Техническое описание транзакции Биткоина

В блокноте Jupyter мы собираемся создать, сгенерировать цифровую подпись и привести пример биткойн-транзакции.

Создание криптоидентификации

Сначала генерируется совершенно новый криптографический идентификатор, который представляет собой закрытую, общедоступную пару ключей. Биткойн использует криптографию с эллиптической кривой вместо чего-то более распространенного, такого как RSA, для защиты транзакций.

В дополнение к фактической кривой определяется точка генератора, которая является просто некоторой фиксированной "начальной точкой" в цикле кривой, которая используется для запуска "случайного блуждания" по кривой. Генератор является общеизвестной и согласованной константой.

Наконец, порядок порождающей точки G известен и фактически является "размером набора", с которым мы работаем в терминах целых кортежей (x, y) в цикле вокруг кривой.

Мы готовы сгенерировать наш закрытый ключ. Закрытый ключ (или "секретный ключ", как я буду называть его в дальнейшем) - это просто случайное целое число, которое удовлетворяет 1 <= ключ < n (напомним, что n - это порядок G)

Секретный ключ - это целое число, и любой, кто его знает, может контролировать все средства, которыми вы владеете в блокчейне Биткойна, связанные с ним.

Далее генерируется открытый ключ. Открытый ключ - это точка на кривой, которая получается в результате добавления точки генератора к самому себе secret_key times. т.е. мы имеем: public_key = G + G + G + (время секретного ключа) + G = secret_key * G.

Секретный ключ представляет собой целое число, но точка генератора G представляет собой кортеж (x,y), который является точкой на кривой, в результате чего получается открытый ключ кортежа (x,y), опять же точка на кривой. Именно здесь надо фактически определить оператор сложения на эллиптической кривой.

С помощью пары закрытый / открытый ключ генерируется криптоидентификация. Далее надо получить соответствующий адрес

биткоин-кошелька. Адрес кошелька - это не просто сам открытый ключ, но он может быть детерминированно получен из него и имеет несколько дополнительных преимуществ (например, встроенную контрольную сумму). Однако, прежде чем мы сможем сгенерировать адрес, нам нужно определить некоторые хэш-функции. Биткойн использует вездесущий SHA-256, а также RIPEMD-160, о которых будет рассказано в описании алгоритмов.

Итак, мы можем получить ваш биткоин-адрес. Создаем подкласс Point под названием PublicKey, который, опять же, является просто точкой на кривой, но теперь имеет некоторую дополнительную семантику и интерпретацию открытого ключа Биткойна, а также некоторые методы кодирования / декодирования ключа в байты для связи в протоколе Биткойна.

Мы генерируем криптоидентификацию, состоящую из секретного ключа (случайного целого числа), который знаем только мы, и производного открытого ключа, прыгая вокруг эллиптической кривой, используя скалярное умножение генерирующей точки на эллиптической кривой Биткойна. Затем мы также получили соответствующий биткойн-адрес, которым мы можем поделиться с другими, чтобы запросить деньги, и для этого было введено две хэш-функции (SHA256 и RIPEMD160).