Перевод спонсировали: Alex Momot, Andrey Solovey, Mike Last.

Перевод: Валерия Андрус

Переведено с помощью сервиса iwillunderstand.cc

Введение

Сейчас пароль – это основный способ получения доступа к различным локальным и сетевым ресурсам. Это метод, который позволяет идентифицировать и защищать как юзеров, так и ресурсы. Однако у этого метода есть один существенный недостаток – если злоумышлённик украдёт пароль, то он получит доступ сразу ко всем данным владельца учётной записи. А если брать во внимание привычку юзеров пользоваться одним и тем же паролем в нескольких приложениях сразу, то ситуация ещё хуже. Плюс если пароль ещё и слабый, он может подвергнуться так называемой атаке «словарного перебора». Также периодически базы данных паролей время от времени сливаются в открытый доступ, и хакеры получают доступ к миллионам паролям [1].

Даже если пострадавший сможет сменить свой пароль, его аккаунт останется уязвимым, так как электронная почта пользователей обычно защищена тем же паролем, который используется на других ресурсах. Таким образом, пароли малоэффективны в защите данных и сеанса пользователя.

Парольный менеджмент может решить эту проблему. Тот софт, который обеспечивает систему безопасного хранения паролей, обычно встраивается как браузерное расширение, которое может противостоять различным хакерским атакам. Менеджеры паролей зачастую могут генерировать сложные пароли, которые будут уникальными для каждого ресурса, что, в свою очередь, выводит систему на качественно новый уровень безопасности. Очевидный недостаток этого механизма состоит в том, что такая система хранения защищена мастер-паролем (Lastpass, 1Password), и в случае кражи или атаки методом «грубой силы» (bruteforcing) на такой важный пароль, все вышеперечисленные угрозы станут актуальны и для такой системы.

Метод двухфакторной аутентификации (2FA) разработан специально для решения таких проблем с паролями. Если объединить 2FA со стандартной защитой пароля, то сайт будет требовать дополнительную информацию, которая доступна только конкретному юзеру.

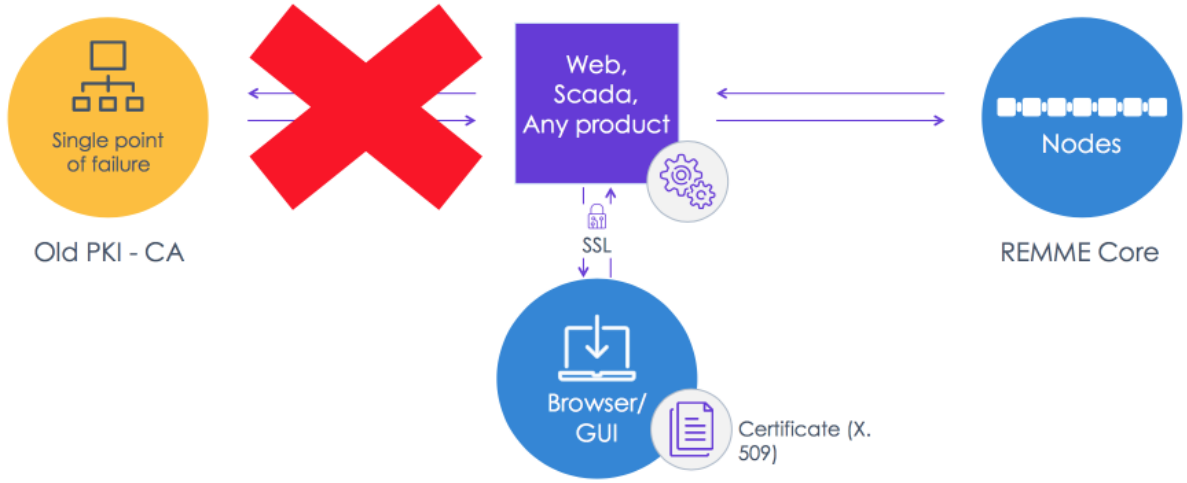
Существуют такие вариации 2FA:

1. Наиболее распространенный вариант – это система одноразовых паролей, которые генерируются каждые N секунд. Обычно идут в комплекте с TOTP протоколом.
2. Одноразовые пароли, которые приходят в текстовом сообщении, как правило, через SMS или мессенджеры.

3. Аппаратный токен.

