Роман А. Шарапов
Московский физико-технический институт,
Институтский пер., 9, г. Долгопрудный, 141701, Россия
e-mail: sharapov.roman@gmail.com, https://orcid.org/0000-0002-2378-0260

ХОЛОДНЫЙ МУЛЬТИВАЛЮТНЫЙ КОШЕЛЕК НА ПЛАТФОРМЕ МКТ

DOI: http://dx.doi.org/10.26583/bit.2019.2.03

Аннотация. В настоящей статье исследуется концепт реализации защищённого мультивалютного холодного кошелька на базе микрокомпьютера с динамически изменяемой архитектурой МКТ. Данное устройство создано на базе новой гарвадской архитектуры, модификации классической гарвардской архитектуры, с использованием неизменяемой памяти (для обеспечения целостности и неизменяемости операционной системы) и блоков сеансовой памяти (для обеспечения корректной работы программного обеспечения). В статье описывается механизм работы блокчейна и блокчейн-транзакций, рассмотрены основные принципы хранения криптовалют. Рассмотрена имплементация в систему алгоритма офлайн-подписи транзакций и протоколов, обеспечивающих иерархическую генерацию ключей по технологии HD Wallet. Предложено разделение функционала мультивалютного кошелька на две группы, предназначенные для исполнения либо в операционной системе, доступной только для чтения либо — доступной и для чтения, и для записи. Расписан сценарий исполнения рабочего цикла спроектированной системы.
Ключевые слова: Новая гарвардская архитектура, динамически изменяемая архитектура, блокчейн, офлайн-подпись транзакций, HD wallet.
Для цитирования: ШАРАПОВ, Роман А. ХОЛОДНЫЙ МУЛЬТИВАЛЮТНЫЙ КОШЕЛЕК НА ПЛАТФОРМЕ МКТ. Безопасность Информационных Технологий, [S.l.], v. 26, n. 3, p. 32-44, 2019. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1215>. Дата доступа: 11 sep. 2019. doi:http://dx.doi.org/10.26583/bit.2019.3.03.

Roman A. Sharapov
Moscow Institute of Physics and Technology,
Insitutskyy per., 9, Dolgoprudnyy, 141701, Russia
e-mail: sharapov.roman@gmail.com, https://orcid.org/0000-0002-2378-0260

Cold multi-currency wallet on the platform MKT

DOI: http://dx.doi.org/10.26583/bit.2019.2.03

Abstract. The paper is exploring the concept of implementing multi-currency secure cold wallet on the basis of the microcomputer with dynamically variable architecture MKT. This device is based on the new Harvard architecture, a modification of the classic Harvard architecture, using immutable memory (to ensure the integrity and immutability of the operating system) and session memory blocks (to ensure the correct operation of the software). The paper describes the work mechanism of blockchain and blockchain transactions, the basic principles of cryptocurrency storage. The implementation of the system algorithm for offline signing of transactions and protocols that provide a hierarchical key generation based on HD Wallet technology are reviewed. It is proposed to divide the functionality of the multi-currency wallet into two groups intended for execution either in the operating system that is read-only or available both for reading and writing. The scenario of execution of the working cycle of the designed system is described.
Keywords: New Harward architecture, dynamically variable architecture, blockchain, offline transaction signature, HD wallet.
For citation: SHARAPOV, Roman A. Cold multi-currency wallet on the platform MKT. IT Security (Russia), [S.l.], v. 26, n. 3, p. 32-44, 2019. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1215>. Дата доступа: 11 sep. 2019. doi:http://dx.doi.org/10.26583/bit.2019.3.03.

Введение

При использовании горячих криптовалютных кошельков из-за постоянного контакта с Сетью, повышается риск компрометации ключей, поэтому для более безопасного хранения криптовалюты и совершения транзакций используют холодные криптовалютные кошельки [1].

Холодный кошелек не имеет постоянного соединения с интернетом и блокчейном и как следствие менее подвержен атакам злоумышленников. Однако рано или поздно появляется необходимость произвести транзакцию с применением закрытых ключей, хранящихся в холодном кошельке, при этом возникают следующие проблемы:

- во-первых, нет гарантии, что операция проводится на доверенном компьютере;
- во-вторых, для выполнения транзакции кошельку необходимо обновить блокчейн, то есть за продолжительное время ему необходимо скачать большой объем информации, и в данных условиях снова повышается риск компрометации ключей, так как за время закачки необходимой информации в систему может скачаться и начать исполняться вредоносный код.

Таким образом, возникает противоречие между потенциальными уязвимостями при хранении ключей и проведении транзакций из горячих и холодных кошельков и потребностями общества в безопасном хранении ключей и безопасном проведении транзакций.

