ФИНТЕХ
АССОЦИАЦИЯ

ZKP

декабрь 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Технология распределенного реестра (ТРР) имеет ряд особенностей, которые отличают ее от других систем и выделяют в отдельный класс. Некоторые из них:
- криптографическая связь данных, хранящихся в распределенном реестре;
- согласование хранимых в распределенном реестре данных;
- наличие механизмов построения бизнес-логики для хранимых в распределенном реестре данных, известных как смарт-контракты.

Для реализации этих отличительных особенностей распределенные реестры должны обладать одним важным свойством: **данные должны быть открыты и доступны для проверки всеми валидаторами (участниками) системы.**

При этом стандарты экономической безопасности требуют, чтобы информационные системы, хранящие и обрабатывающие данные, **обеспечивали конфиденциальность информации.**

В то же время открытость, свойственная распределенным реестрам, и требования к сохранению конфиденциальности, предъявляемые к корпоративным информационным системам, противоречат друг другу, что усложняет применение технологий распределенных реестров в корпоративном секторе.

Для всех разработчиков корпоративных платформ на базе распределенного реестра наступает момент, когда при проектировании архитектуры они наделяют систему свойствами конфиденциальности — и чаще всего в ущерб технологическим особенностям распределенного реестра. В известных корпоративных системах изменяется модель консенсуса или от него отказываются вовсе.

Тем не менее с развитием технологий распределенных реестров развиваются и способы обеспечения конфиденциальности без ущерба для децентрализованной обработки данных и ключевых особенностей ТРР. Можно выделить следующие способы: приватные смарт-контракты, смешивание транзакций, кольцевые подписи, сервисы передачи конфиденциальной информации, гомоморфное шифрование и протоколы с нулевым разглашением. Обзор существующих подходов к обеспечению конфиденциальности в распределенном реестре мы оставим за рамками данного исследования, в Приложении 1 кратко описан каждый подход.

В своей работе мы выделяем и исследуем протоколы с нулевым разглашением как наиболее перспективный путь обеспечения конфиденциальности, способный оказать оказать положительное воздействие на развитие технологий распределенных реестров (от конфиденциальности до масштабируемости решений) и реализацию бизнес-сценариев, которые были невозможны из-за отсутствия сильных свойств конфиденциальности.

ЗАМЕТКА:

Протоколы доказательств с нулевым разглашением позволяют доказать истинность некоторого утверждения, не раскрывая значимой информации по этому утверждению.

ОБ ИССЛЕДОВАНИИ

ЦЕЛИ

Цели исследования – определить актуальное состояние развития протоколов с нулевым разглашением применительно к технологии распределенного реестра, а также установить технологическую зрелость проектов, реализующих такие протоколы. Кроме того, в рамках исследования планируется выявить сложности, связанные с использованием протоколов с нулевым разглашением, при проектировании, разработке и эксплуатации информационных систем.

Результатом исследования станет определение возможности и необходимости стандартизации протоколов данного класса для применения в России, в том числе для применения в сертифицированной блокчейн-платформе Мастерчейн.

ГИПОТЕЗЫ

Мы сформулировали следующие гипотезы о свойствах и применимости протоколов с нулевым разглашением в распределенном реестре:

1. При условии, что протоколы с нулевым разглашением не раскрывают дополнительную информацию при взаимодействии участников, они позволяют усилить конфиденциальность существующих решений, снизив количество метаданных в распределенных реестрах.

2. Так как протоколы с нулевым разглашением можно применять в распределенных реестрах, они позволяют строить принципиально новые способы бизнес-взаимодействия, основанные на криптографически проверяемой информации без ее раскрытия.

ЗАДАЧИ ИССЛЕДОВАНИЯ

1. На основе анализа научных публикаций выделить математические основы и теоретические подходы к построению протоколов с нулевым разглашением.

2. Провести анализ прикладных реализаций протоколов с нулевым разглашением в распределенных реестрах и выявить актуальное состояние применения таких протоколов в корпоративных системах на основе распределенного реестра.

3. На основе анализа прикладных реализаций определить основные сложности проектирования, разработки и эксплуатации протоколов с нулевым разглашением в распределенном реестре.

4. Изучить мнения экспертов рынка технологий распределенного реестра о применимости протоколов с нулевым разглашением.