Использование SSL/TLS сертификатов – это альтернативный способ авторизации юзера в системе, который широко применяется в таких масштабных областях, как банкинг и налоговая система. SSl сертификаты основаны на системе открытых ключей, с помощью которых администраторы могут управлять сертификатами. Современные PKI-системы базируются на так называемой «цепочке доверия»: так, существует основной сертификационный орган, который подписывает сертификаты для других центров. Эти другие центры, в свою очередь, подписывают сертификаты конечным пользователям.

У этой системы тоже есть существенный минус: вся её инфраструктура зависит от доверия к основному и аффилированным сертификационным центрам. Если один из линков или основных сертификатов даёт сбой, вся система теряет свою производительность. Также критически важным моментом остаётся вопрос соотношения централизованной структуры и системы с доверенным сертификационным органом.



Чтобы решить эти проблемы, мы опробовали различные подходы к децентрализованной инфраструктуре открытых ключей, которые основываются на блокчейн-технологии. В нашем понимании, эффективным будет тот подход, благодаря которому конечные потребители смогут управлять своими PKI с высоким уровнем доверия и хорошей отказоустойчивостью (по мере распространения системы). REMME решает следующие проблемы:

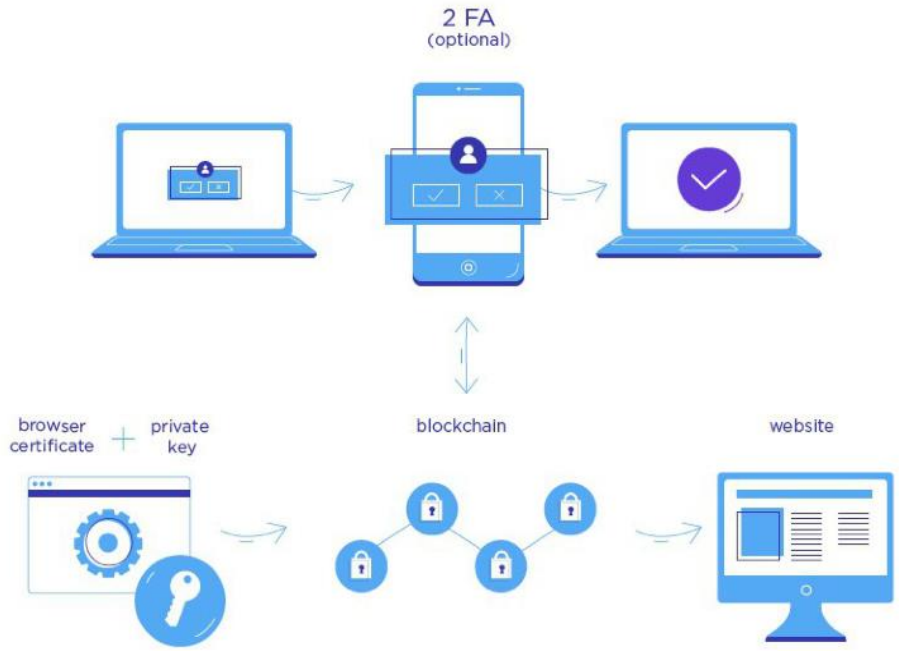
1. Доверия централизованному сертификационному органу (CA[1])
2. Засвеченных слитых ключей CA – сертификационных органов
3. Выдачи фэйковых сертификатов
4. Выдачи «теневых» сертификатов с целью перехвата и перенаправления связей

[1] certification authority

Благодаря решению этих проблем мы сможем установить всеобъемлющий и прозрачный контроль и отслеживать все когда-либо выданные сертификаты, искоренить «теневые» сертификаты и таким образом свести выгоду от перехвата и перенаправления связей к нулю.

Множество организаций (в том числе и Гугл) пыталось решить эти проблемы, но безуспешно [2]. Система блокчейна гарантирует неизменность хранимых в них данных. Это эффективное и практическое решение.

Апробация концепта



Чтобы решить проблемы, которые возникли из-за централизации нынешних PKI систем, мы предлагаем систему, которая базируется на протоколе Биткойна. Этот концепт лёг в основу второй версии REMME. Держите в уме, что тут подразумевается использование самоподписываемого сертификата X.509 [3]. В этой системе софт Биткойна служит следующим целям:

1. Аннуляции сертификатов. Каждый сертификат привязан к выходу определённой биткойн-транзакции. Сертификат признаётся недействительным сразу же после проведения энной транзакции.
2. Аутентификации сертификата. Каждый сертификат хранит цифровую подпись строки (которая подписывается владельцем сертификата; сама

строка называется «канарной»[2]). Это поле задаётся стандартами REMME и биткойн-адресом своего владельца. При этом данные сертификата можно использовать для заполнения вышеупомянутой строки и для проверки подписи с помощью его (сертификата) биткойн-адреса.

