Аннотация

Этот BIP определяет логическую иерархию для детерминированных кошельков на основе алгоритма, описанного в BIP-0032 (далее BIP32) и схемы назначения, описанной в BIP-0043 (далее BIP43).

Этот BIP является частным применением BIP43.

Уровни пути

Мы определяем следующие 5 уровней в пути BIP32:

```
m / purpose' / coin_type' / account' / change / address_index
```

Апостроф в пути указывает на использование жесткого (hardened) производного BIP32.

Каждый уровень имеет специальное значение, описанное в главах ниже.

Назначение

Назначение - это константа, установленная на 44' (или 0x8000002C) согласно рекомендации BIP43. Она указывает, что поддерево этого узла используется согласно этой спецификации.

На этом уровне используется жесткое производное.

Тип монеты

Один мастер-узел (seed) может использоваться для неограниченного числа независимых криптовалют, таких как Bitcoin, Litecoin или Namecoin. Однако, совместное использование одного пространства для различных криптовалют имеет некоторые недостатки.

Этот уровень создает отдельное поддерево для каждой криптовалюты, избегая повторного использования адресов между криптовалютами и улучшая вопросы конфиденциальности.

Тип монеты - это константа, установленная для каждой криптовалюты. Разработчики криптовалют могут запросить регистрацию неиспользованного номера для своего проекта.

Список уже зарегистрированных типов монет находится в главе "Зарегистрированные типы монет" ниже.

На этом уровне используется жесткое производное.

Аккаунт

Этот уровень разделяет пространство ключей на независимые пользовательские идентичности, так что кошелек никогда не смешивает монеты между разными аккаунтами.

Пользователи могут использовать эти аккаунты для организации средств так же, как банковские счета; для пожертвований (где все адреса считаются публичными), для сбережений, для общих расходов и т.д.

Аккаунты нумеруются с индекса 0 в порядке возрастания. Этот номер используется как индекс дочернего элемента в производном BIP32.

На этом уровне используется жесткое производное.

Программное обеспечение должно предотвращать создание аккаунта, если у предыдущего аккаунта нет истории транзакций (то есть ни один из его адресов не был использован ранее).

Программное обеспечение должно обнаружить все использованные аккаунты после импорта seed с внешнего источника. Такой алгоритм описан в главе "Обнаружение аккаунтов".

Цепочка

Константа 0 используется для внешней цепочки и константа 1 для внутренней цепочки (также известных как адреса для сдачи). Внешняя цепочка используется для адресов, которые должны быть видны за пределами кошелька (например, для получения платежей). Внутренняя цепочка используется для адресов, которые не должны быть видны за пределами кошелька и используется для сдачи по транзакциям.

На этом уровне используется публичное производное.

Индекс

Адреса нумеруются с индекса 0 в порядке возрастания. Этот номер используется как индекс дочернего элемента в производном BIP32.

На этом уровне используется публичное производное.

Обнаружение аккаунтов

Когда мастер-seed импортируется с внешнего источника, программное обеспечение должно начать обнаруживать аккаунты следующим образом:

1. Произвести узел первого аккаунта (index = 0)
2. Произвести узел внешней цепочки этого аккаунта
3. Сканировать адреса внешней цепочки; соблюдать предел разрыва, описанный ниже
4. Если транзакции на внешней цепочке не найдены, прекратить обнаружение
5. Если есть транзакции, увеличить индекс аккаунта и перейти к шагу 1

Этот алгоритм успешен, потому что программное обеспечение должно запрещать создание новых аккаунтов, если у предыдущего нет истории транзакций, как описано в главе "Аккаунт" выше.

Обратите внимание, что алгоритм работает с историей транзакций, а не с балансами аккаунтов, так что вы можете иметь аккаунт с нулевым балансом, и алгоритм все равно продолжит обнаружение.

Предел разрыва адресов

Предел разрыва адресов в настоящее время установлен на 20. Если программное обеспечение достигает 20 неиспользуемых адресов подряд, оно предполагает, что дальше нет используемых адресов, и прекращает поиск по цепочке адресов. Мы сканируем только внешние цепочки, потому что внутренние цепочки получают только монеты, которые приходят от связанных внешних цепочек.

Программное обеспечение кошелька должно предупреждать, когда пользователь пытается превысить предел разрыва на внешней цепочке, создавая новый адрес.

Зарегистрированные типы монет

Это стандартные зарегистрированные типы монет для использования на уровне 2 BIP44, описанные в главе "Тип монеты" выше.

Все эти константы используются как жесткое производное.

Индекс	Гексадецимальный	Монета
0	0x80000000	Bitcoin
1	0x80000001	Bitcoin Testnet

Этот BIP не является центральным реестром для зарегистрированных типов монет, пожалуйста, посетите SatoshiLabs, который поддерживает полный список:

SLIP-0044: Registered coin types for BIP-0044

Для регистрации нового типа монеты требуется существующий кошелек, который реализует стандарт, и необходимо создать pull request в вышеуказанный файл.

Примеры

Монета	Аккаунт	Цепочка	Адрес	Путь
Bitcoin	первый	внешняя	первый	m / 44' / 0' / 0' / 0 / 0
Bitcoin	первый	внешняя	второй	m / 44' / 0' / 0' / 0 / 1
Bitcoin	первый	сдача	первый	m / 44' / 0' / 0' / 1 / 0
Bitcoin	первый	сдача	второй	m / 44' / 0' / 0' / 1 / 1
Bitcoin	второй	внешняя	первый	m / 44' / 0' / 1' / 0 / 0
Bitcoin	второй	внешняя	второй	m / 44' / 0' / 1' / 0 / 1
Bitcoin	второй	сдача	первый	m / 44' / 0' / 1' / 1 / 0
Bitcoin	второй	сдача	второй	m / 44' / 0' / 1' / 1 / 1
Bitcoin Testnet	первый	внешняя	первый	m / 44' / 1' / 0' / 0 / 0
Bitcoin Testnet	первый	внешняя	второй	m / 44' / 1' / 0' / 0 / 1
Bitcoin Testnet	первый	сдача	первый	m / 44' / 1' / 0' / 1 / 0
Bitcoin Testnet	первый	сдача	второй	m / 44' / 1' / 0' / 1 / 1
Bitcoin Testnet	второй	внешняя	первый	m / 44' / 1' / 1' / 0 / 0
Bitcoin Testnet	второй	внешняя	второй	m / 44' / 1' / 1' / 0 / 1
Bitcoin Testnet	второй	сдача	первый	m / 44' / 1' / 1' / 1 / 0
Bitcoin Testnet	второй	сдача	второй	m / 44' / 1' / 1' / 1 / 1

Справочник

SLIP-0044: Registered coin types for BIP-0044