5. Определить, есть ли необходимость в стандартизации протоколов, и описать этапы стандартизации, необходимые для использования протоколов в Мастерчейн.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИСПОЛЬЗОВАНИЯ ZKP В БЛОКЧЕЙНЕ

ОСНОВНЫЕ ПОЛОЖЕНИЯ

Доказательство с нулевым разглашением (Zero-knowledge proof, далее ZKP) – это криптографический коммуникационный протокол, который позволяет доказывающей стороне (Prover, P) убедить проверяющего (Verifier, V), что некоторое вычислительное утверждение (Statement, S) является корректным. В распределенном реестре ZKP-протокол может использоваться для сокрытия информации, которая там размещена. [6]

ДОКАЗАТЕЛЬСТВА С НУЛЕВЫМ РАЗГЛАШЕНИЕМ

Доказательство с нулевым разглашением должно обладать следующими свойствами:
1. полнота: при корректном и верном утверждении и соблюдении сторонами условий протокола проверяющая сторона может однозначно убедиться в корректности и истинности утверждения;

2. устойчивость: при ложном утверждении проверяющий сможет убедиться в ложности утверждения;

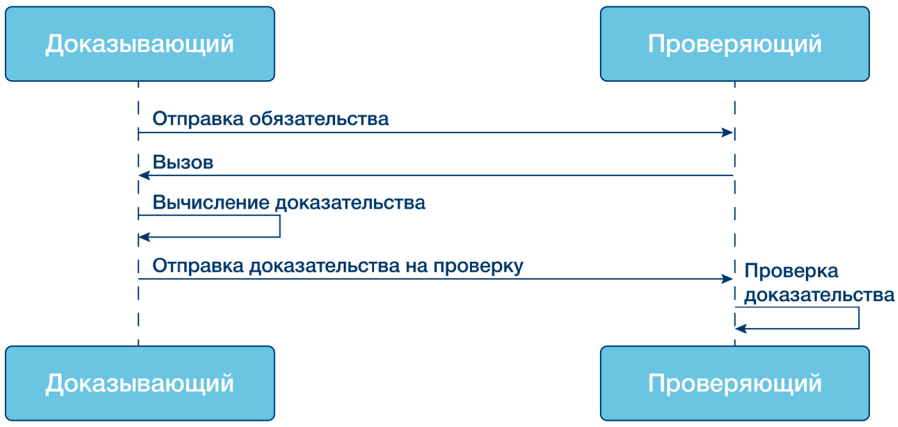
3. нулевое разглашение: проверяющий не сможет получить доступ к проверяемой информации.

Стоит отметить, что ZKP относится к «вероятностным доказательствам». Это класс криптографических протоколов, в которых проверяющая сторона может сделать вывод о том, что утверждение справедливо с некоторой вероятностью [5]. ZKP-протоколы можно поделить на интерактивные и неинтерактивные.

В интерактивных протоколах доказывающий и проверяющий должны быть постоянно на связи (онлайн) для обмена сообщениями (это называется раунды взаимодействия). Таким протоколам свойственна следующая последовательность шагов:

1. доказывающий отправляет проверяющему сообщение, или «обязательство» (англ. commitment);
2. проверяющий отвечает битовой строкой, или «вызовом» (англ. challenge);
3. доказывающий возвращает сообщение с ZK-доказательством, вычисление которого связано с «обязательством», секретным свидетельством и хочет доказать «вызовом»;
4. протокол повторяется произвольное количество раз.

Схема 1. Алгоритм работы интерактивного протокола доказательства с нулевым разглашением



В неинтерактивных протоколах взаимодействие между участниками сводится к одному раунду, а именно пересылке ZK-доказательства от доказывающего к проверяющему:

1. доказывающий генерирует случайное число и на основе него и других параметров создает «вызов», таким образом роль проверяющего на этом шаге заменяется на эвристику Фиата-Шамира [9];
2. далее доказывающий отправляет проверяющему ZK-доказательство, содержащее возможные результаты обменов, которые могли произойти в интерактивном режиме, но не произошли.

Схема 2. Алгоритм работы неинтерактивного протокола доказательства с нулевым разглашением

Применение интерактивных протоколов в распределенном реестре ограничивается необходимостью постоянной онлайн-связи доказывающего и проверяющего для проведения большого количества раундов взаимодействия. На практике такие протоколы не исследуются и не применяются в распределенных реестрах.

Неинтерактивные протоколы лучше подходят для распределенных реестров, так как не требуют отправки сообщений на начальном этапе работы протокола при обмене «обязательством» и «вызовом». Для неинтерактивных протоколов достаточно одного раунда взаимодействия: отправки доказательства и ответа при его проверке.

Особенностью распределенных реестров является то, что все данные хранятся открыто, информация о транзакциях не скрывается – в описании каждой операции содержатся адреса отправителя и получателя и передаваемые данные. При этом нельзя скрыть отдельные транзакции, данные о них или некоторые поля транзакции.

Далее мы рассмотрим проекты на базе распределенного реестра, использующие неинтерактивные протоколы с нулевым разглашением, которые решают задачу сокрытия чувствительной информации.

ОБЗОР МИРОВОГО ОПЫТА ПРИМЕНЕНИЯ НЕИНТЕРАКТИВНЫХ ZKP-ПРОТОКОЛОВ

Интерес к неинтерактивным протоколам с нулевым разглашением обусловлен развитием технологии блокчейн в корпоративном секторе. Также интерес к ZKP-протоколам связан с разработкой этих протоколов экспертным сообществом экосистемы Ethereum.

