УДК 004.056

Е.В. ИСИРОВА, А.В. ПОТИЙ, д-р техн. наук, В.В. СЕМЕНЕЦ, д-р техн. наук

ПРИНЦИПЫ ПОСТРОЕНИЯ ДЕЦЕНТРАЛИЗОВАННОЙ ИНФРАСТРУКТУРЫ ОТКРЫТЫХ КЛЮЧЕЙ

Введение

Успешное внедрение современных технологий электронного управления, электронных доверительных услуг невозможно без создания соответствующей инфраструктуры. Технологичной инфраструктурой реализации упомянутых технологий выступает инфраструктура открытых ключей (ИОК). Использование электронных доверительных услуг с применением цифровой подписи опирается на доверие между субъектами взаимодействия, инфраструктуру открытых ключей и направлено на реализацию модели доверия.

В 2017 году в Украине принят Закон Украины «Об электронных доверительных услугах», который определяет правовые и организационные основы предоставления электронных доверительных услуг, в том числе трансграничных, права и обязанности субъектов правовых отношений в сфере электронных доверительных услуг, порядок осуществления государственного надзора (контроля) за соблюдением требований законодательства в сфере электронных доверительных услуг, а также правовые и организационные основы осуществления электронной идентификации. Для эффективного использования и качественного предоставления таких услуг необходимо решать много технологически сложных задач и технических проблем. В 2004 году в Украине была реализована архитектура ИОК, которая стала базой для использования технологии с открытыми ключами и предоставления услуг по управлению криптографическими ключами. Данная архитектура представляет собой иерархическую систему. Кроме иерархической архитектуры существует еще ряд возможных для использования, которые не были использованы из-за невозможности надежной реализации модели доверия. Цель статьи – предложение новой концепции построения инфраструктуры открытых ключей с использованием технологии blockchain.

Описание существующей инфраструктуры открытых ключей, проблемные вопросы при построении

Инфраструктура открытых ключей (ИОК, англ. PKI – Public Key Infrastructure) – набор средств (технических, материальных, людских и т. д.), распределенных служб и компонентов, в совокупности используемых для поддержки криптозадач на основе закрытого и открытого ключей [1].

В основе PKI лежат использование криптографической системы с открытым ключом и несколько основных принципов [2]:

- закрытый ключ (private key) известен только его владельцу;
- удостоверяющий центр создает электронный документ – сертификат открытого ключа, тем самым удостоверяя факт того, что закрытый (секретный) ключ известен эксклюзивно владельцу этого сертификата, открытый ключ (public key) свободно передается в сертификате;
- никто не доверяет друг другу, но все доверяют удостоверяющему центру;
- удостоверяющий центр подтверждает или опровергает принадлежность открытого ключа заданному лицу, которое владеет соответствующим закрытым ключом.

Основным нормативным документов является стандарт ITU-T X.509 для инфраструктуры открытого ключа и инфраструктуры управления привилегиями (англ. Privilege Management Infrastructure). Он определяет стандартные форматы данных и процедуры распределения открытых ключей с помощью соответствующих сертификатов с цифровыми подписями. Эти сертификаты предоставляются центрами сертификации (англ. Certificate Authority). Кроме того, X.509 определяет формат списка отозванных сертификатов (англ. Certificate revocation lists, CRL), формат сертификатов атрибутов (англ. Attribute certificates) и алгоритм проверки подписи построением пути сертификации (англ. Certification path validation algorithm).

ИОК состоит из ряда подсистем:

- организационно-техническая (включает в себя политику сертификации, регламент);
- подсистема управления списками отозванных сертификатов (уполномоченный на сертификацию, центр регистрации, репозиторий, конечные пользователи);
- подсистема применений ИОК (web-защита, защищенный email, защищенный документооборот, VPN)

Для технологии открытых ключей необходимо, чтобы пользователь открытого ключа был уверен, что этот ключ принадлежит именно тому удаленному субъекту (пользователю или системе), который будет использовать средства шифрования или цифровой подписи. Такую уверенность дают сертификаты открытых ключей. Сертификат имеет ограниченный срок действия. Поскольку пользователь сертификата может самостоятельно проверить его подпись и срок действия, сертификаты могут распространяться через незащищенные каналы связи и серверные системы, а также храниться в кеш-памяти незащищенных пользовательских систем.

