Internet Engineering Task Force (IETF)
Request for Comments: 8061
Category: Experimental
ISSN: 2070-1721

D. Farinacci
lispers.net
B. Weis
Cisco Systems

February 2017

Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality

Конфиденциальность плоскости данных протокола LISP

Аннотация

Этот документ описывает шифрование трафика, инкапсулируемого с использованием протокола LISP[1]. Описан обмен ключами с использованием имеющихся механизмов плоскости управления LISP, а также способы защиты плоскости данных LISP от постороннего наблюдения.

Статус документа

Этот документ не является спецификацией Track и публикуется для проверки, экспериментальной реализации и оценки.

Документ определяет экспериментальный протокол для сообщества Internet. Документ является результатом работы IETF[2] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[3]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке http://www.rfc-editor.org/info/rfc8061.

Оглавление

1. Введение

Этот документ описывает механизм шифрования трафика с инкапсуляцией LISP. Схема описывает способ обмена ключами с помощью имеющихся механизмов плоскости данных LISP, а также способы защиту плосколсти данных от постороннего наблюдения.

Протокол LISP [RFC6830] определяет набор функций, позволяющих маршрутизаторам обмениваться информацией, применяемой для сопоставления немаршрутизируемых идентификаторов конечных точек (Endpoint Identifier или EID) с маршрутизируемыми локаторами (Routing Locator или RLOC). Входные маршрутизаторы туннелей LISP (Ingress Tunnel Router или ITR) и входные маршрутизаторы-посредники (Proxy Ingress Tunnel Router PITR) инкапсулируют пакеты для выходных маршрутизаторов туннелей (Egress Tunnel Router или ETR) и повторно инкапсулирующих маршрутизаторов

[1] Locator/ID Separation Protocol - протокол разделения идентификаторов и локаторов.
[2] Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[3] Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

туннелей (Re-encapsulating Tunnel Router или RTR). Пакеты, приходящие на ITR или PITR, могут быть нешифрованными, что обозначает отсутствие защиты и приватности. Если хост-источник шифрует поток данных, протоколу LISP не требуется шифровать инкапсулируемые пакеты. Однако при передаче хостами нешифрованных (plaintext) пакетов эта схема позволяет зашифровать пользовательские данные для сохранения приватности на пути между инкапсулятором (ITR или PITR) и декапсулятором (ETR или RTR). Шифрованные пакеты являются однонаправленными, а для обратного трафика применяются такие же процедуры, но с другими ключами тех же или иных (при асимметричном пути) xTRs.

Этот документ предъявляет к пространству решений общие требования из [RFC6973], а также указанное ниже.

- Не требуется отдельная инфраструктура открытых ключей (Public Key Infrastructure или PKI), выходящая за рамки архитектуры плоскости управления LISP.
- Обмен ключами **должен** выполняться за один интервал кругового обхода (round-trip), т. е. в обмене могут участвовать лишь 2 пакета.
- Симметричное шифрование в плоскости данных LISP для ускорения.
- По возможности отказ от сторонних привязок доверия.
- Смена ключей при компрометации секретного ключа.
- Поддержка аутентифицированного шифрования с проверкой целостности пакетов.
- Поддержка нескольких шифров с возможностью простого добавления новых алгоритмов.

Выполнение указанных выше требований обеспечивает ряд преимуществ.

- Отказ от PKI сокращает эксплуатационные издержки на поддержание защищённой сети. Управление ключами распределено и не зависит от другой инфраструктуры.
- Доставка пакетов оптимизирована благодаря меньшему числу заголовков. Потери пакетов сокращаются благодаря более эффективному обмену ключами.
- Обеспечивается аутентификация и приватность на основе одного механизма, что сокращает издержки на каждый пакет, повышаю эффективность использования ресурсов.

2. Уровни требований

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

3. Определения терминов

AEAD
: Authenticated Encryption with Associated Data [RFC5116] - аутентифицированное шифрование со связанными данными.

ICV
: Integrity Check Value - значение для проверки целостности.

LCAF
: LISP Canonical Address Format [RFC8060] - канонический формат адреса LISP.

xTR
: Общее обозначение для маршрутизаторов ITR, ETR, RTR, PxTR

4. Обзор

Предложенный в этом документе подход **не** полагается на систему отображения LISP (или иную инфраструктуру ключей) для хранения секретных ключей, обеспечивая более простой и защищённый механизм. Секретный ключ соглауется между ITR и ETR в сообщениях Map-Request и Map-Reply. Когда ITR нужно получить RLOC от ETR, он получит ключевой материал для расчёта общего с ETR секретного ключа.

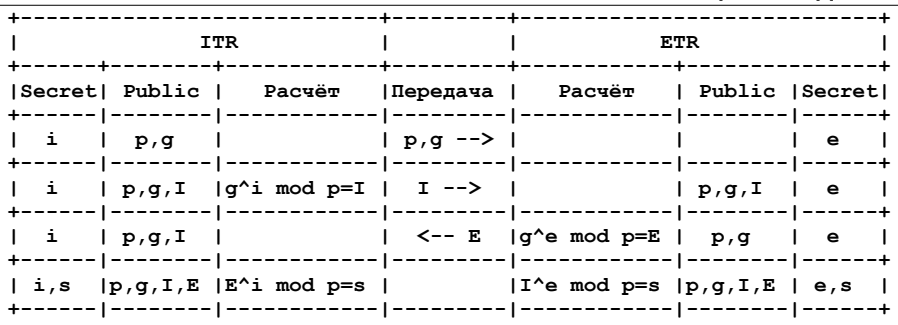
ITR может рассчитать 3 общих секрета с ETR, для которого ITR выполняет инкапсуляцию. Когда ITR шифрует пакет перед инкапсуляцией, он указывает используемый ключ, чтобы маршрутизатор ETR, знающий этот ключ, мог расшифровать пакет после декапсуляции. С помощью идентификаторов ключей в заголовках LISP можно легко менять ключи.

Описанное в этом документе управление ключами является однонаправленным - от ITR (инкапсулятор) к ETR (декапсулятор).

5. Обмен ключами Diffie-Hellman

LISP использует процесс обмена ключами и расчёты Diffie-Hellman [RFC2631] для вычисления общего секрета. Параметры Diffie-Hellman передаются кодами Cipher Suite в сообщениях Map-Request и Map-Reply. Ниже приведено краткое описание процесса.

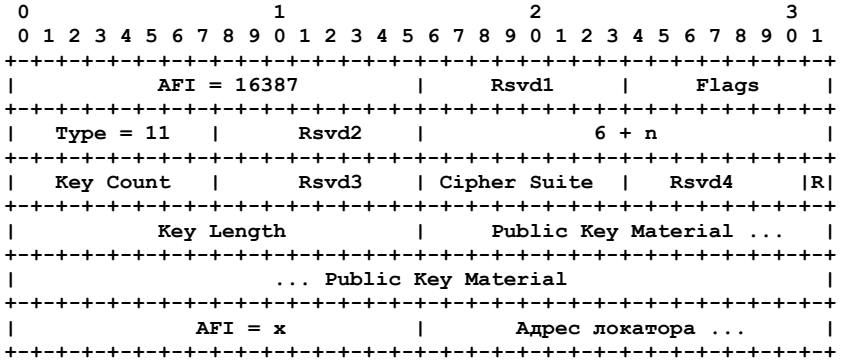
Параметры p и g должны иметь одинаковые значения на маршрутизаторах ITR и ETR. Маршрутизатор ITR рассчитывает открытый ключ I и передаёт его в пакете Map-Request. ETR при получении Map-Request использует параметры p и g для расчёта своего открытого ключа E и передаёт этот клю в сообщении Map-Reply. В этот момент у ETR достаточно сведений для расчёта секретного ключа s с использованием I в качестве основания и своего секретного ключа e в качестве показателя степени. Когда ITR получает Map-Reply, он использует открытый ключ E и свой секретный ключ i для расчёта того же общего секрета, который даст расчёт ETR. Значение p служит модулем при создании общего секрета s (6. Кодирование и передача ключевого материала).



Обмен открытыми ключами для создания общего секретного ключа.

6. Кодирование и передача ключевого материала

Ключевой материал Diffie-Hellman (DH) передаётся в сообщениях Map-Request и Map-Reply, параметры Diffie-Hellman кодируются в LISP Security Key LCAF Type [RFC8060].



Поле Cipher Suite указывает функции обмена ключами DH, шифрования и хэширования.

В поле Key Count указывается число полей {Key-Length, Key-Material}, включённых в LCAF. Максимальное число таких полей - 3 и они указываются как key-id 1, key-id 2 и key-id 3. Бит R не используется в этом варианте применения Security Key LCAF Type и зарезервирован для [LISP-DDT]. Поэтому бит R **следует** сбрасывать (0) при отправке, а при получении он **должен** игнорироваться.

Шифр 0
Резерв

Шифр 1 (LISP_2048MODP_AES128_CBC_SHA256):
Группа DH: 2048 битов MODP [RFC3526]
Шифрование: AES со 128-битовыми ключами в режиме CBC [AES-CBC]
Целостность: интеграция с AEAD_AES_128_CBC_HMAC_SHA_256 [AES-CBC]
Размер IV: 16 байтов
KDF: HMAC-SHA-256

Шифр 2 (LISP_EC25519_AES128_CBC_SHA256):
Группа DH: 256 битов Elliptic-Curve 25519 [CURVE25519]
Шифрование: AES со 128-битовыми ключами в режиме CBC [AES-CBC]
Целостность: интеграция с AEAD_AES_128_CBC_HMAC_SHA_256 [AES-CBC]
Размер IV: 16 байтов
KDF: HMAC-SHA-256

Шифр 3 (LISP_2048MODP_AES128_GCM):
Группа DH: 2048 битов MODP [RFC3526]
Шифрование: AES со 128-битовыми ключами в режиме GCM [RFC5116]
Целостность: интеграция с AEAD_AES_128_GCM [RFC5116]
Размер IV: 12 байтов
KDF: HMAC-SHA-256

Шифр 4 (LISP_3072MODP_AES128_GCM):
Группа DH: 3072 бита MODP [RFC3526]
Шифрование: AES со 128-битовыми ключами в режиме GCM [RFC5116]
Целостность: интеграция с AEAD_AES_128_GCM [RFC5116]
Размер IV: 12 байтов
KDF: HMAC-SHA-256

Шифр 5 (LISP_256_EC25519_AES128_GCM):
Группа DH: 256 битов Elliptic-Curve 25519 [CURVE25519]
Шифрование: со 128-битовыми ключами в режиме GCM [RFC5116]
Целостность: интеграция с AEAD_AES_128_GCM [RFC5116]
Размер IV: 12 байтов
KDF: HMAC-SHA-256

Шифр 6 (LISP_256_EC25519_CHACHA20_POLY1305):
Группа DH: 256 битов Elliptic-Curve 25519 [CURVE25519]
Шифрование: Chacha20-Poly1305 [CHACHA-POLY] [RFC7539]
Целостность: интеграция с AEAD_CHACHA20_POLY1305 [CHACHA-POLY]
Размер IV: 8 байтов
KDF: HMAC-SHA-256

Поле Public Key Material содержит открытый ключ, созданный одним из указанных выше шифров (Cipher Suite). Размер ключа в октетах указывает поле Key Length.

Когда ITR, PITR или RTR передаёт Map-Request, он кодирует свой RLOC в формате Security Key LCAF Type в поле ITR-RLOC. Когда ETR или RTR передаёт Map-Reply, он кодирует свои RLOCs в формате Security Key LCAF Type в поле RLOC-record каждой представляемой записи EID-record.

Если ITR, PITR или RTR передаёт Map-Request с Security Key LCAF Type, а ETR или RTR не хочет получать инкапсулированный трафик с шифрованием, он возвращает Map-Reply без записей RLOC в формате Security Key LCAF Type. Это указывает ITR, PITR или RTR, что шифровать трафик не нужно (маршрутизатор и не сможет его шифровать, поскольку открытый ключ ETR не был получен).

Если ITR или PITR хочет включить несколько key-id в Map-Request, а ETR или RTR хочет использовать лишь часть их, возвращается Map-Reply лишь с этими желаемыми key-id.

7. Общие ключи для плоскости данных

Когда маршрутизатор ITR или PITR получит Map-Reply с воспринятием Cipher Suite, переданного в Map-Request, он готов к созданию ключей плоскости данных, равно как ETR или RTR, возвративший Map-Reply.

Первым шагом является создание общего секрета с использованием предоставленного партнёром открытого ключевого материала DH в сочетании со своим секретным ключевым материалом, как указано в разделе 5. Используемые параметры DH заданы в Cipher Suite, преданном в Map-Request и скопированном в Map-Reply. Полученный общий секрет используется для расчёта ключей AEAD для алгоритмов, заданных в Cipher Suite. Для генерации ключей плоскости данных используется функция вывода ключей (Key Derivation Function или KDF) в режиме счётчика, как указано в [NIST-SP800-108]. Количество выводимого ключевого материала зависит от алгоритмов в Cipher Suite.

Входные данные для KDF указаны ниже.

- Функция KDF - в данном документе это HMAC-SHA-256, но в общем случае задаётся для каждого Cipher Suite.
- Ключ для функции KDF - рассчитанный по методу DH общий секрет.
- Контекст, привязывающий использование ключей плоскости данных к этой сессии. Контекст образуют указанные ниже поля, которые объединяются (конкатенация) и представляются как данные для функции KDF:
  - двухоктетный счётчик с сетевым порядком байтов;
  - строка lisp-crypto с null-символом в конце;
  - значение nonce от ITR из сообщения Map-Request, где был указан Cipher Suite;
  - число требуемых битов ключевого материала (L) в виде двухоктетного значения с сетевым порядком байтов.

Для счётчика в контексте сначала устанавливается значение 1. Когда запрошенный объем ключевого материала превышает размер вывода функции KDF, эта функция вызывается снова и значение счётчика при каждом вызове увеличивается на 1. При получении нужного объёма ключевого материала выполняется его конкатенация и материал используется для создания ключей. Например, для AES со 128-битовыми ключами нужно 16 октетов (128 битов) ключевого материала и HMAC-SHA1-96 требует ещё 16 октетов (128 битов) для поддержки согласованной 128-битовой защиты. Поскольку требуется 32 октета (256 битов) ключевого материала, а функция KDF HMAC-SHA-256 возвращает 256 битов, достаточно одного вызова функции

```
key-material = HMAC-SHA-256(dh-shared-secret, context)
```

где context = 0x0001 || "lisp-crypto" || <itr-nonce> || 0x0100

Значение Cipher Suite, указывающее AES с 256-битовыми ключами требует 32 октета (256 битов) ключевого материала и HMAC-SHA256-128 требует ещё 32 октета (256 битов) для поддержки согласованной 256-битовой защиты. Поскольку нужны 64 октета (512 битов) ключевого материала, а функция KDF HMAC-SHA-256 возвращает 256 битов, нужны 2 вызова

```
key-material-1 = HMAC-SHA-256(dh-shared-secret, context)
```

где context = 0x0001 || "lisp-crypto" || <itr-nonce> || 0x0200

```
key-material-2 = HMAC-SHA-256(dh-shared-secret, context)
```

где context = 0x0002 || "lisp-crypto" || <itr-nonce> || 0x0200

```
key-material = key-material-1 || key-material-2
```

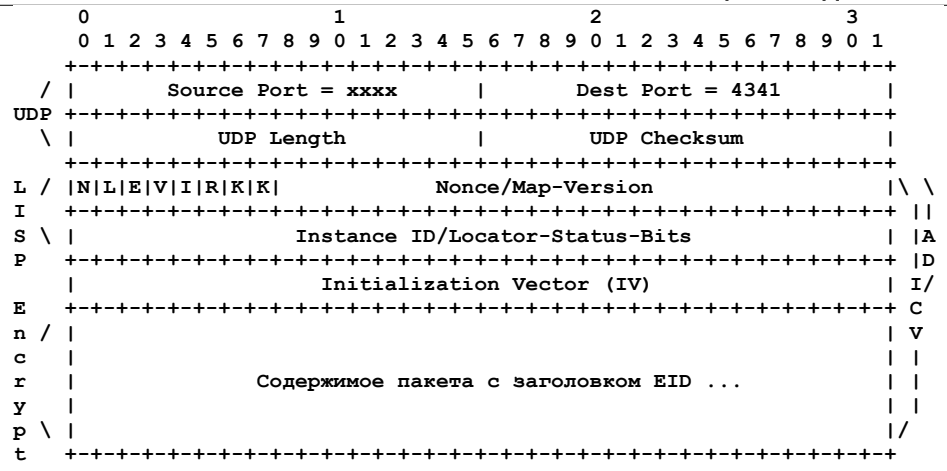
Если размер key-material больше требуемого числа битов (L), используются лишь L старших битов результата.

Из полученного key-material шифры AEAD используют 256 старших битов для AEAD-key. 256-битовый ключ AEAD-key делится на 128-битовый ключ шифрования и 128-битовый ключ проверки целостности в шифре, используемом ITR.

8. Операции плоскости данных

Заголовок инкапсуляции LISP [RFC6830] требуется изменить для кодирования в key-id ключа, применяемого при шифровании.

Когда биты KK имеют значение 00, инкапсулируемый пакет не шифруется. Значения поля KK 1, 2 или 3 указывают key-id секретных ключей, рассчитанных в процессе обмена DH через сообщения Map-Request и Map-Reply. При отличном от 0 поле KK перед данными (payload) помещается вектор инициализации (Initialization Vector или IV), размер которого определяет используемый шифр Cipher Suite. Все заданные этим документом шифры используют AEAD и поле контроля целостности ICV может не включаться в пакет, так как оно указано в шифрованных данных (ciphertext). Дополнительные данные (Additional Data или AD), используемые для ICV, показаны на рисунке и включают заголовок LISP, поле IV и данные пакета (payload).



Биты KK указывают, когда пакет зашифрован и какой ключ применяется.

Когда ITR или PITR получает пакет для инкапсуляции, он сначала выбирает ключ для применения, помещает key-id в заголовок LISP и использует ключ для шифрования всех данных пакета, размещаемых после заголовка LISP. Внешний заголовок, а также заголовки UDP и LISP не шифруются.

На момент создания этого документа существовала открытая рабочая группа для обсуждения необходимости изменения заголовка инкапсуляции данных для шифрования и новых приложений. Этот документ вносит изменения в имеющийся заголовок, чтобы можно было продолжить эксперименты без существенных изменений в плоскости данных. Документ выделяет два не использованных ранее бита (бит R остаётся резервным, как указано в [RFC6830]) для поля KK.

9. Процедуры шифрования и расшифровки

Когда ITR, PITR или RTR инкапсулирует пакет, уже имея рассчитанный ключ AEAD-key (7. Общие ключи для плоскости данных), связанный с RLOC получателя, выполняются указанные ниже процедуры шифрования и инкапсуляции.

1. Инкапсулятор создаёт IV и помещает его перед инкапсулируемым пакетом. Для шифров GCM и ChaCha20 значение IV инкрементируется в каждом пакете (начиная с 1 в первом пакете) и пакет передаётся по RLOC получателя. Для шифров CBC значение IV является случайным для каждого пакета, передаваемого по RLOC получателя. Для шифра ChaCha20 значение IV является 8-байтовым случайным числом, добавляемым в конце 4-байтового счётчика, инкрементируемого в каждом пакете (начиная с 1 в первом пакете).

2. Затем выполняется шифрование с помощью функци AES или ChaCha20 с использованием AEAD-key для содержимого пакета (payload) в соответствии со спецификацией AEAD для шифра Cipher Suite. Шифрование не включает IV, поскольку значение IV должно передаваться в открытом виде, чтобы его можно было использовать при расшифровке. Шифруемое содержимое следует дополнять по размеру до целого числа октетов, которое может потребовать блочный шифр. Результат операции AEAD может включать поле ICV, размер которого определяется соответствующей спецификацией AEAD. Отметим, что AD (т. е. заголовок LISP, в точности совпадающий с добавляемым на следующем этапе, и поле IV) должны предоставляться функции шифрования AEAD как аргумент «associated data».

3. В начало добавляется заголовок LISP. В поле key-id этого заголовка указывается идентификатор ключа, соответствующий использованному при шифровании ключу.

4. В заключение перед шифрованным пакетом добавляется заголовок UDP и внешний заголовок IP и пакет передаётся по RLOC получателя.

Когда ETR, PETR или RTR получает инкапсулированный пакет, выполняются операции декапсуляции и расшифровки.

1. Внешний заголовок IP, а также заголовоки UDP и LISP и поле IV вырезаются из начала пакета. Заголовок LISP и IV сохраняются и передаются операции расшифровки AEAD как аргумент «associated data».

2. Пакет расшифровывается с применением ключа AEAD-key и IV из пакета. Ключ AEAD-key берётся из локального кэша по значению key-id в заголовке LISP. Результатом расшифровки являются открытые данные (plaintext payload), если шифр вернул проверенное значение ICV. В ином случае пакет считается недействительным и отбрасывается. Если спецификация AEAD включает ICV, функция арсшифровки AEAD будет находить значение ICV в шифротексте и сравнивать его с ICV, рассчитанным функцией расшифровки AEAD. Если эти значения ICV не совпадают, пакет считается поделанным.

3. Если пакет не является поддельным, он пересылается после расшифровки по EID получателя.

10. Динамическая смена ключей

Поскольку в сообщениях управления и данных можно указать несколько ключей, ITR может инкапсулировать и шифровать пакеты с использованием конкретного ключа, согласуя с тем же ETR другие ключи. Как только ETR или RTR передаст сообщение Map-Reply, он будет готов к декапсуляции и расшифровке с использованием новых ключей, выведенных с параметрами DH, полученными в Map-Request и возвращенными в Map-Reply.

ITR может в любой момент использовать RLOC-probing для смены ключей или Cipher Suite. При отправке начального Map-Request для заполнения кэша отображений в ITR сообщение Map-Request проходит через систему отображения, где будет отвечать 1 ETR из набора RLOC-set в Map-Reply. Если ITR решит пользоваться другими RLOC из RLOC-set, он должен передать Map-Request напрямую для согласования параметров защиты с ETR. Этот процесс можно использовать для проверки доступности ETR для маршрутизатора ITR при первом добавлении записи map-cache, поэтому ITR может получить сразу статус доступности и ключи в одном обмене Map-Request/Map-Reply.

Событие смены ключей определяется как факт замены маршрутизатором ITR или PITR шифра Cipher Suite или открытого ключа в Map-Request. ETR или RTR сравнивают полученные Cipher Suite и открытый ключ с предоставленными ITR ранее и при наличии изменений рассчитывают новый открытй ключ и Cipher Suite по запросу ITR из Map-Request и возвращают их в Map-Reply. Новый общий секрет можно после этого применять для key-id при шифровании в ITR и расшифровке в ETR. Когда ITR или PITR начинает процесс согласования нового ключа, ему недопустимо применять соответствующий key-id в инкапсулируемых пакетах, пока не получено сообщение Map-Reply от ETR с ожидаемым Cipher Suite (из Map-Request).

Отметим, что при продолжающемся использовании RLOC-probing для проверки доступности RLOC и нежелательности смены ключей ITR или RTR межет отказаться от включения Security Key LCAF Type в Map-Request или указать прежний ключевой материал, полученный в последнем Map-Reply от ETR или RTR. Это указывает ETR или RTR, что смена ключей не требуется.

11. Продолжение работы

По соображениям производительности можно использовать более новые группы Elliptic-Curve Diffie-Hellman (ECDH), как указано в [RFC4492] и [RFC6090] для снижения числа операций CPU при расчёте общих секретных ключей.

Для улучшения защиты а также ускорения программных реализаций будут исследоваться и тестироваться новые подходы к методам шифрования и проверки подлинности. Примерами являются ChaCha20 и Poly1305 [CHACHA-POLY] [RFC7539].

12. Вопросы безопасности

12.1. Поддержка от SAAG

Рабочая группа LISP получила рекомендации по защите от консультативной группы по безопасности (Security Area Advisory Group или SAAG). Группа SAAG участвовала в ранних этапах разработки и её отзывы и предложения включены в документ. Комментарии SAAG приведены ниже.

1. Отказ от использования асимметричных шифров в плоскости данных.
2. Добавление групп ECDH на ранних этапах проектирования.
3. Добавление поля Cipher Suite, поскольку шифры создаются чаще, чем применяющие их протоколы.
4. Рассмотрение новой технологии AEAD для аутентификации и шифрования.

12.2. Угрозы безопасности LISP-Crypto

Поскольку ITR и ETR обмениваются ключами через незащищённые сети общего пользования, атаки с участием человека (man in the middle или MITM) могут обходить обмен ключами и нарушать конфиденциальность плоскости данных. Это может происходить, когда MITM выступает в качестве отправителя Map-Reply и предоставляет свой открытый ключ маршрутизатору ITR, что ведёт его к созданию общего с MITM секретного ключа. Если MITM размещается на пути между ITR и ETR, этот общий секретный ключ можно использовать для расшифровки данных ITR.

Поскольку LISP позволяет защитить сообщения Map-Reply с помощью процесса аутентификации, описанного в [LISP-SEC], ITR может обнаружить подписание MITM сообщения Map-Reply для EID-prefix без полномочий на это. Когда ITR сталкивается с отказом при проверке подписи, он отбрасывает и больше не применяет параметры обмена ключами, избегает использования ETR при инкапсуляции и вносит в системный журнал соответствующую запись для администратора сети. Кроме того, ITR может передать сообщения RLOC-probe по скомпрометированным RLOC для определения доступности полномочного ETR. После успешной проверки подписи в Map-Reply маршрутизатор ITR может начать шифрования и инкапсуляцию пакетов для RLOC этого ETR.

13. Взаимодействие с IANA

В этом документе описан механизм шифрования пакетов с инкапсуляцией LISP на основе процедур обмена ключами Diffie-Hellman. В процессе обмена устройства согласуют используемый Cipher Suite (т. е. функции шифрования э хэширования, служашие для шифрования и расшифровки, а также подписи и проверки пакетов). Для этого зарезервировано 8-битовое поле Cipher Suite в разделе ключевого материала сообщений Map-Request и Map-Reply.

Агентство IANA создало новый реестр (как указано в [RFC5226]) LISP Crypto Cipher Suite. Исходное содержимое реестра приведено в таблице ниже, а новые значения выделяются по процедуре First Come, First Served [RFC5226].

Шифры LISP Crypto.

Значение	***Шифр***	***Описание***
0	Резерв	Раздел 6
1	LISP_2048MODP_AES128_CBC_SHA256	Раздел 6
2	LISP_EC25519_AES128_CBC_SHA256	Раздел 6
3	LISP_2048MODP_AES128_GCM	Раздел 6
4	LISP_3072MODP_AES128_GCM	Раздел 6
5	LISP_256_EC25519_AES128_GCM	Раздел 6
6	LISP_256_EC25519_CHACHA20_POLY1305	Раздел 6

14. Литература

14.1. Нормативные документы

[NIST-SP800-108] National Institute of Standards and Technology, "Recommendation for Key Derivation Using Pseudorandom Functions", NIST Special Publication SP 800-108, DOI 10.6028/NIST.SP.800-108, October 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <http://www.rfc-editor.org/info/rfc2631>.

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, DOI 10.17487/RFC3526, May 2003, <http://www.rfc-editor.org/info/rfc3526>.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <http://www.rfc-editor.org/info/rfc4492>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <http://www.rfc-editor.org/info/rfc5116>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <http://www.rfc-editor.org/info/rfc6090>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <http://www.rfc-editor.org/info/rfc6973>.

[RFC7539] Nir, Y. and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 7539, DOI 10.17487/RFC7539, May 2015, <http://www.rfc-editor.org/info/rfc7539>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <http://www.rfc-editor.org/info/rfc8060>.

14.2. Дополнительная литература

[AES-CBC] McGrew, D., Foley, J., and K. Paterson, "Authenticated Encryption with AES-CBC and HMAC-SHA", Work in Progress, draft-mcgrew-aead-aes-cbc-hmac-sha2-05, July 2014.

[CHACHA-POLY] Langley, A. and W. Chang, "ChaCha20 and Poly1305 based Cipher Suites for TLS", Work in Progress, draft-agl-tls-chacha20poly1305-04, November 2013.

[CURVE25519] Bernstein, D., "Curve25519: new Diffie-Hellman speed records", DOI 10.1007/11745853_14, <http://www.iacr.org/cryptodb/archive/2006/PKC/3351/3351.pdf>.

[DH] Wikipedia, "Diffie-Hellman key exchange", January 2017, <https://en.wikipedia.org/w/index.php?title=Diffie%E2%80%93Hellman_key_exchange&oldid=759611604>.

[LISP-DDT] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "LISP Delegated Database Tree", Work in Progress[1], draft-ietf-lisp-ddt-08, September 2016.

[LISP-SEC] Maino, F., Ermagan, V., Cabellos, A., and D. Saucez, "LISP-Security (LISP-SEC)", Work in Progress[2], draft-ietf-lisp-sec-12, November 2016.

Благодарности

Авторы благодарны Dan Harkins, Joel Halpern, Fabio Maino, Ed Lopez, Roger Jorgensen, Watson Ladd за проявленный интерес, предложения и дискуссии по защите плоскости данных LISP. Отдельная блегодарность руководителю рабочей группы LISP за поддержку этого документа и вклад в раздел «Взаимодействие в IANA».

Авторы выражают особую благодарность Ilari Liusvaara за комментарии и обсуждения. Он поделился своим опытом в части защиты, чтобы сделать lisp-crypto столь защищённым, как это позволяет современная криптография.

Поддержка и предложения рабочей группы SAAG были полезны и заслуживают высокой оценки.

Адреса авторов

Dino Farinacci
lispers.net
San Jose, California 95120
United States of America
Phone: 408-718-2001
Email: farinacci@gmail.com

Brian Weis
Cisco Systems
170 West Tasman Drive
San Jose, California 95124-1706
United States of America
Phone: 408-526-4796
Email: bew@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Опубликовано в RFC 8111. *Прим. перев.*
[2]Опубликовано в RFC 9303. *Прим. перев.*