В системе используются следующие поля сертификата:

UID	Биткойн-адрес владельца сертификата.
L	Цифровая подпись канарного поля, которая делается с помощью биткойн-сайнмесседжа (signmessage).
OU	Идентификатор той транзакции, которая нужна для аннуляции сертификата.
ST	Количество выходов вышеупомянутой транзакции.

Канарная строка формируются по следующему принципу: https://REMME.io/certificate/$CERT_SN/$PUBKEY_HASH/$CERT_OU/$CERT_ST ,

где:

$CERT_SN – серийный номер сертификата;
$PUBKEY_HASH – SHA256 – хэш открытого ключа сертификата;
$CERT_OU – идентификатор используемой транзакции;
$CERT_ST – номер выхода используемой транзакции.

Это поле подписывается с помощью функции сайнмесседжа[3], которая используется при реализации стандартного биткойна (ядро биткойна) и записывается в соответствующее поле сертификата.

Таким образом, весь процес создания REMME-совместимых сертификатов выглядит так:

1. Создайте пару ключей.
2. Создайте транзакцию биткойна, которая будет использоваться для аннуляции сертификата.
3. Создайте сертификат и заполните его поля в соответствии с требованиями REMME.
4. Создайте канарную строку.
5. Подпишите канарную строку с помощью сайнмесседжа.
6. Добавьте канарную строку в соответствующее поле сертификата.

[2] canary string
[3] signmessage с англ.

7. Подпишите сертификат.

А чтобы проверить сертификат, вам понадобится:

1. Получить сертификат.
2. Убедиться, что выход соответствующей транзакции ещё не потрачен (то есть, сертификат ещё актуален).
3. Сформировать канарную строку с помощью данных сертификата.
4. Проверить подпись канарной строки, которая присоединена к сертификату, с помощью функции верификационного сообщения в Биткойне.

Эта система позволяет своим юзерам управлять сертификатами на основе распределённой базы данных Биткойна (Bitcoin's distributed storage ledger). Стоит отметить, что эта система чрезвычайно портативна, потому что она основывается на тех же особенностях, что и большинство ныне существующих блокчейнов.

3. Эволюция REMME

Несмотря на то, что это не первая распределённая система управления сертификатами на основе блокчейн-технологии, первая итерация или REMME всё-таки отличается от системы DNS SSL, которую создали разработчики Namecoin [5] и EMCSSL [6] с помощью технологии Emercoin. Они первыми начали сочетать блокчейн и SSL.

Система DNS SSL позволяет привязывать DNS к хэшу сертификата, а EMCSSL, созданный на основе децентрализованного Emercoin-а специально для хранения key-value bundle, может конвертировать а) серийный номер сертификата в ключ и б) хэш сертификата в хэш-значение[4]. Этот ключ – уникален, а хэш-значение можно изменить. Таким образом, налаживается система управления сертификатами.

REMME отличается от этих двух систем тем, что данные сертификата не сохраняются в начальной точке блокчейна. Аннуляция сертификата должна произойти путём публикации секретного сообщения об аннуляции, которое подписывается закрытым ключом сетевого блокчейн-адреса, как уже упоминалось выше. Эта схема, которая используется для соединения блокчейн-адреса с сертификатом, очень похожа на механизм прокси-сертификата, который описывается в RFC3820 (X.509 Proxy Certificate Profile) [7].

[4] While the DNS SSL system allows the binding of DNS to the certificate hash, the EMCSSL built on the Emercoin decentralized solution for storing the key-value bundle enables the serial number of the certificate to be the key and the certificate hash as the value.

Однако, система, которая основана на существующей криптовалюте, имеет очевидные недостатки - низкую пропускную способность и высокие операционные издержки. Так что наше решение – это выстроить собственную децентрализованную систему с тем финансовым дизайном, который соответствует потребностям продукта.

От такой системы в первую очередь требуется:

1. Привязка ценовой политики к актульной валюте, например, американскому доллару.
2. Высокая пропускная способность блокчейна для оперативного управления сертификатами.
3. Возможность создания частного (закрытого) блокчейна для интеграции в разные организации.

