Internet Engineering Task Force (IETF)
Request for Comments: 5926
Category: Standards Track
ISSN: 2070-1721

G. Lebovitz
Juniper
E. Rescorla
RTFM

June 2010

Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)

Криптографические алгоритмы для опции TCP-AO

Аннотация

Опция аутентификации TCP (TCP Authentication Option или TCP-AO) основана на алгоритмах защиты для обеспечения проверки подлинности в соединениях между конечными точками. Доступно много таких алгоритмов и две системы TCP-AO не смогут взаимодействовать, пока не согласуют использование общего алгоритма. Этот документ задает алгоритмы и атрибуты, которые могут применяться в современных механизмах ручного согласования ключей TCP-AO, и обеспечивает интерфейс для будущих кодов аутентификации сообщений (message authentication code или MAC).

Статус документа

Этот документ содержит проект стандарта Internet (Internet Standards Track).

Документ является результатом работы IETF[1] и представляет собой согласованное мнение членов (сообщества) IETF. Документ был представлен для публичного обсуждения и одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 документа RFC 5741.

Информацию о текущем состоянии этого документа, обнаруженных ошибках и способах обратной связи можно найти по ссылке http://www.rfc-editor.org/info/rfc5926.

Оглавление

1. Введение

Этот документ дополняет [RFC5925]. Подобно многим современным протоколам защиты, TCP-AO позволяет пользователям применять разные криптографические алгоритмы с учетом своих потребностей.

TCP-AO обеспечивает криптографическую аутентификацию и проверку целостности сообщений между парой конечных точек. Для решения этой задачи применяется код аутентификации сообщений (MAC), основанный на общих ключах. Имеется два разных способа создания MAC. Использование кодов MAC на основе хэша (HMAC) определено в [RFC2104], а MAC на основе шифра (CMAC) - в [NIST-SP800-38B].

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

В этом RFC заданы основные требования к MAC, применяемым в TCP-AO, включая указанные здесь и будущие MAC. Документ задает два алгоритма MAC, требуемых во всех реализациях TCP-AO. Указано также две функции вывода ключей (key derivation function или KDF), используемые для создания ключей трафика, применяемых этими MAC. Эти KDF также обязательны для всех реализаций TCP-AO.

2. Требования

2.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с RFC 2119 [RFC2119].

В документе эти слова выделяются **полужирным** шрифтом, а обычное использование слов не означает уровня требования в соответствии с RFC 2119.

2.2. Требования к алгоритму

Ниже приведена начальная спецификация криптографических требований к TCP-AO и указаны 2 алгоритма MAC и 2 функции KDF, которые **должны** быть реализованы для полного соответствия данному RFC.

В таблицах указаны алгоритмы MAC и функции KDF, требуемые для TCP-AO.

Уровень требования	***Алгоритм аутентификации***
Необходимо	HMAC-SHA-1-96 [RFC2104][FIPS-180-3]
Необходимо	AES-128-CMAC-96 [NIST-SP800-38B][FIPS197]

Уровень требования	***Функция создания ключей (KDF)***
Необходимо	KDF_HMAC_SHA1
Необходимо	KDF_AES_128_CMAC

Разъяснения по части обязательности двух алгоритмов MAC приведены в разделе 4.

2.3. Требования к будущим алгоритмам MAC

Опция TCP-AO предназначена для обеспечения криптографической гибкости, поэтому данный документ включает рекомендации для будущих алгоритмов MAC и KDF при использовании с TCP-AO. В частности, будущим алгоритмам MAC **следует** обеспечивать защиту не менее 2^{48} сообщений с вероятностью конфликта не более 10^{-9}.

3. Выбранные алгоритмы

TCP-AO для каждого соединения использовать 2 класса криптографических алгоритмов, заданных этим документом.

(1) Функции создания ключей (KDF), которые являются псевдослучайными функциями (pseudorandom function или PRF) и применяют Master_Key с некоторыми зависящими от соединения входными данными PRF для создания ключей трафика (Traffic_Key), подходящих для проверки подлинности и целостности отдельных сегментов TCP, как описано в TCP-AO.

(2) Алгоритмы MAC, которые принимают ключ и сообщение, создавая тег аутентификации, служащий для проверки целостности и подлинности сообщения.

В TCP-AO эти алгоритмы всегда применяются в паре и каждый алгоритм MAC **должен** задавать KDF для своего использования. Однако функцию KDF **можно** применять с несколькими алгоритмами MAC.

