УДК 004.056.3

МЕТОД ВОССТАНОВЛЕНИЯ КРИПТОКОШЕЛЬКА С ПОМОЩЬЮ SEED-ФРАЗЫ

Карачун М.А., Бессонов А.В., Черкесова Л.В., Короченцев Д.А., Поркшеян В.М.
ФГБОУ ВПО «Донской государственный технический университет», Ростов-на-Дону, e-mail: karachun.m@mail.ru

Статья посвящена обзору технологии восстановления криптокошелька с помощью специальных «seed-фраз», а также её реализации на языке JavaScript. Актуальность статьи обосновывается нарастающими темпами развития и применения криптовалют в сфере экономики, а также сложностью освоения всех принципов работы с данной валютой для рядового пользователя. В ходе работы авторы рассматривают существующие меры безопасности, необходимые для защиты криптокошельков от угроз утери, хищения в случае взлома, либо потери аутентификационных данных. Учитывая все особенности работы криптовалют, а также механизм их оборота и хранения, становится ясным, что необходима разработка специальных мер и алгоритмов для обеспечения целостности и сохранности средств пользователей данной цифровой валюты. Основное содержание исследования составляет анализ алгоритма формирования «seed-фраз» из уже имеющихся алгоритмов генерации ключей и его применение на примере существующего криптокошелька с полным восстановлением всех транзакций. Выделяются и описываются характерные особенности взаимодействия с данной технологией как от лица пользователя, так и от лица разработчика. К главным достоинствам данного метода относятся: безопасность, простота в использовании и интеграции, кроссплатформенность.

Ключевые слова: криптовалюта, криптокошелек, меры безопасности, seed-фраза, мастер-ключ

THE METHOD OF RESTORING A CRYPTOWALLET USING A SEED PHRASE

Karachun M.A., Bessonov A.V., Cherkesova L.V., Korochentsev D.A., Porksheyan V.M.
Don State Technical University, Rostov-on-Don, e-mail: karachun.m@mail.ru

The article is devoted to the review of the recovery technology of the cryptowallet with the help of special «seed-phrases», as well as its implementation in the JavaScript language. The relevance of the article is justified by the growing pace of development and use of cryptocurrency in the economic sphere, as well as the complexity of mastering all the principles of working with this currency for the average user. In the course of the work, the authors examine the existing security measures, which are necessary to protect the cryptographic wallet from threats of loss or theft in the event of hacking or loss of authentication data. Taking into account all the features of the operation of cryptocurrencies, as well as the mechanism of their circulation and storage, it becomes clear that the development of special measures and algorithms is necessary to ensure the integrity and security of the users of this digital currency. The main content of the study is an analysis of the algorithm for the formation of "seed-phrases" of the already existing key generation algorithms and its application using the example of an existing cryptowallet with full recovery of all transactions. Characteristic features of interaction with this technology both on behalf of the user and on behalf of the developer are identified and described. The main advantages of this method are: security, ease of use and integration, cross-platform.

Keywords: cryptocurrency, cryptocurrency wallet, safety measures, seed phrase, master key

Криптовалюты всё более нарастающим темпом занимают важную позицию как элемент финансовой системы современного мира. Но вместе с масштабами их использования и оборота, растут и риски, связанные с их кражей, уничтожением и потерей. Криптоналичность не имеет никаких воплощений, и ее невозможно хранить в сейфе или в другом укромном месте – криптокоды существуют только в памяти жестких дисков. С одной стороны, без вашего пароля доступ к таким деньгам не может получить никто, что исключает возможные кражи [1]. В то же время потеря пароля или самого жесткого диска приведет к потере ваших финансовых средств. Ввиду этих обстоятельств необходима разработка и внедрение специальных методов защиты, позволяющих обеспечить их безопасность и сохранность при проведении операций и хранении.

Многие меры защиты, используемые для криптовалют, схожи с мерами, применяемыми для защиты аккаунтов на сайтах, приложениях, например: двухфакторная авторизация, использование длинных и сложных паролей, хранение паролей на бумажном носителе, использование разных паролей для разных кошельков. Но есть и специфичные меры, к которым относят: создание backup сервера для хранения и восстановления кошелька, хранение кошелька в офлайн-режиме, использование отдельных специально предназначенных аппаратных средств для проведения транзакций [2].

Из написанного выше следует, что ввиду высоких рисков потери или кражи криптовалюты изучение методов защиты криптокошельков является необходимым условием для их полноценного и уверенного использования.