Данное противоречие можно разрешить, если в качестве доверенной исполняемой среды использовать облачный микрокомпьютер с динамически изменяемой архитектурой МКТ, который позволяет размещать ПО в памяти с физически устанавливаемым доступом read-only [2], что исключает ее искажение и обеспечивает неизменность среды функционирования.

В настоящей работе предложен концепт решения этой задачи и рассмотрена реализация локального холодного кошелька на платформе МКТ.

1. Хранение криптовалюты и совершение транзакций

Рассмотрим механизм работы блокчейна и блокчейн-транзакций.

В блокчейн имплементирована программа с внутренним хранилищем в которой хранится реестр монет, который представляет собой массив ассоциаций аккаунтов с суммами. Полный доступ к данному реестру имеет только эта программа, она же предоставляет всем желающим возможности просматривать суммы на аккаунтах и переводить сумм между аккаунтами, предварительно выполнив проверку на наличие у обратившегося доступа к тому аккаунту, с которого он пытается передать деньги, а также на не превышение суммы на балансе этого аккаунта. Проверка доступа основывается на проверке закрытого ключа из классической пары шифрования с открытым ключом. Если проверка прошла, то запрос на транзакцию будет отослан в Сеть, после чего запрос будет принят или отвергнут на основании алгоритма консенсуса, принятого в данном блокчейне [6].

Перечислим ключевые тезисы, на которые будем опираться при построении защищенной системы:

- Криптовалюта как деньги — это не более чем счетчик, записанный в реестре валют блокчейн-программы [3].
- Криптовалюта хранится не в кошельке, а в реестре валюты в блокчейне [3].
- У кого есть закрытый ключ от аккаунта, на котором лежат деньги, тот может переводить деньги в независимости от мнения оригинального владельца пары ключей.

- В случае кражи ключей, и вывода денег, злоумышленника вычислить крайне трудно, практически невозможно.
- Хранить криптовалюту в безопасности означает хранить в безопасности закрытые ключи.
- Неважно, откуда или с какого типа кошелька идет перевод денег, важно, чтобы программа получила из блокчейна корректную идентификационную информацию [4].
- «Холодный кошелек» означает, что кошелек не требует постоянного подключения к интернету. А горячий, соответственно, требует подключения к сети, что делает его более уязвимым для атак, так как каналы связи или сервера могут быть скомпрометированы [1].
- Холодные кошельки в пассивном режиме (без соединения с интернетом и без совершения транзакций) являются безопасными, при условии, что компьютер, на котором хранятся закрытые ключи, является доверенным, и мы можем гарантировать, что у злоумышленников нет к нему доступа.
- Пока не проведен ни один платеж с холодного кошелька (ни разу не использован приватный ключ) — холодный криптокошелек будет столь же надежен, как и аппаратный или бумажный.
- Проверять получение платежей на свой адрес можно на сторонних сервисах контроля блокчейна по запросу на открытый ключ пользователя.
- После проведения первой транзакции с холодного кошелька, кошелек перестает быть холодным в чистом виде (из пассивного режима на короткий период переходит в активный), и возрастает риск компрометации ключа. Чем больше хранимая сумма — тем больше риск.
- После совершения транзакции и отключения интернета риск компрометации ключа снижается, но остается, так как злоумышленники, применяя аналитические методы и вредоносное программное обеспечение, все равно могут получить доступ к закрытым ключам.
- В большинстве криптовалют действует система, аналогичная биткойну: каждая транзакция должна быть потрачена полностью. То есть если у Алисы на аккаунте есть 5 BTC и она хочет перевести Бобу 1 BTC, то она должна сделать транзакцию с двумя выходами: один для Боба (1 BTC) и второй для «сдачи» для Алисы на ее собственный адрес (4 BTC) [5].
- Послетранзакционный риск можно нивелировать, если использовать одноразовые ключевые пары и после каждой транзакции переводить «сдачу» на свой другой, только что созданный аккаунт. В такой схеме аккаунты (открытые ключи и ключевые пары) являются одноразовыми, и после первой же произведенной транзакции больше не используются.

Снижение и ликвидация риска компрометации ключа во время соединения с интернетом являются основными задачами этой работы.

2. Офлайн-подпись транзакций

Офлайн-подпись транзакций — это модификация метода холодного хранения, которая применяется для уменьшения риска компрометации в момент совершения транзакции:

- Система состоит из двух компьютеров, один из них всегда отключен от интернета, второй — подключен к сети

- На первом компьютере с помощью кошелька создаем пару ключей, импортируем открытый ключ на второй компьютер. После чего на этот адрес можно перевести необходимую сумму для хранения.
- Когда возникает необходимость в переводе, то на втором компьютере создается неподписанная транзакция (указываются адреса отправителя и получателя, а также объём средств), которая затем копируется на съемный носитель и импортируется на первый компьютер, неподключенный к интернету.
- На первом компьютере транзакция подписывается после чего точно также на съемном носителе импортируется на первый компьютер и отсылается в Сеть.