При построении ИОК необходимо решать проблемные вопросы на нескольких уровнях:

– правовом (регулирование взаимоотношений между участниками процессов сертификации);
– системном (обоснование выбора архитектуры с учетом решаемых задач);
– процедурно-функциональном (определение основных функциональных требований к системе сертификации, установление перечня услуг центров сертификации);
– функционально-техническом (определение функциональной структуры, физической топологии, обоснование требований безопасности);
– техническом (обоснование выбора аппаратных средств для центров сертификации, в том числе средств криптографической защиты).

Основные угрозы для систем такого типа:

– отказ от выполнения действий;
– подделка сертификата.

Для обеспечения доверия необходимо обеспечить функционирование системы в рамках актуальной модели доверия. Стандарт X.509 [1] предполагает возможное использование следующих моделей доверия:

– строгая иерархия уполномоченных на сертификацию;
– нестрогая иерархия уполномоченных на сертификацию;
– иерархия на базе политик;
– модель распределенного доверия;
– четырехсторонняя модель доверия;
– модель доверия вокруг пользователя;
– web-модель доверия.

На сегодняшний день подавляющее ИОК построены на основе строгой иерархии уполномоченных на сертификацию (рис. 1)

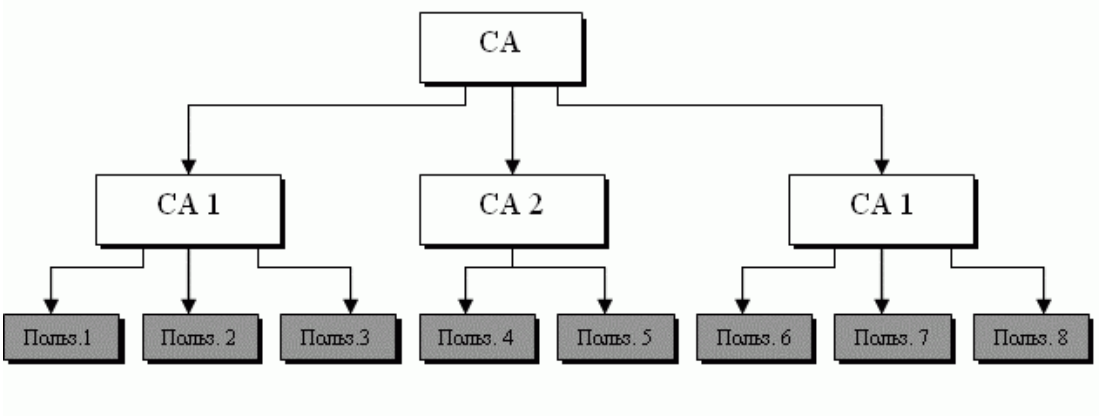

Рис. 1. Иерархическая структура ИОК [2]

Однако такая структура имеет недостатки:

– вся безопасность системы зависит от корневого сертификата центрального уполномоченного на сертификацию. В случае его компрометации все сертификаты в системе являются скомпрометированными;

– пользователи фактически не распоряжаются своими идентификационными данными, при необходимости внести какие-либо корректировки необходимо обращаться к уполномоченному на сертификацию;

– отсутствие интероперабельности системы. Сертификаты, выпущенные разными уполномоченными, не могут быть использованы параллельно;

– отсутствие однозначного соответствия между пользователем и сертификатом, поскольку для одного пользователя может быть выпущено несколько сертификатов.

Другие модели доверия либо слабо распространены, либо не используются вовсе. Однако анализ показал, что с помощью новой технологии blockchain могут быть надежно реализованы и другие модели доверия, в частности модель доверия вокруг пользователя. Рассмотрим ее подробнее.

Модель доверия вокруг пользователя [1, 2]. В такой модели доверия пользователь самостоятельно отвечает за решения каким сертификатам доверять, а какие считать ненадежными. Эти решения зависят от ряда факторов. Первичным источником доверия являются сертификаты родственников, друзей, знакомых, т.е. тех, кого пользователь знает лично (т.е. первичная идентификация проводится самостоятельно пользователем).

Доверие, сконцентрированное вокруг пользователя, иллюстрирует известная система Pretty Good Privacy (PGP) (рис. 2). Пользователь А может решить: доверять сертификату В (на основе доверия к цепочке сертификатов от пользователя D к пользователю С и пользователю В) или отвергнуть сертификат В, аргументируя это тем, что к "неизвестному" пользователю В ведет слишком много связей от "знакомого" пользователя D.