Консенсусный алгоритм

Консенсусный алгоритм – это ключевая часть системы из-за того, что она обеспечивает высокую пропускную способность, которая так нужна для её эффективного функционирования. Широко применяемый PoW-консенсус тут не подходит, потому что для его функционирования нужно слишком много вычислений, которые тормозят работу всей системы [8]. Таким образом, нужно искать другие более шустрые алгоритмы, которые не нуждаются в излишних вычислениях. Также такой консенсусный алгоритм должен сам по себе мотивировать юзеров задерживаться в онлайне и проводить транзакции как можно чаще.

Сейчас известны такие эффективные алгоритмы, как:

1) Proof-of-Stake (PoS), или доказательство доли владения. Этот алгоритм смотрит на то, как много криптовалюты есть у клиента: чем её больше, тем выше вероятность формирования этим юзером нового блока. Есть и пороговое количество монет – во многих системах с этим алгоритмом юзеру нужно всегда обладать определённым «прожиточным минимумом». Недостаток этого метода состоит в высоком уровне угрозы атаки двойного расходования. Пытаясь нивелировать эту угрозу, некоторые криптовалюты начали развивать новые подходы, почву для которых подготовил PoS. Например, NEM пытается задавать другие параметры, такие как аптайм[5], но с ростом количества монет на узле эта фишка утрачивает свою эффективность [9].
2) Proof-of-Authority (PoA), или доказательство полномочий. В рамках этого алгоритма в блокчейне создаётся список определённых узлов, которым разрешается создавать новые блоки. С таким фиксированным списком узлов

[5] время непрерывной работы вычислительной системы - Википедия

будет очень просто создать детерминированный консенсусный алгоритм, согласно которому узлы будут выбираться в таком же фиксированном порядке. Этот метод по природе своей больше подходит для частных блокчейнов, но его можно экстраполировать и на открытую систему с высокой производительностью. Например, протокол Clique позволяет добавлять и удалять узлы со списка с помощью проведения голосования, в котором участвуют авторизованные узлы [10].

Существует одна характерная черта, которая присуща сразу всем вышеупомянутым консенсусным алгоритмам. У всех них есть набор заранее заданных узлов, которые соответствуют определённым требованиям плюс заданный порядок узлов, в соответствии с которым формируется новый блок. Эти параметры задаются самим алгоритмом, а не вычислительной способностью или чем-то ещё. Так как мы планируем использовать систему как и публичный сервис, так и интегрированную приватную систему, мы должны обеспечить техническую сторону вопроса обоим вариантам. Для частной системы нужно использовать PoA, потому что этот алгоритм предлагает фиксированный список узлов, поддерживающих сеть. Что касается координации открытой системы, нужно расширить консенсусный алгоритм PoA таким образом, чтобы каждый узел мог «самовыдвинуться» в список авторизованных узлов и автоматически попасть туда в соответствии с определёнными критериями вроде такого-то числа монет на балансе, аптайма и так далее.

Ценовая политика

Так как, помимо всего прочего, нужно привязать цены к фиатной валюте, платформе нужны какие-то методы, с помощью которых она будет обновлять цены на свои услуги с привязкой к курсу токена. Мы предлагаем такие методы:

1) Использовать доверенный узел, принадлежащий REMME, для обновления цен, которые уже записаны в блокчейне. Адрес этого узла будет привязан к программному обеспечению.
2) Каждый узел формирует собственные ценовые списки на основании определённого алгоритма, а цены, в свою очередь, обновляются тем узлом, который создаёт новый блок. Проверка корректности цен происходит в рамках работы верификационного алгоритма блока.

3.3. Миграция токена между блокчейнами

Для подготовки работы миграционного механизма необходимо наладить межблокчейновую миграцию токена. Это позволит клиентам пользоваться токеном REM, который был изначально выпущен на платформе Эфириума, в блокчейне REMME. Что касается нынешнего положения проекта, тут оптимальный вариант – это подобрать централизованную систему по типу

криптоплатформы Waves. Идея следующая. Один из узлов отслеживает транзакции как в Эфириуме, так и в REMME. Когда пользователь пришлёт токены узлу Эфириума (со своего REMME-адреса, который записан в метаданных – может понадобиться умный контракт), аналогичное количество токенов придет на аккаунт клиента в REMME. Система будет работать по аналогичному принципу и при переылке REM в блокчейн Эфириума [11].

