асимметричным ключом кратко обсуждается в главе 3.3), и могут быть в любое время верифицированы с помощью соответствующего открытого ключа.

**3.3 Асимметричное шифрование**

………………………………………………………………………………………………..

**Источник:**

D. Yaga, P. Mell, N. Roby, K. Scarfone «Blockchain Technology Overview». National Institute of Standards and Technology, U.S. Department of Commerce, 2018. <https://doi.org/10.6028/NIST.IR.8202>

**Перевод:** Алексей Виноградов.