В данной статье будет рассмотрен метод восстановления криптокошелька с помощью seed-фразы (seed-ключа). Seed-фраза представляет собой последовательность из 12 (или другого количества, в зависимости от реализации) английских слов. Главным достоинством метода восстановления с помощью seed-фразы является простота использования и высокая надежность, так как seed-фраза дает возможность сгенерировать в любой момент времени криптокошелек, который по своей сути является последовательностью закрытых ключей, длиной 256 бит каждый. Пример приватного ключа в формате WIF: 5J3mBbAH58CpQ3Y5RNJpPUK. Из него генерируется открытый ключ длиной 512 бит. Затем от открытого ключа вычисляется хэш. Данный хэш является bitcoin адресом, вы можете сообщить его любому желающему перевести вам криптовалюту. Пример bitcoin адреса: 1CiBY4gUTcHxjELDLcds59pjMy2aoNP4fy.

Цель исследования: изучение seed-фраз и их реализация на языке JavaScript.

В процессе достижения поставленной цели авторами были сформулированы и успешно решены следующие задачи: реализация seed-ключа, восстановление кошелька с помощью seed-ключа.

Материалы и методы исследования

Большинство криптокошельков имеют встроенную опцию для восстановления. Восстановление происходит следующим образом: кошелек генерирует пользователю набор случайных английских слов, зачастую их 12, иногда 18 или 24. Пример такой последовательности слов: «happy radiance january light day swine mirror paper shirt screen yellow table». На первый взгляд такая цепочка представляет собой бессмысленный набор случайных слов, но на самом деле она является ключом к восстановлению кошелька вне зависимости от количества транзакций либо средств, хранящихся на нём [3].

Сам же seed-ключ можно хранить на любом виде носителей информации: листок бумаги, гравировка на металле, хранить в памяти. Самым важным условием будет являться то, что нужно обеспечить доступность и сохранность самого носителя, тогда можно не беспокоиться о сохранности своих денег в случае пожара, вывода оборудования из строя и каких-либо других происшествий.

В принципе для использования кошелька будет достаточно одной лишь seed-фразы, так как кошелек можно будет восстановить в любой точке мира, лишь бы имелся доступ к сети Интернет.

Восстановление кошелька происходит довольно просто: загружается программа-кошелёк с поддержкой данной технологии. Список подходящих кошельков довольно широкий, так как сама технология является распространенной. Примеры таких кошельков: Airbitz, Bither, BreadWallet, Coin.Space, Coinomi, CoPay, Electrum, Exodus, GreenAddress, Jaxx, Ledger, MultiBit, Mycelium, Simple Bitcoin Wallet, Trezor. Большинство из этих кошельков потребуют создания seed-ключа перед открытием первого кошелька. В некоторых из них необходимо искать эту опцию самостоятельно.

Сам seed-ключ не дает полного представления о кошельке и его содержимом. Существует криптографическая процедура, которая называется «иерархически детерминированный кошелёк». Она переводит seed-ключ в Мастер-ключ, из которого все остальные ключи развёртываются в «детерминистическом» порядке. Так как seed-ключ является математически генерируемой детерминированной последовательностью [4], то он работает независимо от типа кошелька. Единственная проблема состоит в том, что существует два типа стандартов. В то время как ключевая фраза (seed) и Мастер-ключ одинаково стандартизированы, существует два широко применяемых алгоритма восстановления ключей и адреса кошелька, они называются BIP32 и BIP44. Так что, если бы seed был создан, используя кошелек с поддержкой BIP44, вроде Bither, то при попытке восстановления его при помощи кошелька с BIP32, типа Electrum, будет открыт пустой кошелёк. Поэтому важно знать, какой кошелек подходит для восстановления из вашей seed-фразы. Кошельки с соответствующими им алгоритмами представлены в таблице.

Применяемые алгоритмы
генерации seed-фраз в кошельках

Кошелек	BIP32	BIP44
AirBitz	x	
BreadWallet	x	
Coin.Space	x	
CoPay		x
Electrum	x	
Exodus		x
Jaxx		x
Ledger		x
MultiBit	x	x
Trezor		x

Помимо этого, существует несколько реализаций метода. Например, некоторые кошельки вроде Ledger используют фразу

из 24 слов, тогда как другие, вроде Exodus, будут использовать SEED из 12 слов. SEED из 12 слов несовместим с кошельком, поддерживающим SEED из 24 слов. Также не все кошельки используют и понимают тот же словарный набор, а у некоторых отсутствует правильная реализация путей развёртывания; например Exodus может восстановить лишь первые 4 адреса, созданные с помощью Мастер-ключа, а на Coin.Space нет возможности восстановить адреса, созданные при помощи другого кошелька, и так далее.

