Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра компьютерной безопасности

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
канд. техн. наук, доцент
_____ В. Н. Тренькаев
“ 17 ” января 20 22 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА (ДИПЛОМНАЯ РАБОТА)

АНАЛИЗ БЕЗОПАСНОСТИ ПРОТОКОЛА ОБМЕНА КЛЮЧАМИ DISCV5

по специальности 10.05.01 Компьютерная безопасность,
специализация (профиль) «Анализ безопасности компьютерных систем»

Семенова Екатерина Вадимовна

Руководитель ВКР
канд. тех. наук, доцент
_____ Колегов Д. Н.
“ 17 ” января 20 22 г.

Автор работы
студент группы № 1165
_____ Семенова Е. В.
“ 17 ” января 20 22 г.

Томск–2022

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра компьютерной безопасности

УТВЕРЖДАЮ
Руководитель ООП
канд. техн. наук, доцент
_____ В. Н. Тренькаев
“ 07 ” 10 20 21 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы специалиста обучающемуся
Семеновой Екатерине Вадимовне
Фамилия Имя Отчество обучающегося

по специальности 10.05.01 Компьютерная безопасность, специализация (профиль) «Анализ безопасности компьютерных систем»

1 Тема выпускной квалификационной работы
Анализ безопасности протокола обмена ключами discv5

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис / деканат — 17.01.2022 б) в ГЭК — 28.01.2022

3 Исходные данные к работе:
Объект исследования — протокол discv5
Предмет исследования — протокол обмена ключами в протоколе discv5
Цель исследования — повышение свойств безопасности протокола обмена ключами discv5

Задачи:
— исследовать безопасность протокола обмена ключами discv5;
— разработать дизайн модифицированного протокола обмена ключами discv5 с требуемыми свойствами безопасности;
—разработать спецификацию модифицированного протокола обмена ключами discv5.

Методы исследования:
методы криптографии, анализ протоколов, моделирование угроз, методы компьютерной безопасности

Организация или отрасль, по тематике которой выполняется работа, — децентрализованные системы блокчейн

4 Краткое содержание работы

В работе планируется построить модифицированный вариант протокола обмена ключами discv5 с необходимыми свойствами безопасности, а также привести проект спецификации для нового протокола. В первой главе будут рассмотрены структура протокола обмена ключами discv5 и анализ его безопасности. Во второй главе описана структура разработанного протокола выработки общего сессионного ключа discv5 с требуемыми свойствами безопасности с использование шаблона IK в Noise. Соответственно в этой же главе будет рассмотрен фреймворк Noise, а также дана оценка его шаблоном исходя из свойств предъявляемых к протоколу discv5. В третье главе будет предложен проект спецификации для модифицированного протокола обмена ключами discv5.

Научный руководитель выпускной квалификационной работы, доцент кафедры КБ _____ / Колегов Д. Н./

Задание принял к исполнению, студент группы № 1165 _____ / Семенова Е. В./

АННОТАЦИЯ

Актуальность: в настоящее время большое распространение получили децентрализованные сети. Такие сети обеспечивают существенно больший уровень отказоустойчивости и бескомпромиссную простоту масштабирования. Однако, для такого типа сетей критически важно обеспечивать безопасность протокола взаимодействия участников между собой, а также простоту присоединение новых участников к сети.

Уже существующие стандартные решения, такие как TLS и IPsec, хотя и обеспечивают необходимый уровень безопасности, не всегда могут быть применимы ввиду конкретных технических ограничений. Так, например, в платформе Ethereum используется собственный протокол Node Discovery protocol v5 (discv5), который помимо выполнения заданных требований безопасности (защита транспорта, аутентификация сторон и др.) обеспечивает работу в сетях с негарантированной доставкой сообщений, а также имеет эффективную криптографию, как с точки зрения реализации, так и затрачиваемых вычислительных ресурсов.

Однако, как и любой собственный криптографический протокол, discv5 требует пристального внимания к используемом в нём криптографическим решения, поскольку именно на них базируется надежность всей сети Ethereum.

Объект исследования: протокол discv5.

Предмет исследования: протокол обмена ключами discv5.

Цель работы: повышение свойств безопасности протокола обмена ключами discv5.

Задачи работы:

— исследовать безопасность протокола обмена ключами discv5;

— разработать дизайн модифицированного протокола обмена ключами discv5 с требуемыми свойствами безопасности;

— разработать спецификацию модифицированного протокола обмена ключами discv5.

Результат: модифицированный вариант протокола discv5 со свойствами защиты от чтения назад и симметричной аутентификацией.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ . . . 3

1 Описание протокола discv5 . . . 6
1.1 Основные определения и обозначения . . . 6
1.2 Механизм открытого распределения ключей discv5 . . . 7
1.3 Выработка сессионного ключа в протоколе discv5 . . . 14
1.4 Анализ безопасности протокола discv5 . . . 15
1.4.1 Аутентификация сторон . . . 16
1.4.2 Защита от чтения назад . . . 16

2 Модификации протокола discv5 . . . 18
2.1 Описание фреймворка Noise . . . 18
2.1.1 Описание паттернов протоколов в Noise . . . 20
2.2 Анализ паттернов Noise . . . 22
2.2.1 Описание протокола обмена ключами в discv5 в терминах Noise . . . 22
2.2.2 Анализ протокола для выбора паттерна Noise для обеспечения защиты от чтения назад с сохранением свойств протокола . . . 24
2.2.3 Особенности паттерна IK в Noise . . . 26
2.2.4 Модификации паттерна IK . . . 27
2.3 Модификация механизма открытого распределения ключей . . . 28
2.4 Модификация механизма выработки сессионного ключа . . . 31

3 Проект спецификации модифицированного протокола discv5 . . . 33
3.1 Обработка пакетов . . . 34

ЗАКЛЮЧЕНИЕ . . . 38

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ . . . 39

ВВЕДЕНИЕ

С увеличением числа «умных» устройств, имеющих доступ в сеть, а главное опирающихся на этот доступ для выполнения своих прямых функций, критически важным становится обеспечение устойчивости компьютерных сетей. Однако, классическая клиент-серверная архитектура, использовавшаяся до недавнего времени повсеместно, по своей сути является централизованной сетью с единой точкой отказа — сервером. Ещё одним недостатком централизованных сетей, помимо трудности обеспечения доступности сервера, является сложность их масштабирования. Горизонтальное масштабирование сервера в централизованных сетях требует введения дополнительных сущностей, накладных расходов и не всегда осуществимо на практике.

Логическим развитием централизованных сетей (а также их прямой противоположностью) являются сети децентрализованные. В таких сетях каждый участник может одновременно выступать как сервером, так и клиентом, взаимодействуя с другими узлами сети напрямую, peer-to-peer (P2P). Соответственно, сама структура P2P сетей предполагает отказоустойчивость и легкость масштабирования: все участники сети равноправны и эквивалентны, а потому вышедший из строя узел сети легко может быть заменен любым из доступных в данный момент участников.

Однако, обеспечение работы децентрализованных сетей, в свою очередь, требует также повышенного внимания к построению протокола взаимодействия её участников между собой. Для любой P2P сети критически важно обеспечить безопасное и простое добавление в сеть новых участников, а также защитить уже существующих участников сети, обеспечив, например, защиту от имперсонации — возможности злоумышленнику выдать себя за существующий узел сети.

Так, например, платформа Ethereum 2.0 [1], являющаяся по своей сути P2P сетью для распределенного выполнения произвольных верифицируемых и отказоустойчивых алгоритмов (смарт-контрактов), использует для обеспечения отказоустойчивости сети протокол Node Discovery Protocol v5 [2] (будем далее для краткости обозначать его как discv5). Протокол discv5 обеспечивает возможность подключения новых участников к сети Ethereum, возможность общения любых двух активных участников друг с другом, а также поиск активных участников сети, обладающих заданными свойствами.

Вполне закономерно, что в соответствии с требованиями безопасности, все общение между участниками сети должно быть зашифровано, а сами участники сети аутентифицированы друг перед другом. Для обеспечения данных свойств протокол discv5 описывает процедуру выработки любыми двумя участниками сети общего сессионного ключа — т.н. процедуру распределения ключей. Именно протоколы распределения ключей определяют большинство свойств получаемых на их основе (т.е. с использованием выработанного криптографического ключа) криптографических протоколов.

Кроме того, важно отметить, что использование стандартных криптографических протоколов, обеспечивающих описанный выше уровень безопасности транспорта в сети, таких как (D)TLS или IPsec, в discv5 если не невозможно, то как минимум нецелесообразно. Так, некоторыми из требований, предъявляемых к дизайну discv5 являются возможность работы в сетях с негарантированной доставкой сообщений (например, по протоколу UDP), достаточная легковесность криптографических примитивов, а также стойкость к атакам отказа в обслуживании на основе умножения трафика (amplification DoS attacks).

Несмотря, однако, на отказ от стандартов, дизайн протокола discv5 с точки зрения распределения ключей делает его доказуемо безопасным относительно пассивного злоумышленника [3]. Тем не менее, в нём присутствуют недостатки, которые делают его уязвимым для активного злоумышленника. Например, предлагаемый в discv5 протокол распределения ключей не предполагает защиты от чтения назад, поэтому, если злоумышленнику станет известна секретная часть долговременного ключа получателя, то прошлые сообщения могут быть расшифрованы.

Тем не менее, в современной практической криптографии давно известны и применяются подходы, позволяющие сохраняя относительную простоту и устойчивость протокола обеспечить также и внушительные свойства безопасности протокола. Одним из таких подходов является формирование протокола распределения ключей при помощи Noise Protocol Framework (далее Noise) [4].

Так, фреймворк Noise позволяет используя предопределенные его авторами шаблоны (паттерны) построить протокол распределения ключей, удовлетворяющий заданными свойствами безопасности, и при этом сохраняющий достаточную простоту и верифицируемость. Примером такого подхода может являться описание сетевого протокола nQUIC [5], являющегося полноценной альтернативой QUIC-TLS, и предоставляющего при этом ряд дополнительных криптографических свойств, как например, защита от чтения назад и устойчивость к атакам имперсонации в результате компрометации ключа (Key Compromise Impersonation).

Целью дипломной работы является повышение свойств безопасности протокола обмена ключами discv5. Для достижения данной цели требуется решить следующие **задачи**:

— исследовать безопасность протокола обмена ключами discv5;
— разработать дизайн модифицированного протокола обмена ключами discv5 с требуемыми свойствами безопасности;
— разработать спецификацию модифицированного протокола обмена ключами discv5;

В главе 1 рассматривается структура протокола обмена ключами discv5, проводится анализ его безопасности. В результате описаны уязвимости, обнаруженные в нем: асимметричная аутентификация сторон, отсутствие прямой секретности.

В главе 2 описана структура разработанного протокола выработки общего сессионного ключа discv5 с требуемыми свойствами безопасности с использование паттерна IK в Noise. Соответственно в этой же главе рассматривается фреймворк Noise и дается оценка его паттернам исходя из свойств предъявляемых к протоколу discv5.

В главе 3 предложен проект спецификации для модифицированного протокола обмена ключами discv5.

Итогом дипломной работы служит модифицированный вариант протокола обмена ключами discv5 с необходимыми свойствами безопасности, а именно с симметричной аутентификацией сторон и защитой от чтения назад. Для этого используется наиболее подходящий паттерн IK фреймворка Noise. Также приводится проект спецификации для нового протокола.

1 Описание протокола discv5

В рамках данной главы будет описан протокол обмена ключами discv5. Для начала введем некоторые обозначения и определения для удобства повествования.

1.1 Основные определения и обозначения

Определение 1. Протокол — описание распределенного алгоритма, в процессе выполнения которого два (или более) участника последовательно выполняют определенные действия и обмениваются сообщениями [6].

Определим процесс распределения ключей в рамках протокола.

Определение 2. Распределение ключей — протокол совместной выработки пользователями (общего) ключа секретного путем обмена сообщениями по открытому каналу связи. Протокол должен исключать возможность получения информации о ключе посторонними, а также любым участником до завершения им действий, предусмотренных протоколом [7].

Основные обозначения, используемые в работе, представлены в таблице 1.