Ошибочно считается, что этот метод является абсолютно безопасным, потому что Интернет и закрытые ключи никогда не контактируют. Однако при импорте транзакции на носитель, даже при соблюдении всех мерах безопасности, все равно остается вероятность загрузки исполняемого вредоносного кода и на съемный носитель, и на сам компьютер, что может привести к подмене данных в транзакции, краже или повреждению закрытого ключа.

3. Иерархическая генерация ключей

Рассмотрим технологию, позволяющую удобным образом реализовать одноразовые ключевые пары.

Deterministic wallet — это кошелек, особенность которого состоит в том, что есть возможность из одного секрета (master seed) породить сколько угодно пар ключей для электронной подписи [6]. Это позволяет использовать новые адреса для каждого входящего платежа и сдачи, при этом все порожденные из основного секрета личные ключи друг с другом никак не связаны, то есть нельзя проследить связь между порожденными адресами (определить, что все они принадлежат одному пользователю), а имея порожденный личный ключ, нельзя восстановить изначальный общий секрет. Стандартизированный подход к кодированию основного секрета расписан в протоколах семейства BIP (Bitcoin Improvement Protocol) [7].

Детерминистические кошельки бывают двух типов:

- Простой детерминистический кошелек: Основной секрет здесь конкатенируется с индексом дочернего ключа, который мы хотим получить, после чего конкатенированные данные хешируются, с помощью хеш-функции SHA-256. Кошелек имеет некоторый seed, из которого напрямую генерируется множество личных ключей. Их количество может быть ограничено только размерностью индекса, который конкатенируется к секрету перед хешированием. Обычно это 4 байта, а это около 4 миллиардов уникальных ключей.
- Иерархически детерминистический кошелек (hierarchical deterministic wallets, HD wallets): На каждом уровне иерархии узел порождения имеет три объекта: личный ключ (private key), открытый ключ (public key) и код цепочки (chain code), который используется для порождения следующего уровня иерархии. Ключ, из которого порождают, называется родительским (parent), а порожденный ключ называется дочерним (child). Генерации ключей происходят по стандарту BIP32, в котором определены принципы работы этих кошельков [3].

Стоит отметить, что получение из родительского открытого ключа дочернего личного невозможно. То есть в данных системах можно выполнить следующие операции по развертыванию ключа:

- Master seed --> master key.
- Private parent key --> private child key.
- Private parent key --> public child key.

- public parent key --> public child key.

Рассмотрим следующий подход — hardened derivation. Это метод, не позволяющий рассчитывать дочерние открытые ключи из соответствующего родительского открытого ключа [7]. От обычного порождения отличается тем, что в обычном в качестве сообщения функции HMAC используется конкатенация сериализованной точки на эллиптической кривой, в качестве родительского открытого ключа, а в hardened derivation используют сериализацию родительского личного ключа. Если использовать этот подход, то злоумышленник, имея на руках родительский открытый ключ, не сможет рассчитать дочерние открытые ключи, и, следовательно, он не сможет вычислить адреса и их связь с полученным родительским ключом, что гарантирует дополнительный уровень анонимности.

Перечислим основные протоколы из семейства BIP, активно применяемые при проектировании криптовалютных кошельков с применением технологии HD Wallet:

- **BIP32** (hierarchical deterministic wallets) [8].

BIP32 — это протокол, определяющий иерархические ключи и способы их внутренней организации, в виде древовидной структуры. Также в этом протоколе описан способ разделения ключей на те, что будут использоваться для основных транзакций, и на те, что нужны для возвращения сдачи.

- **BIP39** (mnemonic code for generating deterministic keys) [9].

Данный протокол применяется при развертывании ключей. Он представляет собой кодирование основного секрета в мнемоническую фразу — набор обычных слов, который легко запомнить. При этом при вводе и выводе фразы есть возможность проверить контрольную сумму, то есть с большой вероятностью выявить ошибку, если такая имеется.

- **BIP43** (Purpose Field for Deterministic Wallets) [6] и **BIP44** (Multi-Account Hierarchy for Deterministic Wallets) [10].

BIP43 предполагает запись в первый уровень иерархии ключа номера улучшения, которое предлагает новый путь порождения или улучшения (m/bip_number'/*). BIP44, использует особенность предыдущего предложения, то есть для первого уровня иерархии записывается индекс 44, а в индексе второго уровня записывается определенное значение, которое будет соответствовать типу монеты, которую мы используем для данного кошелька. Теперь в одном кошельке могут разворачиваться и использоваться ключи для разных валют, то есть кошелек становится мультивалютным.