3.1. Функции создания ключей (KDF)

Ключи трафика TCP-AO Traffic_Key выводятся с помощью функции KDF, которая использует текущий ключевой материал, как показано ниже.

```
Traffic_Key = KDF_alg(Master_Key, Context, Output_Length)
```

Параметры выедения ключа рассмотрены ниже.

KDF_alg
: Конкретная псевдослучайная функция (PRF), служащая базой для создания данного Traffic_Key.

Master_Key
: В ручном режиме ключей TCP-AO этот ключ известен обоим партнерам, получающим его через тот или иной интерфейс своей настройки. Master_Key служит затравкой (seed) для KDF. Предполагается, что это человеко-читаемый заранее распространенный ключ (pre-shared key или PSK), что предполагает его переменный размер. Ключам Master_Key **следует** быть случайными (например, не следует выбирать их пользователю). Для обеспечения взаимодействия управляющий интерфейс, с помощью которого настраивается PSK, **должен** воспринимать строки ASCII и **следует** также разрешать указание произвольных двоичных строк в шестнадцатеричном формате. **Могут** поддерживаться и другие методы настройки.

Context
: Двоичная строка с информацией, относящейся к конкретному соединению, для данного выводимого ключевого материала, как описано с параграфе 5.2 [RFC5925].

Output_Length
: Размер (в битах) ключа, создаваемого данной KDF. Этот размер должен соответствовать требованиям алгоритма MAC, использующего результат PRF в качестве затравки.

При вызове KDF создается строка битов размером Output_Length на основе значение Master_Key и context. Затем этот результат может служить криптографическим ключом для любого алгоритма, принимающего ключи размером Output_Length. KDF **может** задавать максимальное значение параметра Output_Length.

3.1.1. Конкретные KDF

Этот документ определяет два алгоритма KDF с соответствующими алгоритмами PRF, как показано ниже:

- KDF_HMAC_SHA1 на основе PRF-HMAC-SHA1 [RFC2104][FIPS-180-3];
- KDF_AES_128_CMAC на основе AES-CMAC-PRF-128 [NIST-SP800-38B][FIPS197].

Обе эти функции KDF основаны на итерационном режиме KDF, заданном в [NIST-SP800-108]. Это означает использование базовой псевдослучайной функции (PRF) с фиксированным размером вывода (128 битов в случае AES-CMAC и 160 для HMAC-SHA1). KDF дает на выходе произвольное число битов за счет работы PRF в «контейнерном» режиме, где каждый вызов PRF использует свой входной блок данных, определяемый счетчиком блоков.

Создание входного блока показано ниже.

```
( i || Label || Context || Output_Length )
```

где

||
: Для любых X || Y символы || представляют конкатенацию (слияние) двоичных строк X и Y.

i
: Счетчик, являющийся двоичной строкой, которая служит вводом для каждой операции PRF в режиме со счетчиком (counter mode). Счетчик i представляется одним октетом. Число итераций будет зависеть от конкретного значения Output_Length, желаемого для данного MAC. Отсчет i всегда начинается с 1.

Label
: Двоичная строка, четко указывающая назначение выводимого данной KDF ключевого материала. Для TCP-AO используется строка ASCII «TCP-AO», где последним символом является заглавная буква O (не 0). Хотя это может показаться лишним в спецификации, поскольку TCP-AO описывает лишь один вызов KDF, строка включена для соответствия FIPS 140.

Context
: Аргумент, предоставляемый интерфейсу KDF в соответствии с параграфом 3.1 .

Output_Length
: Число битов одного ключа, создаваемого KDF. Output_length указывается двумя октетами и задает размер, требуемый алгоритмом MAC, который будет применять результат PRF в качестве затравки (seed).
Вывод множества вызовов PRF просто объединяется (конкатенация). Для Traffic_Key значения от множества вызовов PRF объединяются и отсекаются до нужного размера Traffic_Key. Например, при использовании KDF_HMAC_SHA1 со 160-битовой внутренней PRF для генерации 320 битов данных можно вызвать PRF дважды, с i=1 и i=2. Результатом будет объединение полного вывода первого вызова с результатом второго. Например,

```
Traffic_Key =
       KDF_alg(Master_Key, 1 || Label || Context || Output_length) ||
       KDF_alg(Master_Key, 2 || Label || Context || Output_length)
```

Если требуемое число битов не кратно размеру вывода PRF, результат последнего вызова PRF отсекается до нужного размера.