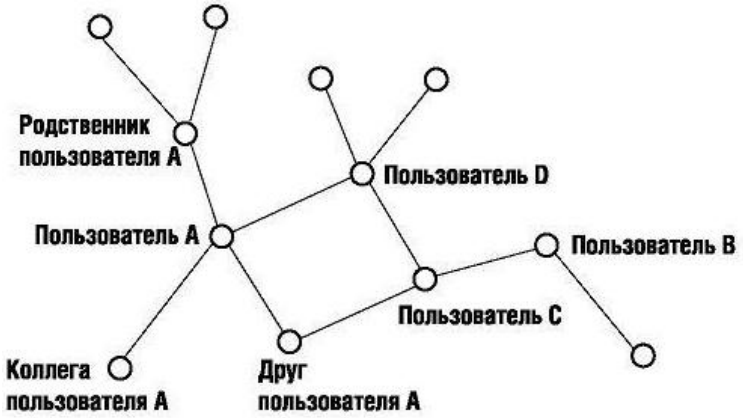

Рис. 2. Модель доверия вокруг пользователя [2]

В силу своей зависимости от действий и решений пользователей модель доверия, сконцентрированного вокруг пользователя, может использоваться только в узком и высокотехнологичном сообществе, но она нежизнеспособна в обычном сообществе, в котором многие пользователи не имеют достаточных знаний о безопасности и технологии PKI. Более того, эта модель не подходит для тех сфер (корпоративной, финансовой, правительственной), где необходим контроль за тем, с кем взаимодействуют и кому доверяют пользователи.

Далее предложим вариант, как можно избежать указанных недостатков с применением технологии blockchain. Кратко опишем саму технологию blockchain.

Технология blockchain

Блокче́йн (англ. blockchain или block chain) – выстроенная по определенным правилам непрерывная последовательная цепочка блоков, содержащих информацию [4].

Блок транзакций – специальная структура для записи группы транзакций в системе Биткойн и аналогичных ей. Транзакция считается завершенной и достоверной («подтвержденной»), когда проверены ее формат и подписи, и когда сама транзакция объединена в группу с несколькими другими и записана в специальную структуру – блок.

Содержимое блоков может быть проверено, так как каждый блок содержит информацию о предыдущем блоке. Все блоки выстроены в одну цепочку, которая содержит информацию обо всех совершенных когда-либо операциях в базе. Самый первый блок в цепочке – первичный блок (англ. genesis block) – рассматривается как отдельный случай, так как у него отсутствует родительский блок.

Блок состоит из заголовка и списка транзакций (рис. 3). Заголовок блока включает в себя свой хеш, хеш предыдущего блока, хеши транзакций и дополнительную служебную информацию. Для транзакций в блоке используется древовидное хеширование, аналогичное формированию хеш-суммы для файла в протоколе BitTorrent. Транзакции, кроме начисления комиссии за создание блока, содержат внутри параметра input ссылку на транзакцию с предыдущим состоянием данных [4].

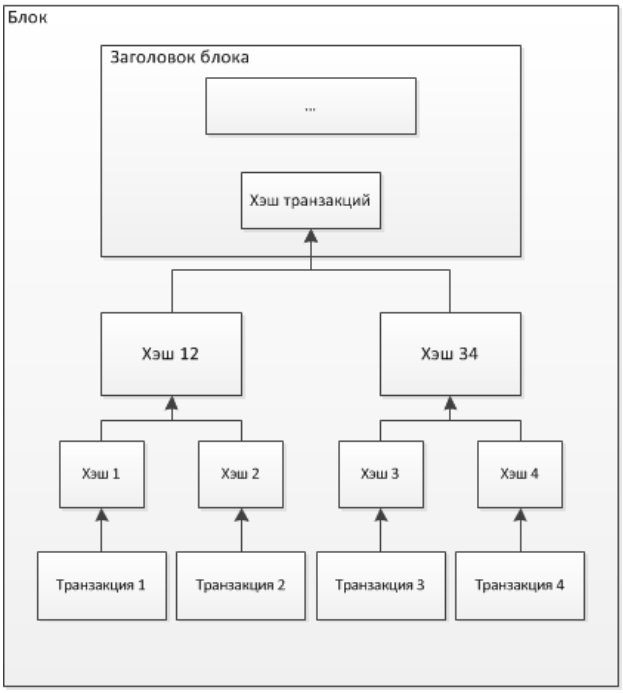

Рис. 3. Структура блока

Созданный блок будет принят остальными пользователями, если числовое значение хеша заголовка равно или ниже определенного числа, величина которого периодически корректируется. Так как результат хеширования функции SHA-256 считается необратимым, на данный момент нет алгоритма получения желаемого результата, кроме случайного перебора. Если хеш не удовлетворяет условию, то в заголовке изменяется параметр *nonce* и хеш пересчитывается. Обычно требуется большое количество пересчетов. Когда вариант найден, узел рассылает полученный блок другим подключенным узлам, которые проверяют блок. Если ошибок нет, то блок считается добавленным в цепочку, и следующий блок должен включить в себя его хеш.