Таким образом, технология HD Wallet с применением метода hardened derivation позволяет оперировать очень большим (до 4 млрд) количеством аккаунтов в разных блокчейн-системах, сохраняя полную анонимность платежей, при этом сохраняя удобство и возможность автоматически посылать сдачу от транзакций на новые адреса. При этом от пользователя требуется всего лишь знать один мастер-ключ (или мнемоническая фраза, при использовании стандарта BIP39). Решение является очень удобным для пользователя и эффективным с точки зрения эффективности и безопасности, кроме того, оно обеспечивает мультивалютность кошелька.

4. МКТ и новая гарвардская архитектура

Вспомним классические подходы при построении архитектуры компьютерных систем:

- Архитектура фон Неймана – реализована практически во всех настольных компьютерах: Команды и данные не разделяются, а передаются по единому общему каналу.

- Гарвардская архитектура – реализована практически во всех планшетных компьютерах и телефонах: потоки команд и данных параллельны и независимы, что требует более сложной организации процессора, но обеспечивает более высокое быстродействие.

Данные классические архитектуры уязвимы, так как в них есть возможность изменения последовательности команд и данных независимо от того, размещены они в одной памяти или разделены [12]. Как следствие – возможность для несанкционированного вмешательства в логику работы с помощью вредоносного программного обеспечения.

Запись в долговременную память можно заблокировать, если использовать механизмы контроля целостности данных, причем проверка целостности должна идти до загрузки операционной системы. То есть атаки можно заблокировать с помощью механизмов контроля запуска задач (процессов, потоков) [13].

Чтобы предотвратить эти атаки, необходимо добавить неизменяемую память, разделить потоки команд и данных, исполнить контрольные процедуры в доверенной среде до запуска ОС. В гарвардской архитектуре уже есть разделение потоков команд и данных, поэтому если сделать память неизменяемой и разрешить движение данных и команд только по направлению к процессору, то ВРПО не будет фиксироваться и исполняться на компьютере, однако в этом случае не сможет исполняться и часть обычных программ, которым необходима запись данных в память для нормальной работы [13-14].

Эти противоречия разрешает новая гарвардская архитектура – модификация гарвардской архитектуры с использованием неизменяемой памяти (что избавляет от необходимости использовать сложные механизмы контроля целостности программ и данных до запуска ОС) и блоков сеансовой памяти (для возможности исполнения программ, для большей части которых необходима возможность записи) [13-14].

Есть целая ветка семейства компьютеров с вирусным иммунитетом на базе новой гарвардской архитектуры [12] — компьютеры МКTrusT, которые позволяют работать в одном из двух режимов — защищенном (read only) или незащищенном (read and write) [15]. Работа в этих режимах производится в разных ОС, загружающихся из разных, физически разделенных разделов памяти (то есть взаимовлияние ОС исключено технологически, как исключен и их одновременный запуск). Переключение режимов работы производится с помощью физического переключателя, расположенного на корпусе устройства, то есть необходимый режим выбирает пользователь, и не может выбрать хакер, так как невозможно программное воздействие на выбор режима. Для совершения транзакции пользователю понадобится выйти в интернет, и в таком случае ОС может перестать быть доверенной. Поэтому выход в интернет осуществляется только со второй, незащищенной, ОС.

5. Использование особенностей МКТ для устранения известных уязвимостей

Перечисленные выше ключевые особенности МКТ позволяют решить ряд проблем при работе с криптовалютными кошельками:

1. Офлайн-подпись транзакций: если в качестве офлайн-компьютера использовать МКТ в защищенном read only режиме, то можно гарантировать, что даже при попадании вредоносного ПО в систему, оно не сможет исполниться и таким образом транзакция не будет скомпрометирована.

2. Использование U2F для подписи транзакций: используя защищенную ОС МКТ в качестве доверенной среды, можно быть уверенным, что на входе и выходе U2F устройства будет именно та транзакция, которая нужна
3. Хранение ключей: если устанавливать кошелек в защищенную ОС МКТ, то при создании аккаунта в этом кошельке, можно выбрать опцию сохранения закрытых ключей в постоянную память, и не бояться, что эти ключи будут скомпрометированы.

Кроме того, особенности архитектуры МКТ позволяют часть функционала кошелька исполнять в защищенной среде и часть — в не защищенной. Чтобы эффективно использовать эту опцию, необходимо понять, какую часть функционала кошелька необходимо исполнять в каждой из ОС. Для этого введем два класса операций:

1. Операции, которые необходимо проводить в защищенной (RO) ОС.
2. Операции, которые необходимо проводить в незащищенной (RW) ОС.