Существует двухфакторная генерация seed-фраз. Она заключается в использовании последовательности из 13/25 слов, причем последнее из них создается самим пользователем в качестве пароля. Некоторые кошельки используют собственные дополненные реализации алгоритма генерации seed-ключа, другие используют стандарт BIP39 для его расширения. Но важно понимать, что при потере слова-пароля восстановление кошелька уже не представляется возможным.

Чтобы осознать принцип работы ключевой фразы, нужно больше знать о том, как хранятся криптовалюты. Монеты получаются при помощи адреса. Этот адрес образуется из публичных ключей, а они, в свою очередь, образуются из приватных ключей. Таким образом, сначала кошелёк сгенерирует приватный ключ, затем он разворачивает публичный ключ, а потом он трансформирует публичный ключ в адрес.

Очень простой способ сохранить монеты – просто записать или сохранить приватный ключ в зашифрованный файл. Приватный ключ похож на случайный набор цифр и букв: L3GrBerZZXtTDcAVNiULbN84UVGjX7ezypSCsYYroBDdQDKX1E53.

Для улучшения приватности и большей сохранности от коллизий большинство продвинутых кошельков создают новый адрес при проведении транзакции. В таком случае, чтобы сделать бэкап средств, пришлось бы сохранять приватные ключи каждый раз, когда создаётся новая транзакция.

В связи с этой проблемой, в 2012 г. один из разработчиков «Bitcoin» Питер Вуйле написал функцию, названную «Иерархически Детерминированные Кошельки», известную как BIP32. Он разработал математику, позволяющую создавать Мастер-ключ, из которого все остальные приватные ключи могут быть развёрнуты в предопределённом порядке [5]. То есть, если у нас есть Мастер-ключ A, то из него можно получить ключи a, b, c, d, и так далее, именно в таком порядке. Метод Питера Вуйле основан на криптографии с открытым ключом и эллиптических кривых. Разработчики, которые пришли позднее, например Марек Палатинус, из чешского Биткойн-стартапа Satoshi Labs, разработали инструменты для разворачивания Мастер-ключа из seed в 12 или 24 слова (BIP39), а также инструменты для создания ключей с поддержкой нескольких аккаунтов (BIP44), что будет означать, что Мастер-ключ A производит аккаунты a, b, c и так далее, что является фундаментальной функцией кошельков, направленной на защиту пользовательской анонимности в условиях интенсивного сбора данных с блокчейна.

Рассмотрим пример работы алгоритма генерации seed-фразы. Первым делом вводятся публичный и частные ключи, а также вся дополнительная информация о кошельке (аккаунте), которая предоставляется либо пользователем, либо платежным сервисом. Далее, на основе этой информации при помощи алгоритма BIP39 формируется seed в base64 формате, который преобразуется в последовательность из 12 слов, как показано на рис. 1 и в листинге 1.

Рис. 1. Генерация мнемонической последовательности

Листинг 1

```
function mnemonToSeed(mnemonic, pass) {
    return new Promise((resolve, reject) => {
        try {
          const mnemonBuff = Buffer.from((mnemonic || ‘’).normalize(‘NFKD’), ‘utf8’);
          const saltBuff = Buffer.from(salt((password || ‘’).normalize(‘NFKD’)), ‘utf8’);
           pbkdf2.pbkdf2(mnemonBuff, saltBuff, 2048, 64, ‘sha512’, (err, data) => {
                if (err)
                    return reject(err);
                else
                    return resolve(data);
            });
        }
        catch (error) {
            return reject(error);
        }
    });
}
```

Из этой последовательности можно получить ключ BIP32, по которому можно будет полностью восстановить кошелек, как показано на рис. 2 и в листинге 2.

Рис. 2. Воссоздание BIP32-ключа

Листинг 2

```
function fromSeed(seed, netw) {
    typeforce(typeforce.Buffer, seed);
    netw = netw || BITCOIN;
    const I = crpt.hmacSHA512(Buffer.from(‘Bitcoin seed’, ‘utf8’), seed);
    const ILEFT = I.slice(0, 32);
    const IRIGHT = I.slice(32);
    return fromPrivKey(ILEFT, IRIGHT, network);
}
```

Все операции кошелька также сохраняются в данном ключе, как можно увидеть на рис. 3.