Блоки одновременно формируются множеством «участников». Удовлетворяющие критериям блоки отправляются в сеть, включаясь в распределенную базу блоков. Регулярно возникают ситуации, когда несколько новых блоков в разных частях распределенной сети называют предыдущим один и тот же блок, то есть цепочка блоков может ветвиться. Специально или случайно можно ограничить ретрансляцию информации о новых блоках (например, одна

из цепочек может развиваться в рамках локальной сети). В этом случае возможно параллельное наращивание различных ветвей. В каждом из новых блоков могут встречаться как одинаковые транзакции, так и разные, вошедшие только в один из них. Когда ретрансляция блоков возобновляется, участники начинают считать главной цепочку с учетом уровня сложности хеша и длины цепочки. При равенстве сложности и длины предпочтение отдается той цепочке, конечный блок которой появился раньше (рис. 4). Транзакции, вошедшие только в отвергнутую ветку, теряют статус подтвержденных.

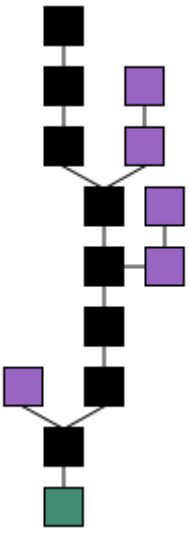

Рис. 4. Цепочка блоков

Таким образом, цепочка блоков содержит историю владения, с которой можно ознакомиться, например, на специализированных сайтах.

Распределенная база данных blockchain формируется как непрерывно растущая цепочка блоков с записями обо всех транзакциях. Копии базы или ее части одновременно хранятся на множестве компьютеров и синхронизируются согласно формальным правилам построения цепочки блоков. Информация в блоках не шифрована и доступна в открытом виде, но отсутствие изменений удостоверяется криптографически через хэш-цепочки (элемент цифровой подписи) [4].

База публично хранит в незашифрованном виде информацию обо всех транзакциях, подписываемых с помощью асимметричного шифрования. Для предотвращения многократной траты используются метки времени, реализованные путем разбиения БД на цепочку специальных блоков, каждый из которых, в числе прочего, содержит в себе хеш предыдущего блока и свой порядковый номер. Каждый новый блок осуществляет подтверждение транзакций, информацию о которых содержит и дополнительное подтверждение транзакций во всех предыдущих блоках цепочки. Изменять информацию в блоке, который уже находится в цепи, непрактично, так как в таком случае пришлось бы редактировать информацию во всех последующих блоках. Таким образом, успешная double-spending атака на практике крайне маловероятна.

Попадание транзакции в блок является подтверждением ее достоверности вне зависимости от наличия других транзакций. Каждый новый блок считается дополнительным «подтверждением» транзакций из предыдущих блоков. Если в цепочке три блока, то транзакции из последнего блока будут подтверждены один раз, а помещенные в первый блок будут иметь три подтверждения. Достаточно дождаться нескольких подтверждений, чтобы свести вероятность отмены транзакции к минимуму.

Если контролировать более 50 % суммарной вычислительной мощности сети, то существует теоретическая возможность при любом пороге подтверждений одни и те же биткойны

передать два раза разным получателям – одна из транзакций будет публичной и подтверждаться в общем порядке, а вторая не будет афишироваться, ее подтверждения будут происходить блоками скрытой параллельной ветви. Лишь через некоторое время сеть получит сведения о второй транзакции, она станет подтвержденной, а первая утратит подтверждения и будет игнорироваться.

Открытость цепочки блоков позволяет внести в произвольный блок изменения. Но тогда потребуется пересчет хеша не только измененного блока, но и всех последующих. Фактически, для такой операции потребуется мощность не меньше той, которая была использована для создания измененного и последующих блоков (то есть всей текущей мощности), что делает такую возможность крайне маловероятной.

Концепция построения PKI на базе технологии blockchain

Частично данная идея нашла воплощение в протоколе безопасности CSMP на основе технологии blockchain, предложенном авторами приложения Crypviser [3].

Модель аутентификации, основанная на технологии blockchain, позволяет пользователям действительно идентифицировать и подтверждать открытые ключи друг друга. Это исключает угрозу «человек посередине» и попытки манипуляции любого рода как на стороне сервера, так и со стороны третьих лиц.