Поддержка сертификатов

В первой версии системы используются сертификаты X.509. Поддерживаются такие юзкейсы:

1. Самоподписанные сертификаты. В этом случае такие данные сертификата, как открытый ключ, подпись, срок действия и информация об аннуляции хранятся в блокчейне.
2. Сертификат, подписанный организацией. В этом случае организация (которая является нашим клиентом) может использовать собственные сертификаты для подписания и управления сертификатами своих клиентов и работников.

Мы планируем интегрировать серверное программное обеспечение для поддержки REMME-сертификатов. Этот софт будет автоматически проверять статус сертификата в блокчейне. Есть несколько вариантов, как сделать это. Например, можно встроить плагин в системы контент-управления или внести самоподписанные сертификаты в список доверенных сертификатов системы. Для этой системы нужно будет использовать такие поля сертификата:

Поле	Использование
Название организации	REMME
UID	Адрес юзера в блокчейне REMME.

2FA

Двухфакторная аутентификация – это дополнительный слой техники безопасности, которым гарантируется, что к аккаунту имеет доступ только его владелец, даже если частный ключ сертификата SSL был засвечен. Мультифакторная аутентификация – это криптографическая система, которая открывает доступ к аккаунту только в случае математически обусловленного соединения нескольких ключей [12]. Как правило, первый ключ генерируется согласно одному фактору, а потом используется для расшифровки второго.

Выбор технологии для второго фактора зависит от характеристики той системы, которая защищена REMME-технологией. Однако, стоит также учитывать, что чем надёжнее метод аутентификации, тем он сложнее и дороже. Таким образом, важно достичь оптимального баланса между надёжностью и сложностью.

Например, если система требует физического присутствия авторизированной персоны, то для прохождения второго фактора лучше всего использовать биометрические данные – отпечатки пальцев или скан сетчатки глаз. Эти данные могут быть использованы для валидации сертификата [13].

Если система пользуется каким-то локальным датчиком, прохождение второго фактора должно происходить посредством физического подключения к локальной сети (то есть, подключения вручную). Если это удалённая система, то будет удобно воспользоваться второстепенным девайсом (телефоном или другим компьютером). Заразить два девайса вредоносной программой сложнее, чем один. Также такой подход поможет защитить аккаунт с засвеченным сертификатом.

Один из самых простых способов двухфакторной авторизации – использование двух разных сертификатов на двух разных девайсах. В этом случае, активация одного сертификата должна подтверждаться активацией второго. Кроме того, можно установить мессенджер на втором девайсе, чтобы получать сообщения с секретными ключами сразу от защищённой системы. В таком случае, надёжность второго фактора эквивалентна надёжности аккаунта в мессенджере. Например, можно использовать Телеграмм (или другие мессенджеры), почту или почту плюс PGP ключ.

Особое внимание стоит уделить стандартному методу TOTP (временный одноразовый пароль[6]), который генерирует одноразовые коды с определёнными временными интервалами (например, каждые 30 секунд). Такая схема используется в приложении Гугл Аутентификатор [14]. Ещё можно использовать чисто аппаратное решение для генерации токенов-аутентификаторов - к примеру, YubiKey [15], Yubico, или Trezor [16].

Структура

В соответствии с требованиями к системе, мы сформировали такие её компоненты:

[6] time-based onetime password

1. Ядро REMME. Главная задача этого компонента – надёжно и безопасно хранить самоподписанные сертификаты плюс отслеживать их статус аннуляции [17]. В случае использования ядра REMME как открытого сервиса, оно будет также отвечать за обработку платежей.

a) В рамках блокчейна REMME существует специфический тип узлов – мастер-узел – который отвечает за организацию создания нового блока. Список мастер-узлов формируется системными администраторами, если это приватная сеть. В случае открытой системы он управляется системой автоматически, как описано в секции «Консенсусные алгоритмы».

b) Любой желающий может запустить узел, в котором хранится целый блокчейн, в целях верификации.

c) Мы предполагаем, что большинство клиентов будут работать в режиме «лёгкого узла». В таком случае клиенту не обязательно хранить весь блокчейн сразу – только ту информацию, которая понадобится для верификации его действий.