3.1.1.1. KDF_HMAC_SHA1

- PRF для KDF_alg: HMAC-SHA1 [RFC2104][FIPS-180-3].
- Использование: HMAC-SHA1(Key, Input).
- Ключ: Master_Key, настраиваемый пользователем и передаваемый KDF.
- Ввод: (i || Label || Context || Output_Length)
- Output_Length: 160 битов.
- Результат: Traffic_Key, используемый TCP-AO в функции MAC.

3.1.1.2. KDF_AES_128_CMAC

- PRF для KDF_alg: AES-CMAC-PRF-128 [NIST-SP800-38B][FIPS197].
- Использование: AES-CMAC(Key, Input).
- Ключ: Master_Key (см. ниже)
- Ввод: (i || Label || Context || Output_Length)
- Output_Length: 128 битов.
- Результат: Traffic_Key, используемый TCP-AO в функции MAC.

Master_Key в текущем механизме ручного распространения ключей TCP-AO является общим секретом, задаваемым администратором. Он передается по отдельному каналу (out-of-band) между двумя устройствами и зачастую между двумя организациями. Общий секрет не обязан иметь размер 16 октетов и его размер может меняться. Однако для AES_128_CMAC требуется ключ размером в точности 16 октетов (128 битов). Можно было бы потребовать от администраторов ввода Master_Key размером 128 битов с достаточным уровнем случайности при использовании AES_128_CMAC, но это создаст ненужную нагрузку на разработчиков и администраторов. Данная спецификация **рекомендует** при развертывании использовать в качестве Master_Key случайные строки размером 128 битов, но не требует этого.

Для работы с переменным размером Master_Key используется механизм, описанный в разделе 3 [RFC4615]. Сначала используется AES_128_CMAC с фиксированным ключом, содержащим лишь нули, в качестве «экстрактора случайности», а общий секрет Master_Key (MK) служит входным сообщением для создания 128-битового ключа Derived_Master_Key (K). Затем результат (K) применяется в качестве ключа при повторном запуске AES-128_CMAC с использованием реальных входных данных для получения ключа трафика Traffic_Key (TK), применяемого в MAC для сообщения, как показано ниже.

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                          KDF-AES-128-CMAC                             +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                       +
+ Ввод   : MK (Master_Key, общий секрет переменного размера)           +
+         : I (Input, т. е. входные данные PRF)                         +
+         : MKlen (размер MK в октетах)                                 +
+         : len (размер M в октетах)                                    +
+ Вывод  : TK (Traffic_Key, 128-битовое псевдослучайное значение)      +
+                                                                       +
+-----------------------------------------------------------------------+
+ Переменная: K (128-битовый ключ для AES-CMAC)                         +
+                                                                       +
+ Этап 1.   Если MKlen = 16                                            +
+ Этап 1a.      K := MK;                                               +
+ Этап 1b.  Иначе                                                      +
+               K := AES-CMAC(0^128, MK, MKlen);                        +
+ Этап 2.   TK := AES-CMAC(K, I, len);                                 +
+           возврат TK;                                                 +
+                                                                       +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

На этапе выводится 128-битовый ключ K для AES-CMAC:

- если Master_Key (MK) заданный администратором, имеет размер 128 битов, просто используется этот ключ;
- если ключ имеет иной размер, выводится ключ K с помощью алгоритма AES-CMAC, использующего 128-битовую строку нулей в качестве ключа и MK как входного сообщения (этап 1b).

На этапе 2 снова применяется алгоритм AES-CMAC с использованием ключа K и реального сообщения I.

На выходе этого алгоритма будет значение TK (Traffic_Key) размером 128 битов, подходящее для использования в функции MAC с каждым сегментом TCP в соединении.

3.1.1.3. Рекомендации для пользовательских интерфейсов в части KDF

В этом параграфе дано предлагаемое представление KDF в реализации пользовательских интерфейсов (UI). Следование приведенным ниже рекомендациям упростит взаимодействие и развертывание.

UI **следует** указывать выбор KDF_HMAC_SHA1 просто как SHA1.

UI **следует** указывать выбор KDF_AES_128_CMAC просто как AES128.

Эти значения отражены в исходном реестре IANA.

UI **следует** использовать KDF_HMAC_SHA1, как принятый по умолчанию выбор настроек TCP-AO. KDF_HMAC_SHA1 в данное время предпочтительней по причине широкой поддержки в большинстве реализаций.

3.2. Алгоритмы MAC

Каждый MAC_alg для TCP-AO имеет в своем определении три указанных ниже элемента.

KDF_Alg
Имя алгоритма TCP-AO KDF, используемого для создания Traffic_Key.