Таблица 1 — Основные обозначения

№	Обозначение	Пояснение
1	A, B	классические обозначения участников протоколов [6]: A — Алиса, B — Боб
2	id_A	идентификатор участника A
3	E^A_{pub}, E^A_{priv}	эфемерная пара ключей для участника A
4	S^A_{pub}, S^A_{priv}	долговременная пара ключей для участника A
5	SK_{AB}	общий сессионный ключ между A и B
6	$AES_{GCM}(key, nonce, pt, ad)$	аутентифицированное симметричное шифрование/расшифрование открытых данных pt с помощью AES в режиме GSM с ключом key, случайным одноразовым числом $nonce$ и дополнительными данными аутентификации ad
7	$secp256k1(key, pt)$	алгоритм цифровой подписи на основе эллиптической кривой для данных pt с использованием ключа key
8	N_A	запись об узле A в формате EIP-778
9	$A \to B$	сообщение в рамках распределения ключа, которое будет передаваться от участника A к участнику B
10	$x\|\|y$	конкатенация элементов x и y

Будем называть *инициатором* сторону, которая начинает сеанс протокола, отправляя первое сообщение протокола другой стороне, которая называется *получателем*. В рамках дипломной работы будем считать что Алиса является инициатором, а Боб получателем.

Также дадим несколько определений тесно связанных с процессом получения новых ключей.

Определение 3. *Функция диверсификации KDF* — отображение, позволяющее порождать производные ключи и ключевой материал по корневому ключу и произвольным данным с использованием псевдослучайной функции [8].

Определение 4. *Код аутентификации сообщения на основе хеш-функции HMAC* — механизм обеспечения аутентичности информации на основе симметричного ключа, построенный с использованием хэш-функции [8].

Определение 5. *HKDF (HMAC Key Derivation Function)* — функция диверсификации KDF на основе HMAC. В данном случае HMAC выступает в качестве механизма извлечения псевдослучайного ключа из входных данных.

1.2 Механизм открытого распределения ключей discv5

Протокол discv5 обеспечивает возможность подключения новых участников к сети Ethereum, возможность общения любых двух активных участников друг с другом, а также поиск активных участников сети, обладающих заданными свойствами.

Задача распределения ключей , необходимых для функционирования криптографической системы, является одной из центральных в криптографии, поскольку в большинстве случаев стойкость такой системы основана на недоступности ключей. Поэтому протоколы распределения ключей задают большинство свойств, предъявляемых к криптографическим протоколам. Они должны обеспечивать не только конфиденциальность, но и аутентификацию всех аспектов взаимодействия: ключа, источника, отправителя и получателя, времени сеанса, чтобы гарантировать, что никакой другой участник, кроме заранее определенного второго участника (и, возможно, других доверенных участников), не мог получить доступа ни к одному секретному ключу [6].

Дизайн протокола discv5 с точки зрения распределения ключей делает его доказуемо безопасным относительно пассивного злоумышленника, однако в нём присутствуют недостатки, которые делают его уязвимым для активного злоумышленника. Так, например, в протоколе отсутствует защита от чтения назад, поэтому, если злоумышленнику станет известна секретная часть долговременного ключа получателя, то прошлые сообщения могут быть расшифрованы.

Прежде чем описать подробнее недостатки присущие discv5 необходимо рассмотреть подробнее структуру его механизма открытого распределения ключей,

Каждый участник протокола представлен записью узла в формате EIP-778 [9]. Эта запись содержит в себе данные которые позволяют однозначно определить узел в сети. Для записи об узле A в формате EIP-778 используется обозначение N_A (строка 9, таблица 1).

Структура пакета, передаваемого от участника к участнику, представлена в виде схемы на рисунке 1.

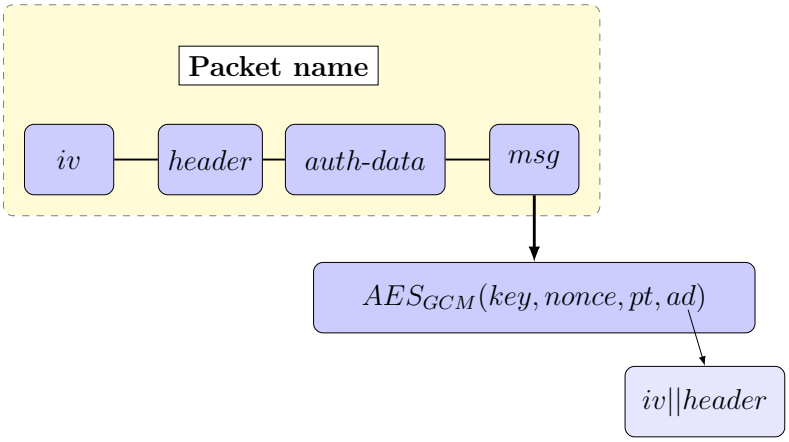


Рисунок 1 — Структура пакета. Здесь iv — вектор инициализации — случайная или псевдослучайная последовательность символов уникальная для каждого пакета; $header$ — заголовок сообщения, структура которого одинакова для всех сообщений; $auth\text{-}data$ — дополнительные данные аутентификации, которые передаются с каждым сообщением, но их структура различна; msg — зашифрованное сообщение (строка 7, таблица 1)

Для аутентификации используется протокол «запрос-ответ» (challenge-response authentication) [10].

1. $A \rightarrow B$ (заявка — witness);
2. $A \leftarrow B$ (запрос — challenge);
3. $A \rightarrow B$ (ответ — response);

После выполнения такого цикла протокола участник B может принять решение о доверии к участнику A. Идея лежащая в основе такого семейства протоколов состоит в том, что одна сторона задает вопрос («запрос»), а другая сторона должна предоставить действительный ответ для аутентификации. Чаще всего такая аутентификация нужна для подтверждения владения ключом без его непосредственной передачи в рамках протокола.

Также аутентификация «запрос-ответ» может помочь решить проблему выработки общего сеансового ключа шифрования. Одним из вариантом получения ключей будет использование секрета — некоторого значения, известного только участнику протокола (например, пароль). Так, значение запроса и секрет могут быть объединены с помощью специальной функции для создания сеансового ключа шифрования. Это осо-

бенно эффективно против атаки «человек посередине», поскольку злоумышленник не сможет получить ключ шифрования, используя только запрос и ответ, но не зная при этом секрета, и, следовательно, не сможет расшифровать поток данных.

При работе с discv5 могут быть рассмотрены две возможные ситуации взаимодействия участников в сети. Предпосылкой к этому является то, что для установления соединения с Бобом инициатору связи необходимо иметь данные о нем: его идентификатор id_B, а также его открытый долговременный ключ — S^B_{pub}. Если Алиса уже не первый раз связывается с Бобом, то необходимости в распределение ключей нет, так как у них уже выработан сессионный ключ SK_{AB} и таким образом первый вариант событий при взаимодействии двух узлов это просто отправить сообщение зашифрованное на общем сессионном ключе.

Второй же вариант учитывает следующие ситуации, во-первых, если Алиса впервые подключается к сети и тогда о нем никаких данных нет и в том числе общих сессионных ключей с другими узлами. Также возможна ситуация, что у узлов уже есть информация друг о друге, но она невалидная или устаревшая, тогда распределение ключей также является необходимым.

Опишем шаги, выполняемые в рамках протокола распределения ключей. Каждый шаг будет себя включать отправку сообщения между участниками, а также действия совершаемые для формирования сообщений для каждого шага.

Шаг 1. Алиса оправляет пакет содержащий в себе сообщение, зашифрованное либо сеансовым ключом, если Алиса уже является участником сети и связывалась с Бобом, либо случайным ключом, если это первое подключение Алисы.

Данные аутентификации данного пакета включают в себя идентификатор Алисы id_A. На рисунке 2 подробно описана структура пакета. В качестве зашифрованного сообщения в пакете от Алисы придет запрос к Бобу о том, что она хочет узнать данные о других узлах сети.

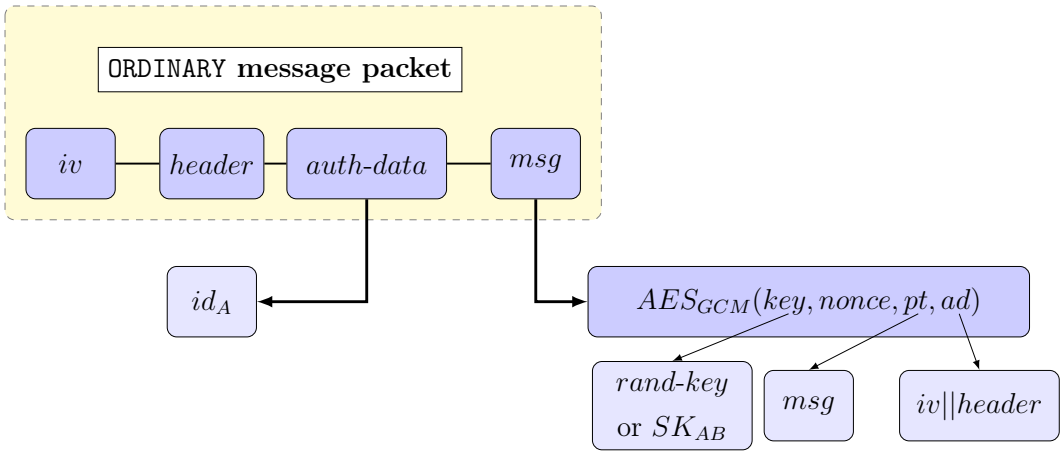


Рисунок 2 — Структура первого сообщения протокола, которое передается от Алисы к Бобу. Здесь id_A — идентификатор Алисы

Шаг 2. Опишем подробнее что происходит на данном шаге, опираясь на краткую схему на рисунке 3. Боб получает сообщение и извлекает идентификатор Алисы id_A из данных аутентификации.

Если Боб имеет сеансовые ключи от предыдущей связи с Алисой, он пытается расшифровать сообщение, находящееся в пакете. Если расшифрование и аутентификация сообщения завершатся успешно, то необходимости в распределении ключей нет, и Боб может просто ответить на запрос.

Если у Боба нет сеансовых ключей или же расшифрование или аутентификация завершились ошибкой, то Боб должен инициировать протокол распределения ключей, ответив пакетом `WHOAREYOU`.

В таком пакете не будет содержаться зашифрованное сообщение. Стоит отметить, что именно данный пакет, а не пакет, полученный на шаге 1 является запросом в схеме «запрос-ответ».

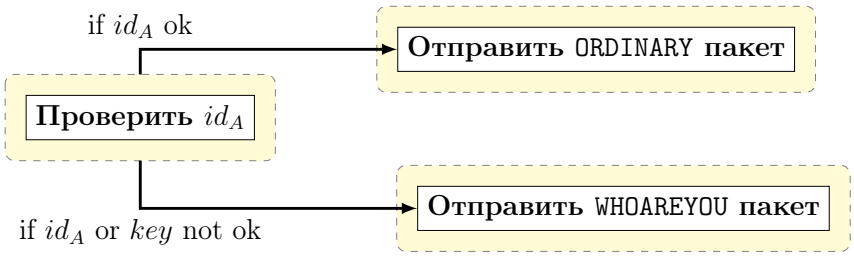


Рисунок 3 — Краткая схема шага 2. Сначала Боб проверяет валидность идентификатора Алисы id_A. Далее в зависимости от ситуации отправляет, либо пакет `ORDINARY` как ответ на запрос, либо же `WHOAREYOU` пакет, который является началом протокола открытого распределения ключей

Структура пакета `WHOAREYOU` представлена на рисунке 4. Данные аутентификации будут содержать в себе следующие две значения.

1. $nonce_{chl}$ — случайное одноразовое число, необходимое для избежания атаки повтора в рамках схемы «запрос-ответ»;
2. *enr-seq* — номер записи об узле инициатора из таблицы Боба. Если же такой записи нет, то *enr-seq* выставляется равным 0.

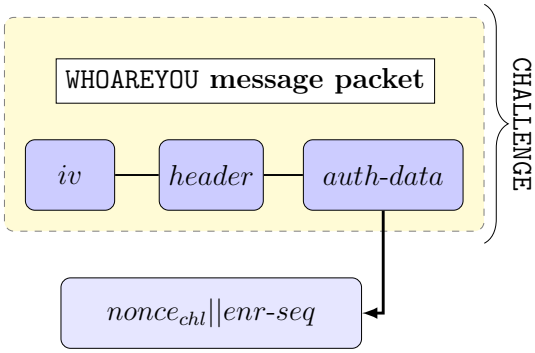


Рисунок 4 — Структура пакета `WHOAREYOU`, который передается от Боба к Алисе. Здесь *enr-seq* — номер записи об узле инициатора из таблицы Боба; $nonce_{chl}$ — случайное одноразовое число

Боб также должен хранить запись об узле Алисы и запрос, которым является весь пакет `WHOAREYOU` в течение короткого времени после отправки Алисе, потому что они понадобятся на шаге 4.