К первому классу отнесем все функции, где так или иначе задействованы закрытые ключи или персональные данные пользователя, так как эти данные являются наиболее ценными и, как следствие, наиболее вероятными целями для хакеров, и поэтому для этих функций необходимо обеспечить максимально возможную безопасную среду исполнения.

Ко второму классу отнесем те функции, для исполнения которых необходима запись данных в постоянную память (так как в этом классе применяется read and write ОС), а также необходим интернет (соединение с которым повышает вероятность атаки на систему, а поэтому не желательно оперировать ценными данными в этой ОС).

Проведем следующий анализ функционала:

- опишем каждую функцию;
- укажем, нужен ли интернет для ее корректной работы;
- укажем, есть ли потенциальные уязвимости (если применять функцию на обычном компьютере) и какие;
- на основе предыдущих трех пунктов решим — к какому классу причислить эту функцию.

Результаты анализа занесем в табл. 1.

Перечислим операции, необходимые для работы проектируемой системы, и разделим их на два класса:

- **Операции, которые необходимо проводить в защищенной (RO) ОС:**
  - Генерация ключей, мастер-ключа или мнемонической фразы.
  - Развертывание ключей из мастер-ключа или мнемонической фразы.
  - Создание нового аккаунта в кошельке.
  - Подпись транзакции через интерфейс кошелька.
  - Подпись транзакции через U2F.
  - Хранение ключей в постоянной памяти.
  - Аутентификация через U2F.
- **Операции, которые необходимо проводить в незащищенной (RW) ОС:**
  - Чтение баланса из скачанного блокчейна.
  - Скачивание и обновление блокчейна.
  - Чтение баланса из блокчейна в интернете.
  - Создание транзакции.
  - Трансляция транзакции в блокчейн-сеть.

Выполнение этих операций в соответствующих ОС поможет минимизировать риски компрометации.

Таблица. 1 Сравнительный анализ функционала кошелька

Название функции	**Описание**	**Есть ли потенциальные уязвимости?**	**Класс функции**
Генерация ключей, мастер-ключа или мнемоническ-кой фразы	Генератор псевдослучайных чисел выдает на выходе, в зависимости от выбранного режима, либо ключевую пару, либо seed, из которого дальше по протоколу BIP32 будут развернуты остальные ключевые пары и по протоколу BIP39 этот seed будет представлен в виде мнемонической фразы	Если среда и приложение доверенные — то нет	Класс 1
Развертыва-ние ключей из мастер-ключа или мнемоническ-кой фразы	Пользователь вводит ключ или мнемоническую фразу, а кошелек на основе введенных данных выдает список всех ключевых пар пользователя, после чего ими можно пользоваться	Если среда и приложение доверенные — то нет	Класс 1
Создание нового аккаунта в кошельке	После генерации и развертывания ключей, на выбор пользователя будет создан файл аккаунта в кошельке. В этом файле будет записано, что за данным пользователем закреплены соответствующие открытые ключи, и этот же файл будет хранить историю операций.	Хранить историю операций и открытые ключи в постоянной памяти компьютера — это потенциально не безопасное решение, так как если среда недоверенная, то при компрометации такого файла злоумышленник может узнать, например, что определенный набор открытых ключей принадлежит одному человеку	Класс 1
Чтение баланса из скачанного блокчейна	Идет чтение из блокчейна, расположенного в постоянной памяти. По завершению операции на выходе получаем баланс адреса на момент последнего обновления блокчейна	Если исключить вероятность наличия вредоносного кода в скачанном блокчейне, то сама операция чтения не является уязвимой	Класс 2
Скачивание обновление блокчейна	Обновление блокчейна, для последующего совершения транзакции или чтения баланса аккаунта	Из интернета в течение продолжительного времени качается большой объем данных (в случае если блокчейн не обновлялся достаточно давно), поэтому есть вероятность закачки вредоносного кода	Класс 2
Чтение баланса из блокчейна в интернете	Чтобы не скачивать гигабайты данных, идет чтение из блокчейна, расположенного в интернете, то есть идет обращение к серверам или соседним узлам. По завершению операции на выходе получаем баланс аккаунта на момент последнего обновления блокчейна	За время соединения с интернетом в систему может попасть вредоносный код. Также есть вероятность чтения данных из ложных источников в случае компрометации каналов связи	Класс 2

Название функции	**Описание**	**Есть ли потенциальные уязвимости?**	**Класс функции**
Создание транзакции	Пользователь вводит адрес отправителя, адрес получателя и количество валюты, которую нужно перевести. В случае если сумма переводится не полностью, то необходимо добавить адрес для возвращения сдачи. Также в транзакцию необходимо добавить ряд служебной информации, включая адреса и хэши транзакций, с которых были получены средства. Для корректной работы этого этапа необходимо обновить блокчейн до актуального состояния. На выходе получаем сформированную, но не подписанную транзакцию	Есть возможность того, что злоумышленник может подменить данные в транзакции на ложные. С учетом того, что этому этапу предшествовал этап обновления блокчейна, в системе мог оказаться вредоносный код	Класс 2
Подпись транзакции через интерфейс кошелька	Чтобы подписать транзакцию, пользователю необходимо ввести закрытый ключ в соответствующее поле (или идет чтение из файла, в котором в зашифрованном виде хранятся закрытые ключи), после чего система произведет все необходимые операции и вычисления	Подмена транзакции на ложные данные. Компрометация закрытого ключа	Класс 1
Трансляция транзакции в блокчейн-сеть	Подписанная транзакция посылается в Сеть, на ближайший доступный узел	Злоумышленник может послать в Сеть, подписанную в предыдущем пункте транзакцию с ложными данными. Остальные атаки зависят от надежности канала связи и узлов, на которые идет распространение информации	Класс 2
Аутентификация через U2F	В случае если был создан кошелек с двухфакторной аутентификацией, для входа в свой аккаунт пользователь должен вставить в компьютер U2F устройство	Если среда и приложение доверенные, то нет	Класс 1
Подпись транзакции через U2F	На вход в U2F устройство посылается сформированная и неподписанная транзакция. U2F устройство посылает в компьютер подписанную транзакцию	На вход устройству может послаться транзакция с фальшивыми данными или с вредоносным кодом	Класс 1
Хранение ключей	Закрытые ключи хранятся на жестком диске в зашифрованном виде	Ключи лучше не хранить в постоянной памяти, но если возникает такая необходимость, то делать это можно только в защищенной ОС	Класс 1

6. Сценарий работы системы

Опишем работу системы на верхнем логическом уровне. Начало работы с мультивалютным кошельком выглядит следующим образом:

- Заходим в защищенную ОС и создаем кошелек.
- Импортируем необходимые открытые ключи для последующего применения в не защищенной ОС.
- Заходим в незащищенную ОС и импортируем созданные в предыдущем пункте открытые ключи в кошелек.
- Когда возникает необходимость совершить транзакцию с одного из адресов, то система в незащищенной ОС должна:
  - Обновить блокчейн.
  - Сформировать транзакцию.
  - Экспортировать сформированную транзакцию на отчуждаемый носитель.
- Сформированная в предыдущем пункте транзакция на отчуждаемом носителе импортируется в защищенную ОС.
- В защищенной ОС транзакция подписывается одним из вариантов:
  - С помощью U2F устройства.
  - С помощью закрытого ключа, хранимого в зашифрованном виде в файле в постоянной памяти.
  - С помощью закрытого ключа, введенного пользователем вручную в соответствующее поле интерфейса.
- После этого подписанная транзакция экспортируется через отчуждаемый носитель обратно в незащищенную ОС.
- В незащищенной ОС подписанная транзакция транслируется в Сеть

Схема рабочего цикла кошелька приведена на рис. 1

Таким образом, мы получаем систему, в которой все операции, производимые с закрытыми ключами, проводятся в отдельной защищенной операционной системе с постоянной памятью, доступной только чтения. Благодаря технологии иерархически детерминированного кошелька все ключевые пары являются одноразовыми, то есть после каждой транзакции закрытый ключ больше не используется, причем, даже если злоумышленник перехватит все использованные ключи, он не сможет вычислить остальные.

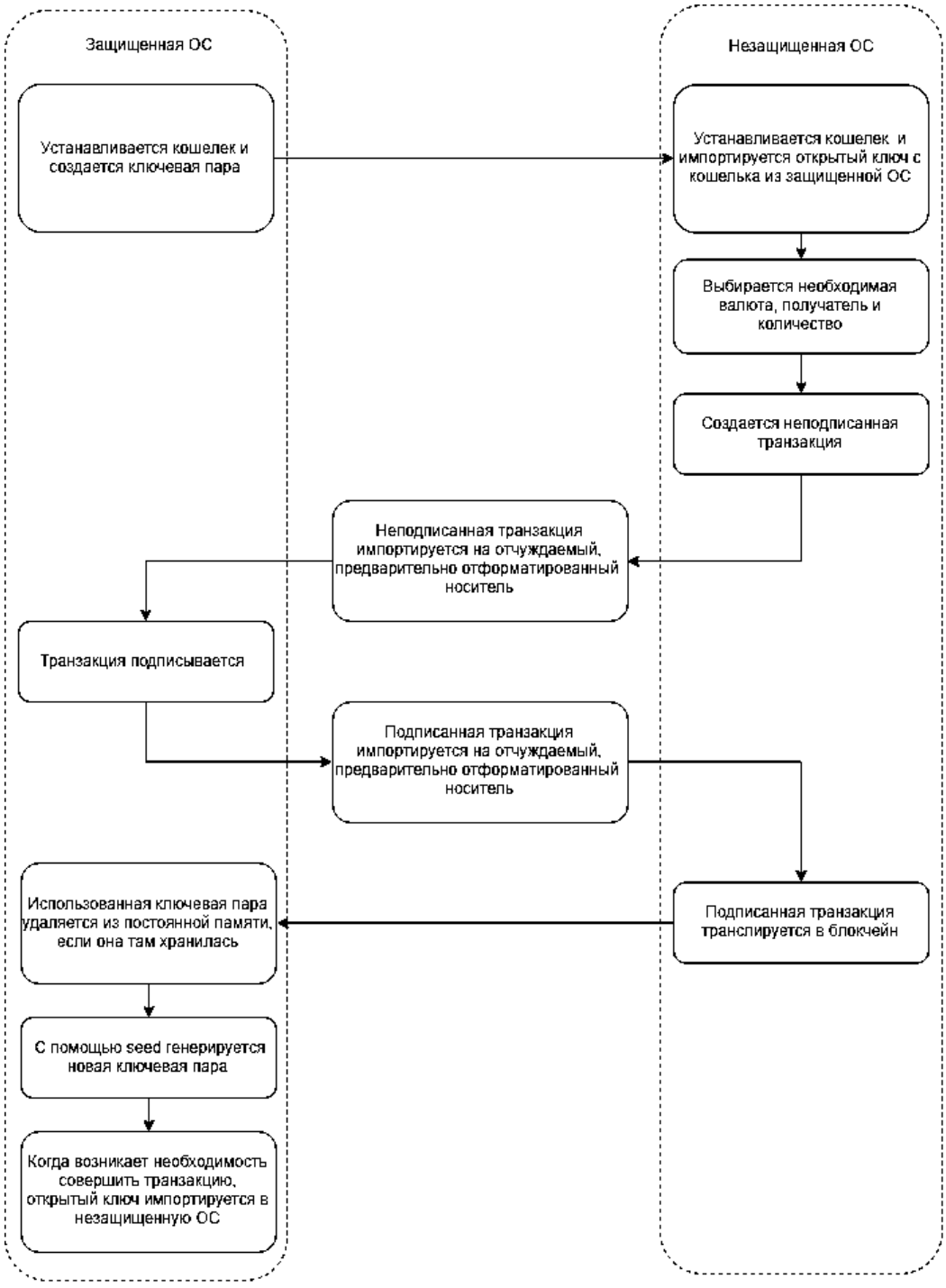

Рис. 1. Схема работы проектируемой системы
(Fig. 1 Working scheme of the designed system)

Заключение

В данной статье рассмотрена одна из возможных реализаций холодного мультивалютного кошелька на базе компьютера с динамически изменяемой архитектурой МКТ: были рассмотрены ключевые аспекты системы, выбран алгоритм развертки ключей, предложен вариант разделения функционала мультивалютного холодного кошелька на

две группы, каждая из которых предназначена для исполнения в своей среде, и описан цикл работы системы на логическом уровне.

СПИСОК ЛИТЕРАТУРЫ:

1. Cold storage [Электронный ресурс]. — Электрон. текстовые дан. — URL: https://en.bitcoinwiki.org/wiki/Cold_storage, свободный (дата обращения: 20.06.19).
2. Конявский, В.А. Компьютер с «вирусным иммунитетом» [Электронный ресурс]. В.А. Конявский. — Электрон. текстовые дан. — URL: http://www.okbsapr.ru/konyavskiy_2015_2.html, свободный (дата обращения: 20.06.19).
3. Bitcoin in a nutshell — Blockchain [Электронный ресурс]. — Электрон. журн. — URL: https://habr.com/ru/post/320176/ (дата обращения: 20.06.19).
4. Как на самом деле работает протокол Биткоин [Электронный ресурс]. — Электрон. журн. — URL: https://habr.com/ru/post/222493/ (дата обращения: 20.06.19).
5. Bitcoin: A Peer-to-Peer Electronic Cash System [Электронный ресурс]. Satoshi Nakamoto. — Электрон. журн. — URL: https://bitcoin.org/bitcoin.pdf (дата обращения: 20.06.19).
6. Hierarchical deterministic Bitcoin wallets that tolerate key leakage [Электронный ресурс]. Gus Gutoski. — Электрон. журн. — URL: https://eprint.iacr.org/2014/998.pdf (дата обращения: 20.06.19).
7. Иерархическая генерация ключей [Электронный ресурс]. — Электрон. текстовые дан. — URL: https://habr.com/company/distributedlab/blog/413627/, свободный (дата обращения: 20.06.19).
8. bip-0032 [Электронный ресурс]. — Электрон. текстовые дан. — URL: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki, свободный (дата обращения: 20.06.19).
9. bip-0039 [Электронный ресурс]. — Электрон. текстовые дан. — URL: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, свободный (дата обращения: 20.06.19).
10. bip-0043 [Электронный ресурс]. — Электрон. текстовые дан. — URL: https://github.com/bitcoin/bips/blob/master/bip-0043.mediawiki, свободный (дата обращения: 20.06.19).
11. bip-0044 [Электронный ресурс]. — Электрон. Текстовые дан. — URL: https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki, свободный (дата обращения: 20.06.19).
12. Конявский В.А. Компьютер с вирусным иммунитетом. Информационные ресурсы России. 2015. № 6. С. 31–34.
13. Конявский В.А. Мобильный компьютер с аппаратной защитой доверенной операционной системы: Патент на полезную модель № 147527. 10.11.2014. Бюл. № 31.
14. Конявский В.А. Защищенный микрокомпьютер МК-TRUST — новое решение для ДБО. Национальный банковский журнал. — 2014. — № 3. — С. 105.
15. Работа с одного рабочего места в двух разных контурах защищенности [Электронный ресурс]. — Электрон. текстовые дан. — URL: http://www.okbsapr.ru/sol17.html, свободный (дата обращения: 20.06.19).

REFERENCES:

[1] Cold storage [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://en.bitcoinwiki.org/wiki/Cold_storage, svobodnyj (accessed: 20.06.2019) (in Russian).
[2] Konjavskij, V.A. Komp'juter s «virusnym immunitetom» [Jelektronnyj resurs] V.A. Konjavskij. — Jelektron. tekstovye dan. — URL: http://www.okbsapr.ru/konyavskiy_2015_2.html, svobodnyj (accessed: 20.06.2019) (in Russian).
[3] Bitcoin in a nutshell — Blockchain [Jelektronnyj resurs]. — Jelektron. zhurn. — URL: https://habr.com/ru/post/320176/ (accessed: 20.06.2019) (in Russian).
[4] Kak na samom dele rabotaet protokol Bitkoin [Jelektronnyj resurs]. — Jelektron. zhurn. — URL: https://habr.com/ru/post/222493/ (accessed: 20.06.2019) (in Russian).
[5] Bitcoin: A Peer-to-Peer Electronic Cash System [Jelektronnyj resurs]. Satoshi Nakamoto. — Jelektron. zhurn. — URL: https://bitcoin.org/bitcoin.pdf (accessed: 20.06.2019) (in Russian).
[6] Hierarchical deterministic Bitcoin wallets that tolerate key leakage [Jelektronnyj resurs]. Gus Gutoski. — Jelektron. zhurn. — URL: https://eprint.iacr.org/2014/998.pdf (accessed: 20.06.2019) (in Russian).
[7] Ierarhicheskaja generacija kljuchej [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://habr.com/company/distributedlab/blog/413627/, svobodnyj (accessed: 20.06.2019) (in Russian).
[8] bip-0032 [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki, svobodnyj (accessed: 20.06.2019) (in Russian).

[9] bip-0039 [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, svobodnyj (accessed: 20.06.2019) (in Russian).
[10] bip-0043 [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://github.com/bitcoin/bips/blob/master/bip-0043.mediawiki, svobodnyj (accessed: 20.06.2019) (in Russian).
[11] bip-0044 [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki, svobodnyj (accessed: 20.06.19).
[12] Konjavskij V.A. Komp'juter s virusnym immunitetom. Informacionnye resursy Rossii. 2015. № 6. S. 31–34 (in Russian).
[13] Konjavskij V.A. Mobil'nyj komp'juter s apparatnoj zashhitoj doverennoj operacionnoj sistemy: Patent na poleznuju model' № 147527. 10.11.2014. Bjul. № 31 (in Russian).
[14] Konjavskij V.A. Zashhishhennyj mikrokomp'juter MK-TRUST — novoe reshenie dlja DBO. Nacional'nyj bankovskij zhurnal. — 2014. — № 3. — S. 105 (in Russian).
[15] Rabota s odnogo rabochego mesta v dvuh raznyh konturah zashhishhennosti [Jelektronnyj resurs]. — Jelektron. tekstovye dan. — URL: http://www.okbsapr.ru/sol17.html, svobodnyj (in Russian).

Поступила в редакцию – 25 июня 2019 г. Окончательный вариант – 17 августа 2019 г.
Received – Juny 25, 2019. The final version – August 17, 2019.