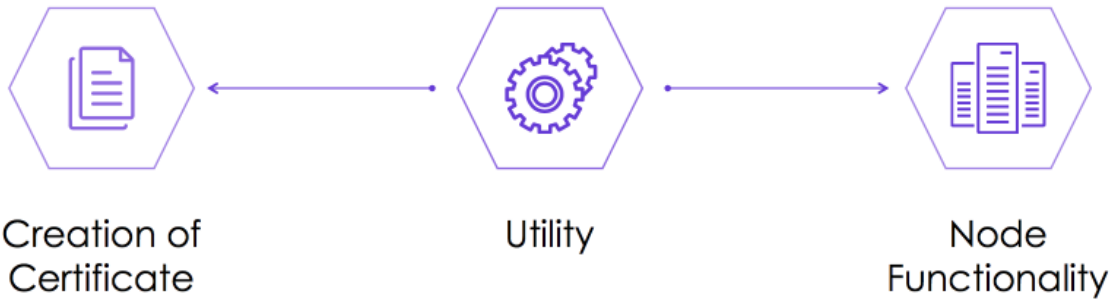
2) Софт клиента (серверная интеграция) нужна для верификации сертификатов, предоставленных REMME.

3) Биткойн-привязочная (anchoring) система для бóльшей безопасности.

4) Оракулы для апдейта ценовых списков. Они нужны нам для привязки цены к фиатной валюте (это обязательно). Вторая причина – системе нужно знать цены на биткойны и эфир из-за тесной связи с ними.

5) Отдельная система для межблоковой миграции токена.

Модель токена



REM-токен необходим для проведения всех внутренних операций в экосистеме, например, таких (это не исчерпывающий список):

1. Запуска процесса генерации сертификатов (энное число токенов блокируется для транзакций аннуляции).

2. Аннуляции сертификата.

3. Установления узла (эта возможность будет доступна после Q3 2018 - об этом читайте детальнее в секции «Дорожная карта»).

4. Сбора комиссии за трансферы токена между блокчейнами.

5. Сбора комиссии за обмен токенами между юзерами в блокчейне REMME.

Структура комиссии зависит от сумм транзакции:

1. Если токенсейлу удастся собрать менее 5 миллионов долларов США, то:

70% комиссии будет храниться на аккаунте мастер-узла, который создал рассматриваемый блок, а 30% перейдёт сразу в REMME. Эти 30% уйдут на управление транзакциями между блокчейнами и такими сервисами, как Биткойн- якоринг (anchoring) и так далее.

2. В случае сбора от 5 до 10 миллионов долларов США на том же токенсейле:

80% комиссии останется на аккаунте мастер-узла, который создал рассматриваемый блок, а 20% перейдут в REMME напрямую и будут использоваться для управления сервисами.

3. В случае сбора более 10 миллионов долларов США на том же токенсейле:

То же самое, что и в случае выше, только соотношение комиссии тут будет 90% к 10%.

Варианты использования (юзкейсы) пользования токенами

Вариант 1. Клиенты из сферы традиционной экономики (банк) хотят воспользоваться REMME. Они переводят REMME деньги в фиатную валюту, а потом REMME меняет эти средства на REM-токены, а часть процентной доли токена отправляет узлам в качестве награды за выдачу сертификата.

Вариант 2. Криптобиржа хочет воспользоваться REMME. Она покупает токены напрямую у владельцев на той бирже, которая поддерживает REM-токены, а потом отправляет часть процентной доли узлам в качестве награды за выдачу сертификата.

Поскольку сертификатов будет требоваться всё больше и больше, цена токена будет расти из-за ограниченного количества всех токенов. Как уже упоминалось выше, цена сертификата привязана к фиатной валюте, поэтому она имеет нелинейную зависимость.

Дорожная карта

1. Q4, 2015:

– Кристаллизация идеи и проверка концепта

2. 2016:

– Ядро REMME MVP V 0.1 с 2FA, которое базируется на блокчейне Emercoin-а, появляется в Телеграмме
– 5 экспериментальных проектов

3. Q2 2017:

– Победитель на хакатоне Майкрософта «Blockchain Intensive»
– Меморандум о взаимопонимании с Укринмашем
– Стратегическое партнёрство с Infopulse

4. Q3 2017:

– Ядро REMME MVP V 0.2 становится CRL-инфраструктурой на основе биткойн-блокчейна, а сертификаты генерируются в браузере
– Релиз «Белой книги» 1.0
– Запуск экспериментального проекта REMME's 2018

5. Q4 2017:

– 4 декабря, 2017: Пресейл для белого списка сообщества
– Релиз «Белой книги» 2.0
– Публичный релиз GitHub

6. Q1, 2018:

– Публичная альфа-версия ядра REMME
– Аудит безопасности продукта и пентестинг (тестирование на проникновение)
– Расширение команды разработчиков программного обеспечения
– Запуск пилотной программы REMME's 2018
– Фаза публичной продажи
– Правовая структура

7. Q2, 2018:

– На этой стадии будут вводиться частные блокчейны для интеграции с общеорганизационными системами[7]. Мы планируем, что частные блокчейны смогут:

a) Быть хранилищем данных о сертификатах, которое основывается на блокчейне;

b) помогать интегрировать различные системы клиентов;

c) проводить привязку к биткойну (anchoring) для повышения безопасности системы.

– Свободный доступ к интеграции с вебсайтами и веб-приложениями (pen source integration libraries for websites and web applications
– Дополнительные вариации 2FA, например, Signal, Status, WeChat, Trezor
– Пентестинг и аудит безопасности продукта
– 20+ интеграций

[7] enterprise systems на англ.

8. Q3, 2018:

– Публичное тестирование. На этом этапе юзеры смогут по собственному желанию стать владельцами мастер-узла. Этих «самовыдвиженцев» будут обрабатывать актуальные мастер-узлы, которые (в случае их соответствия всем критериям) внесут их в список одобренных узлов. Также на этом этапе благодаря развитию PoA мастер-узлы будут регулярно добавляться разработчиками REMME с того же списка одобренных узлов. Разработчики REMME будут обновлять цены в централизованном порядке с помощью доверенного узла (см. «Ценовую политику»). Новая цена окончательно закрепится на своём месте после десяти подтверждений того блока, в которым она была впервые зафиксирована. Плюс на этой стадии произойдут межблокчейновые переводы токенов, так что каждый, который приобрел REMME-токены, сможет извлечь наибольшую выгоду от пользования сервисом.
– Интеграция ядра REMME с системами Active Directory и SCADA
– Расширение децентрализованной экосистемы узлов
– Пентестинг и аудит безопасности продукта
– Открытие офиса продаж в Лондоне. Расширение команды маркетологов: проведение набора продажников-специалистов, которые будут отвечать за европейский вектор развития. Создание саппорта.
- 50+ интеграций

9. Q4, 2018:

– На этой стадии планируется релиз публичной системы. К этому моменту процесс управления мастер-узлами в публичном сервисе станет полностью автоматическим: разработчики REMME больше не учавствуют в этом.
– Также мы запланировали обновление консенсусного алгоритма. Производительность и стабильность операции (или время безотказной работы – аптайм) мастер-узлов оцениваются каждые N секунд. При появлении нового блока (что происходит каждые m секунд), псевдорандомный алгоритм выбирает один из узлов в соответствии с его рейтингом (чем выше оценивается узел, тем выше вероятность, что он будет выбран), а система ждёт появления блока, под которым подписывается этот узел. Если выбранный узел затягивает с созданием нового блока более, чем k секунд, то алгоритм автоматически переходит к следующему узлу. Таким образом, система обеспечивает максимальное соответствие заданным параметрам и мотивирует мастер-узлы оставаться в системе
– Создание приложения DApps для IoT поверх ядра REMME (с прицелом на автомобильные и умные города)
– Пентестинг и аудит безопасности продукта

– Открытие офиса продаж в Нью-Йорке: набор продажников-специалистов для развития американского вектора развития, набор саппорта и расширение команды маркетологов в целом
– 100+ интеграций

10.Q1, 2019

– Открытие офиса продаж в Токио и Сингапуре: набор продажников-специалистов для развития азиатского вектора развития, набор саппорта и расширение команды маркетологов в целом

– проведение специализированных мероприятий, посвященных вопросу кибербезопасности, раз в три месяца

– специальные уроки и курсы по кибербезопасности в Украине для подготовки специалистов и развития экосистемы REMME в целом.

7. Варианты использования (юзкейсы)

7.1. Пример 1: управляемая система с общедоступной информацией

Сервисы криптобирж пользуются REMME для авторизации клиента, чтобы заменить стандартную схему логина и пароля. На ней сидят есть 25,000 клиентов, из которых 20,000 активны. Нагрузка – 2 тысячи посетителей в день.

Прибыль: сервис вносит предоплату за определённое количество сертификатов (со сроком действия в один год), так что конечным юзерам не понадобится оплачивать сертификаты. Каждый год сервис заказывает и оплачивает новые сертификаты для своих клиентов.

Валидность сертификата: только для сервиса.

2FA: REMME обеспечивает сервис с релевантным софтом для реализации метода 2FA на основе сообщений.

Полномочия юзера:
– Создавать сертификат
– Если секретный ключ был засвечен – срочно отзывать сертификат
– Автоматически получать новый сертификат при окончании срока действия старого
– Выбирать приоритетный метод 2FA.

Полномочия администратора:

- Обеспечивать надёжную платёжную систему (с REMME-токенами)
- Мониторинг системы (выдача и аннуляция номеров сертификатов)
- Менеджмент первичных (root) сертификатов
- Выдавать новый сертификат при окончании срока действия старого

7.2. Пример 2: частный блокчейн

Государственная компания хочет интегрировать REMME для авторизации клиентов на своих внутренних сервисах.

Потенциал: до 1,000 клиентов.

Прибыль за интеграцию и поддержку.

Валидность сертификата: только для этого сервиса; также для разных компонентов системы используются разные сертификаты.

2FA: аппаратный ключ.

Полномочия юзера:

- Генерировать сертификат
- Срочно отзывать сертификат, если секретный ключ был засвечен

Полномочия администратора:

- Мониторинг системы (номеров выданных и аннулированных сертификатов, активных сессий)
- Управление первичными (root) сертификатами
- Управление сертификатами разных компонентов системы
- Выдача нового сертификата при окончании срока действия старого
- Срочно аннулировать сертификаты юзеров
- Редактировать данные клиентов, которые хранятся в блокчейнах.

8. Выводы

Цель системы безопасности REMME – помочь защитить конфиденциальные данные компаниям, которые связаны с развитием инфраструктуры и медицинской техники, финансовым и блокчейн-компаниям, IoT-организациям и подобным. REMME будет продвигаться как технология управления Инфраструктурой Открытых Ключей (PKI[8]), которая основана на сертификатном стандарте X.509. Этот стандарт, в свою очередь, пользуется SSL/TLS для защиты всего канала от атаки.

REMME не сохраняет данные сертификата в блокчейне на начальной точке. Аннуляция сетификата происходит посредством публикации секретного сообщения об аннуляции, которое подписывается частным ключом блокчейн-адреса сети[9]. Токен REM нужен для выполнения внутренних операций в рамках экосистемы, так что он функционирует как «utility» токен.

Консенсусный гибрид PoS+PoA используется для улучшения пропускной способности, безопасности и масштабируемости сети. Благодаря этому подходу клиентам не придётся ждать нескольких подтверждений, поскольку сертификат начинает действовать сразу же после создания транзакции.

Двухфакторная аутентификация обеспечивает дополнительный слой защиты, ведь это принцип, согласно которому доступ всегда есть только у владельца аккаунта, даже если частные сертификатные ключи SSL/TLS были засвечены. Выбор технологии для второго фактора зависит от характеристик системы, которая защищена технологией REMME.

По мере развития системы REMME внушительное количество стандартных и известных компонентов уже доказало свою эффективность и прошло проверку временем. Для REMME блокчейн-технология – это транспортное средство и консенсусный способ решения всё той же проблемы – децентрализации.

[8] distributed Public Key Infrastructure ("PKI")
[9] private key of the network blockchain address.

9. Ссылки

1. Verizon data breach investigation report
2. http://www.certificate-transparency.org/
3. X.509 specification
4. https://en.wikipedia.org/wiki/Warrant_canary
5. https://namecoin.org/
6. http://emercoin.com/EMCSSL
7. https://www.ietf.org/rfc/rfc3820.txt
8. Proof-of-work description
9. Proof-of-stake description
10. Ethereum Clique PoA protocol description
11. Waves gate to Ethereum
12. https://www.miracl.com/miracl-labs/m-pin-a-multi-factor-zeroknowledgeauthentication-protocol
13. http://www.cse.lehigh.edu/prr/Biometrics/Archive/Papers/Uludag05.pdf
14. https://www.google.com/intl/ru/landing/2step/features.html
15. https://www.yubico.com/products/yubikey-hardware/
16. https://trezor.io/
17. Our Github repositories