В зависимости от конфигурации проекты, реализующие протокол с нулевым разглашением, могут по-разному использовать криптографические примитивы, свои строительные блоки, комбинируя их согласно заложенным требованиям и прикладному применению.

Таблица 1. Примеры использования неинтерактивных ZKP-протоколов в распределенных реестрах

Название	Платформа	Уровень	Описание	Протокол	Ссылки	Git
ING-Bank ZKRP	Ethereum, Corda	Протокол	Реализации протоколов **Bulletproofs, Zero Knowledge Range Proof** (ZKRP) и **Zero Knowledge Set Membership** (ZKSM).	Bulletproofs, ZKRP, ZKSM	https://www.ingwb.com/media/2667860/zero-knowledge-range-proofs.pdf	https://github.com/ing-bank/zkrp
Nightfall	Ethereum	Прикладной	Набор инструментов, совмещающий ZoKrates и смарт-контракты стандартов ERC-20 и ERC-721 для выполнения конфиденциальных транзакций с этими токенами. Токсичный отход автоматически удаляется при генерации артефактов. **Недостатки:** требуется доверенная сторона для прохождения доверенной установки; ключи доказательств не рекомендуется хранить в блокчейне, так как они очень большие.	zkSNARK	https://raw.githubusercontent.com/EYBlockchain/nightfall/master/doc/whitepaper/nightfall-v1.pdf	

Название	Платформа	Уровень	Описание	Протокол	Ссылки	Git
ZSL	Quorum	Прикладной	Совместный проект команды J.P. Morgan и ZCash для реализации протокола создания смарт-контрактов (в т.ч. нереестровых) с нулевым разглашением (z-contract) и токенов с нулевым разглашением (z-token).	zkSNARK	https://github.com/ConsenSys/quorum/wiki/ZSL	https://github.com/ConsenSys/zsl-q/tree/master/docs
ZoKrates	Ethereum	Инструменты разработки	Набор инструментов для использования zk-SNARK в сети Ethereum. ZoKrates позволяет разработчикам создавать доказательства, а затем проверять их при помощи Solidity, что дает возможность проверять доказательства в децентрализованных приложениях (DApp) на основе цепочки блоков Ethereum.	zkSNARK	https://zokrates.github.io/	https://github.com/Zokrates/ZoKrates
QEDIT	Произвольные	Инструменты разработки	SDK для разработки корпоративных систем, основанных на блокчейне и использующих протоколы с нулевым разглашением. Представляет фреймворк для построения внереестрового программного обеспечения для создания цифровых активов с соблюдением полной конфиденциальности данных транзакций с цифровыми активами.	Произвольные	https://raw.githubusercontent.com/QED-it/zkinterface/master/zkInterface.pdf	https://github.com/QED-it
ZK Sync	Ethereum	Прикладной	Проект по увеличению производительности блокчейна Ethereum. Используется подход Rollup, в котором вычисления и хранение происходят вне цепочки. Для каждого Rollup-блока генерируется и проверяется ZK-доказательство в основной цепи.	zkSNARK	https://zksync.io/faq/tech.html#zkrollup-architecture	https://github.com/matter-labs/zksync
Zcoin	Zcoin	Криптографическая система	Криптовалюта общего назначения, позволяющая скрывать владельцев криптовалюты.	Sigma	https://eprint.iacr.org/2014/764.pdf	https://github.com/zcoinofficial
StarkWare	Ethereum, StarkWare	Криптографическая схема	Реализация zk-STARK от создателей протокола.	zkSTARK	https://starkware.co/resources/	https://github.com/starkware-libs/ethSTARK
Monero	Monero	Криптографическая система	Криптовалюта общего назначения, позволяющая создавать анонимные транзакции посредством кольцевых подписей.	Ring Sing, Bulletproofs	https://eprint.iacr.org/2020/312.pdf	https://github.com/monero-project/monero
PIVX	PivX	Криптографическая система	Криптовалюта общего назначения с возможностью осуществления конфиденциальных транзакций.	zkSNARK	https://pivx.org/white-papers-2/	https://github.com/PIVX-Project

Название	Платформа	Уровень	Описание	Протокол	Ссылки	Git
Zinc	Ethereum	Инструменты разработчика	Язык программирования для создания доказательств с нулевым разглашением для Ethereum и систем доказательств общего назначения.	zkSNARK	https://zinc.zksync.io/index.html	https://github.com/matter-labs/zinc
Zcash	ZCash	Криптографическая система	Криптовалюта общего назначения, позволяющая проводить полностью анонимные транзакции без раскрытия взаимодействующих сторон и предмета взаимодействия.	zkSNARK	https://raw.githubusercontent .com/zcash/zips/master/protocol/protocol.pdf	https://github.com/zcash/zcash
Tornado.cash	Ethereum	Криптографическая система	Повышает конфиденциальность транзакций, разрывая связь между адресами отправителя и получателя. Протокол использует смарт-контракт, принимающий депозиты ETH, которые могут быть сняты по другому адресу. При этом нет никакого способа связать вывод средств с депозитом, что обеспечивает полную конфиденциальность.	zkSNARK	https://tornado.cash/Tornado.cash_whitepaper_v1.4.pdf	https://github.com/tornadocash
Zether	Ethereum, Quorum	Прикладной	Механизм обеспечения конфиденциальности (суммы транзакций скрыты) и анонимности (отправитель и получатель транзакций скрыты) для анонимных Zether (ZSC) смарт-контрактов в сети Ethereum.	Bulletproofs	https://eprint.iacr.org/2019/191.pdf	https://github.com/ConsenSys/anonymous-zether
AZTEC (Anonymous Zero-knowledge Transactions with Efficient Communication)	Ethereum	Протокол	Протокол второго уровня, разрабатываемый для Ethereum с целью проведения транзакций с нулевым разглашением передаваемого сообщения. Использует концепцию «Записка». Поддерживает протокол скрытого адреса, который можно использовать для маскировки связи между «владельцем» и любыми его идентификационными артефактами в блокчейне. Текущая реализация не позволяет скрывать отправителя и получателя, как и в mimblewimble не скрывается графа транзакций.	ZKRP, Plonk (разрабатывается)	https://github.com/AztecProtocol/AZTEC/blob/master/AZTEC.pdf	https://github.com/AztecProtocol
Idemix	Hyperledger fabric	Прикладной	Позволяет скрыть данные пользователя при его входе в систему и проверке данных. Основан на схеме слепой подписи, реализация построена на основе работ по схемам подписи, основанных на спаривании.	Система доказательств Camenisch	https://eprint.iacr.org/2016/663.pdf	https://github.com/hyperledger/fabric/tree/master/idemix

В большинстве представленных проектов используются неинтерактивные протоколы с нулевым разглашением zk-SNARK. Они являются самыми развитыми и изученными в технологиях распределенных реестров, в том числе и в экосистеме Ethereum, базовой для российской блокчейн-платформы Мастерчейн.

ZK-SNARK

zk-SNARK [7] – это криптографический протокол неинтерактивного доказательства знания с нулевым разглашением. Он позволяет доказывать, что вычислительное утверждение удовлетворяет некоторой системе ограничений, выраженной в виде арифметической схемы, при этом не раскрывая способа получения этого утверждения. Подчеркнем особенность zk-SNARK: он обладает достаточно коротким доказательством и не требует много времени на проверку. Именно эта особенность стала ключевой для успеха этого протокола в технологиях распределенного реестра.

Протокол zk-SNARK состоит из 3 функций: **G**, **P** и **V**.

Функция **G** (генератор ключей), принимает параметр **λ** (также называемый «токсичный отход»), программу **C** (необходимая для расчета «**W**» функция с параметрами). Затем генерируется два ключа: «ключ доказывающего» (proving_key) и «ключ проверяющего» (verifing_key).

$$(proving\_key, verifing\_key) = G(\lambda, C).$$

Функция **P** принимает на вход три параметра: ключ доказывающего (proving_key), случайное, общедоступное значение **X** и доказываемое знание **W**itness (которое не раскрывается). На выходе функция **P** создает и возвращает доказательство «proof».

$$proof = P(proving\_key, x, witness).$$

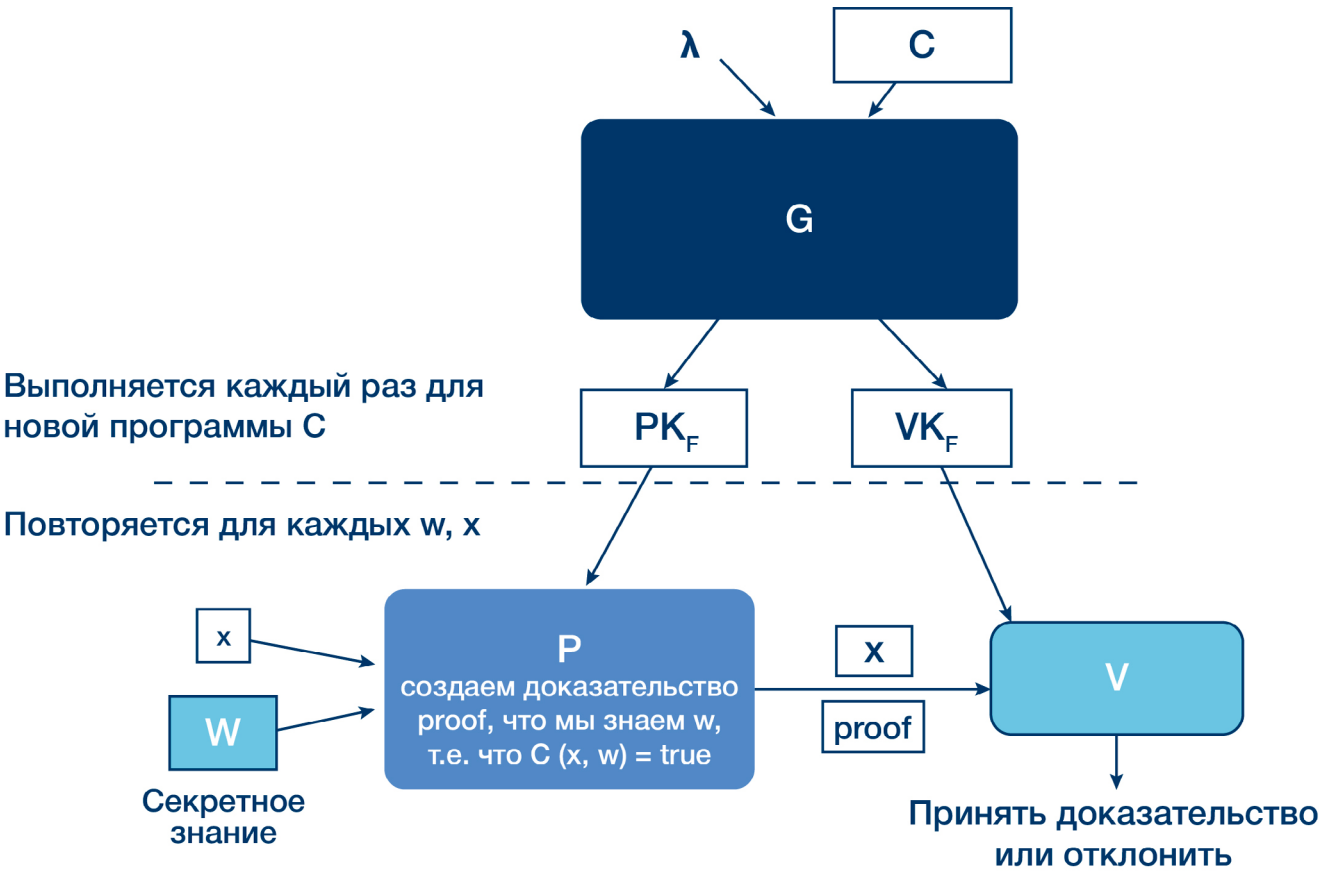
Функция **V** принимает на вход три параметра: ключ проверяющего (verifing_key), общедоступное значение **X** и доказательство (**proof**). На выходе функция **V** возвращает логическую переменную **True** или **False**, указывая таким образом на истинность или ложность доказательства.

$$\{TRUE \mid FALSE\} = V(vk, x, prf).$$

Используемый в функции **G** параметр **λ** - это так называемый «токсичный отход», который должен быть конфиденциальным и при возможности сразу же уничтожен вместе с носителем, так как обладая им, можно подделывать любые доказательства. Поддельные доказательства возвращают значение TRUE независимо от того, действительны ли они на самом деле и знает ли доказывающий секретное значение witness.

В общем виде zk-SNARK состоит из компонентов, схематично обозначенных ниже:

Схема 3. Компоненты и механизм работы протокола zk-SNARK



Для лучшего понимания условий работы протокола нам необходимо дополнительно обозначить специфику среды, в которой исполняется протокол zk-SNARK.

ОСОБЕННОСТИ ZK-SNARK В ETHEREUM

Проверка доказательства в zk-SNARK состоит из операций на эллиптических кривых. В частности, проверяющему требуется скалярное умножение и сложение на группе точек эллиптических кривых, а также билинейное спаривание (вычислительно более сложной операции).

Базовая платформа, на которой основан Мастерчейн, предоставляет реализацию этих операций в виде предварительно скомпилированных контрактов, использующих специальные эллиптические кривые – alt_bn128. Это дает возможность реализовать ZKP-схемы в коде смарт-контрактов. При этом сами алгоритмы создания и проверки

доказательств zk-SNARK не реализованы в Ethereum, из-за чего возникают следующие проблемы:

1. трудности создания сложных ZKP-схем на смарт-контактах. Криптографические операции и криптографические схемы требуют значительного количества операций, тогда как на размер кода смарт-контракта налагаются жесткие ограничения;

2. необходимость создавать отдельные параметры доверенной установки для каждого нового смарт-контракта;

3. потребность в ручной реализации всех алгоритмов.

ЗАМЕТКА:

Ethereum использует Тьюринг-полную виртуальную машину Ethereum (EVM), в которой операции с эллиптической кривой чрезвычайно сложны для выполнения. Для предоставления этих возможностей в EVM существуют так называемые предварительно скомпилированные смарт-контракты, которые реализуют необходимые операции.

Поэтому на этапе прикладного построения ZKP-протокола перед разработчиками стоит выбор:

1. использовать уже готовые инструменты для автоматического создания смарт-контрактов с необходимыми преобразованиями для проверки доказательств, например, ZoKrates и Zinc;

2. реализовывать протокол на уровне смарт-контрактов, решая сопутствующие сложности построения ZKP-системы в терминах EVM;

3. модифицировать блокчейн-узел, использовав одну из криптографических библиотек [8] для создания своей криптографической схемы. При этом предстоит добавить соответствующие модули для криптографических схем и возможность работать с добавленной функциональностью через прекомпилированные смарт-контракты.

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ, РАЗРАБОТКИ И ЭКСПЛУАТАЦИИ СИСТЕМ С ZKP-ПРОТОКОЛАМИ

При проектировании и разработке бизнес-приложений с использованием существующих библиотек и проектов, стоит учитывать несколько особенностей.

УРОВЕНЬ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

В ходе выполнения протокола zk-SNARK участники используют операции (сложение, умножение, спаривание) на эллиптических кривых alt_bn128, которые были добавлены в платформу Ethereum. Аналогичных эллиптических кривых в стандартах ГОСТ нет. Поэтому все проекты, использующие возможности встроенной в Ethereum криптографии (например, наборы инструментов ZoKrates), не смогут быть запущены на платформе Мастерчейн либо в иной системе, реализующей спецификацию Ethereum и построенной на ГОСТ-криптографии.

УРОВЕНЬ РАЗРАБОТКИ СЕРВИСА

Существующие инструменты разработки криптосистем на базе распределенного реестра с возможностью генерации смарт-контрактов, поддерживающих создание и проверку доказательств, подразумевают использование специализированных предметно-ориентированных языков программирования (далее ПОЯ). Такие ПОЯ используют примитивные типы данных. Использование ПОЯ при разработке сложной бизнес-логики требует перевода абстракций бизнес-сущностей и соответствующих структур данных в примитивные типы, что является трудоемкой и сложновыполнимой задачей.

УРОВЕНЬ СОЗДАНИЯ ZKP-АРТЕФАКТОВ

Для работы ZKP-протокола необходимо сгенерировать ключ проверки и ключ доказательства. Генерацией ключей должна заниматься доверенная сторона. В зависимости от конфигурации протокола функции доверенной стороны на себя может брать протокол конфиденциального вычисления [10].

В zk-SNARK при генерации ключей используется секретный параметр («токсичный отход»), который необходимо уничтожить, так как его можно использовать для создания ложных доказательств, которые будут приняты проверяющим. Поэтому для генерации ключей рекомендуется использовать протокол конфиденциальных вычислений, который позволяет не полагаться на честность единственного участника генерирующего артефакты протокола. В этом протоколе достаточно одной доверенной стороны, чтобы убедиться в надежности созданных ключей. Эти ключи передаются сторонам бизнес-процесса, если они не участвовали в генерации ключей. Ключи могут передаваться вместе с экземпляром узла распределенного реестра.

УРОВЕНЬ ВЗАИМОДЕЙСТВИЯ С PP

В распределенном реестре записываются все транзакции, которые изменяют состояние смарт-контракта. При вызове участником метода смарт-контракта, даже если полезная нагрузка зашифрована, можно предположить, что участник является заинтересованной стороной в бизнес-процессе.

ВОЗМОЖНОЕ РЕШЕНИЕ

1. Участники бизнес-процесса отправляют транзакции с зашифрованной полезной нагрузкой (ZKP-доказательство) оператору ИС, оператор ИС выступает техническим автором транзакций в РР. Недостаток решения: оператор ИС является SPOF для участников бизнес-процесса.

2. Шифровать все структуры данных в смарт-контрактах. Участники бизнес-процесса будут отправлять транзакции, авторство будет зафиксировано в распределенном реестре, но конкретный документ, с которым работает участник, будет скрыт. Недостаток решения: реализация всего смарт-контракта на ПОЯ. Вероятно, потребуется дать возможность проверки наличия некоторого значения в списке (подход ZKSM) для регулятора.

ПРОИЗВОДИТЕЛЬНОСТЬ

Выполнять криптографические операции (операции на эллиптических кривых) в смарт-контрактах неэффективно. К коду смарт-контрактов предъявляются жесткие требования, так как они хранятся в распределенном реестре, а их выполнение происходит при проверке корректности транзакций в EVM на каждом узле сети распределенного реестра. При этом криптографические операции являются вычислительно затратными, поэтому увеличивают размер кода смарт-контрактов.

ВОЗМОЖНОЕ РЕШЕНИЕ

модификация узла распределенного реестра, подключение дополнительных криптографических библиотек для создания криптографических схем и использование прекомпилированных контрактов.

ВЫВОДЫ

В результате проведенного исследования можно отметить, что протоколы с нулевым разглашением активно исследуются и разрабатываются участниками рынка распределенных реестров, так как имеют определенный потенциал в повышении конфиденциальности. Однако стоит заметить, что многие проекты носят исключительно академический характер, заметно отсутствие развитых программных продуктов для создания и использования систем с применением доказательств с нулевым разглашением.

В экосистеме Ethereum проводятся активные прикладные исследования неинтерактивных протоколов с нулевым разглашением. В основном исследуются, разрабатываются и используются инструменты сообщества zk-SNARK. Это связано с тем, что исторически zk-SNARK были первыми на рынке, для них проведено большее количество исследований и разработано множество вспомогательных утилит. В рамках нашего исследования нам удалось проверить инструмент ZoKrates (Приложение 2) сообщества zk-SNARK.

Отметим, что у протокола zk-SNARK есть отличительная особенность — «доверенная установка». Мы считаем, что для приватных и консорциумных сетей это не должно стать существенной проблемой. Корректность проведения такой процедуры критически важна для будущей безопасности данных в сети, поэтому ей нужно уделить особое внимание – для этого разработаны отдельные протоколы. В расчете на будущее необходимо рассматривать протоколы, которые не требуют проведения процедуры «доверенной установки», или провести дополнительное исследование протоколов с т.н. обновляемой доверенной установкой, которая выдвигается экспертным сообществом как замена «статической доверенной установке».

zk-STARK — второй активно исследуемый сообществом подход, в котором отсутствует «доверенная установка» и который является квантово устойчивым, но для него существует не так много инструментов. Технология zk-STARK позволяет ускорить процесс обмена информацией и устраняет необходимость предварительной доверительной настройки. Краткость системы гарантирует быстрый процесс верификации и небольшой размер доказательств. При этом в сообществе zk-STARK инструменты по созданию и проверке доказательств в значительной степени отсутствуют.

Другой подход – Bulletproofs – позиционируется как самый простой из неинтерактивных протоколов, при этом он относится к виду ZKRP. Особенность и ограничение данного вида протоколов — проверка вхождения числа в некоторый числовой набор, что накладывает ограничения на использование в бизнес-кейсах, которые могут требовать иных видов проверки знания. Данный вид чаще всего применяется в распределенных реестрах с моделью UTXO. Как и в случае с zk-STARK, инструменты создания и проверки доказательств в значительной степени отсутствуют.

Исходя из вышеизложенного, на текущий момент имеет смысл рассматривать протокол zk-SNARK – из-за богатства его возможностей.

Сложности использования zk-SNARK в российской сертифицированной платформе Мастерчейн связаны в первую очередь с тем, что криптографические наборы, применяемые в популярных наборах инструментов zk-SNARK, не сертифицированы в Российской Федерации, а стандартизированных наборов, эквивалентных используемым в инструментах zk-SNARK эллиптическим кривым, нет.

По мнению экспертов, ключевая особенность неинтерактивного протокола с нулевым разглашением применительно к распределенному реестру заключается в том, что участники бизнес-процесса отправляют в распределенных реестр только результат вычислений, при этом распределенный реестр не содержит в открытом виде непосредственных бизнес-данных, над которыми производятся вычисления. Данная особенность протокола позволяет:

1. повысить пропускную способность из-за проведения всех вычислений вне распределенного реестра, так как отпадает необходимость ждать достижения консенсуса по всем операциям в транзакциях;
2. усилить конфиденциальность информации о сделках, так как данные не хранятся в распределенном реестре в открытом виде.

Перечисленные возможности могут быть полезны во многих бизнес-кейсах, построенных на распределенных реестрах, в том числе на российской платформе Мастерчейн, и позволят подкрепить сильные свойства платформы дополнительными возможностями по повышению конфиденциальности. Особую значимость они приобретают при реализации бизнес-кейсов с большим количеством участников сделок.

Опрошенные эксперты отметили, что финансовый сектор и отрасль в целом, потенциально заинтересованы в реализации и стандартизации протоколов с нулевым разглашением, так как это позволит повысить конфиденциальность операций с одной стороны, и доказуемую корректность операций (соответствие требованиям регулятора) – с другой.

Ассоциация ФинТех приглашает всех заинтересованных участников рынка начать исследования протоколов с нулевым разглашением, а также начать работу по созданию государственных стандартов криптографических наборов для применения в протоколах с нулевым разглашением в финансовом секторе. Для этого необходимо провести совместные тематические исследования специализированных эллиптических кривых и операций над ними при исполнении протокола с нулевым разглашением.

ССЫЛКИ

1. Подводные камни сертификации блокчейн-решений | Открытые системы. СУБД

2. Р 1323565.1.012-2017 «Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации»

3. Приказ ФСБ РФ от 09.02.2005 N 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)» (с изменениями и дополнениями)

4. Zokrates/ZoKrates: A toolbox for zkSNARKs on Мастерчейн

5. Запеченков С.В. Криптографические протоколы и их применение в финансовой и коммерческой деятельности

6. zkSNARKs in a nutshell | Ethereum Foundation Blog

7. What are zk-SNARKs?

8. https://github.com/ZKProofs/ZKProofs.github.io/blob/master/index.md#recent-zero-knowledge-proving-systems

9. Fiat–Shamir heuristic

10. Протокол конфиденциального вычисления — Википедия

ПРИЛОЖЕНИЕ 1. СПОСОБЫ ДОСТИЖЕНИЯ КОНФИДЕНЦИАЛЬНОСТИ В РАСПРЕДЕЛЕННОМ РЕЕСТРЕ

ПРИВАТНЫЕ СМАРТ-КОНТРАКТЫ — высокоуровневая технология, подразумевающая использование смарт-контрактов, внутреннее состояние которых сокрыто от неавторизованных пользователей. Если сокрытие состояния смарт-контрактов совершается на уровне распределенного реестра, используется гомоморфное шифрование. Также сокрытие внутреннего состояния смарт-контракта может совершаться вне реестра, с привязкой контрольной суммы состояния к распределенному реестру для проверки корректности изменений состояния смарт-контракта вне реестра, при этом в ряде случаев используется протокол с нулевым разглашением. Используется для реализации сложной бизнес-логики и защиты от несанкционированного доступа. Пример использования: Quorum ZSL, Sawtooth.

СМЕШИВАНИЕ ТРАНЗАКЦИЙ — техника, использующая особенности базовой учетной модели и модели распределенного реестра для сокрытия путей владения базовой для системы криптовалютой. В зависимости от используемого подхода техника полностью разрывает связь между отправителем и получателем криптовалюты или смешивает ее, усложняя анализ цепочки владения. Пример использования: Tornado.cash, CoinJoin.

КОЛЬЦЕВЫЕ ПОДПИСИ — вариант реализации электронной подписи, когда сообщение подписывается участником некоторой группы подписантов без раскрытия, кем именно. Используется для создания «конфиденциальных транзакций», авторство которых требуется сокрыть. Гипотетически, техника, используемая в конфиденциальных транзакциях на основе кольцевой подписи, поддается анализу и определению автора, если не соблюдаются определенные техники безопасности. Пример использования: Monero Ring Confidential Transaction, Corda.

СЕРВИСЫ ПЕРЕДАЧИ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ — внереестровые приложения, которые создают защищенные каналы связи для передачи сообщений между взаимодействующими участниками. Контроль доступа к защищаемой информации, ее целостность и корректность обеспечивается при помощи распределенного реестра. Хранение информации происходит на узлах отправителя и получателя. Пример использования: СПКС, Besu Orion.

ГОМОМОРФНОЕ ШИФРОВАНИЕ — специальный вид шифрования, позволяющий совершать математические действия над зашифрованным текстом без раскрытия данных. Гомоморфное шифрование реализуется через запись специально подготовленных (зашифрованных) данных в публичных смарт-контрактах, доступ к данным имеют только специально авторизованные участники. Действия над зашифрованными данными осуществляются при помощи обычных транзакций. Используется для сокрытия балансов пользователей, но не скрывает адреса отправителей и получателей. Пример использования: Zether, Monero (аддитивное гомоморфное шифрование).

ПРИЛОЖЕНИЕ 2. ОПИСАНИЕ НАБОРА ИНСТРУМЕНТОВ ZOKRATES

Набор инструментов для использования zk-SNARK в сети Ethereum. ZoKrates позволяет разработчикам создавать доказательства, а затем проверять их при помощи Solidity, что дает возможность проверять доказательства в децентрализованных приложениях (DApp) на основе цепочки блоков Ethereum.
В обновлении EVM Byzantium были внесены некоторые улучшения, такие как добавление специальной эллиптической кривой, скалярное умножение и проверка спаривания эллиптической кривой alt_bn1287, необходимые для выполнения проверок zk-SNARK.

На текущий момент ZoKrates находится в активной разработке и нужно учитывать некоторые особенности применения этого инструментария:

- язык написания специальных программ имеет свои особенности;
- используемая по умолчанию схема доказательств пластична;
- для избежания пластичности разработчик может использовать другие схемы доказательств;
- ZoKrates не поддерживает этап «церемония», когда происходит доверенная установка;
- доверенной установкой занимается проверяющий, потому он должен быть доверенной стороной.

ПРЕДНАЗНАЧЕНИЕ : набор инструментов для разработки криптографических систем на базе распределенного реестра

ZKP ПРОТОКОЛ : zk-SNARKs

ПРЕИМУЩЕСТВА : автоматическая генерация всех артефактов, генерация смарт-контракта на языке Solidity для проверки доказательств

НЕДОСТАТКИ : маловыразительный язык DSL, доверенную установку проводит проверяющий

ИСТОЧНИК : https://zokrates.github.io/

GITHUB : https://github.com/Zokrates/ZoKrates

АВТОРЫ

Илья Дружинин
Аналитик-исследователь

Алексей Цветков
Руководитель разработки платформы Мастерчейн

Алексей Трошичев
Менеджер проекта Мастерчейн

Анатолий Конкин
Руководитель направления «Развитие технологии распределенного реестра»

Петр Каламбет
Системный инженер

ZKP

декабрь 2020

ФИНТЕХ
АССОЦИАЦИЯ