Идея основана на способности технологии blockchain к децентрализованному распространению и управлению открытыми ключами. Поскольку blockchain представляет собой децентрализованную базу данных, она содержит информацию о соответствии уникального идентификатора каждого пользователя и первую половину его открытого ключа *(id: first_half (PK))*. Сервер Crypviser (CV-сервер) содержит информацию о соответствии между уникальным ID пользователя и значением второй половины его открытого ключа *(id: second_half (PK))*.

Первоначальная аутентификация. В ходе регистрации учетной записи приложение Crypviser генерирует уникальное значение ID и первоначальный секретный SK (Shared Key – общий ключ) на локальном устройстве пользователя. Открытый ключ является производным из SK. Эти ключи постоянно используются лишь в целях первоначальной идентификации. CV-сервер, действующий в качестве узла blockchain, имеет собственную пару ключей.

Регистрация учетной записи. По завершении процесса регистрации новая пара ключей генерируется на устройстве пользователя, связанном с учетной записью. В то же время уникальный хеш *CrypID* генерируется на различных источниках энтропии, например частичные хеши симметричного ключа, используемого для защиты локальной базы данных и парольной фразы пользователя. Вторая часть открытого ключа, *CrypID* и идентификатор пользователя *(id: second_half (PK):CrypID)* передаются на CV-сервер с помощью установленного безопасного соединения. Идентификатор пользователя, который был сгенерирован на этапе первоначальной аутентификации, предназначен для обеспечения анонимности на стороне CV-сервера.

Интеграция blockchain. Для записи первой половины первоначального открытого ключа в blockchain пользователь проводит транзакцию (отправляет токены для аутентификации) в пользу CV-сервера. Транзакция содержит метаданные со значением первой половины *first_half* (PK – открытый ключ), которые прописываются в реестре blockchain. После этого записанная часть открытого ключа пользователя может быть проверена на CV-сервере и на сторонах пользователей для исключения попыток атаки «человек посередине» при передаче половины открытого ключа через сеть.

Лишь владелец секретного ключа может «потратить» токен, решив специальную криптографическую «задачу», связанную со сложными расчетами с помощью CrypID. Это означает, что CV-сервер обеспечивает действие первой части открытого ключа пользователя, прописанного в blockchain.

Для подтверждения и проверки первой части открытого ключа, прописанного в blockchain на стороне пользователя, CV-сервер аналогичным образом отправляет пользователю токены для аутентификации. Приложение Crypviser выполняет аналогичные алгоритмы для проверки аутентичности записанной части своего открытого ключа.

Таким образом, CV-сервер и пользователь одновременно проверяют подлинность половины первоначального открытого ключа пользователя. Безопасность частичного значения открытого ключа, записанного в реестре, обеспечивается другими узлами с помощью функции распределения данных.

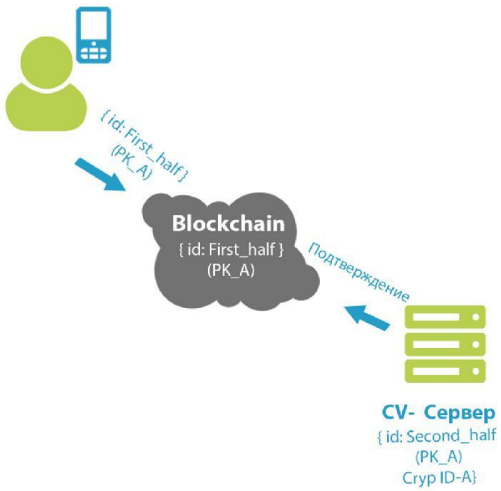

Рис. 5. Распространение открытого ключа [3]

Аутентификация открытого ключа. Алгоритм распространения и проверки достоверности открытого ключа между сторонами описан ниже:

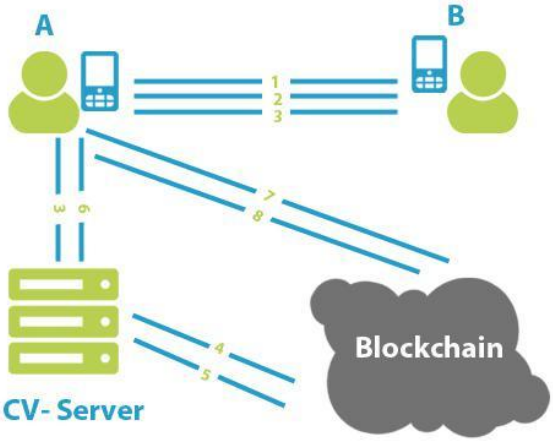

Рис. 6. Аутентификация открытого ключа

1. Сторона А хочет инициировать новый сеанс связи с криптографической защитой и отправляет сообщение следующего содержания:

(Nonce_A, timestamp_A)

2. Сторона В отправляет следующий ответ:

(Nonce_B, timestamp_B,E[(timestamp_A, Nonce_A, id_B, hash(id:PK_B))SK_B]CrypID_B)SK_B,

где *E[(timestamp_A, Nonce_A,id_B, hash(id:PK_B))SK_B]CrypID_B* – вывод зашифрованных данных с CrypID, принадлежащим Стороне В.

Важно отметить, что данные сначала подписываются ключом Стороны В, а затем шифруются.

3. Пользователь А получает сообщение и пересылает его как запрос CV-серверу через защищенный TLS-канал:

$$E[(timestamp\_A, Nonce\_A, id\_B, hash(id: PK\_B))SK\_B]CrypID\_B),$$

4. CV-сервер с CryptID Стороны В расшифровывает зашифрованный текст и проверяет соответствие идентификатора Стороны В, сохраненного локально, его ID, полученному из дешифрованных данных;

5. Затем он получает первую часть открытого ключа из реестра blockchain по идентификатору Стороны В и объединяет ее со второй частью открытого ключа, хранящейся в локальной базе данных, а также проверяет цифровую подпись полученных данных.

6. CV-сервер подписывает сообщение своим секретным ключом, шифрует его с помощью CrypID Стороны А и отправляет следующие данные Стороне А:

$$E[(\text{‘}OK\text{’}, timestamp\_A, Nonce\_A, id\_B, hash(id: PK\_B),$$

$$second\_half(PK\_B))SK\_S]CrypID\_A.$$

7. Получив сообщение, Сторона А выполняет следующие действия:

- дешифрует данные, полученные с CV-сервера, и проверяет цифровую подпись;
- сравнивает значения timestamp_A и Nonce_A с ранее отправленными Стороне В. Значение timestamp_A должно находиться в допустимом временном интервале, а значение Nonce_A должно совпадать.

8. Сторона А запрашивает blockchain и получает первую часть открытого ключа, которая принадлежит Стороне В.

- получает весь открытый ключ Стороны В, объединив полученную часть открытого ключа от blockchain и CV-сервера
- проверяет цифровую подпись всего пакета, полученного от пользователя В ранее;
- рассчитывает хеш *(id:PK_B)*) и сравнивает его со значением хеша данных, полученных от CV-сервера.

9. В случае успешного завершения всех проверок Сторона А считает, что PK_B изначально принадлежит Стороне В. Затем Сторона А отправляет Стороне В следующее сообщение:

$$E[(timestamp\_B, Nonce\_B, id\_A, hash(id:PK\_A))SK\_A]CrypID\_A$$

Сторона В использует тот же алгоритм для получения и аутентификации открытого ключа стороны А.

Таким образом, мы видим, что CV-сервер по сути выступает в роли третьей доверенной стороны, однако его функции резко сокращены по сравнению с функциями третьей доверенной стороны в системе, основанной на модели доверия строгой иерархии.

Кроме того, с использованием такого решения исключается возможность атаки типа «man in the middle».

«Отказ» от третьей доверенной стороны

Технология blockchain, на наш взгляд, способна самостоятельно обеспечивать доверие в системе.

По сути, blockchain – это журнал с фактами (реестр фактов), реплицируемый на несколько компьютеров, объединенных в сеть равноправных узлов (P2P). Фактами может быть что угодно, от денежных операций и до подписания контента. Члены сети – анонимные лица, называемые узлами. Все коммуникации внутри сети используют криптографию, чтобы надежно идентифицировать отправителя и получателя. Когда узел хочет добавить факт в журнал, в сети формируется консенсус, чтобы определить, где этот факт должен появиться в журнале; этот консенсус называется блоком.

Эта идея может быть воплощена для реализации инфраструктуры открытых ключей без построения строгой иерархии уполномоченных на сертификацию. Что, в свою очередь, резко снизит расходы на содержание громоздкой системы иерархической структуры.

Основные принципы децентрализованной ИОК.

1. Каждый пользователь (пользователь выступает узлом) хранит свою ключевую пару самостоятельно. Сертификат открытого ключа передается вместе с подписанным сообщением.
2. Запись о транзакции по законам blockchain хранится в распределенной базе.
3. Блок транзакций содержит реестр состояний сертификата.
4. При проверке правильности транзакции (фактически действительности сертификата открытого ключа) проверяющему необходимо проследить реестр состояния сертификата отправителя вплоть до его первой публикации (аналогичные действия проходят в системе Bit-Coin для проверки наличия «средств» на «счету» клиента, т.е. исключение «двойного расходования»).
5. Первичная идентификация нового пользователя является обязательной и должна быть надежно подтверждена. Для этой и только для этой цели необходим доверенный узел (аналог уполномоченного на сертификацию в иерархической структуре). Его роль будет состоять в первичном выпуске сертификата нового пользователя, а также в случаях, необходимых для изменения статуса сертификата. После первой транзакции, совершенной этим пользователем, обращение к доверенному узлу больше не возникает, кроме ситуаций, требующих изменения статуса сертификата данного пользователя. Т. е. данный узел будет обеспечивать новых пользователей «родительским» блоком («genesis block»), для того, чтобы уже существующие узлы могли проверять статусы сертификата нового пользователя. Целесообразным представляется возложить эту роль на государственную структуру.

Введем следующие условные обозначения:

M – сообщение,

$Sign$ – цифровая подпись отправителя,

H- криптографическая хеш-функция,

$Sert$ – сертификат открытого ключа отправителя,

ID – уникальный идентификатор отправителя, выданный ему при первичной идентификации,

$Status$ – статус сертификата открытого ключа отправителя,

$Sign$ – цифровая подпись.

Первичная идентификация пользователя. Как уже упоминалось, первичная идентификация должна проводиться государственной структурой (доверенным узлом). При обращении к которой пользователю будет выдан его уникальный идентификатор ID и соответствующий ему сертификат открытого ключа $Sert$. Следует отметить, что доверенный узел не хранит у себя ID пользователя, более того, он его не знает.

Первая транзакция нового пользователя должна быть обращена к доверенному узлу для того, чтобы последующие могли ссылаться на нее по законам blockchain. Так как для надежного подтверждения транзакции необходимо вычисление 3 – 5 блоков, следующих за блоком с данной транзакцией, рекомендуется оправлять транзакцию не только к одному представителю доверенного узла, а к нескольким (оператор регистрации, оператор сертификации, администратор безопасности).

После прохождения первичной идентификации данные распространяются в распределенную базу данных, в которой они хранятся в следующем виде:

Данные, хранящиеся в виде таблицы в распределенной БД (blockchain)

$H(Sert,ID)$	$H(Sert,Status)$	$Status$

Алгоритмы формирования и проверки подписи. Алгоритм формирования подписи не отличается от существующего и зависит только от типа используемой подписи.

Алгоритм проверки подписи в ИОК на основе тенологии blockchain

Отправитель генерирует транзакцию:

$$M;\ Sign;\ H(Sert, ID);\ Sert;\ Status \quad (1)$$

Ниже приведен вариант адаптированного протокола:

1. {«hash»:« »,
2. «ver»:1,
3. «vin_sz»:1,
4. «vout_sz»:1,
5. «lock_time»:0,
6. «size»: ,
7. «in»:[
8. {«prev_out»:
9. {«hash»:« …»,
10. «n»:0},
11. «scriptSig»:«… …»}],
12. «out»:[
13. {«value»:« »,
14. «scriptStatus»:« »}]}

Здесь строка 1 содержит хэш оставшейся части транзакции, выраженной в шестнадцатеричном виде. Это используется в качестве идентификатора транзакции.

Строка 2 указывает на версию протокола.

Строки 3 и 4 указывают на то, что транзакция имеет один ввод и один вывод соответственно.

Строка 5 содержит значение lock_time, которое может быть использовано для контроля, когда транзакция будет завершена. Если lock_time установлен в 0, это означает, что транзакция немедленно завершена.

Строка 6 содержит размер (в байтах) транзакции.

Строки с 7 по 11 определяют входные данные к операции. В частности, строки с 8 по 10 говорят нам, что ввод должен быть взят с вывода из предыдущей сделки с соответственной хэш-суммой, выраженной в шестнадцатеричном формате, n = 0 указывает на то, что это будет первый вывод из той транзакции. Строка 11 содержит подпись отправителя, затем пробел, а затем соответствующий открытый ключ в шестнадцатеричном формате.

Строки с 12 по 14 определяют выходные данные. В частности, строка 13 говорит нам о значении вывода: статус сертификата. Строка14 это отображение языка сценариев и адреса строки.

Алгоритм проверки состоит из двух этапов:

I этап заключается в проверке цифровой подписи $Sign$ на основании сертификата открытого ключа отправителя $Sert$. Если эта проверка выполнена успешно (т.е. цифровая подпись наложена именно при помощи личного ключа, которому соответствует предоставленный сертификат открытого ключа отправителя), необходимо переходить к этапу 2 для удостоверения, что данный сертификат открытого ключа отправителя действительно принадлежит отправителю.

II этап состоит из таких шагов:

1. Получаем значение и адрес поля $Status$ из таблицы на основании полученного от отправителя $H(Sert, ID)$;
2. Вычисляем $H'(Sert,\ Status)$;

3. Получаем значение и адрес поля $Status^{/}$ из таблицы на основании $H^{/}(Sert,\ Status)$;

4. Если значение и адрес $Status = Status^{/}$, проверка считается успешной.

Предложенная выше система обладает рядом преимуществ:

– значительное снижение затрат на содержание громоздкой иерархической структуры уполномоченных на сертификацию;

– пользователь самостоятельно контролирует свои идентификационные данные и способен немедленно сообщить о необходимости их корректировки (компрометации);

– нивелирование угрозы «man in the middle»;

– «исчезновение цели» для направленной атаки. В отличие от иерархической структуры, когда главными мишенями для злоумышленников были центры сертификации ключей, в данном случае отсутствует явная цель для атаки, т.к. записи хранятся распределенно и, по сути, злоумышленник вынужден атаковать всю сеть целиком, а не конкретный узел;

– система может быть использована не только непосредственно для услуги электронной подписи, но и для обеспечения электронной идентификации граждан;

– выход из строя одного или нескольких узлов не приводит к остановке системы;

– отсутствие необходимости делать и хранить резервные копии;

– интероперабельность системы заключается в том, что сертификаты, выпущенные различными уполномоченными на сертификацию, могут легко использоваться в единой системе;

– легкая масштабируемость, т.к. добавление нового пользователя (нового узла) происходит без изменений основных принципов функционирования архитектуры.

Выводы

1. Анализ показал, что стойкость ИОК на базе технологии blockchain будет превышать стойкость централизованной системы. Следует понимать, что речь идет не о криптостойкости, а именно о резильентности системы в целом.

2. Энергетические затраты, необходимые для реализации атаки на систему, будут составлять 50 % от вычислительной мощности такой системы. Нарушителю необходимо будет атаковать всю систему целиком. Соответственно, для того, чтобы иметь 50 %-й шанс на успех в решении одного блока, ему необходимо будет располагать вычислительной мощностью равной вычислительной мощности всей остальной системе. Кроме того, рекомендация 3 – 5 ступенчатого подтверждения резко и значительно снижает его шансы, т.к. для этого ему будет необходима вычислительная мощность, которая существенно превышаюшает вычислительную мощность всей системы. Таким образом, стойкость системы повышается с ростом числа узлов (пользователей).

3. В существующей ИОК данная концепция позволит ликвидировать проблему совместимости сертификатов, выпущенных различными уполномоченными на сертификацию.

4. Применение изложенного подхода позволит облегчить переход на новые алгоритмы подписей, в частности на постквантовые, в которых стойкость зависит не от криптопериода ключа (3 года, 5 лет), а от количества наложенных подписей (например в hash based подписях). Исходя из этого, blockchain технология позволит более рационально управлять сертификатами открытых ключей.

5. Более того, для некоторых алгоритмов hash based подписей (таких как подписи Merkle, XMSS) взможны упрощения алгоритмов (исключения из протокола сертификата открытого ключа и использование вместо него пути аутентификации), что позволит уменьшить объем передаваемых данных, т.к. путь аутентификации должен быть передан в любом случае согласно алгоритмов подписи Merkle, XMSS.

Список литературы:

1. ISO/IEC 9594-8 ITU-T Rec. X.509 «Основные положення сертификации ключем и сертификации атрибутов».

2. Инфраструктура открытых ключей: технологии, архитектура, построение и внедрение : учеб. пособие / А.В. Потий, А.В. Леншин, Л.С. Сорока, В.И. Есин, Б.И. Мороз. Днепропетровск : Академия пограничной службы Украины, 2011. 202 с.

3. CrypViser GmbH Whitepaper

4. Michael Nielsen How the Bitcoin protocol actually works http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Харьковский национальный
университет имени В.Н.Каразина;
Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 05.03.2018