Path Toggle	Address Toggle	Private Key Toggle
m/44'/0'/0'/0/0	ikZ25f8ic3oua5Cdx4nPYcCauGnqCcTU8J	PhDF6UeU4Dud9fwKUvW4BVqAxPVWPtEbzPosbejYNWUG4HfTSQLH
m/44'/0'/0'/0/1	iXbpMqtcFsKpsFviUXBHfJr9mD8puNvC5j	PkmWWnPY9Gag2eJUQnMU9NWfb3iCBYUgyL1fLMYNjJf1GgEYCv6M
m/44'/0'/0'/0/2	ihJVQRqbJKRygsBwTZFd2gpd26x8dNpfsi	PmWBRvuEitZfE4jy88qwSRGyfHaqForWY78PYesU3JP9MxGWPycN
m/44'/0'/0'/0/3	icanzq8jQNy9BY8ptjWQKa7EkaxFNfj4kj	Pjwpa8padA5P8xEb6UjvaGEAyHH3EYMx5sisCsUy6wFQEpy4kPq7

Рис. 3. Список транзакций, полученный из BIP32-ключа

Результаты исследования и их обсуждение

Отметим достоинства приведенной выше реализации метода:

1. Безопасность – так как в основе метода лежит криптография с открытым ключом на эллиптических кривых, то при наличии открытого ключа злоумышленник не сможет найти соответствующий закрытый ключ более эффективно, чем путем решения проблемы дискретного логарифма на эллиптических кривых (предполагается, что потребуется 2^{128} операций над точками эллиптической кривой) [6, 7]. Также возможность хранения кошелька в виде seed-фразы позволяет свести защиту своей криптовалюты к обеспечению конфиденциальности последовательности из двенадцати слов.

2. Простота использования для пользователя, которому необходимо только запустить программу. Та в свою очередь сгенерирует кошелек и соответствующую ему seed-фразу, которую пользователь в дальнейшем может хранить как ему удобно (например, записать на бумаге или запомнить).

3. Кроссплатформенность – написанный на JavaScript код можно запустить на любом компьютере с установленным современным браузером.

4. Простота внедрения данного метода в любое программное средство для работы с криптовалютой.

Заключение

В итоге можно сделать вывод, что данный алгоритм позволяет пользователю хранить свой кошелек в виде последовательности из нескольких слов, из которых он гарантированно может восстановить кошелек в полной целостности. От пользователя требуется только лишь обеспечить необходимые меры по защите seed-фразы от преднамеренной кражи и случайной утери, тогда он будет обладать мощным механизмом по защите своего криптокошелька. В статье также была показана реализация метода генерации кошелька с использованием seed-фразы, которая обеспечивает необходимую безопасность, простоту использования и внедрения, а также является кроссплатформенной.

Список литературы

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2009. [Electronic resource]. URL: http://bitcoin.org/bitcoin.pdf (date of access: 25.05.2019).

2. Mauro Conti, Sandeep Kumar E., Chhagan Lal, Sushmita Ruj. A Survey on Security and Privacy Issues of Bitcoin. IEEE Communications Surveys & Tutorials. 2018. vol. 20. no. 4. P. 3416–3452. DOI: 10.1109/COMST.2018.2842460.

3. Mukhopadhyay U., Skjellum A., Hambolu O., Oakley J., Yu L., Brooks R. A brief survey of cryptocurrency systems. 14th Annual Conference on Privacy, Security and Trust (PST) (Auckland, New Zealand, 12–14 December 2016). DOI: 10.1109/PST.2016.7906988.

4. Khovratovich D., Jason Law. BIP32-Ed25519 Hierarchical Deterministic Keys over a Non-linear Keyspace // IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (Paris, France, 26–28 April 2017). DOI: 10.1109/EuroSPW.2017.47.

5. Pieter Wuille. BIP 32: Hierarchical deterministic wallets. 2012. [Electronic resource]. URL: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki (date of access: 25.05.2019).

6. Жданов О.Н., Чалкин В.А. Эллиптические кривые: Основы теории и криптографические приложения. М.: «ЛИБРОКОМ», 2012. 193 с.

7. Болотов А.А., Гашков С.Б., Фролов А.Б. Элементарное введение в эллиптическую криптографию // Протоколы криптографии на эллиптических кривых. М.: «КомКнига», 2012. 306 с.