Key_Length
Требуемый размер Traffic_Key (в битах) для использования с данным MAC.

MAC_Length
Окончательное число битов в поле TCP-AO MAC. Это может быть усеченный вывод функции MAC.

Коды MAC рассчитываются для TCP-AO, как показано ниже.

```
MAC = MAC_alg(Traffic_Key, Message)
```

где

- MAC_alg - применяемый алгоритм MAC;
- Traffic_Key - переменная, содержащая результат KDF;
- Message - сообщение, аутентифицируемое в соответствии с параграфом 5.1 [RFC5925].

Этот документ задает 2 варианта алгоритма MAC для генерации MAC, используемых TCP-AO:

- HMAC-SHA-1-96 на основе [RFC2104] и [FIPS-180-3];
- AES-128-CMAC-96 на основе [NIST-SP800-38B][FIPS197]

Оба алгоритма обеспечивают высокий уровень защиты и эффективность. Алгоритм AES-128-CMAC-96 потенциально эффективней, особенно при аппаратной реализации, но HMAC-SHA-1-96 шире применяется в протоколах Internet и в большинстве случаев может поддерживаться с минимальными изменениями или совсем без изменения развернутого в настоящее время кода и устройств.

Следует отметить важный аспект в выборе алгоритмов для TCP-AO, связанный с отсечкой выводе MAC до 96 битов. AES-128-CMAC-96 создает 128 битов MAC, а HMAC SHA-1 - 160. Вывод MAC отсекается до 96 битов для обеспечения разумного компромисса между безопасностью и размером сообщений для включения в поле опции TCP-AO.

3.2.1. Использование HMAC-SHA-1-96

По определению для HMAC [RFC2104] требуется криптографическая хэш-функция. SHA1 служит такой функцией для аутентификации и проверки целостности сегментов TCP с помощью HMAC.

Для HMAC-SHA-1-96 применяется 3 фиксированных элемента:

- KDF_Alg - KDF_HMAC_SHA1;
- Key_Length - 160 битов;
- MAC_Length - 96 битов

в

```
MAC = MAC_alg (Traffic_Key, Message)
```

HMAC-SHA-1-96 для TCP-AO имеет значения:

- MAC_alg - HMAC-SHA1
- Traffic_Key - переменная с результатом KDF;
- Message - сообщение для аутентификации в соответствии с параграфом 5.1 в [RFC5925].

3.2.2. Использование AES-128-CMAC-96

В контексте TCP-AO использование AES-128-CMAC-96 реально говорит о применении алгоритма AES-128 для основанного на шифровании кода MAC в соответствии с [NIST-SP800-38B].

Для AES-128-CMAC-96 применяется 3 фиксированных элемента:

- KDF_Alg - KDF_AES_128_CMAC;
- Key_Length - 128 битов;
- MAC_Length - 96 битов

в

```
MAC = MAC_alg (Traffic_Key, Message)
```

AES-128-CMAC-96 для TCP-AO имеет значения:

- MAC_alg - AES-128-CMAC-96 [NIST-SP800-38B];
- Traffic_Key - переменная с результатом KDF;
- Message - сообщение для аутентификации в соответствии с параграфом 5.1 в [RFC5925].

4. Вопросы безопасности

Этот документ наследует все вопросы безопасности, рассмотренные в TCP-AO [RFC5925], AES-CMAC [RFC4493] и HMAC-SHA-1 [RFC2104].

Безопасность основанных на криптографии систем зависит от строгости выбранного криптографического алгоритма и ключей, используемых этим алгоритмом. Защита зависит также от устройства протокола, используемого системой для предотвращения путей обхода криптографии.

Следует принять меры по обеспечению непредсказуемости выбираемых ключей, избегая заведомо слабых ключей для применяемого алгоритма. В [RFC4086] представлена полезная информация по генерации ключей и криптостойким случайным значениям.

Отметим, что в составе KDF_AES_128_CMAC для PRF требуется 128-битовый (16 байтов) ключ для затравки. Однако для удобства развертывания и администрирования решено отказаться от требования вводить 16-байтовый ключ Master_Key. Поэтому задана программа субключа, которая может обрабатывать Master_Key переменного размера, в том числе меньше 16 байтов. Однако это **не** означает, что администраторы могут применять слабые ключи, им рекомендуется следовать [RFC4086], как указано выше. Документ просто пытается «огородить глупость» там, где это возможно.