Шаг 3. Данный шаг протокола состоит из нескольких этапов, обозначенных на рисунке 5. Алиса получает сообщение от Боба, что означает, что Боб доступен и готов к протоколу распределения ключей.

Для выработки нового ключа и формирования пакета нужно предпринять следующие действия.

Для начала Алиса генерирует новую пару эфемерных ключей (E^A_{pub}, E^A_{priv}).

Далее на паре E^A_{priv} и S^B_{pub} при помощи операции Диффи–Хеллмана с использованием HKDF (HMAC Key Derivation Function) [11] вырабатывается сессионный ключ, подробнее этот процесс описан в разделе 1.3.

Кроме того данный пакет содержит в себе ответ согласно схеме «запрос-ответ». В рамках протокола она обозначается как идентификационная подпись *id-signature* размера *sig-size*, помимо ответ на запрос она также доказывает, что инициатор контролирует секретный ключ, которым подписана запись об его узле. Подпись также предотвращает атаку повтора для протокола распределения ключей.

Для формирования подписи используется секретная часть долговременного ключа Алисы S^A_{priv}, а само сообщение включает в себя запрос (пакет `WHOAREYOU` отправленный на шаге 2), открытую часть эфемерного ключа узла Алисы E^A_{pub}, идентификатор получателя, а также фиксированное сообщение `const = discovery v5 identity proof`.

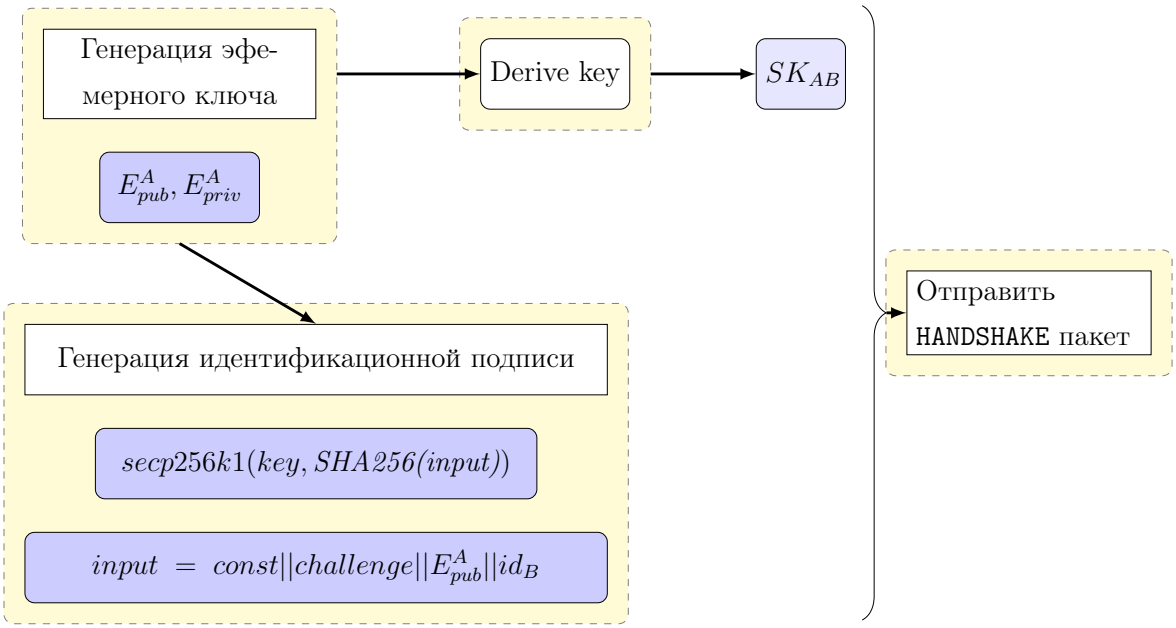


Рисунок 5 — Краткая схема шага 3

Если говорить о составе пакета HANDSHAKE предложенного на рисунке 6, то открытая часть эфемерного ключа размера *eph-size* Алисы включается в пакет в раздел *auth-data*.

Далее на этом же шаге Алиса добавляет описание своего узла N_A в формате EIP-778 [9]. Это поле может быть опущено, если *enr-seq* для WHOAREYOU достаточно недавний, то есть когда он совпадает с текущим порядковым номером узла инициатора.

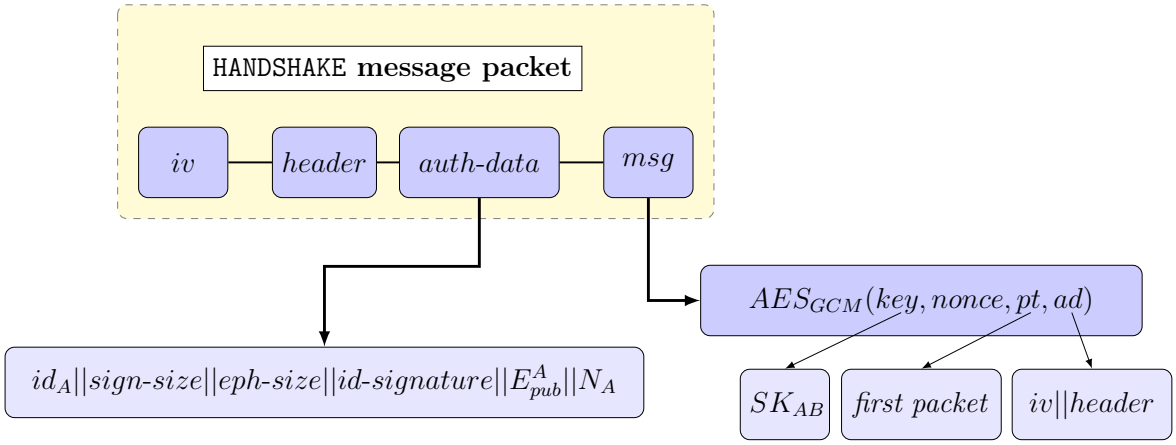


Рисунок 6 — Структура пакета, который передается от Алисы к Бобу. Здесь id_A — идентификатор Алисы; *sign-size* — размет идентифкационной подписи *id-signature*; *eph-size* — размер эфемерного ключа E^A_{pub}; N_A — описание узла Алисы в формате EIP-778

Зашифрованное сообщение в пакете HANDSHAKE в качестве открытого текста включает в себя пакет отправленный на шаге 1. Важной особенностью этого шага является

то, что сообщение от Алисы к Бобу будет зашифровано уже на их новом сессионном ключе.

Шаг 4. После получение пакета `HANDSHAKE` в соответствие с рисунком 7 Боб первую очередь проверяет наличие записи об узле в данном пакете, если пришли новые данные, то прежде чем выполнять последующие действия получатель должен проверить подпись данной записи.

Затем Боб проверяет идентификационную подпись на соответствие открытому ключу из записи об Алисе.

После этого Боб может выполнить вывод ключа, используя свой собственный долговременный закрытый ключ и эфемерный открытый ключ из пакета `HANDSHAKE`. Используя полученные сеансовые ключи, он пытается расшифровать сообщение, содержащееся в пакете.

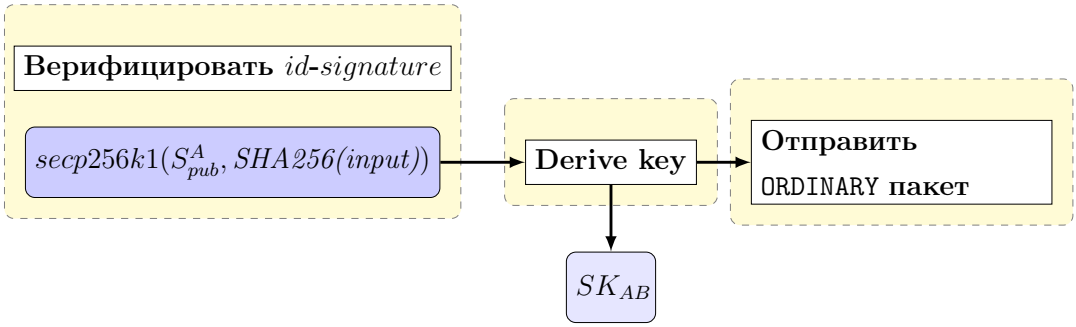


Рисунок 7 — Краткая схема шага 4

Если сообщение может быть расшифровано и аутентифицировано, Боб считает новые ключи сеанса действительными. После этого он отвечает на сообщение пакетом, который содержит информацию о близлежащих узлах. Структура пакета представлена на рисунке 8.

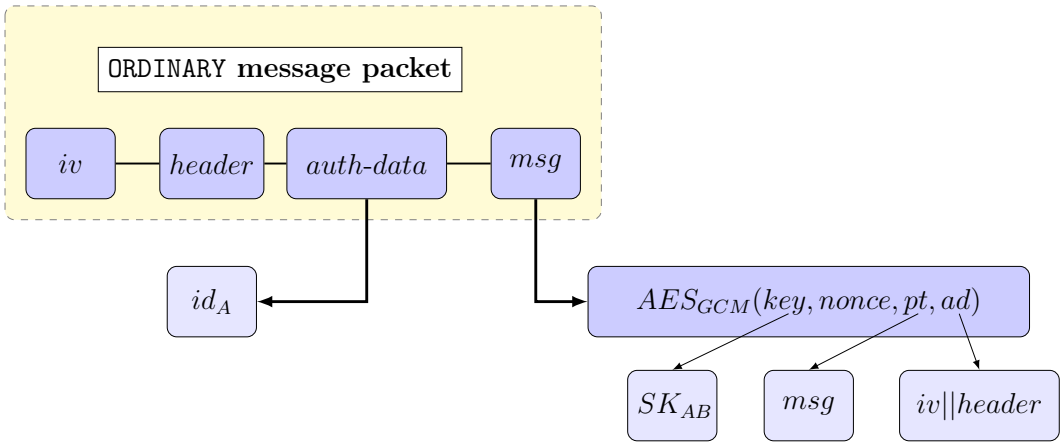


Рисунок 8 — Структура пакета, который передается от Алисы к Бобу. Здесь id_A — идентификатор Алисы

Шаг 5. На последнем шаге Алиса проверяет, что может расшифровать полученное от Боба сообщение. Успех означает, что сессионный ключ между участниками протокола выработан успешно.

1.3 Выработка сессионного ключа в протоколе discv5

Как и говорилось ранее, для получения новых сессионных ключей инициатор генерирует эфемерную пару ключей на эллиптической кривой, которую указал получатель в качестве схемы протокола. В качестве примера предположим, что запись узла Боба использует схему «v4» [2]. В этом случае ключи будут вырабатываться на кривой secp256k1 [12].

В описанном выше протоколе распределения ключей для получения сессионного ключа необходимо выполнить операцию Диффи–Хеллмана на эллиптических кривых. В качестве пар ключей будут использоваться эфемерный ключ инициатора протокола, то есть Алисы, и долговременный ключ получателя, то есть Боба.

На рисунке 9 представлена краткая схема процесса выработки сессионного ключа в протоколе discv5.

Отметим, что для создания сессионного ключа в качестве криптографической соли будет использована строка следующего виде, содержащая в себе фиксированный текст, а также идентификаторы узлов: `«discovery v5 key agreement»` || id_A || id_B.

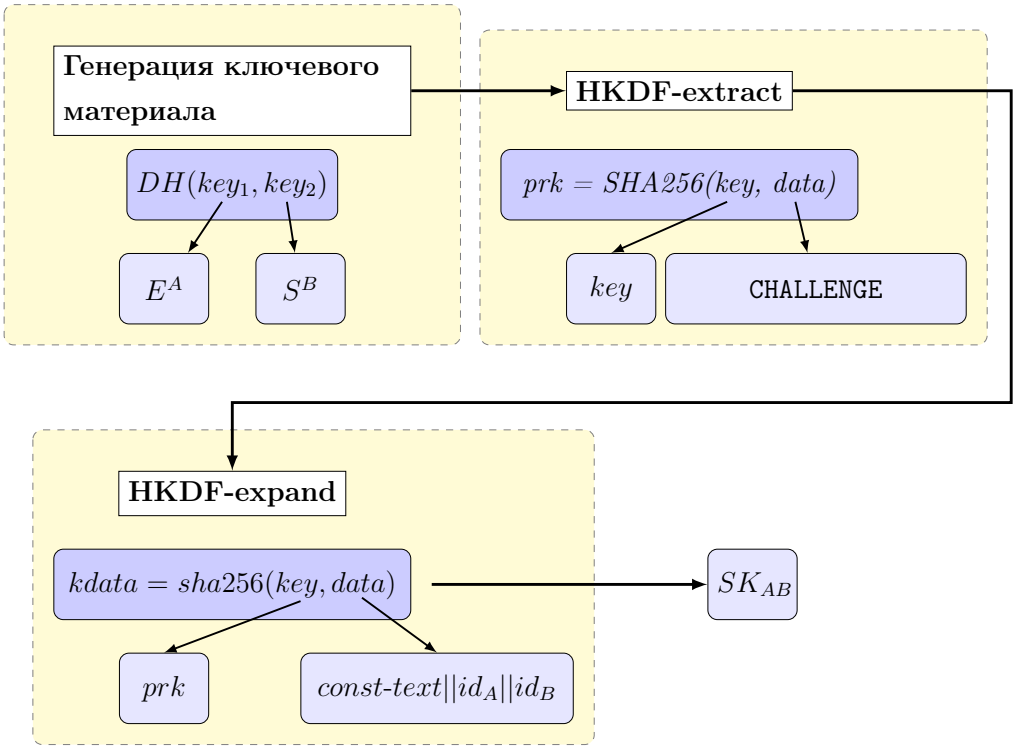


Рисунок 9 — Схема выработки ключа

Процесс выработки нового сессионного ключа включает в себя два этапа. Первым делом из исходного ключевого материала извлекается псевдослучайный ключ фиксированной длины. Важное свойство этой функции заключается в том, что созданный с ее помощью ключ выглядит независимым от источника и является равномерно распределённым (в статистическом и вычислительном смысле). Его цель - «сконцентрировать» энтропию входного ключа в короткий, но криптографически стойкий ключ.

На втором этапе из полученного ранее псевдослучайного ключа создается сессионный ключ заданной длины. В рамках HKDF для этого этапа используется криптографическая соль — случайное (не обязательно секретное) значение, используемое для усложнения определения прообраза функции, создающей псевдослучайный ключ.

Рассмотрим каким образом и с помощью каких данный происходит выработка сессионного ключа в протоколе discv5.

На этапе извлечения псевдослучайного ключа фиксированный длины будет вычислен хеш с использованием ключа полученного после описанной выше операции Диффи–Хеллмана, а в качестве данных используется запрос от Боба, то есть по факту пакет из шага 2 протокола распределения ключей из раздела 1.2.

Далее полученный сессионный ключ разбивается на два ключа одинаковой длины — первая половина будет считаться сессионным ключом инициатора протокола, а вторая часть ключом получателя.

1.4 Анализ безопасности протокола discv5

В первую очередь необходимо подчеркнуть, что в данной работе речь пойдет непосредственно о криптографических аспектах безопасности протокола discv5. Как и любой P2P протокол, discv5 подвержен атакам, характерным для распределенных протоколов. Такие атаки, как правило, направлены на создание и подключение к действующей сети узлов, подконтрольных злоумышленнику, либо же на перехват контроля или имперсонацию под действующих участников сети, то есть на создание как можно большего участка сети под контролем злоумышленника. Классическим примером таких атак являются атака Сивиллы [13] и атака затмения [14].

Однако, доказуемо безопасная защита от такого рода атак невозможна, а практические решения для обеспечения достаточного уровня безопасности опираются на конкретные технические решения в основе защищаемых протоколов. Так, например, классические методы защиты от атаки затмения опираются на построение доверенных наборов IP-адресов, а хорошо изученная контрмера против атаки Сивиллы состоит в том, чтобы сделать создание идентификаторов для новых узлов сети вычислительно дорогостоящим. Очевидно, подобные меры хоть и увеличивают сложность атаки, однако не могут считаться абсолютными.

Анализ же криптографических свойств протоколов, а в особенности протоколов распределения ключей, напротив, область хорошо изученная. Так, идентифицировав конкретные недостатки криптографического протокола, можно гарантировано и доказуемо устранить их, использовав общепринятые проверенные подходы и механизмы безопасности. Как уже было упомянуто ранее, большая часть свойств безопасности любого протокола защиты данных обеспечивается механизмом распределения ключей, используемым в данном протоколе. Потому, остановимся подробнее на особенностях механизма обмена ключами в рамках протокола discv5.

1.4.1 Аутентификация сторон

Аутентификация субъекта (Аутентификация сторон, Peer Entity Authentication) — проверка с подтверждением подлинности одной из сторон наличия или полномочий (посредством представленных доказательств и/или документов) идентичности второй стороны, участвующей в выполнении протокола, а также того, что она действительно принимает участие в выполнении текущего сеанса протокола. Обычно она осуществляется посредством набора данных, который мог быть сгенерирован только вторым участником (например, как отклик на запрос) [6]. Так, в алгоритме Диффи–Хеллмана для обеспечения аутентификации какой-либо стороны необходимо при согласовании ключей использовать секретную часть долговременного ключа соответствующей стороны.

Однако, как известно из спецификации протокола discv5 и главы 1.2, в протоколе обмена ключами, используемом в discv5, участники используют различные механизмы для аутентификации себя в рамках протокола. Так, инициатор общения в discv5 аутентифицирует себя используя подпись части сообщения секретной частью своего долговременного ключа, а получатель – используя свой долговременный ключ при вычислении сессионного ключа с использованием протокола Диффи–Хеллмана.

Такая асимметричность аутентификации сторон, во-первых, выглядит избыточной, ведь при анализе безопасности протокола discv5 авторы предполагают наличие только пассивного злоумышленника, а протокол Диффи–Хеллмана сам по себе предоставляет достаточный уровень безопасности в данных ограничениях. При условии же наличия активного нарушителя, то есть нарушителя, имеющего возможность как модифицировать сообщения в рамках протокола, так и инициировать сеансы протоколы самостоятельно, подобная асимметричность аутентификации делает протокол небезопасным с точки зрения выполнения свойства прямой секретности.

1.4.2 Защита от чтения назад

В криптографии прямая секретность (Forward secrecy), также известная как защита от чтения назад, это особенность конкретных протоколов согласования ключей,

которая дает уверенность в том, что компрометация долговременных ключей не приводит к компрометации старых сеансовых ключей. [6]

Эффективной методикой для достижения прямой секретности при согласовании ключа с помощью алгоритма Диффи–Хеллмана является использование эфемерных ключей. Однако, в протоколе обмена ключами в рамках discv5 общий сессионный ключ получается при использовании только долговременного ключа получателя и эфемерного ключа инициатора, что, соответственно, приводит удовлетворению свойства прямой секретности только для инициатора. Таким образом, если долговременный секретный ключ получателя будет скомпрометирован, то все прошлые сообщения, зашифрованные на ключе, полученном с использованием скомпрометированного, могут быть расшифрованы. Более того, далее пассивный наблюдатель также сможет расшифровывать все будущие сообщения, отправляемые ответчику, налету.

Авторами discv5 утверждается, что достижение симметричной взаимной аутентификации сторон без использования статических ключей в операциях Диффи–Хеллмана нецелесообразно, поскольку это увеличивает количество сообщений, требуемых для выработки сессионного ключа, а значит замедляет функционирование сети.

Тем не менее, использование современных криптографических механизмов, таких как Noise Protocol Framework, позволяет обеспечить взаимную аутентификацию сторон и сохранение свойства прямой секретности без увлечения числа сообщений в протоколе обмена ключами. Далее, в дипломной работе, будет представлена модификация протокола обмена ключами в discv5 с использованием Noise, обладающая указанными свойствами.

2 Модификации протокола discv5

В рамках данной главы будет описана модификация протокола обмена ключами discv5 с обеспечением свойств прямой секретности и взаимной аутентификации. Для этого необходимо будет рассмотреть структуру фреймворка Noise, а также провести анализ его паттернов.

2.1 Описание фреймворка Noise

Noise Protocol Framework [4] (далее фреймворк Noise или просто Noise) — это фреймворк для описания протоколов, основанных на обмене ключами с помощью Диффи–Хеллмана. Ключевой особенностью Noise является то, что каждый ключ полученный в процессе выполнения операций Диффи–Хеллмана влияет на внутреннее состояние протокола, а значит и на сессионные ключи, которые будут выработаны в результате его (протокола) выполнения. Особенное распространение Noise получил в областях, где использование более тяжеловесных протоколов, таких как TLS или IPSec, является избыточным.

Каждый криптографический протокол полученный с помощью Noise описывает процедуру получения общего сессионного ключа парой его участников. Так, участники в процессе выполнения протокола обмениваются между собой сообщениями, последовательно выполняя операции Диффи–Хеллмана и добавляя их результат в ключевой материал для будущего сессионного ключа. Результатом выполнения протокола, полученного на основе Noise, является сессионный ключ, который одинаков у обоих участников протокола и может далее использоваться для симметричного шифрования, а также некоторый набор свойств безопасности, установленных в процессе выполнения протокола, таких например, как аутентификация сторон, сокрытие их идентификационных данных и др.

Под получением на основе фреймворка Noise криптографического протокола обмена ключами понимается следующий процесс. Из предложенных авторами Noise шаблонов будущих протоколов (handshake patterns) выбирается тот, который удовлетворяет требуемым свойствам безопасности и чьи ограничения допустимы для разрабатываемого протокола. Далее выбранный шаблон протокола инстанциируется набором конкретных криптографических примитивов, то есть, для заданной последовательности действий выбирается конкретная ключевая система, позволяющая производить операции Диффи–Хеллмана, а также алгоритмы хеширования и симметричного шифрования. Полученная в результате последовательность шагов вместе с набором конкретных криптографических примитивов и будет являться выработанным протоколом. Особенно ценным в работе с Noise Protocol Framework является тот факт, что он не налагает дополнительных ограничений на выбираемые криптопримитивы, позволяя разработчи-

ку протокола использовать произвольные криптопримитивы, удовлетворяющие требуемым стандартам или вычислительным характеристикам.

Предложенный авторами Noise набор шаблонов (называемых в рамках фреймворка и далее по тексту паттернами) представлен в виде последовательности сообщений, каждое из которых состоит из последовательности токенов — символов, означающих заданную последовательность криптографических операций. Авторы определяют следующие виды токенов, которые могут быть включены в сообщения:

— «`e`»: Участник генерирует новую эфемерную пару ключей и добавляет открытую часть эфемерного ключа в сообщение в незашифрованном виде.

— «`s`»: Участник добавляет открытую часть долговременного ключа в сообщение, шифруя его с использованием промежуточного ключа (в случае, если ключ не был определён на начало операции, шифрование не производится).

— «`ee`», «`es`», «`se`», «`ss`» : Операции Диффи–Хеллмана, произведённые с использованием указанных ключей участников протокола. Первая буква токена определяет тип ключа инициатора (эфемерный `e`, или долговременный `s`), вторая буква — соответствующий тип ключа получателя. Полученное в результате вычисления операции Диффи–Хеллмана используется для обновления промежуточного ключа шифрования.

В качестве примера рассмотрим как в терминах Noise описывается классический протокол Диффи–Хеллмана. Для начала сформируем описание сообщений протокола при помощи описанных выше токенов. Полученный паттерн протокола будет выглядеть следующим образом:

$\longrightarrow$ `e`

$\longleftarrow$ `e`, `ee`

Переводя данное описание с языка Noise получается следующая последовательность действий Алисы (инициатора) и Боба (получателя). Алиса генерирует новую эфемерную пару ключей и отправляет Бобу открытую часть ключа `e`. Боб, получив сообщение Алисы, также генерирует пару эфемерных ключей и отправляет Алисе открытую часть ключевой пары. Далее, Боб выполняет операцию Диффи–Хеллмана (`ee`), используя свой закрытый эфемерный ключ и открытый ключ Алисы, полученный в первом сообщении. Результат проведенной операции Диффи–Хеллмана используется Бобом для формирования промежуточного ключа, который, в виду отсутствия других сообщений, становится сессионным. Алиса же в свою очередь, получив сообщение от Боба, выполняет симметричную операцию Диффи–Хеллмана, используя полученный от Боба открытый и собственный закрытый ключ, после чего использует результат операции для формирования сессионного ключа. Таким образом, в результате выполнения протокола и Алиса, и Боб имеют одинаковый сессионный ключ, который далее может использоваться для построения защищенного канала связи.

Описанный выше паттерн протокола не обладает никакими дополнительными свойствами безопасности, кроме, непосредственно, возможности выработать общий сессионный ключ на основе публичных частей ключевых пар. Ценность же Noise заключается в том, что помимо универсального языка описания подобных шаблонов протоколов, авторами Noise предложен целый ряд готовых шаблонов, для каждого из которых описан набор свойств безопасности, которые они предоставляют. Остановимся подробнее на характере предложенных паттернов протоколов.

2.1.1 Описание паттернов протоколов в Noise

В рамках Noise выделяют два типа паттернов — односторонние и интерактивные. Ключевой особенностью односторонних протоколов является то, что фактически все они состоят из единственного сообщения, отправляемого инициатором протокола. После обработки этого единственного сообщения оба участника протокола вырабатывают общий сессионный ключ и готовы к установлению зашифрованного канала связи.

Очевидно, что такой способ формирования общего ключа невозможен без некоторого дополнительного знания участников друг о друге: например, знания инициатором открытой части долговременного ключа получателя. Данное «дополнительное знание» называется в рамках Noise «пред-сообщением» и обозначается в паттернах следующим образом:

`NAME:`

$\longrightarrow$ `pre − messages`

...

$\longleftarrow$ `messages`

где

— `pre-messages` — пред-сообщения, то есть информация, передающаяся участниками вне протокола по отдельному каналу связи;

— `messages` — сообщения, в виде набора токенов, описывающих операции, производимые в рамках выполнения протокола;

— `NAME` — имя паттерна, которое однозначно определяет набор сообщений в рамках протокола, включая как обычные, так и пред-сообщения.

В Noise авторы определяют следующие односторонние паттерны:

— `N` — долговременный ключ отправителя не передается и не участвует в выработке сессионного ключа;

— `K` — долговременный ключ отправителя известен получателю заранее, то есть передается в пред-сообщениях и участвует в выработке сессионного ключа;

— `X` — долговременный ключ отправителя передается в рамках протокола и участвует в выработке сессионного ключа.

Стоит отметить, что во всех вышеописанных паттернах по-умолчанию предполагается, что инициатору известна публичная часть долговременного ключа получателя, поскольку в противном случае проведение операции Диффи–Хеллмана за 1 шаг протокола невозможно. Таким образом, приведенные выше односторонние паттерны отличаются только наличием и вариантами передачи долговременного ключа инциатора.

Поскольку необходимость безопасной передачи пред-сообщений накладывает существенные требования к формируемому криптографическому протоколу (например, наличие безопасной возможности обмена публичными ключами), то, как правило, больший интерес представляют интерактивные паттерны.

Также, как и в случае с односторонними паттернами, имя паттерна однозначно определяет весь набор сообщений в рамках протокола. Однако, ввиду возможности использовать более одного сообщения, имя интерактивного паттерна состоит уже из двух букв, каждая из которых описывает различные варианты использования долговременных ключей обоих сторон в протоколе.

Первая буква в имени интерактивного паттерна описывает взаимодействие с долговременным ключом инициатора, причем значение букв совпадает с таковым в односторонних паттернах везде, за исключением `I`, которая неопределена для односторонних паттернов:

- `N` — долговременный ключ инициатора не передается и не участвует в выработке сессионного ключа;
- `K` — долговременный ключ инициатора известен получателю заранее, то есть передается в пред-сообщениях, и участвует в выработке сессионного ключа;
- `X` — долговременный ключ инициатора передается в рамках протокола и участвует в выработке сессионного ключа;
- `I` — долговременный ключ инициатора передается в рамках протокола немедленно и участвует в выработке сессионного ключа.

Вторая буква в имени интерактивного паттерна, соответственно, описывает взаимодействие с долговременным ключом получателя:

- `N` — долговременный ключ получателя не передается и не участвует в выработке сессионного ключа;
- `K` — долговременный ключ получателя известен инициатору заранее, то есть передается в пред-сообщениях и участвует в выработке сессионного ключа;
- `X` — долговременный ключ получателя передается в рамках протокола и участвует в выработке сессионного ключа.

Таким образом, существует 12 возможных комбинаций букв в имени интерактивных паттернов: NN, NK, NX и т.д.. Каждая из таких комбинаций называется в рамках Noise «фундаментальным» паттерном, описана авторами фреймворка при помощи токенов и задает конкретный набор свойств безопасности. Именно из данных 12 фунда-

ментальных паттернов будет происходит выбор одного, обеспечивающего необходимые свойства безопасности для нового протокола обмена ключами на базе discv5.

2.2 Анализ паттернов Noise

В рамках данного раздела необходимо проанализировать какой из предлагаемых Noise паттернов протоколов обмена ключами, во-первых, удовлетворяет всем необходимым для использования в рамках discv5 требованиям и обладает необходимыми (не меньшими, чем у оригинального протокола) свойствами безопасности, а, во-вторых, привносит максимальное число дополнительных свойств безопасности без ущерба эффективности (в рамках, например, числа сообщений) оригинальному протоколу.

Первым шагом при выборе паттерна логично сделать рассмотрение оригинального протокола обмена ключами из discv5, дабы в дальнейшем пользоваться при выборе общей терминологией.

2.2.1 Описание протокола обмена ключами в discv5 в терминах Noise

Напомним, что все паттерны, описанные в рамках Noise, однозначно определяются своим названием, которое, в свою очередь, задаёт набор участвующих в протоколе эфемерных и долговременных ключей. Следовательно, для описания оригинального протокола обмена ключами в discv5 в терминах Noise, в первую очередь, необходимо определить характер использования ключей в сообщениях протокола.

Из описания протокола в главе 1 видно, что

— ключ получателя известен инициатору до непосредственно момента обмена сообщениями (то есть, в терминах Noise, передаётся в пред-сообщении);

— для формирования сессионного ключа, который далее будет использоваться для шифрования сообщений протокола discv5, используется результат операции Диффи–Хеллмана между эфемерной ключевой парой со стороны инициатора и долговременной ключевой парой со стороны получателя.

Таким образом, описанный способ применения ключей **для формирования сессионного ключа** однозначно соответствует одностороннему паттерну `N`, то есть, в терминах Noise:

`N:`

$\longrightarrow$ `s`

...

$\longleftarrow$ `e, es`

Так, не смотря на то, что оригинальный протокол обмена ключами описывался в большем количестве шагов (а значит и отправленных сообщений), непосредственно в

формировании сессионного ключа участвует только сообщение с шага 3, что соответствует определению одностороннего паттерна.

Спецификация Noise определяет для паттерна `N` следующие свойства безопасности:

1. **Отсутствие аутентификации инициатора**. Так как долговременный ключ инициатора не используется в операциях Диффи–Хеллмана, данное сообщение протокола не аутентифицирует личность инициатора и может быть отправлено любым участником, включая активного злоумышленника.
2. **Защита от чтения назад сохраняется только при условии конфиденциальности долговременного ключа получателя**. Поскольку при получении сессионного ключа используется только долговременный ключ получателя, компрометация данного ключа может привести к расшифрованию как всех прошлых, так и всех будущих сообщений, зашифрованных на полученном в рамках данного выполнения протокола сессионном ключе.
3. **Отсутствие защиты от атаки повтора**. Отсутствие у принимающей стороны эфемерной ключевой пары, и, соответствие, привносимой ей в сессионный ключ случайности, позволяет активному злоумышленнику повторять данное сообщение, заставляя получателя производить дорогостоящие операции Диффи–Хеллмана.

Необходимо отметить, что 2 из 3 указанных выше недостатков паттерна `N` исправляются в рамках протокола discv5 следующим образом:

1. Инициатор аутентифицирует себя, отправляя в шаге 3 помимо эфемерного ключа также и идентификационную подпись, полученную на своём долговременном ключе.
2. С целью защиты от атаки повтора, в данные для вычисления идентификационной подписи также включается сообщение, отправленное получателем на шаге 2 (делая тем самым сообщение на шаге 3 не самостоятельным, а зависящим от предыдущего состояния протокола discv5).

Последнее же из свойств безопасности, а именно защита от чтения назад, никак не усиливается дополнительными данными в рамках протокола discv5, а потому компрометация долгосрочного ключа получателя приводит к компрометации всех сообщений, зашифрованных на соответствующем сессионном ключе.

Авторами Noise, однако, также описывается целый ряд паттернов, которые предлагают более сильные свойства безопасности, чем паттерн `N`, включая в том числе и защиту от чтения назад. В следующем подразделе будет произведен анализ предлагаемых авторами Noise паттернов и выбор паттерна, удовлетворяющего требованиям протокола discv5, и при этом, предлагающего наилучшие свойства безопасности.

2.2.2 Анализ протокола для выбора паттерна Noise для обеспечения защиты от чтения назад с сохранением свойств протокола

Здесь, опираясь на ранее данное описание протокола discv5, выберем один из описанных в Noise паттернов, дающий наилучшие свойства безопасности. Как уже было упомянуто, авторами Noise приводится 3 односторонних и 12 интерактивных фундаментальных паттернов, плюс различные модификации последних.

Выберем для начала один из предложенных фундаментальных паттернов, опираясь на свойства и требования протокола discv5, а затем рассмотрим предлагаемые авторами Noise модификации выбранного паттерна.

В первую очередь отметим, что ни один из односторонних паттернов Noise, в виду своей природы, не может удовлетворять свойству защиты от чтения назад, которое хотелось бы привнести в протокол discv5. Отсутствие защиты от чтения назад в односторонних паттернах связано с невозможностью принимающей стороны использовать свой эфемерный ключ в процессе получения сессионного ключа, поскольку использование эфемерного ключа получателем потребует ответного сообщения отправляемого инициатору, что противоречит определению одностороннего паттерна. Сформулируем тогда первое утверждение относительно выбора подходящего паттерна:

Замечание 1. Поскольку ни один из односторонних паттернов не обладает свойством защиты от чтения назад, то исключим из рассмотрения все односторонние паттерны Noise и будем фокусироваться на рассмотрении только интерактивных.

Следующей важной характеристикой протокола discv5 является то, что открытая часть долговременного ключа получателя должна быть известна инициатору заранее. При этом, стоит отметить, что данный ключ используется не только для аутентификации получателя при установлении соединения, но и как способ идентификации участников протокола на его логическом уровне, поэтому необходимость знать ключ получателя присутствует вне зависимости от используемого механизма обмена ключами. Исходя из этого факта логично сфокусироваться на паттернах, содержащих долговременный ключ получателя в пред-сообщениях, поскольку «повторная» передача открытой части ключа приведет только к бессмысленному увеличению размера сообщений, а полный отказ от использования долговременного ключа получателя усложнит или вовсе сделает невозможным его аутентификацию в рамках протокола. Таким образом мы можем сократить пространства выбора втрое:

Замечание 2. Ввиду того, что протокол discv5 предполагает знание инициатором открытой части долговременного ключа, в том числе в функциональных (некриптографических) целях, при выборе подходящего паттерна Noise достаточно рассмотреть только те паттерны, в которых долговременный ключ получателя передается заранее

в пред-сообщениях. В соответствие с принятыми в Noise обозначениями, имена таких паттернов будут оканчиваться на `K`: `NK`, `KK`, `XK`, `IK`.

Поскольку одной из функциональных целей протокола discv5 является подключение к сети Ethereum новых участников, то предположение о том, что получатель знает долговременный ключ инициатора является в корне неверным и противоречит самой сути протокола discv5. Следовательно:

Замечание 3. Паттерны Noise, в рамках которых предполагается, что открытая часть долговременного ключа инициатора известна получателю до начала общения, противоречат идее протокола discv5 и должны быть исключены из рассмотрения. Единственным из паттернов, использующих долговременный ключ инициатора в пред-сообщениях среди паттернов, перечисленных в замечании 2, является паттерн `KK`. Исключим его из рассмотрения.

Как уже было упомянуто ранее, открытые части долговременных ключей участников являются также и их идентификаторами в рамках протокола discv5, потому их передача в рамках протокола discv5 необходима. Таким образом, нецелесообразным видится и использования паттернов Noise, в рамках которого долговременный ключ вообще не используется для получения сессионного ключа: использование долговременного ключа инициатора в операциях Диффи–Хеллмана позволяет устранить асимметричность аутентификации сторон, выявленную в discv5 в главе 1 (аутентификация источника происходит за счет механизма подписи, а получателя за счет использование своего долговременного ключа при выработке общего сессионного ключа).

Замечание 4. Для ликвидации асимметричности аутентификации сторон в протоколе discv5 необходимо использовать долговременный ключ инициатора в операциях Диффи–Хеллмана и, следовательно, результирующем сессионном ключе. Соответственно, среди выделенных в замечании 2 паттернов в дополнение ко всем предыдущим замечаниям будет исключен также и паттерн `NK`, поскольку он не предполагает использование долговременного ключа инициатора.

Помимо свойств безопасности к dicsv5 предъявляется также и ряд нефункциональных требований:

— необходимость устойчивости к атакам повтора и отказа в обслуживании требует, чтобы инициатор передал открытую часть своего долговременного ключа уже в первом сообщении, поскольку она будет использована для проверки подписи под данными, обеспечивающими защиту от указанных атак (подробнее см. шаг 4 протокола discv5 в главе 1);

— необходимость эффективной работы в сетях с негарантированной доставкой данных делает крайне нежелательным увеличение количества сообщений в протоколе обмена ключами, поскольку дополнительное сообщение усложнит

обработку сообщений протокола в условиях их неупорядоченного и негарантированного получения.

Необходимость удовлетворять данным нефункциональным требованиям также вносит дополнительные ограничения в возможный набор выбираемых паттернов:

Замечание 5. Среди предложенных в замечании 2 паттернов два предполагают передачу долговременного ключа в момент обмена сообщениями: `XK` и `IK`. При этом, в рамках паттерна `XK` открытая часть долговременного ключа инициатора будет передана не в первом же сообщении протокола, а только после взаимного обмена эфемерный ключами, что кроме всего прочего увеличивает число шагов протокола обмена ключами на 1. Данная особенность паттерна `XK` делает его несовместимым с оригинальным протоколом discv5, но не привносит дополнительно значимых свойств безопасности, потому его использование также кажется нецелесообразным.

Таким образом, исходя из замечаний 1–5, единственный паттерн, удовлетворяющий необходимым свойствам безопасности и при этом не приводящий к существенным изменениям оригинального протокола dicv5, это паттерн `IK`, описывающийся следующим образом:

`IK`:

$\longleftarrow$ `s`

...

$\longrightarrow$ `e`, `es`, `s`, `ss`

$\longleftarrow$ `e`, `ee`, `se`

2.2.3 Особенности паттерна IK в Noise

Важной особенностью Noise является то, что все интерактивные паттерны, в которых используются долговременные ключи обоих сторон, в итоге обеспечивают одинаковые свойства относительно аутентификации инициатора и конфиденциальности дополнительных данных в сообщении. Соответственно, паттерн `IK` не является исключением.

После завершения протокола и получения общего сессионного ключа инициатор устойчив к атакам имперсонации в результате компрометации ключа (key-compromise impersonation), поскольку во время обмена сообщениями участниками производятся операции Диффи–Хеллмана на парах эфемерный-долговременный ключ (`es` или `se`). Таким образом, если соответствующие секретные ключи безопасны, то аутентификация инициатора не может быть подделана даже при условии компрометации его долговременного ключа.

Кроме того, для результирующего протокола на основе паттерна `IK` будет выполняться свойство совершенной секретности. Это утверждение базируется на том факте, что при формировании сессионного ключа шифрования используются одновременно

результаты вычисления операций Диффи–Хеллмана как на паре эфемерных ключей (`ee`), так и на паре эфемерный-долговременный ключ (`es`). Таким образом, при предположении, что эфемерные ключи безопасны, а получатель не является активным злоумышленником, который украл долговременный секретный ключ получателя, полезная нагрузка не может быть расшифрована.

Все интерактивные паттерны в Noise позволяют вместе с сообщениям самого паттерна передавать дополнительные данные, которые могут передаваться как в открытом виде, так и в зашифрованном в зависимости от наличия у участника на момент передачи промежуточного ключа (то есть, была ли на данный момент выполнена хотя бы одна операция Диффи–Хеллмана, поскольку именно их результат является ключевым материалом для ключей шифрования).Как правило, свойства безопасности сообщений до завершения обмена ключами слабее, чем итоговые свойства безопасности, которые будут обеспечены после получения общего сессионного ключа. Так, открытая часть долговременного ключа инициатора в рамках паттерна `IK` передается в первом же сообщении. Однако, несмотря на то, что она зашифрована на промежуточном ключе, полученном в результате проведения операции Диффи–Хеллмана между эфемерным ключом инициатора и статическим ключом получателя (`es`), данное шифрование не обеспечивает защиты от чтения назад (Forward Secrecy): если злоумышленник получит секретный долговременный ключ получателя, он может расшифровать открытый ключ инициатора. Однако, важно отметить, что последнее утверждение относительно отсутствия защиты от чтения назад справедливо только для первого сообщения протокола. Компрометация закрытого долговременного ключа получателя после выполнения протокола (и даже после выполнения первого его шага) уже не приводит к нарушению данного свойства, то есть защита от чтения назад сообщений самого протокола discv5 сохраняется.

2.2.4 Модификации паттерна IK

Помимо 12 фундаментальных интерактивных паттернов авторы Noise предлагают также для каждого из этих паттернов набор модификаций, влияющих на порядок выполнения операций Диффи–Хеллмана с долговременными ключами. Фактически, такие модификации откладывают аутентификацию сторон до «следующего сообщения». Описываемые модификации обозначаются авторами Noise с помощью добавления 1 после одного из символов (или же обоих символов) в имени паттерна, указывая на то что аутентификация инициатора и/или получателя было отложена для следующего сообщения.

Рассмотрим все возможные модификации паттерна `IK`, а именно когда аутентификация отложена для инициатора, для получателя и когда оба пользователя аутентифицируются в следующем сообщении:

I1K:	IK1:	I1K1:
$\longleftarrow$ s	$\longleftarrow$ s	$\longleftarrow$ s
...	...	...
$\longrightarrow$ e, es, s	$\longrightarrow$ e, s	$\longrightarrow$ e, s
$\longleftarrow$ e, ee	$\longleftarrow$ e, ee, se, es	$\longleftarrow$ e, ee, es
$\longrightarrow$ se		$\longrightarrow$ se

Как видно из представленных выше схем, два из трех отложенных паттернов, а именно I1K и IK1 увеличивают число сообщений, необходимых для установления общего сессионного ключа на 1, что, как уже было упомянуто ранее, нецелесообразно с точки зрения необходимости работы в сетях с негарантированной доставкой данных.

Оставшийся паттерн, IK1, не увеличивает числа сообщений в протоколе, однако откладывает вычисление операции Диффи–Хеллмана между эфемерным ключом отправителя и долговременным ключом получателя, а значит и получение промежуточного ключа шифрования. Это приводит к тому, что открытый долговременный ключ отправителя, а также дополнительные данные в рамках протокола discv5, будут переданы в первом сообщении без какой бы то ни было криптографической защиты.

Таким образом, ни одна из модификаций оригинального паттерна IK с отложенной аутентификацией сторон не улучшает свойств результирующего протокола discv5 на базе паттерна Noise. На этом примем паттерн IK как оптимальный со всех описанных точек зрения выбор, и опишем на его базе модификацию механизма открытого распределения ключей в протоколе discv5.

2.3 Модификация механизма открытого распределения ключей

Как может выглядеть адаптированный протокол с паттерном IK. Важно отметить, что первые два шага будут полностью повторять изначальный протокол, можно считать, что они не участвуют в Noise.

Изменения протокола затрагивают только пакеты с шага 3. Именно они будут описаны далее. Но стоит помнить, что пакет с шага 2 должен быть сохранен участниками так как он используется в качестве запроса, а пакет с шага 1 используется как сообщение для шифрования в пакете на шаге 3.

Шаг 3. Алиса получает сообщение от Боба что означает, что Боб доступен и готов к протоколу распределения ключей.

В отличие от оригинального протокола передачи ключей в discv5 сообщение в пакете будет зашифровано не сессионным ключом, а так называемым промежуточным ключом CK_{AB} который будет получен от комбинации значений операций Диффи–Хеллмана с использованием HKDF 1.3.

Рассмотрим подробнее краткую схему, представленную на рисунке 10.

Для начала Алиса генерирует новую пару эфемерных ключей (E^A_{pub}, E^A_{priv}).

Далее выполняется две операции Диффи–Хеллмана: первая на паре ключей E^A_{priv} и S^B_{pub}, а вторая на паре S^A_{priv} и S^B_{pub}.

Кроме того данный пакет содержит в себе ответ согласно схеме «запрос-ответ». В рамках протокола она обозначается как идентификационная подпись *id-signature* размера *sig-size*, помимо ответ на запрос она также доказывает, что инициатор контролирует секретный ключ, которым подписана запись об его узле. Подпись также предотвращает атаку повтора для протокола распределения ключей.

Для подписи используется секретный часть долговременного ключа Алисы, а само сообщение включает в себя запрос (пакет `WHOAREYOU` отправленный на шаге 2), Открытая часть эфемерного ключа узла Алисы, а также идентификатор получателя.

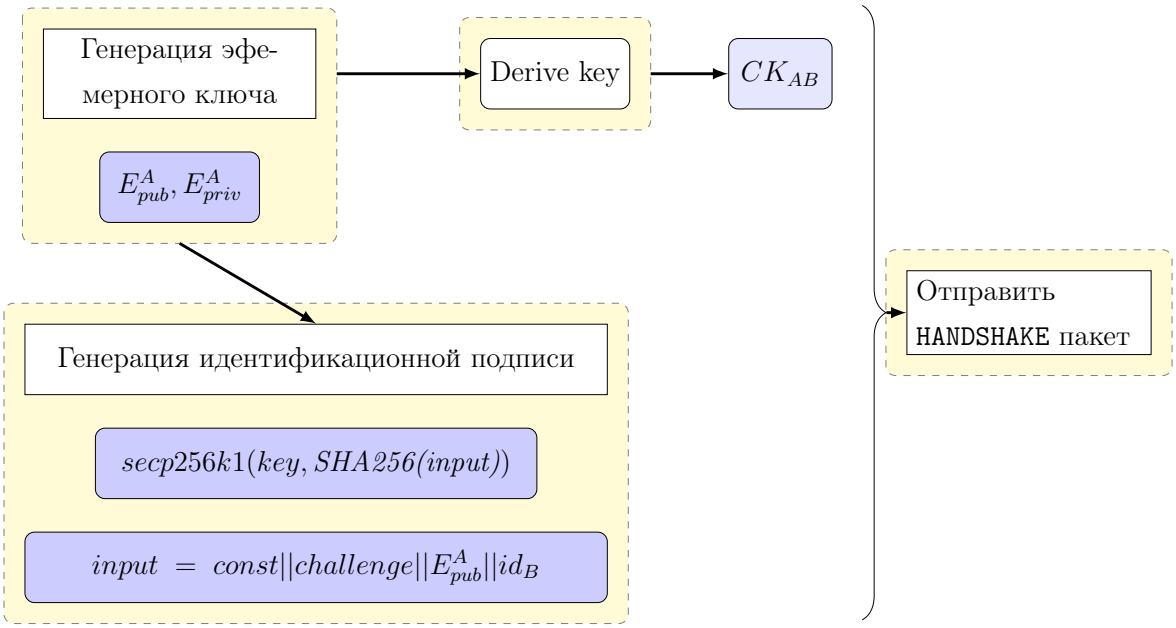


Рисунок 10 — Краткая схема шага 3

Если говорить о составе пакета `HANDSHAKE`, представленного на рисунке 11, то открытая часть эфемерного ключа размера *eph-size* Алисы включается в пакет в раздел *auth-data*.

Далее на этом же шаге Алиса добавляет описание своего узла N_A в формате EIP-778 [9]. Это поле может быть опущено, если *enr-seq* для `WHOAREYOU` достаточно недавний, то есть когда он совпадает с текущим порядковым номером узла инициатора. Если *enr-seq* равен нулю, запись должна быть отправлена. Зашифрованное сообщение в пакете `HANDSHAKE` в качестве открытого текста включает в себя пакет отправленный на шаге 1.

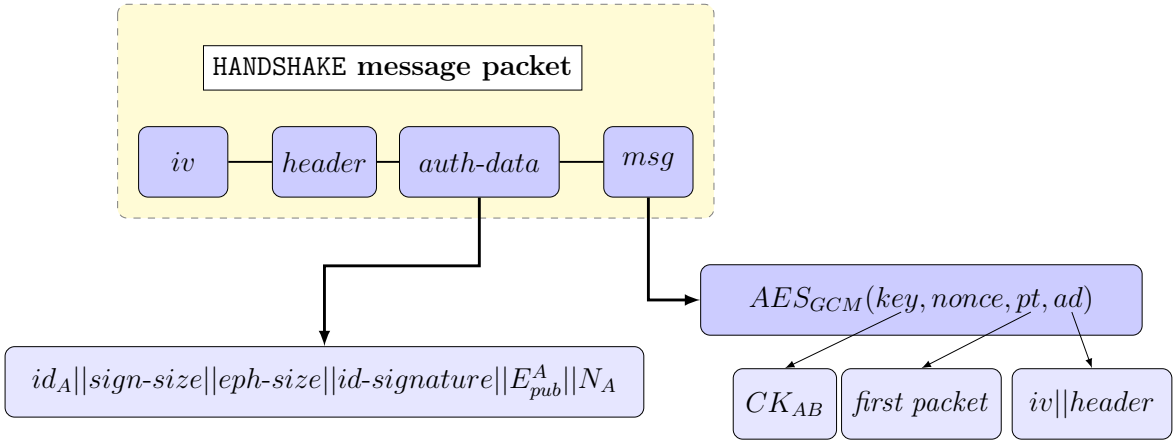


Рисунок 11 — Структура пакета, который передается от Алисы к Бобу. Здесь id_A — идентификатор Алисы; *sign-size* — размет идентифкационной подписи *id-signature*; *eph-size* — размер эфемерного ключа E^A_{pub}; N_A — описание узла Алисы в формате EIP-778; CK_{AB} — промежуточный ключ, полученный после шага 3

Шаг 4. Согласно рисунку 12 после получение пакета `HANDSHAKE` Боб первую очередь проверяет наличие записи об узле в данном пакете, если пришли новые данные, то прежде чем выполнять последующие действия получатель должен проверить подпись данной записи.

Затем Боб проверяет идентификационную подпись на соответствие открытому ключу из записи об Алисе.

После этого Боб может выполнить вывод промежуточного ключа, используя свой собственный долговременный закрытый ключ, а также эфемерный и долговременный открытые ключи из пакета `HANDSHAKE`. Используя полученные сеансовые ключи, он пытается расшифровать сообщение, содержащееся в пакете.

После этого Боб генерирует новую пару эфемерных ключей (E^B_{pub}, E^B_{priv}).

Выполняет поочередно операции Диффи–Хеллмана на парах ключей E^A_{pub} и E^B_{priv}, а затем на ключах S^A_{pub} и E^B_{priv}. Ключ полученный после данных операций будет являться требуемым сессионным ключом.

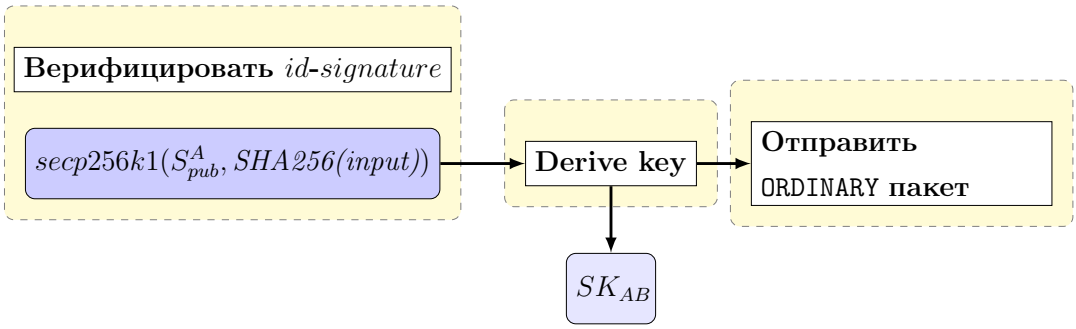


Рисунок 12 — Краткая схема шага 4

После выполнения всех этих операция Боб добавляет к пакету, отправляемому Алисе, зашифрованное на новом сессионном ключе сообщение, а также в данные аутентификации добавляет открытую часть своего эфемерного ключа E^B_{pub} размера *eph-size*. Вид данного пакета представлен на рисунке 13.

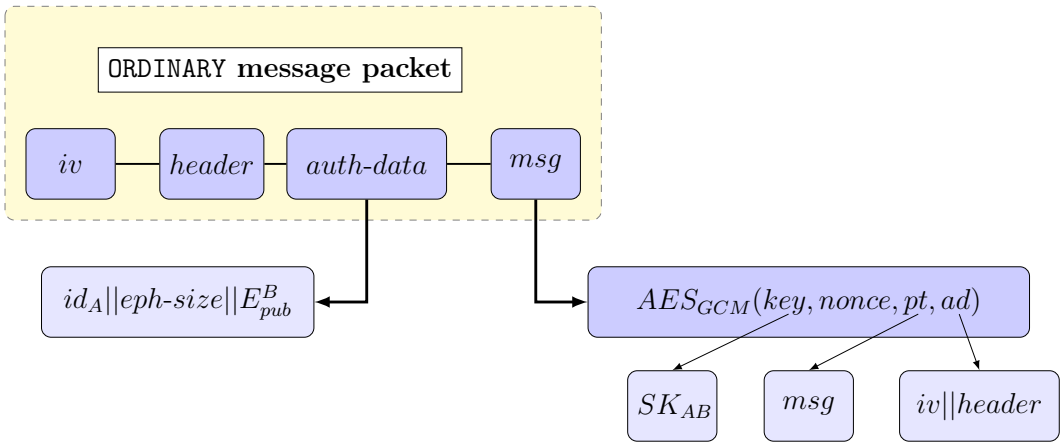


Рисунок 13 — Структура пакета, который передается от Алисы к Бобу. Здесь id_A — идентификатор Алисы; *eph-size* — размер эфемерного ключа E^{pub}

Шаг 5. Получив сообщение от Боба Алиса теперь также может сформировать сессионный ключ, выполнив соответственно операции Диффи–Хеллмана на парах ключей E^A_{priv} и E^B_{pub}, а затем на ключах S^A_{priv} и E^B_{pub}. После этого Алиса проверяет, что может расшифровать полученное сообщение от Боба и это означает, что сессионный ключ между участниками протокола выработан успешно.

2.4 Модификация механизма выработки сессионного ключа

В модифицированном протоколе помимо сессионного ключа также будут вырабатываться промежуточные ключи. Некоторые из них будут использоваться не только как материал для следующего ключа, но и в качестве ключа шифрования для сообщения.

Как уже описывалось выше в разделе 2.2 новый сессионный ключ зависит от различных комбинаций ключей инициатора и отправителя. Это обеспечивает более высокий уровень безопасности по сравнению с изначальным протоколом

Тем не менее процесс формирование промежуточного ключа по факту также осуществляется путем применения HKDF к различному ключевому материалу.

Вычисление этого ключевого материала будет производится путем применения функции DH к различным комбинациям ключей Алисы и Боба, в соответствии с таблицей 2.

Таблица 2 — Этапы вычисления итогового ключа

Этап	Итоговый ключ	Генерация ключевого материала	HKDF-extract $prk = sha256(key, data)$			HKDF-expand	
			key	data	prk	key	data
1	CK_1	$k_1 = DH(E^A, S^B)$	k_1	CHALLENGE	prk_1		const-text
2	CK_2	$k_2 = DH(S^A, S^B)$	k_2	CK_1	prk_2		const-text
3	CK_3	$k_3 = DH(E^A, E^B)$	k_3	CK_2	prk_3		const-text
4	SK_{AB}	$k_4 = DH(S^A, E^B)$	k_4	CK_3	prk_4		const-text

Для первого этапа выработки ключа из исходного ключевого материала извлекается псевдослучайный ключ исходной длины. Важно что получение первого промежуточного ключа CK_1 по факту совпадает с получением сессионного ключа в изначального протоколе обмена ключами discv5. Для остальных же ключей в первом этапе в качестве дополнительных данных для хэширования будет использоваться соответствующий предыдущий ключ (колонка «Итоговый ключ» в таблице 2).

Второй этап HKDF-expand выполняется аналогично описанию в разделе 1.3. В качестве криптографической соли будет использована строка следующего виде, содержащая в себе фиксированный текст, а также идентификаторы узлов: `«discovery v5 key agreement»` || id_A || id_B. Ключ же соотвественно берется как выходное значение из первого этапа.

Отметим, что CK_3 используется для шифрования сообщения на шаге 3, а общий сессионный ключ будет выработан теперь только на шаге 4. Тем не менее это не увеличивает время приема-передачи, поскольку только на шаге 4 будет передаваться ответ на изначальное сообщение, те условно значимый текст. На третьем же шаге сообщение в себе содержит зашифрованный первый пакет и по факту не влияет на время приема передачи.

3 Проект спецификации модифицированного протокола discv5

В разделе 2.3 был описан новый модифицированный протокол с теоретической точки зрения, в этой же главе будет будет предложен проект спецификации протокола, опираясь на который можно уже будет построить референсную реализацию.

Суть изменений может быть описана следующим образом:

1. В отличие от исходного протокола, описанного в разделе 1.2, где на шаге 3 шифрование сообщения происходило уже новым выработанным сессионным ключом, в модифицированном протоколе шифрование на шаге 3 производится промежуточным ключом.
2. Соответственно, необходимо внести изменения в формирование нового сессионного ключа, согласно описанию в разделе 2.4.
3. На шаге 4 при формирование пакета от Боба к Алисе в модифицированном протоколе в данные аутентификации помимо идентификатора инициатора необходимо добавить открытую часть эфемерного ключа Боба E^B_{pub} размера *eph-size*.

Последний пункт изменений требует добавления нового типа пакета.

Напомним, что в исходном протоколе выделяли три вида пакетов: `ORDINARY`, `HANDSHAKE`, `WHOAREYOU`. В протоколе discv5 механизм открытого распределения ключей подразделялся на следующие шаги, определяемые типом пакета:

шаг 1 `ORDINARY`;

шаг 2 `ORDINARY`или `WHOAREYOU`;

шаг 3 `HANDSHAKE`;

шаг 4 `ORDINARY`;

шаг 5 `ORDINARY`.

В модифицированном протоколе изменения, вносимые на четвёртом шаге, требуют введения нового типа пакета. Это обусловлено тем, что необходимо изменить данные аутентификации для того чтобы была возможность передать эфемерный ключ получателя.

При данной модификации на шаге 4 в ответ на `HANDSHAKE`отправляется новый пакет `HANDSHAKERESP`. А модифицированный механизм открытого распределения ключей примет следующий вид:

шаг 1 `ORDINARY`;

шаг 2 `ORDINARY`или `WHOAREYOU`;

шаг 3 `HANDSHAKE`;

шаг 4 `HANDSHAKERESP`;

шаг 5 `ORDINARY`.

В следующем разделе

3.1 Обработка пакетов

Для начала опишем параметры заголовка которые одинаковые для всех пакетов и их вычисление далее не будет описываться.

1. Вектор инициализации (iv) — случайная или псевдослучайная последовательность символов для каждого пакета.
2. Идентификатор протокола (protocol-id). В данном случае discv5. Получатель не должен отвечать на пакеты с несоответствующим идентификатором протокола.
3. Версия протокола — 0x002
4. Флаг — идентификатор который отвечает за типа пакета и также определяет кодировку данных для аутентификации, которая отличается в зависимости от типа пакета. Выделяют следующие типы флагов:
   — 0 — `ORDINARY` пакет.
   — 1 — `WHOAREYOU` пакет.
   — 2 — `HANDSHAKE` пакет.
   — 3 — `HANDSHAKERESP` пакет.
5. nonce – (случайное однократно используемое число) — случайное 96-битное число.
6. auth-size — размер данных аутентификации, следующих за заголовком.

Заголовок пакет шифруется с использованием известного ключа для предотвращения обнаружения и блокирования протокола средствами анализа трафика. Для шифрования используется $AES_{CTR}(key, iv, pt)$ — неаутентифицированное симметричное шифрование/расшифрование открытых данных pt, с помощью AES в режиме CTR с ключом key и вектором инициализации iv.

$$packet = iv||masked\text{-}header||message$$
$$masked\text{-}header = AES_{CTR}(masking\text{-}key, iv, header)$$
$$masking\text{-}key = dest\text{-}id[: 16]$$

Обработка пакетов `ORDINARY`и `WHOAREYOU` соответствует оригинальному протоколу. Также отметим, что данные сообщения, обозначаемые как pt, будут опускаться, если они не являются специальными в рамках протокола открытого распределения ключей

Далее будем рассматривать ситуацию когда Алиса впервые подключается к участнику и общего сессионного ключа у них нет.

Как уже было сказано в главах 1 и 2 при описании механизма открытого распределения ключей discv5 пакет отправленный на шаге 2 является запросом в протоколе аутентификации типа «запрос-ответ». Будем обозначать его следующим образом:

$challenge = iv||header||authdata$

Используемые алгоритмы

$DH\text{-}Generate() = (key_{pub}, key_{priv})$ — функция генерации ключевой пары, соответствующей эллиптической кривой secp256k1 [12].

$ecdh(key_{pub}, key_{priv})$ создает ключевой материал, используемые далее в HKDF с помощью протокола Диффи–Хеллмана на эллиптической кривой. Открытая часть ключа умножается на секретную часть ключ для создания секретного ключа $k = key_{pub} * key_{priv}$. Результат представляется в виде 33 байт и равен $y||k.x$, где y равно 0x02, когда $k.y$ четный или 0x03, когда k.y — нечетный.

Обработка пакета на шаге 3

Формирование полей заголовка пакета описано в таблице 3. Верхняя строка и первый столбец соответствуют количеству байт, которое могут занимать различные поля. Далее описывается формирование остальных полей пакета из таблицы 3, а также вычисления выполняемые инициатором и получателем.

Таблица 3 — Структура пакета, отправляемого на шаге 3

<table>
<tr><td></td><td>0</td><td>4</td><td>8</td><td>12</td><td>16</td></tr>
<tr><td>0</td><td colspan="4">iv</td></tr>
<tr><td>16</td><td colspan="2">protocol-id</td><td>version</td><td>flag</td><td>nonce</td></tr>
<tr><td>32</td><td></td><td>auth-size</td><td colspan="3">src-id</td></tr>
<tr><td>48</td><td colspan="4"></td></tr>
<tr><td>64</td><td colspan="2"></td><td>sig-size</td><td>eph-size</td><td>id-signature</td></tr>
<tr><td>80</td><td colspan="4"></td></tr>
<tr><td>96</td><td colspan="4"></td></tr>
<tr><td>112</td><td colspan="3"></td><td colspan="2">eph-pub-key</td></tr>
<tr><td>128</td><td colspan="4"></td></tr>
<tr><td>144</td><td colspan="3"></td><td colspan="2">record</td></tr>
<tr><td>...</td><td colspan="4"></td></tr>
<tr><td>288</td><td colspan="3"></td><td colspan="2">msg</td></tr>
</table>

Действия инициатора

На шаге 3 инициатор формирует и отправляет получателю сообщение с помощью следующих преобразований:

$$
\begin{aligned}
&auth\text{-}size = len(header) + sig\text{-}size + eph\text{-}size + len(record)\\
&E^A_{pub}, E^A_{priv} = DH\text{-}generate()\\
&packet.eph\text{-}pub\text{-}key = E^A_{pub}\\
&k_1 = ecdh(E^A_{priv}, S^B_{pub})\\
&prk_1 = sha256(k_1, challenge)\\
&CK_1 = sha256(prk_1, const\text{-}text)\\
&k_2 = ecdh(S^A_{priv}, S^B_{pub})
\end{aligned}
$$

$$prk_2 = sha256(k_2, CK_1)$$
$$CK_2 = sha256(prk_2, const\text{-}text)$$
$$packet.msg = AES_{GCM}(CK_2, nonce, first\text{-}packet, iv||header)$$
$$id\text{-}signature\text{-}text = "discovery\ v5\ identity\ proof"$$
$$id\text{-}signature\text{-}input = id\text{-}signature\text{-}text||challenge\text{-}data||eph\text{-}pub\text{-}key||id_B$$
$$id\text{-}signature = sign(sha256(id\text{-}signature\text{-}input))$$

Действия получателя

В свою очередь получатель при получении пакета, выполняет следующие вычисления для проверки корректности и формирования промежуточного ключа.

$$E^A_{pub} = packet.eph\text{-}pub\text{-}keyS^A_{pub} = packet.N_A.pub\text{-}keyk_1 = ecdh(E^A_{pub}, S^B_{priv})$$
$$prk_1 = sha256(k_1, challenge)$$
$$CK_1 = sha256(prk_1, const\text{-}text)$$
$$k_2 = ecdh(S^A_{pub}, S^B_{priv})$$
$$prk_2 = sha256(k_2, CK_1)$$
$$CK_2 = sha256(prk_2, const\text{-}text)$$
$$id\text{-}signature\text{-}text = "discovery\text{-}v5\text{-}identity\text{-}proof"$$
$$id\text{-}signature\text{-}input = id\text{-}signature\text{-}text||challenge\text{-}data||eph\text{-}pub\text{-}key||id_B$$
$$packet.id\text{-}signature\ ? = sign(sha256(id\text{-}signature\text{-}input))$$

В итоге у обоих сторон в конце обработки сообщения на шаге три будет выработан одинаковый промежуточный ключ CK_2

Обработка пакета на шаге 4

Формирование полей заголовка пакета описано выше. Далее описывается формирование остальных полей пакетов из таблицы 4, а также вычисления выполняемые инициатором.

Таблица 4 — Структура пакета, отправляемого на шаге 4

<table>
<tr><td></td><td>0</td><td>4</td><td>8</td><td>12</td><td>16</td></tr>
<tr><td>0</td><td colspan="5">iv</td></tr>
<tr><td>16</td><td colspan="2">protocol-id</td><td>version</td><td>flag</td><td>nonce</td></tr>
<tr><td>32</td><td colspan="2"></td><td>auth-size</td><td colspan="2">src-id</td></tr>
<tr><td>48</td><td colspan="5"></td></tr>
<tr><td>64</td><td colspan="3"></td><td>eph-size</td><td>eph-pub key</td></tr>
<tr><td>80</td><td colspan="5"></td></tr>
<tr><td>96</td><td colspan="4"></td><td>msg</td></tr>
</table>

Действия получателя

На шаге 4 получатель отправляет инициатору ответный пакет следующего содержания.

$$
\begin{aligned}
&E^B_{pub}, E^B_{priv} = DH\text{-}generate() \\
&packet.eph - pub - key = E^B_{pub} \\
&k_3 = ecdh(E^A_{pub}, E^B_{priv}) \\
&prk_3 = sha256(k_3, CK_2) \\
&CK_3 = sha256(prk_3, const - text) \\
&k_4 = ecdh(E^A_{pub}, E^B_{priv}) \\
&prk_4 = sha256(k_4, CK_3) \\
&SK_{AB} = sha256(prk_4, const - text) \\
&packet.msg = AES_{GCM}(SK_AB, nonce, pt, iv||header)
\end{aligned}
$$

Действия инициатора

Далее инициатор протокола при получении пакета, выполняет следующие вычисления.

$$
\begin{aligned}
&k_3 = ecdh(E^B_{pub}, E^A_{priv}) \\
&prk_3 = sha256(k_3, CK_2) \\
&CK_3 = sha256(prk_3, const - text) \\
&k_4 = ecdh(E^B_{pub}, S^A_{priv}) \\
&prk_4 = sha256(k_4, CK_3) \\
&SK_{AB} = sha256(prk_4, const - text)
\end{aligned}
$$

Таким образом, у обоих участников протокола в конце обработки сообщения на шаге 4 будет выработан одинаковый сессионный ключ SK_{AB}.

На этом протокол обмена ключам закончен, участники могут приступать к защищенному обмену сообщениями.

ЗАКЛЮЧЕНИЕ

В рамках данной работы было выполнено описание и анализ безопасности протокола обмена ключами используемого в Ethereum Node Discovery Protocol v5. В результате анализа, в рассматриваемом протоколе был выявлен ряд недостатков, приводящих к потенциальным криптографическим уязвимостям.

Так, отсутствие защиты от чтения назад позволяет злоумышленнику, завладевшему секретной частью долговременного ключа получателя, расшифровать все прошлые сообщения, полученные с использованием скомпрометированного ключа. Решением данной проблемы может быть взаимная аутентификация сторон при распределении ключей.

С учетом предложенных исправлений недостатков механизма обмена ключами discv5 в рамках дипломной работы было произведено проектирование и разработана спецификация протокола выработки общего сессионного ключа с использованием Noise Protocol Framework.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. *Buterin V.* A Next-Generation Smart Contract and Decentralized Application Platform. — 2013.
2. Node Discovery Protocol v5. — URL: `https://github.com/ethereum/devp2p/blob/master/discv5/discv5.md` (дата обращения: 17.01.2022).
3. Node Discovery Protocol v5. Rationale. — URL: `https://github.com/ethereum/devp2p/blob/master/discv5/discv5-rationale.md` (дата обращения: 17.01.2022).
4. *Perrin T.* The Noise Protocol Framework. — URL: `https://noiseprotocol.org/noise.pdf` (дата обращения: 17.01.2022).
5. NQUIC: Noise-Based QUIC Packet Protection / M. Hall-Andersen [и др.] // Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC. — Heraklion, Greece : Association for Computing Machinery, 2018. — с. 22—28. — (EPIQ'18). — ISBN 9781450360821. — DOI: `10.1145/3284850.3284854`. — URL: `https://doi.org/10.1145/3284850.3284854` (дата обращения: 17.01.2022).
6. *Черемушкин А. В.* Криптографические протоколы. Основные свойства и уязвимости: учебное пособие. — 2009.
7. *Б.А П.* Словарь криптографических терминов //. — Изд. "МЦНМО", Москва, 2006. — ISBN 5-04057-257-X.
8. *ТК 26 «Криптографическая защита информации» Т. комитет по стандартизации.* Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. — URL: `https://tc26.ru/standard/rs/%D0%A0%2050.1.113-2016.pdf` (дата обращения: 17.01.2022).
9. Ethereum Node Records. — URL: `https://github.com/ethereum/devp2p/blob/master/enr.md` (дата обращения: 17.01.2022).
10. *Tilborg H.* Encyclopedia of cryptography and security. — New York : Springer, 2011. — ISBN 978-1-4419-5906-5.
11. *Krawczyk D. H.*, *Eronen P.* HMAC-based Extract-and-Expand Key Derivation Function (HKDF) //. — RFC Editor, 05.2010. — (Request for Comments ; 5869). — DOI: `10.17487/RFC5869`. — URL: `https://rfc-editor.org/rfc/rfc5869.txt`.
12. *Wei Bi Xiaoyun Jia M. Z.* A Secure Multiple Elliptic Curves Digital Signature Algorithm for Blockchain. — URL: `https://arxiv.org/pdf/1808.02988.pdf` (дата обращения: 17.01.2022).

13. *Douceur J. (* The Sybil Attack // Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS). — 01.2002. — URL: https://www.microsoft.com/en-us/research/publication/the-sybil-attack/ (дата обращения: 17.01.2022).

14. Eclipse Attacks on Bitcoin's Peer-to-Peer Network / E. Heilman [и др.] // 24th USENIX Security Symposium (USENIX Security 15). — Washington, D.C. : USENIX Association, 08.2015. — с. 129—144. — ISBN 978-1-939133-11-3. — URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman (дата обращения: 17.01.2022).

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Семенова Катя
Проверяющий: Семенова Катя (seemenkina@gmail.com / ID: 6754389)

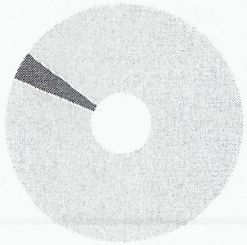
Отчет предоставлен сервисом «Антиплагиат» - users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 12
Начало загрузки: 23.01.2022 14:45:28
Длительность загрузки: 00:00:01
Имя исходного файла: semenova-text.pdf
Название документа: semenova-text
Размер текста: 77 кБ
Символов в тексте: 80955
Слов в тексте: 10094
Число предложений: 371

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 23.01.2022 14:45:30
Длительность проверки: 00:00:04
Комментарии: не указано
Модули поиска: Интернет Free



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
3,64%	0%	0%	96,36%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,45%	239_109_216_0_0.600_63632183 ko`chirish http://library.tuit.uz	06 Дек 2020	Интернет Free
[02]	0,73%	Криптографические протоколы: основные свойства и уязвимости https://yandex.ru	16 Окт 2019	Интернет Free
[03]	0%	176_207_1_0_0.600_64624527 Криптографические протоколы: основные свойства и уязвимости http://lib.tsu.ru	05 Дек 2020	Интернет Free

Еще источников: 7
Еще заимствований: 1,46%