Этот документ посвящен выбору криптографических алгоритмов для использования с TCP-AO. Алгоритмы, для которых в документе сказано «**необходимо** реализовать», к настоящему времени взломать не удалось, а криптографические исследования позволяют надеяться, что они останутся защищенными в обозримом будущем. Для некоторых алгоритмов с течением времени может измениться представление о защищенности, которое имелось на момент создания этого документа. Следует ждать, что документ будет время от времени обновляться, поэтому для внедрения нужно выбирать наиболее свежую версию документа.

Разъяснение включения 2 обязательных алгоритмов MAC

Два алгоритма MAC и две соответствующих функции KDF указаны как обязательные в результате обсуждений в рабочей группе TCPM и консультаций с руководителями направления Security (Area Director). Алгоритм SHA-1 был выбран в силу его широкой распространенности, достаточной в настоящее время стойкости и разумных вычислительных издержек, поэтому он отнесен к категории **должно** для современной опции TCP-AO. Строгости SHA-1 HMAC должно быть достаточно в обозримом будущем, даже с учетом отсечки до 96 битов.

Недавно обнаруженные уязвимости других MAC (например, MD5 и HMAC MD5) не имеют практического значения для HMAC-SHA-1, но эти типы анализа набирают обороты и могут со временем стать причиной обхода HMAC при достаточном развитии с течением времени. Проблемы безопасности, вызвавшие замену SHA-1 на SHA-256 в цифровых подписях [HMAC-ATTACK] не делают сразу же SHA-1 уязвимым для применения SHA-1 в режиме HMAC.

AES-128 CMAC считается строже SHA-1, но пока еще не так широко распространен. Алгоритм AES-128 CMAC также **должен** быть реализован для стимуляции производителей использовать его и обеспечения запасного варианта MAC на случай компрометации (взлома) SHA-1.

5. Взаимодействие с IANA

Агентство IANA создало и разместило на сайте (http://www.iana.org) реестр Cryptographic Algorithms for TCP-AO Registration Procedure: RFC Publication after Expert Review

Исходно в реестр включены две записи, приведенные в таблице.

Алгоритм	*Документ*
SHA1	[RFC5926]
AES128	[RFC5926]

6. Благодарности

Спасибо Eric "EKR" Rescorla за огромный вклад и комментарии, включая трудное переписывание раздела 3.1.x, что привело к его включению в число авторов документа.

Спасибо Paul Hoffman, чья работа [RFC4308] иной раз служила источником для копирования при создании первой версии.

Спасибо Tim Polk, за письмо, резюмирующее рекомендации SAAG для TCPM по двум алгоритмам хэширования для TCP-AO, значительные части которого были помещены в разные разделы документа.

Спасибо Jeff Schiller, Donald Eastlake и рабочей группе IPsec, чьи документы [RFC4307] и [RFC4835] послужили консультацией и частично использованы в требованиях раздела 2 этого документа.

(Иными словами, автор является лишь редактором чужих текстов, из которых состоит документ.)

Eric "EKR" Rescorla и Brian Weis прояснили проблемы со входными данными PRF для функций KDF. EKR также оказал существенную помощь в структурировании текста, а также в выборе хороших криптографических решений.

Спасибо рабочей группе TCPM, которая терпеливо рецензировала документ в течение 2 лет.

7. Литература

7.1. Нормативные документы

[FIPS-180-3] FIPS Publication 180-3, "Secured Hash Standard", FIPS 180-3, October 2008.

[FIPS197] FIPS Publications 197, "Advanced Encryption Standard (AES)", FIPS 197, November 2001.

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

[NIST-SP800-38B] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", SP 800-38B, May 2005.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, June 2006.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

7.2. Дополнительная литература

[HMAC-ATTACK] "On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1", <http://www.springerlink.com/content/00w4v62651001303>, 2006, <http://eprint.iacr.org/2006/187>.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4307] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, December 2005.

[RFC4308] Hoffman, P., "Cryptographic Suites for IPsec", RFC 4308, December 2005.

[RFC4615] Song, J., Poovendran, R., Lee, J., and T. Iwata, "The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4615, August 2006.

[RFC4835] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, April 2007.

Адреса авторов

Gregory Lebovitz

Juniper Networks, Inc.

1194 North Mathilda Ave.

Sunnyvale, CA 94089-1206

US

Phone:

EMail: gregory.ietf@gmail.com

Eric Rescorla

RTFM, Inc.

2064 Edgewood Drive

Palo Alto, CA 94303

US

Phone: 650-678-2350

EMail: ekr@rtfm.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru