Network Working Group
Request for Comments: 3748
Obsoletes: 2284
Category: Standards Track

B. Aboba
Microsoft
L. Blunk
Merit Network, Inc
J. Vollbrecht
Vollbrecht Consulting LLC
J. Carlson
Sun
H. Levkowetz, Ed.
ipUnplugged
June 2004

Расширяемый протокол аутентификации (EAP)

Extensible Authentication Protocol (EAP)

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Тезисы

В этом документе определен расширяемый протокол аутентификации (EAP[1]) - модель проверки подлинности, которая поддерживает множество методов проверки. EAP обычно работает непосредственно на базе протоколов канального уровня типа PPP[2] или IEEE 802 и не требует использования протокола IP. EAP обеспечивает собственную поддержку повторов передачи и избавления от дубликатов, основанную на гарантированном порядке доставки нижележащего уровня. Фрагментация в EAP не поддерживается, однако отдельные методы EAP могут реализовать свои средства фрагментации.

Этот документ отменяет действие RFC 2284. Перечень отличий от RFC 2284 приведен в Приложении A.

Оглавление

[1]Extensible Authentication Protocol
[2]Point-to-Point Protocol

1. Введение

В этом документе определен расширяемый протокол аутентицикации (EAP) - модель проверки подлинности, которая поддерживает множество методов проверки. EAP обычно работает непосредственно на базе протоколов канального уровня типа PPP или IEEE 802 и не требует использования протокола IP. EAP обеспечивает собственную поддержку повторов передачи и избавления от дубликатов, но она основана на гарантированном порядке доставки нижележащего уровня. Фрагментация в EAP не поддерживается, однако отдельные методы EAP могут реализовать свои средства фрагментации.

EAP может использоваться на выделенных каналах и в коммутируемых устройствах как в проводных, так и в беспроводных средах. В настоящее время протокол EAP реализован на хостах и маршрутизаторах, которые соединены через устройства коммутации или по телефонным линиям с использованием протокола PPP [RFC1661]. Протокол также может быть реализован в коммутаторах и точках доступа, использующих протоколы IEEE 802 [IEEE-802]. Инкапсуляция EAP в проводных средах IEEE 802 описана в стандарте [IEEE-802.1X], а инкапсуляция в беспроводных ЛВС - в стандарте [IEEE-802.11i].

Одним из преимуществ архитектуры EAP является ее гибкость. EAP служит для выбора конкретного механизма аутентификации, обычно после того, как проверяющая сторона (authenticator) запрашивает дополнительную информацию для определения конкретного метода аутентификации, который будет применяться. Вместо требования обновлять проверяющую сторону для поддержки каждого нового метода аутентификации EAP разрешает использование внутреннего сервера аутентификации, который может реализовать те или иные методы аутентификации, а проверяющая сторона просто пропускает через себя все или некоторые методы проверки подлинности партнеров.

В этом документе требования к проверяющей стороне не зависят от того, работает она в проходном (pass-through) режиме или нет. Когда требования относятся к проверяющей стороне или к внутреннему серверу аутентификации (в зависимости от того, где завершается аутентификация EAP), будет использоваться термин «сервер EAP».

1.1. Спецификация требований

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

1.2. Терминология

Ниже определены часто используемые в этом документе термины.

Authenticator - проверяющая сторона (аутентификатор)
Инициирующая аутентификацию EAP сторона соединения. Термин authenticator используется в стандарте [IEEE-802.1X] и в данном документе его трактовка совпадает с принятой в этом стандарте.

Peer - партнер
Сторона соединения, отвечающая на запрос аутентификации. В стандарте [IEEE-802.1X] для обозначения этой стороны соединения используется термин Supplicant.

Supplicant - проверяемая сторона
Сторона соединения, отвечающая на запрос аутентификации. Трактовка этого термина в данном документе совпадает с трактовкой в стандарте [IEEE-802.1X]. Кроме того, в этом документе для обозначения отвечающей на запрос аутентификации стороны используется термин peer (партнер).

Backend authentication server - внутренний сервер аутентификации

Внутренний сервер аутентификации представляет собой объект, обеспечивающий услуги аутентификации для проверяющей стороны. При использовании этого сервера он обычно выполняет методы EAP для проверяющей стороны. Такая же терминология применяется в стандарте [IEEE-802.1X].

AAA

Проерка подлинности (Authentication), полномочий (Authorization) и учет (Accounting). Протоколы AAA для работы с EAP включают RADIUS [RFC3579] и Diameter [DIAM-EAP]. В этом документе термины «сервер AAA» и «внутренний сервер аутентификации» используются, как синонимы.

Displayable Message - отображаемое сообщение

Понятная человеку строка символов. Кодирование сообщений **должно** следовать формату преобразований UTF-8 [RFC2279].

EAP server - сервер EAP

Объект, завершающий использование метода аутентификации EAP с проверяемой стороной. При использовании внутреннего сервера аутентификации сервер EAP является частью проверяющей стороны. При работе проверяющей стороны в проходном (pass-through) режиме сервер EAP размещается на внутреннем сервере аутентификации.

Silently Discard - отбрасывание без уведомления

Отбрасывание пакета без дальнейшей обработки. Реализации **следует** обеспечивать возможность протоколирования таких событий с записью содержимого отбрасываемых пакетов и учитывать такие события в записях статистики.

Successful Authentication - успешная аутентификация

В контексте этого документа успешная аутентификация означает обмен сообщениями EAP, в результате которого проверяющая сторона принимает решение о предоставлении доступа проверяемой стороне, а та принимает решение об использовании этого доступа. Решение проверяющей стороны обычно включает два аспекта - аутентификацию и проверку полномочий, проверяемая сторона может быть аутентифицирован проверяющей, но не получить доступа в соответствии с политикой проверяющей стороны.

Message Integrity Check (MIC) - проверка целостности сообщения

Функция хэширования (с ключом), используемая для аутентификации и защиты целостности данных. Обычно используется термин MAC[1], но в спецификациях IEEE 802 и в данном документе применяется акроним MIC во избежание путаницы с контролем доступа к среде - Medium Access Control.

Cryptographic Separation - криптографическое разделение

Два ключа (x и y) считаются «криптографически разделенными», если наличие всех сообщений, переданных с использованием протокола, не позволяет определить x по y или y по x без «взлома» некоторых криптографических докущений. В частности, это определение допускает наличие у злоумышленника информации о всех nonce, переданных открытым текстом, а также всех предсказуемых значених счетчиков, используемых протоколом. Криптографический взлом будет обычно требовать обращения необратимой (one-way) функции или предсказания значения криптографического генератора псевдослучайных чисел без знания секрета. Иными словами, если ключи разделены криптографически, не существует способа сокращения расчета x из y или y из x и противник должен выполнить расчет, эквивалентный по объему полному поиску значения секретного ключа.

Master Session Key (MSK) - основной ключ сеанса

Ключевой материал, создаваемый сервером и проверяемой стороной EAP, который экспортируется методом EAP. MSK имеет размер не менее 64 октетов. В существующих реализациях сервер AAA, действующий в качестве сервера EAP, доставляет MSK проверяющей стороне.

Extended Master Session Key (EMSK) - расширенный основной ключ сеанса

Дополнительный ключевой материал, создаваемый клиентом и сервером EAP, который экспортируется методом EAP. Размер EMSK составляет не менее 64 октетов. EMSK не используется совместно с проверяющей стороной или третьими сторонами. EMSK зарезервирован на будущее и в настоящее время не используется.

Result indications - индикация результата

Метод обеспечивает индикацию результата, если после передачи и получения последнего сообщения:

1) проверяемая сторона понимает была ли она аутентифицирована сервером и был ли сервер аутентифицирован ею;

2) сервер понимает аутентифицировал ли он другую сторону и был ли аутентифицирован ею.

В случаях, когда успешной аутентификации достаточно для предоставления доступа, стороны будут также знать о желании другой стороны предоставить доступ. Это бывает не всегда и доступ проверяемой стороны может быть отвергнут по причине отсутствия нужных полномочий (например, ограничение сессии) или по иным причинам. Поскольку обмен EAP осуществляется между проверяемой стороной и сервером, на решение о предоставлении доступа могут влиять и друге узлы (например, AAA-прокси). Более подробное обсуждение этого вопроса содержится в параграфе 7.16.

1.3. Применимость

Протокол EAP был разработан для аутентификации в системах доступа, где недоступен уровень IP. Использование EAP для иных целей (например, для передачи больших объемов данных) **не рекомендуется**.

Поскольку EAP не требует связности IP, этот протокол лишь обеспечивает гарантированный транспорт для протоколов аутентификации и не более того.

[1]Message Authentication Code - код аутентификации сообщения.

EAP является пошаговым протоколом и в каждый момент в сети находится только один пакет. В результате протокол EAP не обеспечивает эффективной передачи больших объемов данных, в отличие от транспортных протоколов типа TCP [RFC793] или SCTP [RFC2960].

Хотя EAP поддерживает повторную передачу, он предполагает наличие упорядоченной доставки на нижележащем уровне и поэтому обработка пакетов, доставленных с нарушением порядка, не поддерживается.

Поскольку EAP не поддерживает фрагментации и сборки фрагментов, методам EAP, генерирующим элементы, размер которых превышает минимальное значение EAP MTU, требуется поддерживать свою фрагментацию и сборку.

Хотя методы аутентификации типа EAP-TLS [RFC2716] обеспечивают поддержку фрагментации и сборки, определенные в этом документе методы EAP не поддерживают этого. В результате при превышении пакетом EAP размера MTU для канала, эти методы сталкиваются с проблемами.

Аутентификация EAP инициируется сервером (проверяющей стороной), тогда как многие протоколы аутентификации инициируются клиентом (проверяемой стороной). В результате может потребоваться добавление для алгоритма аутентификации одного или двух дополнительных сообщений (по крайней мере один период кругового обхода) при работе через EAP.

При поддержке аутентификации на основе сертификатов число дополнительных периодов кругового обхода может многократно возрасти по причине фрагментации цепочек сертификатов. В общем случае фрагментированный пакет EAP будет требовать для передачи столько периодов кругового обхода, сколько было создано фрагментов. Например, цепочка сертификатов размером 14960 октетов будет требовать 10 периодов кругового обхода при передаче 1496-октетных EAP MTU.

При работе EAP с протоколами нижележащего уровня, вносящими существенные потери, или при наличии таких потерь на пути между проверяющей стороной и сервером аутентификации методы EAP, требующие множества периодов кругового обхода, могут сталкиваться с трудностями. В таких случаях предпочтительно использовать методы EAP с меньшим числом интервалов кругового обхода.

2. Расширяемый протокол аутентификации (EAP)

Аутентификационный обмен EAP осуществляется в описанном ниже порядке.

[1] Проверяющая сторона передает пакет запроса (Request) на аутентификацию партнера. Пакет имеет поле Type для указания запрашиваемой информации. Примерами типов запроса могут служить Identity, MD5-challenge, и т. П. Тип MD5-Challenge близко соответствует протоколу аутентификации CHAP [RFC1994]. Обычно, проверяющая сторона будет передавать начальный запрос Identity, однако этот запрос не является обязательным и **может** быть пропущен. Например, такой запрос может не потребоваться, когда аутентификация определяется портом, к которому подключен партнер (выделенная линия, выделенное устройство или коммутируемая линия), или аутентификация выполняется иным путем (по отождествления вызывающей станции или MAC-адресу, в поле Name отклика MD5-Challenge и т. П.).

[2] Проверяемая сторона передает пакет отклика (Response) в ответ на пригодный запрос. Как и пакет запроса, отклик имеет поле Type с аналогичным назначением и применением.

[3] Проверяющая сторона передает дополнительный пакет Request, а партнер отвечает на него пакетом Response. Последовательность запросов и откликов повторяется, пока это требуется. Протокол EAP работает в «пошаговом» режиме, поэтому новый запрос не может быть передан до получения пригодного отклика. Проверяющая сторона отвечает за повторную передачу запросов, как описано в параграфе 4.1. После заданного числа повторов передачи проверяющей стороне **следует** завершить EAP-транзакцию. Проверяющей стороне **недопустимо** передавать пакет Success или Failure при повторе передачи или отсутствии отклика от партнера.

[4] Транзакция продолжается, пока не станет ясно, что проверяющая сторона не может аутентифицировать партнера (неприемлемые отклики на один или множество запросов) и в результате она **должна** передать пакет EAP Failure (код 4). Транзакция может также продолжаться, пока проверяющая сторона не решит, что она успешно завершила аутентификацию. В этом случае аутентификатор **должен** передать пакет EAP Success (код 3).

Преимущества

- Протокол EAP может поддерживать множество механизмов аутентификации без предварительного согласования используемого механизма.
- Серверы доступа NAS[1] (например, коммутатор с портом доступа) не обязаны понимать каждый метод доступа и **могут** действовать как проходные агенты для внутреннего сервера аутентификации. Поддержка проходного режима является необязательной. Проверяющая сторона **может** аутентифицировать локальных партнеров и в то же время работать в проходном режиме для нелокальных партнеров и непонятных методов аутентификации.
- Разделение проверяющей стороны и внутреннего сервера аутентификации упрощает управление свидетельствами и реализацию политики принятия решений.

Недостатки

- При использовании с PPP протокол EAP требует добавления нового типа аутентификации в PPP LCP[2], что ведет к необходимости обновления реализаций протокола PPP. Это также является отклонением от предыдущей модели аутентификации в PPP с согласованием конкретного механизма аутентификации в процессе LCP. Аналогично, реализациям коммутаторов и точек доступа требуется поддержка [IEEE-802.1X] для использования EAP.

[1]Network Access Server - сервер доступа в сеть.
[2]Link Control Protocol - протокол управления каналом. П*рим. Перев.*

- Когда проверяющая сторона отделена от внутреннего сервера аутентификации, усложняется анализ защиты и распространение ключей (если оно требуется).

2.1. Поддержка последовательности методов

Транзакция EAP **может** использовать последовательность методов. Типичным примером является запрос Identity, за которым следует один метод аутентификации EAP (например, MD5-Challenge). Однако проверяющая сторона и ее партнер **должны** использовать только один метод аутентификации (тип 4 или выше) в транзакции EAP, после чего проверяющая сторона **должна** передать пакет Success или Failure.

После того, как партнер передал пакет Response того же типа, который был указан в изначальном запросе, проверяющей стороне **недопустимо** передавать запросы других типов (за исключением Notification-Request) до завершения финального цикла данного метода и **недопустимо** передавать запрос любого типа для дополнительного метода после завершения работы исходного метода аутентификации. Партнер, получивший такой запрос, **должен** трактовать его как непригодный и отбрасывать без уведомления. В результате повторные запросы Identity не поддерживаются.

Проверяемой стороне **недопустимо** передавать пакет Nak (обычный или расширенный) в ответ на запрос после того, как был передан изначальный отклик, отличный от Nak. Поскольку атакующий может предавать обманные пакеты EAP Request, проверяющей стороне при получении неожиданного пакета Nak **следует** отбрасывать его, делая запись в журнал событий.

Множество методов аутентификации в одной транзакции EAP не поддерживается по причине уязвимости для MITM[1]-атак (см. Параграф 7.4) и несовместимости с существующими реализациями.

Когда используется один метод аутентификации EAP, но внутри него применяются другие методы («туннелирование» методов), запрет на использование множества методов аутентификации не применим. Такие «туннелированные» методы выглядят с точки зрения EAP, как один метод. Совместимость с ранними версиями может быть обеспечена за счет того, что партнер, не понимающий «туннелированный» метод, может ответить на начальный запрос EAP пакетом Nak (обычным или расширенным). Для предотвращения уязвимостей «туннелированные» методы **должны** поддерживать защиту от MITM-атак.

2.2. Модель мультиплексирования EAP

Концептуально реализация EAP состоит из перечисленных ниже компонент.

[a] **Нижележащий уровень** отвечает за передачу кадров EAP между проверяющей стороной и ее партнером. EAP работает на основе множества протоколов, включая PPP, проводные ЛВС IEEE 802 [IEEE-802.1X], беспроводные ЛВС IEEE 802.11 [IEEE-802.11], UDP (L2TP [RFC2661] и IKEv2 [IKEv2]) и TCP [PIC]. Поведение протоколов нижележащего уровня рассматривается в разделе 3.

[b] **Уровень EAP** получает и передает пакеты EAP через нижележащий уровень, обеспечивает детектирование дубликатов и повторную передачу, а также доставляет и принимает сообщения EAP между уровнями проверяющей и проверяемой сторон EAP.

[c] **Уровни проверяющей и проверяемой сторон EAP**. На основе поля Code уровень EAP демультиплексирует пакеты EAP на уровень проверяющей и проверяемой сторон EAP. Обычно реализация EAP на данном хосте будет поддерживать функциональность проверяемой или проверяющей стороны, но возможно и совмещение этих функций. В последнем случае в реализации EAP присутствуют уровни проверяющей и проверяемой сторон.

[d] **Уровень методов EAP** реализует алгоритмы аутентификации, принимает и передает сообщения EAP через уровни проверяющей и проверяемой сторон EAP. Поскольку сам протокол не обеспечивает поддержки фрагментации, эта задача возлагается на методы EAP, как описано в разделе 5.

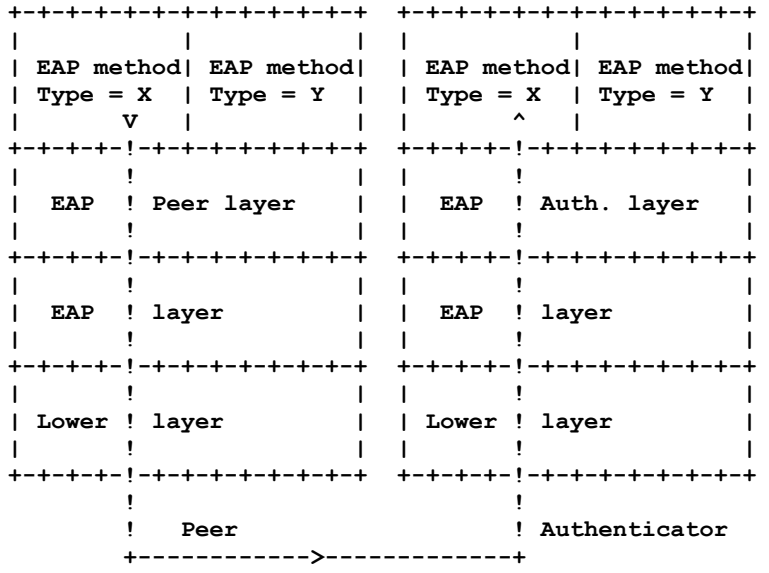


Рисунок 1. Модель мультиплексирования EAP.

Рисунок 1 иллюстрирует модель мультиплексирования EAP. Отметим, что реализация не обязана строго следовать этой модели - важно, чтобы поведение «в проводе» соответствовало модели.

[1]Man-in-the-Middle – перехват и изменение пакетов с участием человека в точке перехвата. П*рим. Перев.*

В EAP поле Code используется подобно номеру протокола в IP. Предполагается, что уровень EAP демультиплексирует входящие пакеты EAP по значению поля Code. Принятые пакеты EAP с кодами 1 (Request), 3 (Success) и 4 (Failure) доставляются уровнем EAP на уровень партнера EAP, если тот реализован. Пакеты EAP с кодом 2 (Response) доставляются уровню проверяющей стороны EAP, если он реализован.

Поле Type используется подобно номеру порта в UDP или TCP. Предполагается, что уровни проверяющей стороны и ее партнера демультиплексируют входящие пакеты EAP по значению типа и доставляют их только методу EAP, соответствующему данному типу. Реализация метода EAP на хосте может регистрировать принятые пакеты от уровня проверяющей стороны или ее партнера, а также от обоих уровней в зависимости от поддерживаемой роли.

Поскольку для методов аутентификации EAP может оказаться желательным доступ к Identity, реализациям **следует** делать запросы и отклики Identity доступными для методов аутентификации (типа 4 и выше), а не только методу Identity. Тип Identity рассматривается в параграфе 5.1.

Отклик Notification используется только в качестве подтверждения партнером запроса Notification, но не подтверждает его обработку или вывод для пользователя. Не следует предполагать, что содержимое запросов и откликов Notification доступно другим методам. Тип Notification рассмотрен в параграфе 5.2.

Методы Nak (тип 3) и Expanded Nak (тип 254) используются для согласования метода. Партнеры отвечают на начальный запрос EAP для неприемлемого типа откликом Nak (тип 3) или Expanded Nak (тип 254). Не следует предполагать, что содержимое откликов Nak доступно другим методам. Типы Nak рассмотрены в параграфе 5.3.

Пакеты EAP с кодами Success и Failure не включают поля Type и не доставляются методам EAP. Коды Success и Failure рассматриваются в параграфе 4.2.

С учетом сказанного здесь сообщения Success, Failure, отклики Nak Response, запросы и отклики Notification **недопустимо** использовать для передачи данных, адресованных методам EAP.

2.3. Проходной режим

При работе в проходном режиме[1] проверяющая сторона просматривает поля Code, Identifier и Length, как описано в параграфе 4.1. Полученные от партнера пакеты EAP пересылаются уровню аутентификации внутреннего сервера аутентификации, а пакеты от этого сервера, предназначенные партнеру, пересылаются последнему.

Хост, получивший пакет EAP, может выполнить по отношению к пакету только одну из трех операций - обработать, отбросить или переслать пакет. Решение о пересылке обычно принимается по результатам проверки полей Code, Identifier и Length. Работающая в проходном режиме реализация **должна** быть способна пересылать пакеты, полученные от партнера с кодом 2 (Response), внутреннему серверу аутентификации. Она также **должна** быть способна принимать пакеты EAP от внутреннего сервера и пересылать партнеру пакеты EAP с кодами 1 (Request), 3 (Success) и 4 (Failure).

Если проверяющая сторона не поддерживает локально один или несколько методов аутентификации, способных проверять подлинность партнера, поля заголовка уровня методов EAP (Type, Type-Data) не проверяются в процессе принятия решения о пересылке. Когда проверяющая сторона поддерживает локальные методы аутентификации, она **может** проверять поле Type, чтобы определить необходимость обработки или пересыли пакета. Соответствующие этой спецификации реализации проходного режима **должны** по умолчанию пересылать пакеты EAP любого типа.

Пакеты EAP с кодами 1 (Request), 3 (Success) и 4 (Failure) демультиплексируются уровнем EAP и доставляются уровню партнера. Поэтому если хост не реализует уровень партнера EAP, эти пакеты отбрасываются без уведомления. Аналогично пакеты EAP с кодом 2 (Response) демультиплексируются уровнем EAP и доставляются уровню проверяющей стороны. Если хост не реализует уровень проверяющей стороны EAP, пакеты отбрасываются без уведомления. Поведение проверяемого узла в проходном режиме не рассматривается в спецификации и не поддерживается протоколами AAA типа RADIUS [RFC3579] и Diameter [DIAM-EAP].

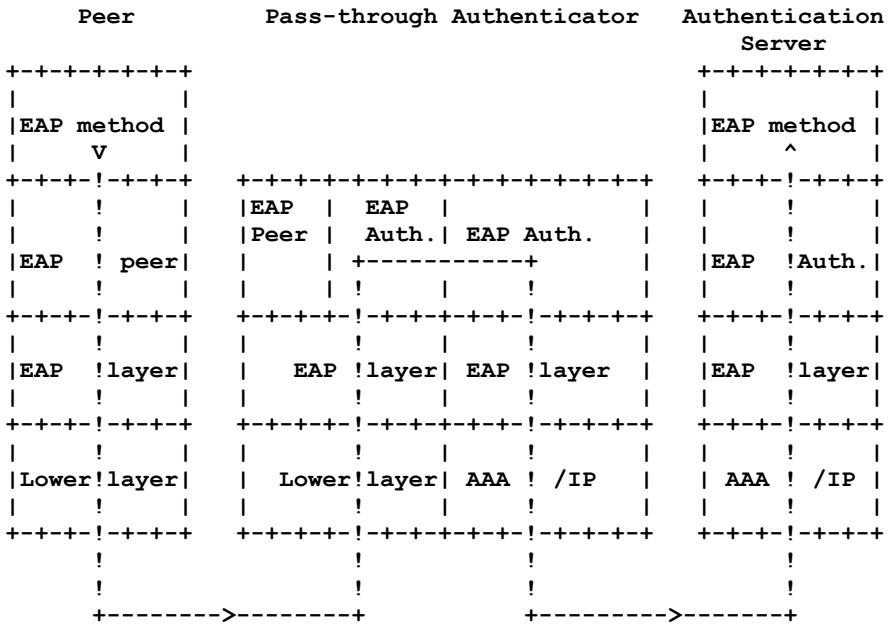


Рисунок 2. Проверяющая сторона в проходном режиме.

Рисунок 2 иллюстрирует модель пересылки.

[1]Pass-through authenticator.

Для сессий, где проверяющая сторона работает в прозрачном режиме, результат аутентификации **должен** определяться только на основе индикации Accept/Reject, переданной внутренним сервером аутентификации. **Недопустимо** определять результат по содержимому пакета EAP, переданного с индикацией Accept/Reject, или отсутствию такой индикации в инкапсулированном пакете EAP.

2.4. Взаимная аутентификация

Поскольку протокол EAP является одноранговым, может возникать независимая и одновременная аутентификация во встречных направлениях (в зависимости от возможностей нижележащего уровня). Обе стороны канала могут действовать, как проверяющие и проверяемые одновременно. В таких случаях обеим сторонам необходимо реализовать уровни проверяющей и проверяемой стороны EAP. Кроме того, метод EAP на обеих сторонах должен поддерживать функциональность проверяющего и проверяемого узла.

Хотя EAP поддерживает взаимную аутентификацию[1] сторон, некоторые реализации EAP, методов, протоколов AAA и канального уровня могут не поддерживать такой режим. Некоторые методы EAP могут поддерживать асимметричную аутентификацию с запросом одного типа свидетельств для проверяющей стороны и другого типа для ее партнера. Хосты, поддерживающие взаимную аутентификацию с таким методом, должны быть обеспечены свидетельствами обоих типов.

Например, EAP-TLS [RFC2716] представляет собой протокол «клиент-сервер», в котором для клиента и сервера обычно используются разные профили сертификатов. Это предполагает для хоста, поддерживающего взаимную аутентификацию с использованием EAP-TLS, необходимость реализовать уровни проверяющей и проверяемой стороны EAP, поддерживать роли обеих сторон в реализации EAP-TLS и обеспечивать подходящие сертификаты для каждой роли.

Протоколы AAA типа RADIUS/EAP [RFC3579] и Diameter EAP [DIAM-EAP] поддерживают только работу в режиме проходной проверяющей стороны. Как было отмечено в параграфе 2.6.2 [RFC3579], сервер RADIUS отвечает на Access-Request, инкапсулированный в пакет EAP-Request, Success или Failure откликом Access-Reject. Следовательно, он не поддерживает работы в режиме проходного партнера.

Даже при использовании метода, поддерживающего взаимную аутентификацию сторон и индикацию результата, некоторые аспекты могут потребовать двух аутентификаций EAP (по одной в каждом направлении). Эти аспекты включают перечисленное ниже.

[1] **Поддержка двухстороннего создания ключей на нижележащем уровне**. Нижележащие уровни типа IEEE 802.11 могут обеспечивать лишь одностороннюю генерацию и доставку временных сеансовых ключей. Например, согласование ключа группы, определенное в [IEEE-802.11i], является односторонним, поскольку в режиме инфраструктуры IEEE 802.11 только точки доступа (AP[2]) передают групповой и широковещательный трафик. В специальном[3] режиме IEEE 802.11, где любой из партнеров может передавать групповой/широковещательный трафик, требуется два односторонних обмена ключами групп. Вследствие присущих архитектуре ограничений это также ведет к необходимости использования индивидуальной адресации при создании ключей и обмена EAP в каждом направлении.

[2] **Поддержка «разрубания узлов» на нижележащем уровне**. Нижележащие уровни типа IEEE 802.11 ad hoc не поддерживают «разрубания узлов», когда два хоста инициируют аутентификацию друг друга, что приводит в результате лишь к одной аутентификации. Это приводит к тому, что даже при поддержке двухстороннего согласования групповых ключей в 802.11 могут выполняться две аутентификации (по одной для направления).

[3] **Политика партнера**. Методы EAP могут поддерживать индикацию результата, позволяя партнеру показать серверу EAP в рамках метода, что он успешно аутентифицировал сервер, а серверу выполнить аналогичную индикацию для партнера. Однако проверяющая сторона в проходном режиме не может быть уверена, что партнер принял представленные сервером EAP свидетельства, пока эта информация не будет представлена проверяющей стороне по протоколу AAA. Проверяющей стороне **следует** интерпретировать получение ключа в пакете Accept, как индикацию успешной аутентификации сервера ее партнером.

Однако возможно, что политика доступа партнера оказалась неподходящей в начальном обмене EAP даже при взаимной аутентификации. Например, проверяющая сторона EAP может не показывать наличие полномочий для работы в роли проверяющей и проверяемой сторон. В результате партнер может потребовать дополнительной аутентификации в обратном направлении, даже если сервер EAP аутентифицирован им.

3. Поведение нижележащего уровня

3.1. Требования к нижележащему уровню

Ниже перечислены допущения EAP о нижележащем уровне.

[1] **Транспорт без гарантии доставки**. В EAP проверяющая сторона продолжает передавать запросы, пока на них не получено откликов, так что EAP не полагается на гарантию доставки на нижележащем уровне. Поскольку EAP определяет свое поведение для повторов передачи, возможно (хотя и не желательно) повторение передачи на уровне EAP и на нижележащем уровне при обеспечении тем гарантий доставки.

Отметим, что пакеты Success и Failure не передаются повторно. Без гарантий нижележащего уровня и при заметном количестве ошибок эти пакеты могут теряться, что будет приводить к тайм-аутам. Следовательно, для реализаций желательно повысить свою устойчивость к потере пакетов Success и Failure, как описано в параграфе 4.2.

[1] **Детектирование ошибок**. Хотя EAP не предполагает гарантированной доставки на нижележащем уровне, протокол опирается на детектирование ошибок этого уровня (например, CRC, Checksum, MIC и т. П.). Методы EAP могут не включать MIC или рассчитывать контрольную сумму не для всех полей пакета EAP (таких, как Code, Identifier, Length или Type). В результате без детектирования ошибок на нижележащем уровне

[1]В оригинале используется термин «peer-to-peer operation». П*рим. Перев.*
[2]Access Point.
[3]Ad hoc.

незамеченные ошибки могут попадать в поля заголовков уровня EAP или уровня методов EAP, приводя к отказам при аутентификации.

Например, метод EAP TLS [RFC2716] рассчитывает значение MIC только для поля Type-Data и трактует несовпадение MIC, как критическую ошибку. Без детектирования ошибок на нижележащем уровне этот метод и аналогичные ему методы не смогут работать надежно.

[2] **Защита**. EAP не требует от нижележащего уровня поддержки услуг защиты типа конфиденциальности пакетов, аутентификации, целостности и защиты от повторного использования пакетов. Однако при доступности таких услуг для динамического получения ключевого материала могут использоваться методы EAP, поддерживающие Key Derivation (см. Параграф 7.2.1). Это позволяет связать аутентификацию EAP с последующими данными и обеспечить защиту от изменения данных, повторного использования пакетов или использования ложных пакетов (spoofing). Более подробное рассмотрение приведено в параграфе 7.1.

[3] **Минимальное значение MTU**. EAP может работать с нижележащими уровнями, которые обеспечивают размер EAP MTU не менее 1020 октетов.

EAP не поддерживает определение MTU для пути, а фрагментация и сборка не поддерживаются ни EAP, ни определенными в этой спецификации методами Identity (1), Notification (2), Nak Response (3), MD5-Challenge (4), One Time Password (5), Generic Token Card (6) и расширенным Nak (254).

Обычно партнер EAP получает информацию о EAP MTU от нижележащего уровня и устанавливат подходящее значение для размера кадров EAP. Когда проверяющая сторона работает в проходном режиме, у сервера аутентификации нет прямого пути определения EAP MTU и поэтому он полагается на получение информации от проверяющей стороны (например, с помощью атрибута Framed-MTU, описанного в параграфе 2.4 [RFC3579]). Тогда как методы типа EAP-TLS [RFC2716] поддерживают фрагментацию и сборку, методы EAP, изначально разработанные для использования с протоколом PPP, где гарантируется поддержка MTU размером 1500 октетов для кадров управления (см. Параграф 6.1 [RFC1661]), могут не поддерживать функций фрагментации и сборки.

Методы EAP могут предполагать по умолчанию минимальное значение EAP MTU в 1020 октетов. Методам EAP **следует** включать поддержку фрагментации и сборки, если размер их элементов данных превышает указанное минимальное значение EAP MTU.

EAP является пошаговым протоколом, что ведет к снижению эффективности при использовании фрагментации. Следовательно, если нижележащий уровень поддерживает фрагментацию и сборку (как в случае работы EAP по протоколу IP), использование фрагментации на этом уровне может оказаться предпочтительней фрагментации в EAP. Этого можно достигнуть путем задания излишне большого значения EAP MTU, что приведет к использованию фрагментации и сборки на нижележащем уровне.

[4] **Возможность дублирования пакетов**. Нижележащий уровень с гарантированной доставкой будет обеспечивать уровню EAP поток пакетов, не содержащий дубликатов. Отсутствие дубликатов на нижележащем уровне желательно, но не требуется. Поле Identifier позволяет проверяющей стороне и ее партнеру обнаруживать дубликаты пакетов.

[5] **Гарантии порядка доставки**. EAP не требует монотонного роста значения поля Identifier и для корректной работы ему нужно сохранение порядка доставки на нижележащем уровне. Изначально протокол EAP разрабатывался для использования с протоколом PPP, в спецификации которого [RFC1661] (раздел 1) сказано:

> «Протокол PPP предназначен для простых каналов, передающих пакеты между двумя партнерами. Эти каналы обеспечивают полнодуплексную двухстороннюю передачу и предполагается, что порядок доставки пакетов сохраняется.»

Нижележащий транспорт для EAP **должен** сохранять порядок доставки между отправителем и получателем для заданного уровня приоритета (гарантии сохранения порядка в [IEEE-802]).

Нарушение порядка доставки обычно будет приводить к отказу аутентификации EAP и необходимости повторного запуска процедур аутентификации. В среде, где нарушение порядка является обычным делом, столь же обычными будут и отказы при аутентификации EAP. **Рекомендуется** использовать EAP только с нижележащими уровнями, обеспечивающими сохранение порядка доставки. Использование EAP с транспортом raw IP или UDP **не рекомендуется**. Инкапсуляция EAP в RADIUS [RFC3579] удовлетворяет требованиям по сохранению порядка, поскольку протокол RADIUS гарантирует такое сохранение.

3.2. Использование EAP с протоколом PPP

При организации связи через канал PPP каждая из сторон соединения сначала передает пакеты LCP для настройки канала передачи данных в фазе Link Establishment. После того как канал будет организован, PPP может использовать необязательную фазу аутентификации (Authentication) перед переходом в фазу Network-Layer Protocol.

По умолчанию аутентификация не является обязательной. Если аутентификация для соединения желательна, реализация **должна** задать опцию настройки протокола аутентификации[1] в фазе организации соединения.

Если в фазе Authentication проверка подлинности партнера была выполнена, сервер может использовать аутентификационные данные при выборе опций для согласования на сетевом уровне.

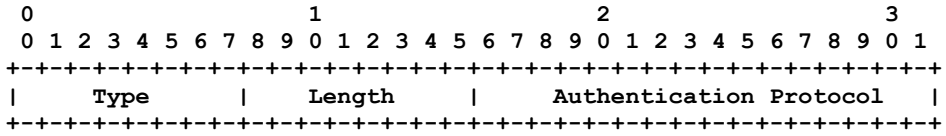
При реализации с PPP протокол EAP не выбирает конкретный механизм аутентификации в фазе PPP Link Control, а оставляет этот выбор для фазы Authentication. Это позволяет проверяющей стороне запросить больше информации перед определением конкретного механизма аутентификации. Кроме того, это позволяет использовать «внутренний» сервер, который реально поддерживает различные механизмы, тогда как проверяющая сторона PPP просто передает через себя аутентификационный обмен. Фазы организации канала и аутентификации PPP, а также опция Authentication Protocol Configuration определены в спецификации протокола PPP [RFC1661].

[1] Authentication Protocol Configuration Option

3.2.1. Формат опции настройки протокола аутентификации для PPP

Формат опции PPP Authentication Protocol Configuration для согласования EAP показан на рисунке. Поля опции передаются слева направо.

В поле Information кадра PPP Data Link Layer, где поле протокола имеет шестнадцатеричное значение C227 (PPP EAP), инкапсулируется один пакет EAP.



Type

3

Length

4

Authentication Protocol

Шестнадцатеричное значение C227 показывает протокол EAP.

3.3. Использование EAP с протоколами IEEE 802

Инкапсуляция EAP в кадры IEEE 802 определена в стандарте [IEEE-802.1X]. Инкапсуляция IEEE 802 для EAP не включает PPP и протокол IEEE 802.1X не поддерживает согласований для канального или сетевого уровня. В результате этого при использовании IEEE 802.1X нет возможности согласования не входящих в EAP механизмов аутентификации типа PAP или CHAP [RFC1994].

3.4. Индикация нижележащего уровня

Надежность и безопасность индикации нижележащего уровня зависит от этого уровня. Поскольку протокол EAP не привязан к определенным средам, наличие защиты на нижележащем уровне не принимается во внимание при обработке сообщений EAP.

Партнер, получивший индикацию успеха от нижележащего уровня (см. Параграф 7.2), в целях повышения надежности **может** предположить, что пакет Success был потерян и вести себя так, будто этот пакет был получен. Это включает и игнорирование пакета Success в некоторых условиях, как описано в параграфе 4.2.

Обсуждение некоторых вопросов надежности и безопасности индикации нижележащего уровня в PPP, проводных сетях IEEE 802 и беспроводных сетях IEEE 802.11 приведено в параграфе 7.12. Канальный уровень.

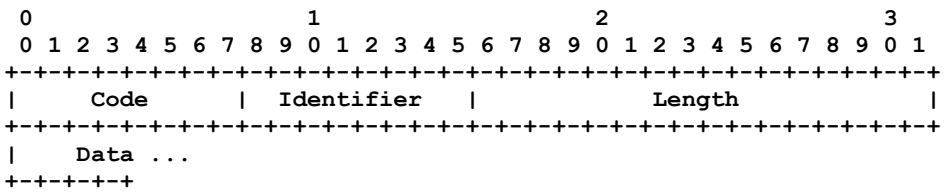
После завершения аутентификации EAP проверяемая сторона обычно будет передавать и принимать данные через проверяющую сторону. Желательно обеспечить гарантию того, что элементы этих данных будут совпадать с теми, которые использовались при успешной аутентификации EAP. Для решения этой задачи требуется, чтобы нижележащий уровень обеспечивал на уровне отдельных пакетов защиту целостности, аутентификацию и защиту от повторного использования пакетов. Эти услуги нужно связать с ключами, созданными в процессе аутентификации EAP. В противном случае последующий трафик может быть изменен, подменен или использован повторно.

Когда ключевой материал для шифронабора нижележащего уровня обеспечивается EAP, согласование шифронабора и активация ключей контролируются нижележащим уровнем. Для случая PPP шифронаборы согласуются в ECP так, что ключи, полученные из аутентификации EAP, невозможно использовать до завершения ECP. Поэтому начальный обмен EAP невозможно защитить с использованием шифронабора PPP, хотя повторная аутентификация EAP может быть защищена.

В средах IEEE 802 начальная активация ключей также обычно происходит после завершения аутентификации EAP. Поэтому начальный обмен EAP обычно невозможно защитить с помощью криптографии нижележащего уровня, хотя повторная аутентификация и обмен перед аутентификацией могут быть защищены.

4. Формат пакетов EAP

В этом разделе описан формат пакетов EAP. Поля передаются слева направо.



Code

Однооктетное поле Code показывает тип пакета EAP и может принимать значения:

1 Request — запрос;
2 Response — отклик;
3 Success — успех;
4 Failure — отказ.

Поскольку в EAP определены только коды 1-4, пакеты EAP с другими значениями кода **должны** отбрасываться обеими сторонами без уведомелния.

Identifier

Однооктетное поле Identifier используется для обеспечения соответствия между запросами и откликами.

Length

Двухоктетное поле Length показывает размер (в октетах) пакета EAP с учетом полей Code, Identifier, Length и Data. Октеты, выходящие за пределы указанного размера, следует трактовать, как заполнение канального уровня - на приеме эти данные **должны** игнорироваться. Сообщения, в которых значение поля Length превышает размер полученного пакета, **должны** отбрасываться без уведомления.

Data

Поле Data не является обязательным, а его формат зависит от типа пакета (значения поля Code).

4.1. Запросы и отклики

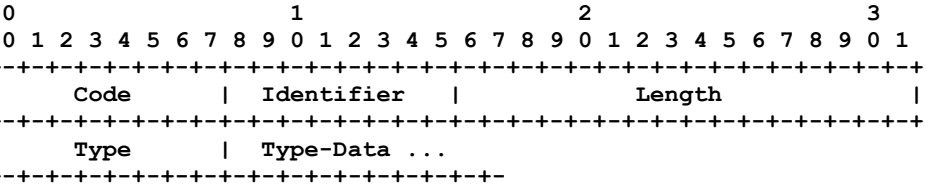
Описание

Пакеты Request (запрос, код 1) передаются партнеру проверяющей стороной. Каждый пакет Request имеет поле Type, которое показывает тип запрашиваемой информации. Дополнительные пакеты Request **должны** передаваться, пока не будет получен пригодный отклик, не завершится отсчет числа попыток или нижележащий уровень не сообщит об отказе.

Повторные запросы **должны** передаваться с тем же значением поля Identifier, чтобы их можно было отличить от новых запросов. Содержимое поля данных зависит от типа запроса. Партнер **должен** передавать пакет Response в ответ на корректный запрос. Отклики **должны** передаваться только в ответ на корректные пакеты Request и никогда не должны повторяться по таймеру.

Если партнер получает корректный дубликат запроса, на который уже передан отклик, он **должен** повторить передачу исходного пакета Response без обработки полученного дубликата. Запросы **должны** обрабатываться в порядке их получения, обработка предыдущего запроса **должна** быть завершена до начала работы с новым пакетом Request.

Ниже приводится описание полей пакетов Request и Response. Поля передаются слева направо.



Code

1 для Request;
2 для Response.

Identifier

Поле Identifier занимает 1 октет. Значения идентификатора в передаваемых повторно (по атйм-ауту в ожидании отклика) пакетах запросов **должны** совпадать со значением поля в исходном пакете Request. В новых (не повторных) запросах значение поля Identifier **должно** меняться.

Поле Identifier в пакетах Response **должно** соответствовать значению идентификатора в остающемся незавершенным запросе. Проверяющая сторона при получении пакета Response с идентификатором, не соответствующим остающемуся запросу, **должна** отбрасывать такой пакет без уведомления.

Для того, чтобы не путались новые запросы с повторами предыдущих, выбираемые для каждого нового пакета Request значения поля Identifier должны отличаться только от идентификатора в исходном запросе - уникальности идентификаторов для всех запросов транзакции не требуется. Одним из путей решения этой задачи является увеличение значения поля Identifier на 1 в каждом новом запросе. Рекомендуется для первого запроса выбирать значение идентификатора случайным образом, а не начинать отсчет с нуля, поскольку такое решение затруднит организацию атак.

Поскольку пространство значений идентификаторов для каждой сессии независимо, проверяющая сторона не ограничена 256 одновременными аутентификационными транзакциями. По этой же причине повторная аутентификация в транзакции EAP может продолжаться достаточно долго и не ограничена 256 периодами кругового обхода.

Примечание для разработчиков. Проверяющая сторона несет ответственность за повторную передачу пакетов Request. Если сообщение Request получено из некого источника (например, от внутреннего сервера аутентификации), проверяющая сторона должна сохранить копию такого запроса на случай возможного повтора передачи. Партнер отвечает за детектирование и обслуживание дубликатов запросов до начала какой-либо их обработки, включая передачу третьей стороне. Проверяющая сторона отвечает также за отбрасывание сообщений Response с непригодным значением поля Identifier до какой-либо их обработки, включая пересылку внутреннему аутентификационному серверу для проверки. Поскольку проверяющая сторона может повторить передачу до получения от партнера отклика, она может получить множество пакетов Response с одним идентификатором. Пока проверяющей стороной не будет принят новый запрос, значение поля Identifier не изменяется, поэтому проверяющая сторона пересылает отклики внутреннему серверу аутентификации по одному.

Length

Двухоктетное поле Length показывает размер пакета EAP с учетом полей Code, Identifier, Length, Type и Type-Data. Октеты, выходящие за пределы указанного размера, следует трактовать, как заполнение канального уровня - на приеме эти данные **должны** игнорироваться. Сообщения, в которых значение поля Length превышает размер полученного пакета, **должны** отбрасываться без уведомления.

Type

Поле Type занимает 1 октет и показывает тип запроса или отклика. В каждом пакете EAP Request или Response **должно** быть указано одно значение Type. Исходные спецификации типов описаны в разделе 5 данного документа.

Поле Type в отклике **должно** соответствовать полю типа в запросе или обычному/расширенному Nak (см. Параграф 5.3), показывающему неприемлемость типа запроса для партнера. Партнеру **недопустимо** слать пакет Nak (обычный или расширенный) в ответ на запрос после того, как был передан исходный отклик без Nak. Сервер EAP, получивший пакет Response, который не удовлетворяет перечисленным здесь требованиям, **должен** отбросить его.

Type-Data

Поле Type-Data зависит от типа запроса и отклика.

4.2. Пакеты Success и Failure

Пакет Success передается проверяющей стороной партнеру после завершения метода EAP (тип 4 или выше) для индикации успешно завершенной аутентификации партнера. Проверяющая сторона **должна** передать пакет EAP с Code = 3 (Success). Если партнера аутентифицировать не удается (непригодные отклики на один или множество запросов), после неудачного завершения метода EAP реализация **должна** передать пакет EAP с кодом 4 (Failure). Проверяющая сторона **может** ввести множество запросов до передачи отклика Failure, чтобы предотвратить влияние ошибок (опечаток) пользователя. В пакеты Success и Failure **недопустимо** включать дополнительные данные.

Проверяющей стороне **недопустимо** передавать пакет Failure, если спецификация данного метода не позволяет явно заканчивать работу метода в этой точке. Реализация EAP на проверяемой стороне, получившая пакет Success или Failure, передача которого не разрешена явно, **должна** отбросить этот пакет без уведомления. По умолчанию проверяемая сторона EAP **должна** отбрасывать без уведомления «пьяные» пакеты Success (пакеты Success, переданные сразу после соединения). Это предотвращает для некорректно работающих реализаций проверяющей стороны возможность обойти взаимную аутентификацию за счет передачи пакета Success до завершения транзакции метода EAP.

Примечание для разработчиков. Поскольку пакеты Success и Failure не подтверждаются, проверяющая сторона не передает их повторно и возникает риск потери такого пакета. Проверяемая сторона в таких случаях **должна** вести себя, как описано здесь. Информация по обработке индикации успеха или неудачи от нижележащего уровня была рассмотрена в параграфе 3.4.

Как было отмечено в параграфе 2.1, для транзакции EAP разрешается использовать только один метод аутентификации EAP. Методы EAP могут использовать разные способы индикации. После того, как проверяющая сторона передаст партнеру индикацию отказа, она **должна** передать вслед пакет Failure, независимо от отклика партнера. После того, как проверяющая сторона передаст партнеру индикацию успеха и получит индикацию успеха от того, она **должна** передать вслед пакет Success.

Проверяемая сторона после неудачного завершения метода аутентификации (индикация отказа от проверяющей стороны или нежелание партнера продолжать транзакцию - возможно после передачи индикации негативного результата) **должна** прервать транзакцию и передать индикацию отказа на нижележащий уровень. Проверяемая сторона **должна** отбрасывать без уведомления пакеты Success и **может** отбрасывать без уведомления пакеты Failure. В результате потеря пакета Failure не будет приводить к тайм-ауту.

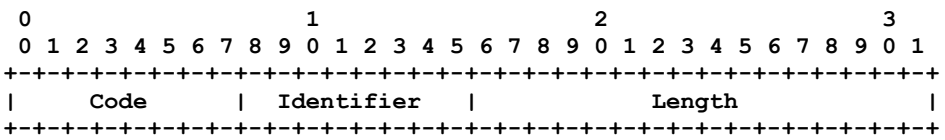
Проверяемая сторона после обмена индикаторами успешного завершения с обеих сторон, **должна** без уведомления отбрасывать пакеты Failure. Если пакет EAP Success не был получен, проверяемая сторона может предположить потерю пакета.

Если проверяющая сторона не передала индикацию результата и партнер желает продолжать транзакцию, он будет ждать пакета Success или Failure после завершения работы метода и отбрасывать такой пакет без уведомления **недопустимо**. Если ни одного из этих пакетов не было получено, проверяемой стороне **следует** прервать транзакцию во избежание длительного ожидания при потере пакета EAP Failure.

Если партнер сталкивается с отказом при попытке аутентифицировать проверяющую сторону, последняя **должна** передать пакет Failure. **Недопустимо** в таких случаях предоставлять доступ, передавая пакет Success. Однако проверяющая сторона **может** опускать аутентификацию партнера в случаях предоставления ограниченного доступа (например, гостевого). В таком случае проверяющая сторона **должна** передать пакет Success.

Когда партнер аутентифицировал проверяющую сторону, но последняя не передала индикацию результата, она **может** отвергнуть доступ, передав пакет Failure, если у партнера нет полномочий доступа в сеть.

Ниже показан формат пакетов Success и Failure. Поля передаются слева направо.



Code

3 для Success;
4 для Failure.

Identifier

Поле Identifier занимает 1 октет и служит для сопоставления с откликами. Значение поля Identifier **должно** соответствовать значению аутентификатора в пакете Response, на который передается ответ.

Length

4

4.3. Поведение при повторе передачи

Поскольку процесс аутентификации часто включает ввод данных пользователем, следует принимать меры предосторожности при выборе стратегии повторной передачи и тайм-аутов. По умолчанию, когда EAP работает с нижележащим уровнем без гарантированной доставки, значение таймера повторной передачи EAP **следует** оценивать динамически. Предлагается повторять передачу не более 3 - 5 раз.

При работе с нижележащим уровнем, обеспечивающим гарантии доставки (например, EAP на основе ISAKMP/TCP, как описано в [PIC]), для таймера повторной передачи на проверяющей стороне **следует** устанавливать бесконечное значение, чтобы на уровне EAP повторной передачи не происходило. Партнер может поддерживать свое значение тайм-аута, чтобы предотвратить бесконечное ожидание запроса.

Когда процесс аутентификации требует пользовательского ввода, время кругового обхода определяется скорей действиями пользователя, чем параметрами сети, поэтому динамическое определение RTO может оказаться бесполезным. В таких случаях для таймера повтора следует устанавливать значение, достаточное для действий пользователя с увеличением в некоторых случаях (например при использовании карт-маркеров) значения тайм-аута (параграф 5.6).

Рекомендации по выбору значения тайм-аута для проверяющей стороны может дать внутренний сервер аутентификации (например, через атрибут RADIUS Session-Timeout).

Для динамической оценки таймера повтора передачи в EAP **рекомендуется** применять алгоритмы оценки SRTT, RTTVAR и RTO, описанные в [RFC2988], включая алгоритм Karn с описанными ниже возможными изменениями.

[a] Для предотвращения одновременной синхронизации множества распределенных систем таймеры повтора передачи задаются с флуктуациями на основе значения RTO с добавлением случайной величины из диапазона -RTOmin/2 - RTOmin/2. **Могут** использоваться и другие способы задания флуктуаций, однако значения **должны** быть псевдослучайными. Обсуждение генерации псевдослучайных значений приведено в работе [RFC1750].

[b] При работе EAP по одному каналу (а не через Internet) **можно** использовать меньшие значения RTOinitial, RTOmin и RTOmax. Рекомендуется задавать значения RTOinitial=1 сек., RTOmin=200 мсек. И RTOmax=20 сек.

[c] При работе EAP по одному каналу (а не через Internet) **можно** сделать оценки для проверяющей стороны, а не для сессии. Это позволяет при оценке более полно использовать информацию о поведении нижележащего уровня.

[d] Реализация EAP **может** сбросить значения SRTT и RTTVAR после многократного снижения значения таймера, поскольку очевидно, что эти значения в такой ситуации не отражают реальность. После сброса SRTT и RTTVAR их следует инициализировать следующим значением RTT[1], полученным в соответствии с уравнением 2.2 в [RFC2988].

5. Типы начальных запросов и откликов EAP

В этом параграфе определен начальный набор типов EAP, используемых в обменах Request/Response. В будущих документах могут быть определены новые типы. Поле Type занимает 1 октет и определяет структуру пакета с запросом или откликом EAP. Первые три типа имеют специальное значение.

1	Identity
2	Notification
3	Nak (только в Response)
4	MD5-Challenge
5	Однократный пароль (OTP)
6	Базовая карточка доступа (GTC)
254	Расширенные типы
255	Для экспериментов

Оставшиеся типы определяют аутентификационные обмены. Типы Nak (3) и Expanded Nak (254) применимы только для откликов и их **недопустимо** использовать в запросах.

Все реализации EAP **должны** поддерживать типы 1-4, определенные в данном документе, **следует** поддерживать также тип 254. Реализации **могут** поддерживать другие типы, определенные здесь или в будущих RFC.

Методы EAP **могут** поддерживать аутентификацию на основе общего секрета. Если таким секретом является вводимая пользователем комбинация символов, реализация **может** поддерживать кодировки символов, отличные от ASCII. В таких случаях обработку ввода следует выполнять с использованием подходящего профиля stringprep [RFC3454] и переводить введенные пользователем данные в строку октетов с использованием кодировки UTF-8 [RFC2279]. Предварительная версия возможного профиля stringprep описана в [SASLPREP].

5.1. Identity

Описание

Тип Identity используется для запроса отождествления партнера. В общем случае проверяющая сторона может ввести запрос этого типа в качестве стартового. В тех случаях, когда предполагается взаимодействие с пользователем на стороне партнера, **может** включаться дополнительное отображаемое сообщение. В ответ на запрос типа 1 (Identity) **следует** передавать отклик типа 1 (Identity).

Некоторые реализации EAP включают различные опции в запрос типа Identity после NUL-символа. По умолчанию реализации EAP **не следует** предполагать, что запрос или отклик типа Identity может быть больше 1020 октетов.

Рекомендуется использовать отклики Identity прежде всего в целях маршрутизации и выбора используемого метода EAP. Методам EAP **следует** включать механизм получения отождествления, чтобы не возникало необходимости опираться на отклик Identity. Запросы и отклики Identity передаются в открытом виде, поэтому атакующий может увидеть отождествление и даже изменить его. Для предотвращения такой угрозы предпочтительно включать в аутентификационный обмен метод EAP, который поддерживает аутентификацию на уровне пакетов, а также обеспечивает защиту целостности, конфиденциальность и предотвращение повторного использования пакетов. Отклик Identity может оказаться непригодным для такого метода - он может быть усеченным или затуманенным для сохранения тайны, а также «декорированным» для маршрутизации. Когда партнер настроен на восприятие только методов аутентификации, поддерживающих защищенные обмен отождествлениями, этот партнер **может** предоставлять сокращенный отклик Identity (например без своего имени в NAI [RFC2486]). Дополнительное рассмотрение этого вопроса содержится в параграфе 7.3.

Примечание для разработчиков. Партнер **может** получать в результате ввода данных пользователем. Проверяющей стороне рекомендуется повторять Identity Request в случаях получения непригодного отождествления или отказа при аутентификации, поскольку пользователи могут ошибаться при вводе. Предлагается повторять Identity Request не менее 3 раз, прежде чем прервать аутентификацию. Для индикации некорректной попытки перед повтором Identity Request **может** передаваться Notification Request. **Возможно** также индицировать отказ в сообщении нового запроса Identity.

[1]Round-Trip Time - время кругового обхода. П*рим. Перев.*

Type
1

Type-Data
Это поле **может** содержать отображаемое сообщение, представленное символами ISO 10646 с кодировкой UTF-8 [RFC2279]. Если Request содержит null-символ, выводиться будет только часть этого поля до символа null. Если значение Identity неизвестно, в отклик Identity следует включать поле нулевого размера. В откликах Identity **недопустимо** завершать поле null-символом. В любом случае размер поля Type-Data определяется из значения поля Length в пакете Request/Response.

Параметры защиты (параграф 7.2) показаны в таблице.

Механизм аутентификации	Нет
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	-
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.2. Notification

Описание
Необязательный тип Notification используется для передачи отображаемых сообщений партнеру от проверяющей стороны. Проверяющая сторона **может** передать партнеру Notification Request в любой момент, когда нет незавершенных запросов, до окончания метода аутентификации EAP. Партнер **должен** ответить на запрос Notification пакетом Notification Response, если спецификация метода аутентификации EAP не запрещает использовать сообщения Notification. В любом случае **недопустимо** передавать пакет Nak Response в ответ на запрос Notification. Отметим, что по умолчанию максимальный размер Notification Request составляет 1020 октетов, что оставляет не более 1015 октетов для выводимого пользователю сообщения.

Метод EAP **может** запрещать передачу сообщений Notification при работе этого метода. В таких случаях партнер **должен** отбрасывать запросы Notification без уведомления с момента, когда на начальный запрос для данного типа был передан отклик того же типа.

Проверяемой стороне **следует** отображать текст сообщения на пользовательской консоли или заносить в системный журнал при невозможности отображения. Тип Notification предназначен для передачи подтверждаемых уведомлений императивного характера, а не для индикации ошибок и поэтому не меняет состояния партнера. Примером использования уведомлений может служить ситуация с передачей предупреждения об грядущем окончании срока действия пароля, близком к нулю номере OTP, неудачной попытке аутентификации и т. п. Во многих случаях не следует требовать передачи сообщения Notification.

Type
2

Type-Data
Поле Type-Data в запросе содержит непустое отображаемое сообщение из символов ISO 10646 в кодировке UTF-8 [RFC2279]. Размер сообщения определяется значением поля Length в пакете Request. **Недопустимо** завершать сообщение null-символом. В ответ на запрос с полем типа, имеющим значение 2 (Notification) **должен** передаваться отклик. Поле Type-Data в отклике является пустым. Отклик следует передавать незамедлительно (независимо от отображения или записи сообщения в системный журнал).

Параметры защиты (см. Параграф 7.2) показаны в таблице.

Механизм аутентификации	Нет
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	-
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.3. Nak

5.3.1. Обычный тип Nak

Описание
Традиционный тип Nak относится только к откликам. Этот тип устанавливается в откликах на запросы, когда желаемый тип аутентификации недоступен. Типы аутентификации могут иметь значение 4 и выше. Пакет Response содержит значение одного или нескольких типов аутентификации, желаемых партнером. Тип 0 используется для

индикации отсутствия предлагаемых типов аутентификации, поэтому проверяющей стороне **не следует** передавать другой запрос после получения отклика Nak, содержащего нулевое значение.

Поскольку традиционный тип Nak применим только к откликам и имеет очень ограниченную функциональность, этот тип **недопустимо** использовать для индикации ошибок общего типа или согласования специфических параметров конкретного метода EAP.

Code

2 для Response.

Identifier

Поле Identifier занимает 1 октет и служит для сопоставления откликов с запросами. Значение поля Identifier в обычном отклике Nak **должно** соответствовать значению Identifier в пакете Request, на который передается отклик.

Length

>=6

Type

3

Type-Data

Когда проверяемая сторона получает запрос для неподходящего типа аутентификации (4-253, 255) или запрос для типа 254 при отсутствии поддержки расширенных типов, **должен** передаваться пакет Nak Response (тип 3). Поле Type-Data отклика Nak (тип 3) **должно** содержать один или множество октетов (по одному октету на тип), показывающих желаемые типы аутентификации, или 0 для индикации отсутствия предложений. Партнер, поддерживающий расширенные типы при получении запроса для неподходящего типа аутентификации (4-253, 255), **может** включить в отклик Nak (тип 3) значение 254 для индикации желания использовать расширенный тип. Если проверяющая сторона может принять это предложение, она будет отвечать на него сообщением Expanded Type Request (тип 254).

Параметры защиты (параграф 7.2) показаны в таблице.

Механизм аутентификации	Нет
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	-
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.3.2. Expanded Nak

Описание

Расширенный тип Nak применим только к откликам. Этот тип **должен** передаваться только в откликах на запросы типа 254 (Expanded Type), когда тип аутентификации не приемлем. Expanded Nak Type использует формат Expanded Type, а отклик содержит один или множество типов аутентификации, желательных для проверяемой стороны (все в формате Expanded Type). Нулевое значение служит для индикации отсутствия предложений. Общий формат расширенного типа описан в параграфе 5.7. Поскольку тип Expanded Nak пригоден только для откликов и имеет очень ограниченную функциональность, его **недопустимо** использовать для индикации ошибок общего плана типа передачи сообщений об ошибках или согласования специфических параметров конкретного метода EAP.

Code

2 для Response.

Identifier

Поле Identifier занимает 1 октет и служит для сопоставления откликов с запросами. Значение поля Identifier в отклике Expanded Nak **должно** соответствовать значению Identifier в пакете Request, на который передается отклик.

Length

>=20

Type

254

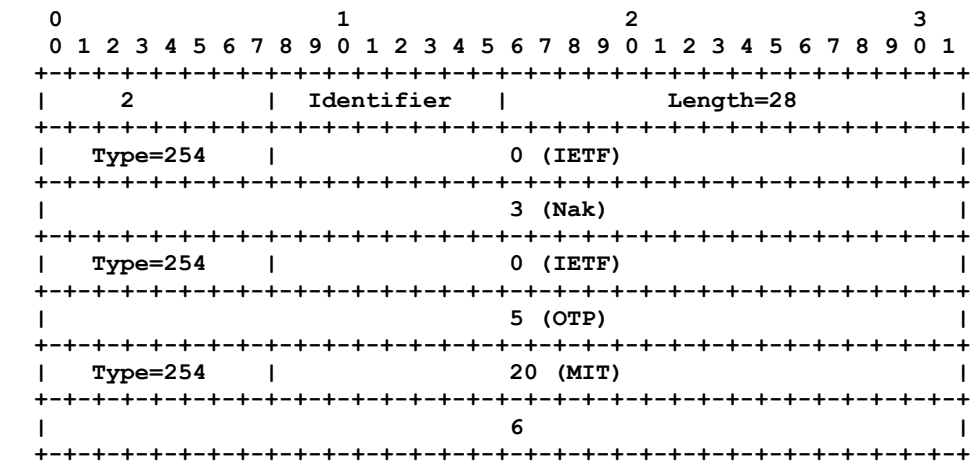
Vendor-Id

0 (IETF)

Vendor-Type

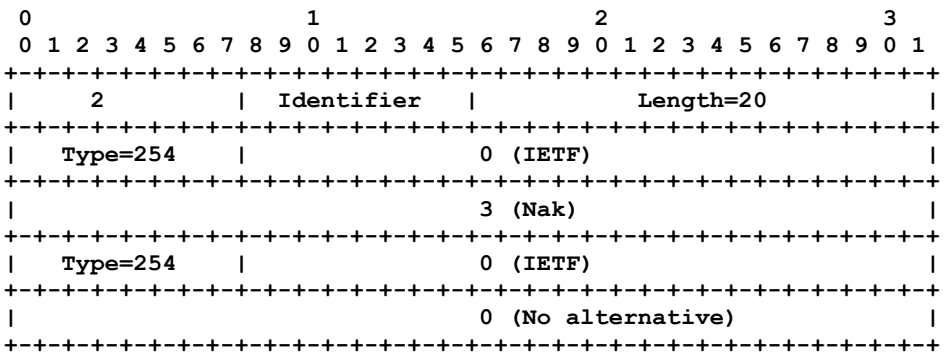
3 (Nak)

Vendor-Data



Тип Expanded Nak передается только в том случае, когда запрос содержит расширенный тип (254), как определено в параграфе 5.7. Поле Vendor-Data отклика[1] Nak **должно** содержать один или множество типов аутентификации (4 и выше) в расширенном формате (8 октетов на тип) или 0 (также в расширенном формате) для индикации отсутствия предложений. Желаемые типы аутентификации могут включать комбинацию типов Vendor-Specific и IETF. Например, расширенный отклик Nak, показывающий предпочтение для OTP (тип 5) и MIT (Vendor-Id=20) расширенного типа 6 будет иметь вид, приведенный на рисунке.

Отклик Expanded Nak, показывающий отсутствие желаемых типов, приведен на рисунке.



Параметры защиты (см. Параграф 7.2) показаны в таблице.

Механизм аутентификации	Нет
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	-
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.4. MD5-Challenge

Описание

Тип MD5-Challenge аналогичен протоколу PPP CHAP [RFC1994] (с MD5 в качестве алгоритма). Запрос содержит сообщение с «вызовом» партнеру. В ответ на запрос **должен** передаваться отклик, который **может** иметь тип 4 (MD5-Challenge), 3 (Nak) или 254 (Expanded Nak). Отклик Nak показывает желаемые партнером типы аутентификации. Реализации проверяемой стороны и сервера **должны** поддерживать механизм MD5-Challenge. Проверяющая сторона, которая работает только в проходном режиме, **должна** разрешать обмен информацией с внутренним сервером аутентификации, который поддерживает MD5-Challenge, хотя сама реализация проверяющей стороны EAP не обязана поддерживать MD5-Challenge. Однако, если проверяющая сторона может быть настроена на аутентификацию партнеров локально (например, не работать в проходном режиме), требование поддержки механизма MD5-Challenge становится актуальным.

Отметим, что использование поля Identifier для типа MD5-Challenge отличается от описанного в [RFC1994]. EAP позволяет повторять передачу запросов MD5-Challenge, тогда как в [RFC1994] сказано, что оба поля Identifier и Challenge **должны** изменяться при каждой передаче Challenge (эквивалент пакета с запросом MD5-Challenge в CHAP[2]).

Примечание. [RFC1994] трактует разделяемый секрет, как строку октетов и не задает способы ввода этой строки в систему (или управляется пользователем). Реализация EAP MD5-Challenge **может** поддерживать ввод парольных фраз, содержащих отличные от ASCII символы. Инструкции по обработке ввода и кодированию в октеты приведены в разделе 5.

Type

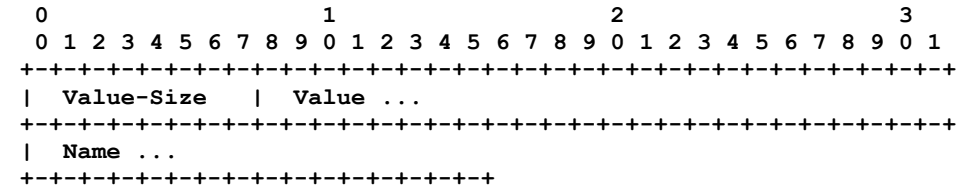
4

[1]Расширенного отклика. П*рим. Перев.*

[2]Challenge Handshake Authentication Protocol.

Type-Data

Содержимое поля Type-Data кратко описано ниже. Информацию по использованию этих полей можно найти в спецификации протокола PPP CHAP [RFC1994].



Параметры защиты (параграф 7.2) показаны в таблице.

Механизм аутентификации	Пароль или общий ключ
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	Нет
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.5. Одноразовый пароль (OTP)

Описание

Система одноразовых паролей (OTP[1]) определена в документах «A One-Time Password System[2]» [RFC2289] и «OTP Extended Responses[3]» [RFC2243]. Запрос содержит вызов OTP в формате, описанном в [RFC2289]. В ответ на запрос **должен** передаваться отклик, типа 5 (OTP), 3 (Nak) или 254 (Expanded Nak). Отклик Nak показывает желаемые партнером типы аутентификации. Метод EAP OTP предназначен для использования только в системе с одноразовыми паролями и его **недопустимо** применять для передачи паролей в открытом виде.

Type

5

Type-Data

Поле Type-Data содержит в запросе «вызов» OTP, как отображаемое сообщение. В откликах это поле используется для 6 слов из словаря OTP [RFC2289]. **Недопустимо** завершать сообщения null-символом. Размер поля определяется из значения поля Length в пакете Request/Reply.

Примечание. [RFC2289] не задает способ ввода пароля пользователем и преобразования пароля в октеты. Реализации EAP OTP **могут** поддерживать пароли с отличными от ASCII символами. Инструкции по обработке ввода и преобразования в октеты содержатся в разделе 5.

Параметры защиты (параграф 7.2) показаны в таблице.

Механизм аутентификации	Одноразовый пароль
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Да
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	Нет
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

5.6. Маркерные карты (GTC)

Описание

Тип GTC[4] определен для использования с различными реализациями маркерных карточек, которые требуют пользовательского ввода. Запрос содержит отображаемое сообщение, а отклик - информацию Token Card, требуемую для аутентификации. Обычно эта информация считывается пользователем с устройства для работы с картами и вводится в форме текста ASCII. В ответ на запрос **должен** передаваться отклик. В отклике **должен** указываться тип 6 (GTC), 3 (Nak) или 254 (Expanded Nak). Отклик Nak показывает желаемые партнером типы аутентификации. Метод EAP GTC предназначен для использования с картами, поддерживающими аутентификацию «запрос - отклик», и его **недопустимо** использовать с открытыми паролями в отсутствие защищенного туннеля с аутентификацией сервера.

[1]One-Time Password.
[2]Система с однократным паролем.
[3]Расширенные отклики OTP.
[4]Generic Token Card - маркерные карты базового типа.

Type
6

Type-Data
Поле Type-Data в запросах содержит непустое отображаемое сообщение (размер определяется значением поля Length в пакете Request). **Недопустимо** завершать сообщение null-символом. В ответ на запрос **должен** передаваться отклик типа 6 (GTC). Отклик содержит данные из карты, нужные для аутентификации.

Реализации EAP GTC **могут** поддерживать отклики с символами, отличными от ASCII. Инструкции по обработке ввода и преобразованию его в октеты приведены в разделе 5.

Параметры защиты (параграф 7.2) показаны в таблице.

Механизм аутентификации	Аппаратный маркер
Согласование шифронабора	Нет
Взаимная аутентификация	Нет
Защита целостности	Нет
Защита от повторов	Нет
Конфиденциальность	Нет
Создание ключей	Нет
Стойкость ключей	-
Устойчивость к атакам по словарю	Нет
Быстрый повтор соединения	Нет
Криптографическая привязка	-
Независимость сессий	-
Фрагментация	Нет
Связывание каналов	Нет

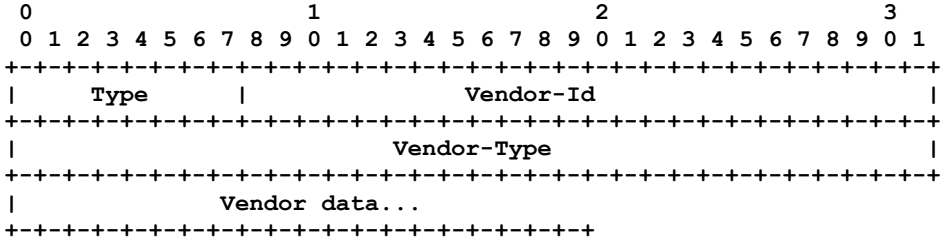
5.7. Расширенные типы

Описание
Поскольку большинство применений EAP связано с фирменной продукцией, введен отдельный тип, позволяющий производителям поддерживать свои расширенные типы, не подходящие для общего применения.

Expanded Type служит также для расширения глобального пространства типов методов за пределы исходных 255 значений. Значение Vendor-Id = 0 отображает исходные 255 возможных типов в пространство из 2^{32}-1 возможных типов (тип 0 используется только в отклике Nak для индикации отсутствия предложений).

Поддерживающие атрибут Expanded реализации **должны** трактовать типы EAP со значением меньше 256 одинаково, независимо от формы их представления - один октет или 32-битовое значение Vendor-Type в Expanded Type с Vendor-Id = 0. Партнеры, не поддерживающие Expanded Type, **должны** передать отклик Nak, как описано в параграфе 5.3.1 и согласовать более подходящий метод аутентификации.

Формат и описание полей Expanded Type показаны ниже. Поля передаются слева направо.



Type
254 для Expanded Type

Vendor-Id
Трехоктетное значение Vendor-Id представляет код SMI Network Management Private Enterprise Code для производителя с использованием сетевого порядка байтов (как выделено IANA). Нулевое значение Vendor-Id зарезервировано для использования IETF с целью расширения глобального пространства типов EAP.

Vendor-Type
Четырехоктетное поле Vendor-Type представляет фирменный тип метода от указанного производителя.

Если Vendor-Id = 0, поле Vendor-Type является расширением и надмножеством существующего пространства типов EAP. Первые 256 типов зарезервированы для совместимости с однооктетными типами EAP, которые уже определены или могут быть определены в будущем. Таким образом, типы EAP от 0 до 255 семантически одинаковы при их указании в однооктетных полях EAP Type или в полях Vendor-Type при Vendor-Id = 0. Из этого правила есть одно исключение - обычные и расширенные пакеты Nak относятся к одному типу, но должны трактоваться по-разному, поскольку имеют различные форматы.

Vendor-Data
Поле Vendor-Data определяется производителем. При Vendor-Id = 0 поле Vendor-Data будет использоваться для доставки содержимого методов EAP с определенными IETF типами.

5.8. Experimental

Описание
Тип Experimental не имеет фиксированного формата и содержимого. Этот тип предназначен для экспериментов с новыми типами EAP и тестов. При использовании этого типа нет никаких гарантий интероперабельности, как отмечено в [RFC3692].

Type
255

Type-Data
Не определено.

6. Взаимодействие с IANA

В этом разделе приведено руководство для агентства IANA[1] в части регистрации значений, связанных с протоколом EAP, в соответствии с BCP 26 [RFC2434].

В EAP имеется два пространства имен, требующих регистрации - коды пакетов и типы методов.

EAP не является протоколом общего назначения и **не следует** выделять значений для целей, не связанных с аутентификацией.

Термины «пространство имен» (name space), «выделенное значение» (assigned value), «регистрация» (registration) трактуются здесь в соответствии с BCP 26.

При выделении значений используются определенные в BCP 26 правила и процедуры: «для частного применения» (Private Use), «обслуживание в порядке очереди» (First Come First Served), «экспертиза» (Expert Review), «требуется спецификация» (Specification Required), «с согласия IETF» (IETF Consensus), «стандартизация» (Standards Action).

Для регистрационных запросов, где следует консультироваться с указанными экспертами, ответственность за выбор таких экспертов ложится на руководителя направления IESG. Смысл заключается в том, что любое выделение значений должно сопровождаться публикацией RFC. Но для выделения значений до публикации RFC может использоваться заключение указанного эксперта, который может одобрить выделение значений, когда станет ясно, что RFC будет публиковаться. Эксперт будет направлять запрос в рассылку рабочей группы EAP (или ее преемника, заданного руководителем направления) для обзора и комментариев, включая Internet-Draft. До истечения 30 дней эксперту следует принять или отвергнуть регистрационный запрос и опубликовать свое решение в рассылке EAP WG[2] или ее преемника, а также проинформировать IANA. Отказ должен быть мотивирован и по возможности в него следует включать конкретные предложения по изменению запроса для того, чтобы он был удовлетворен.

6.1. Коды пакетов

Коды пакетов (Packet Codes) имеют диапазон от 1 до 255, из которого уже выделены значения 1 - 4. Поскольку коды оказывают существенное влияние на интероперабельность, для выделения нового кода требуется стандартизация (Standards Action). Начинать выделение новых кодов следует со значения 5.

6.2. Типы методов

Исходное пространство типов методов EAP имеет диапазон от 1 до 255 и является наиболее дефицитным ресурсом EAP, поэтому выделять новые значения следует осторожно. Типы методов 1 - 45 уже распределены, при этом значение 20 доступно для переопределения. Типы 20 и 46 - 191 могут быть выделены с одобрения указанного эксперта при наличии спецификации (Specification Required).

Выделение блоков типов (более одного типа для данной цели) требует согласия IETF (IETF Consensus). Значения типов методов EAP от 192 до 253 зарезервированы и для их распределения требуется стандартизация.

Тип 254 выделен для Expanded Type. Если Vendor-Id = 0, значение Expanded Type служит для специфических целей данной реализации EAP, когда интероперабельность не имеет значения. При использовании с Vendor-Id = 0 тип метода 254 может также служить для поддержки расширенного пространства типов методов. Значения типов в диапазоне от 256 до 4294967295 могут распределяться после того, как будут распределены все типы 1 - 191, с одобрения указанного эксперта при наличии спецификации.

Тип 255 выделен для экспериментов типа тестирования новых методов EAP перед выделением постоянного типа.

7. Вопросы безопасности

В этом разделе описана базовая модель угроз, а также параметры защиты методов EAP, позволяющие смягчить эти угрозы.

Предполагается, что базовая модель угроз и параметры защиты будут применяться для определения требований к методам EAP при использовании в конкретных средах. Пример такого анализа требований имеется в стандарте [IEEE-802.11i-req]. Раздел описания параметров защиты является обязательным в спецификации метода EAP и требования для некоторых методов могут просто совпадать.

7.1. Модель угроз

Протокол EAP был разработан для использования с PPP [RFC1661], а позднее его адаптировали для проводных сетей IEEE 802 [IEEE-802] в стандарте [IEEE-802.1X]. Впоследствии EAP было предложено использовать в беспроводных ЛВС и Internet. Во всех случаях применения протокола атакующий может получить доступ к каналу, по которому передаются пакеты EAP. Например, атаки на телефонную инфраструктуру описаны в работе [DECEPTION].

Атакующий с доступом к каналу, может организовать множество атак, включая перечисленные ниже.

[1] Попытка раскрытия идентификационных данных пользователей путем перлюстрации трафика.

[2] Попытка изменения или подмены пакетов EAP.

[3] Атаки на отказ служб за счет подмены индикации нижележащего уровня или пакетов Success/Failure, повторного использования пакетов EAP или генерации пакетов с перекрывающимися идентификаторами.

[4] Попытка раскрытия паролей с помощью атак по словарю для собранных в канале данных.

[5] Попытка убедить проверяемый узел присоединиться к сети без доверия путем организации MITM-атаки.

[6] Попытка нарушения процесса согласования EAP для принуждения к выбору более слабого метода аутентификации.

[1]Internet Assigned Numbers Authority.
[2]Working Group – рабочая группа. П*рим. Перев.*

[7] Попытка восстановления ключей при использовании недостаточно стойких способов создания ключей в методах EAP.

[8] Попытка воспользоваться слабостью шифронабора после завершения транзакции EAP.

[9] Попытка снизить уровень стойкости при согласовании шифронабора, используемого для аутентификации EAP.

[10] Предоставление некорректной информации партнеру и/или серверу EAP от имени «проверяющей стороны» за счет использования сторонних механизмов (например, через AAA или протокол нижележащего уровня). К таким атакам относятся также «подмена» проверяющей стороны или предоставление противоречивой информации партнеру или серверу EAP.

В зависимости от нижележащего уровня такие атаки могут требовать доступа к каналу на физическом уровне. При использовании EAP в беспроводных сетях пакеты EAP могут пересылаться проверяющими узлами (например, для предварительной аутентификации) так, что атакующему не нужно находиться в области покрытия проверяющей стороны для организации атаки на проверяемые узлы. При использовании EAP в Internet атаки могут осуществляться на значительно большем удалении.

7.2. Параметры защиты

Для четкой формулировки защиты, обеспечиваемой методом EAP, спецификации методов EAP **должны** включать раздел Security Claims (Параметры защиты), содержащий перечисленные ниже описания.

[a] **Механизм** - указание технологии аутентификации, сертификатов, общих ключей, паролей, карт и т. п.

[b] **Параметры защиты** - параметры защиты, обеспечиваемые методом, с использованием терминов, определенных в параграфе 7.2.1 - взаимная аутентификация, защита целостности, предотвращение повторного использования, конфиденциальность, создание ключей, стойкость к атакам по словарю, быстрое повторное соединение, криптографическое связывание. В разделе «Параметры защиты» спецификации метода EAP **следует** включать описания заявленных параметров. Это может обеспечиваться путем включения обоснования в приложение или указания ссылки на такие обоснования.

[c] **Стойкость ключей** - если метод создает ключи, **должна** быть оценена их стойкость. Эта оценка позволяет потенциальным пользователям метода определить достаточна ли стойкость ключей для планируемых приложений.

Эффективную стойкость ключа **следует** оценивать числом битов следующим образом - если эффективная стойкость ключа составляет N битов, лучшие современные методы восстановления ключей (с вероятностью, которая не пренебрежимо мала) требуют в среднем усилий, сравнимых с выполнением 2^{N-1} операций типового блочного шифрования[1]. Данные о стойкости ключа **следует** сопроводить кратким обоснованием полученного числа битов. В это объяснение **следует** включать параметры, требуемые для достижения заявленной стойкости ключа, основанные на текущем понимании алгоритмов.

Стойкость ключей зависит от используемого для их создания метода. Например, для ключей, созданных на основе общего секрета (пароль или долгосрочный секрет), а также, возможно, некоторой открытой информации (типа nonce), эффективная стойкость ограничена стойкостью долгосрочного секрета (предполагается, что процедура создания ключей проста в плане расчетов). В качестве другого примера отметим, что при использовании алгоритмов с открытым ключом стойкость симметричного ключа зависит от стойкости использованного открытого ключа.

[d] **Описание иерархии ключей**. Методы EAP, создающие ключи, **должны** включать ссылку на спецификацию иерархии ключей или описывать создание ключей MSK и EMSK.

[e] **Наличие уязвимостей**. В дополнение к описываемым параметрам защиты спецификация **должна** указывать, какие из перечисленных в параграфе 7.2.1 параметров защиты **не** включены.

7.2.1. Терминология, связанная с параметрами защиты для методов EAP

В этом параграфе даны определения терминов, используемых при описании средств защиты методов EAP.

Protected ciphersuite negotiation - защищенное согласование шифронабора

Способность метода EAP согласовывать шифронабор, используемый для защиты транзакции EAP, а также защиты целостности согласования. Это не относится к возможности согласования шифронабора, используемого для защиты данных.

Mutual authentication - взаимная аутентификация

Метод EAP, в котором с использованием взаимной блокировки проверяющая сторона выполняет аутентификацию партнера, а тот, в свою очередь, аутентифицирует ее. Два независимых однонаправленных метода аутентификации, используемых в противоположных направлениях не обеспечивают взаимной аутентификации в соответствии с этим определением.

Integrity protection - защита целостности

Обеспечение аутентификации источника данных и защита от несанкционированного изменения пакетов EAP (включая запросы и отклики EAP). При поддержке такой защиты спецификация метода **должна** описывать пакеты и поля EAP, которые будут защищены.

Replay protection - защита от повторного использования пакетов

Защита от повторного использования метода EAP или его сообщений, включая индикацию успешного завершения или отказа.

[1]Понятия «сравнимых усилий» и «типового блочного шифрования» сложно определить точно, однако здесь достаточно разумных предположений. Дополнительную информацию можно найти в работе [SILVERMAN].

Confidentiality - конфиденциальность

Шифрование сообщений EAP, включая запросы и отклики EAP, индикацию успешного завершения или отказа. При поддержке такой защиты **должна** также обеспечиваться защита целостности (см. Параграф 7.3).

Key derivation - создание ключей из чего-либо

Способность метода EAP создавать экспортируемый ключевой материал типа MSK и EMSK. MSK служит лишь для создания ключей, но не для защиты транзакции EAP и последующих данных. EMSK пока служат резервом.

Key strength - стойкость ключа

Если эффективная стойкость ключа составляет N битов, лучшие современные методы восстановления ключей (с вероятностью, которая не пренебрежимо мала) требуют в среднем усилий, сравнимых с выполнением 2^{N-1} операций типового блочного шифрования.

Dictionary attack resistance - устойчивость к атакам по словарю

При использовании парольной аутентификации пароли в общем случае выбираются из небольшого (по сравнению с набором N-битовых ключей) набора, что ведет к возможности организации атак по словарю. Можно говорить о стойкости метода к атакам по словарю, если при использовании пароля в качестве секрета данный метод не позволяет организовать результативную атаку с использованием паролей, имеющихся в словаре атакующего.

Fast reconnect - быстрое повторное соединение

Способность организовать новую защищенную связь при наличии другой защищенной связи или обновить существующую связь более эффективно или за меньшее число периодов кругового обхода.

Cryptographic binding - криптографическая связка

Демонстрация серверу EAP того, что один объект действует в качестве проверяемой стороны EAP для всех методов, выполняемых в рамках туннельного метода. Связка **может** также означать, что сервер EAP демонстрирует партнеру один объект, который действует в качестве сервера EAP для всех методов, выполняемых в рамках туннельного метода. При корректном выполнении связка снижает уязвимость к MITM-атакам.

Session independence - независимость сеансов

Демонстрация того, что пассивные (типа захвата данных транзакции EAP) или активные (включая компрометацию MSK или EMSK) не приводят к компрометации последующих или предшествующих ключей MSK или EMSK.

Fragmentation - фрагментация

Способность метода EAP поддерживать фрагментацию и сборку пакетов. Как отмечено в параграфе 3.1, методам EAP следует поддерживать фрагментацию и сборку, если размер пакетов EAP превышает минимальное значение MTU (1020 октетов).

Channel binding - связывание каналов

Свойство метода EAP, позволяющее сравнить данные, передаваемые по каналу с защитой целостности (например, аутентификаторы конечных тичек), со значениями, передаваемыми с использованием иных механизмов (без применения данного канала) типа AAA или протокола нижележащего уровня.

Примечание. Приведенный список не является исчерпывающим. Могут включаться и другие свойства типа дополнительной защиты от атак на службы.

7.3. Защита Identity

Обмен Identity является в транзакции EAP необязательным. Поэтому можно опустить этот обмен целиком или использовать предлагаемый применяемым методом обмен идентификационными данными после организации защищенного канала.

Однако при поддержке роуминга в соответствии с [RFC2607] может потребоваться нахождение подходящего внутреннего сервера аутентификации до выполнения аутентификационной транзакции. Связанная с областью часть NAI[1] [RFC2486] обычно включается в отклик EAP Identity, чтобы разрешить маршрутизацию аутентификационного обмена на подходящий внутренний сервер. Поэтому, хотя связанная с партнером часть NAI может быть опущена в отклике EAP Identity при наличии прокси или трансляторов, связанная с областью часть может требоваться.

Возможно, что идентификационные данные в отклике Identity будут отличаться от данных, аутентифицированных методом EAP. Такое отличие может быть преднамеренным для сокрытия идентификационной информации. Методу EAP при решении вопроса о предоставлении доступа **следует** использовать аутентифицированное отождествление.

7.4. MITM-атаки

При туннелировании EAP с использованием протокола, опускающего аутентификацию партнера, возникает потенциальная уязвимость к MITM-атакам, более подробно описанная в работах [BINDING] и [MITM].

Как было отмечено в параграфе 2.1, EAP не позволяет использовать нетуннелированые последовательности методов аутентификации. Когда последовательность методов EAP разрешена, партнер может не иметь уверенности в том, что один объект действует в качестве проверяющей стороны во всех методах EAP данной последовательности. Например, проверяющая сторона может завершить метод EAP, а потом передать следующий метод (в последовательности) другому объекту без согласия партнера и даже не информируя его. Аналогично, проверяющая сторона может не иметь подтверждения того, что во всех методах EAP данной последовательности проверяется один партнер.

Туннелирование EAP с использованием другого протокола открывает возможность для атак с подставным проверяющим узлом, туннелирующим EAP легитимному серверу. Когда протокол туннелирования используется для создания ключей, но не требует аутентификации партнера, атакующий, который убедил легитимного партнера соединиться с ним, сможет туннелировать пакеты EAP легитимному серверу с успешным завершением аутентификации и получением ключа. Это позволяет атакующему организовать MITM-атаку, получая доступ в сеть, а также возможность расшифровывать трафик между легитимным партнером и сервером.

Для ослабления таких атак возможен ряд мер, перечисленных ниже.

[1]Network Access Identifier - аутентификатор доступа в сеть.

[a] Требование взаимной аутентификации в механизмах туннелирования EAP.

[b] Требование криптографического связывания между протоколом туннелирования EAP и туннелируемыми методами EAP. При поддержке криптографической связки требуется также механизм для защиты от атак на снижение уровня, позволяющих обойти такую связку. Дополнительную информацию о криптографических связках можно найти в работе [BINDING].

[c] Ограничение методов EAP, которые разрешено использовать без защиты, на основе политики проверяющей стороны и ее партнера.

[d] Отказ от использования туннелей при доступности одного стойкого метода.

7.5. Атаки с изменением пакетов

Хотя методы EAP могут поддерживать аутентификацию источника данных, защиту целостности и предотвращение повторного использования на уровне пакетов, такая поддержка не обеспечивается на уровне EAP.

Поскольку Поле Identifier занимает 1 октет, значение этого поля легко угадать, что дает атакующему возможность инжектировать или повторно использовать пакеты EAP. Атакующий может также изменять заголовки EAP (поля Code, Identifier, Length, Type) в пакетах, где нет защиты заголовков. Это может вести к отбрасыванию пакетов или некорректной их интерпретации.

Для защиты пакетов EAP от изменения, подмены или повторного использования рекомендуется использовать методы, поддерживающие защищенное согласование шифронабора, взаимную аутентификацию, защищенное создание ключей, а также защиту целостности и предотвращение повторного использования пакетов. Определения этих параметров защиты приведены в параграфе 7.2.1.

Для обеспечения защиты могут применяться специфические для метода проверки целостности сообщения (MIC). Если метод EAP поддерживает MIC на уровне пакетов, проверяющие узлы в проходном режиме, серверы аутентификации и проверяемые узлы **должны** проверять MIC. Информацию о непригодных MIC **следует** протоколировать. Трактовка непригодных MIC (критическая или некритическая ошибка) определяется спецификацией метода EAP.

Для методов, обеспечивающих защиту целостности пакетов EAP, **рекомендуется** включать в расчет контрольной суммы все поля заголовка EAP (Code, Identifier, Length, Type, Type-Data).

Поскольку сообщения EAP типов Identity, Notification и Nak не включают своего MIC, может оказаться желательным покрытие MIC метода EAP содержащейся в сообщении информации, а также заголовка каждого сообщения EAP.

Для обеспечения защиты EAP можно также инкапсулировать в защищенный канал, созданный протоколами типа ISAKMP [RFC2408], как делается в [IKEv2] или в TLS [RFC2246]. Однако, как было отмечено в параграфе 7.4, туннелирование EAP может приводить к уязвимости для MITM-атак.

Существующие методы EAP определяют проверки целостности сообщений (MIC), охватывающие более одного пакета EAP. Например, EAP-TLS [RFC2716] определяет MIC через запись TLS, которая может быть разделена на множество фрагментов, в сообщении FINISHED контрольная сумма MIC учитывает предыдущие сообщения. В случаях покрытия MIC для множества пакетов EAP негативный результат проверки MIC обычно трактуется как критическая ошибка.

В EAP-TLS [RFC2716] негативный результат проверки MIC считается критической ошибкой в соответствии со спецификацией TLS [RFC2246]. Однако возможна разработка методов EAP, которые будут поддерживать MIC на уровне пакетов и реагировать на негативный результат проверки целостности простым отбрасыванием пакетов.

В этом документе при описании обработки сообщений EAP предполагается, что проверка MIC на уровне пакетов (если она используется) выполняется до передачи каких-либо откликов и смены состояния хоста, принявшего пакет.

7.6. Атаки по словарю

Парольные механизмы аутентификации типа EAP-MD5, MS-CHAPv1 [RFC2433] и Kerberos V [RFC1510] известны уязвимостями к атакам по словарю. Уязвимости MS-CHAPv1 описаны в [PPTPv1], MS-CHAPv2 - в [PPTPv2], Kerberos - в [KRBATTACK], [KRBLIM] и [KERB4WEAK].

Для защиты от атак по словарю рекомендуется использовать методы аутентификации, устойчивые к таким атакам (см. определение в параграфе 7.2.1).

Если известно, что используемый алгоритм аутентификации уязвим к атакам по словарю, транзакцию можно туннелировать в защищенный канал. Однако туннелирование EAP может приводить к уязвимости для MITM-атак (см. параграф 7.4) и поэтому предпочтительно использовать методы, устойчивые к атакам по словарю.

7.7. Подключение к сети без доверия

В методах EAP, использующих одностороннюю аутентификацию (например, EAP-MD5), партнер не проверяет другую сторону, что делает этого партнера уязвимым для атак с использованием подставных проверяющих узлов. Для устранения этой уязвимости служат методы, поддерживающие взаимную аутентификацию (см. определение в параграфе 7.2.1).

В EAP не требуется выполнения аутентификации в полнодуплексном режиме или использования одного протокола для обоих направлений. Использование своего протокола для каждого направления является совершенно нормальной ситуацией и будет зависеть от согласованного протокола. Однако в общем случае выполнение одной взаимной аутентификации более предпочтительно, нежели проведение двух односторонних проверок (для каждого направления). Это связано с тем, что процедуры односторонней аутентификации, которые не связаны криптографически так, что указывается их принадлежность к одной сессии, могут быть объектом MITM-атак, как описано в параграфе 7.4.

7.8. Атаки на согласование

В таких атаках злоумышленник пытается принудить проверяющую сторону и ее партнера к согласованию менее защищенного метода EAP. Протокол EAP не обеспечивает защиты для откликов Nak, хотя методы могут включать отклики Nak в свои проверки MIC.

На проверяющей стороне (или в связи с ней) не предполагается, что отдельный именованный партнер будет поддерживать выбор методов. Это делает партнера уязвимым к атакам, в результате которых согласуется использование наименее защищенного метода из числа доступных. Поэтому **следует** указывать ровно один метод данного именованного партнера. Если партнеру потребуются другие методы аутентификации при изменившихся обстоятельствах, **следует** использовать иные свидетельства, для каждого из которых предлагается единственный метод аутентификации.

7.9. Поведение при отказе в аутентификации

Взаимодействие EAP с нижележащим уровнем типа PPP и IEEE 802 сильно зависит от реализации.

Например, при отказе в процессе аутентификации некоторые реализации PPP не разрывают соединение, ограничивая вместо этого трафик на сетевом уровне некоторым фильтруемым подмножеством, что дает партнеру возможность предложить обновление секретов или отправить сетевому администратору почтовое сообщение о возникших проблемах. Подобно этому, в случаях, когда аутентификация не прошла и доступ к контролируемому порту не может быть предоставлен, в [IEEE-802.1X] может разрешаться ограниченный трафик через контролируемый порт.

В EAP не предусматривается попыток повтора аутентификации. Однако в PPP машина состояний LCP может заново согласовать протокол аутентификации в любой момент, что позволяет повторить попытку. Аналогично, в IEEE 802.1X проверяемая (Supplicant) или проверяющая (Authenticator) сторона могут повторить аутентификацию в любой момент. **Рекомендуется** не сбрасывать значения счетчиков неудачных попыток, пока аутентификация не завершится успешно или в результате отказа канала.

7.10. Создание ключей

Для сервера EAP и его партнера возможна взаимная аутентификация и создание ключей. Для обеспечения ключевого материала, который будет впоследствии использоваться согласованным шифром, поддерживающий создание ключей метод EAP **должен** экспортировать основной ключ сеанса (MSK) размером не менее 64 октетов и расширенный основной ключ сеанса (EMSK) размером не менее 64 октетов. Методы EAP, создающие ключи, **должны** поддерживать взаимную аутентификацию между сервером EAP и его партнером.

Ключи MSK и EMSK **недопустимо** использовать напрямую для защиты данных - их размер достаточен для создания ключа AAA, который впоследствии служит для создания временных сеансовых ключей (TSK[1]), используемых с выбранным шифронабором. Каждый шифр отвечает за спецификацию создания ключей TSK по ключу AAA.

Ключ AAA создается из материала, экспортируемого методом EAP (MSK и EMSK). Процедура создания ключа выполняется на сервере AAA. Во многих протоколах, использующих EAP, ключи AAA и MSK эквивалентны, но возможны и более сложные механизмы (см. [KEYFRAME]).

Методам EAP **следует** поддерживать «свежесть» ключей MSK и EMSK даже в тех случаях, когда одна из сторон может не иметь высококачественного генератора случайных чисел. **Рекомендуется** каждой стороне передавать значение nonce размером не менее 128 битов, используемое для создания ключей MSK и EMSK.

Методы EAP экспортируют ключи MSK и EMSK, но не TSK, чтобы обеспечить независимость методов EAP от шифра и среды. Ключевой материал, экспортируемый методами EAP, **должен** быть независимым от шифронабора, согласованного для защиты данных.

В зависимости от протокола нижележащего уровня методы EAP могут работать до или после согласования шифронабора, поэтому выбранный шифр может быть не известен методу EAP. За счет предоставления ключевого материала пригодного для любого шифра методы EAP могут использоваться с широким спектром шифров и сред.

Для обеспечения независимости от алгоритмов методам EAP, создающим ключи, **следует** поддерживать (и документировать) защищенное согласование шифра для защиты транзакции EAP между сервером и партнером. Этот набор не совпадает с набором, который проверяющая сторона и партнер согласуют для защиты данных.

Стойкость ключей TSK, используемых для защиты данных, в конечном итоге зависит от стойкости ключей, созданных методом EAP. Если этот метод не может обеспечивать достаточно стойкий ключевой материал, ключи TSK могут взломаны методом тупого перебора (Brute force). Для поддержки систем, требующих стойких ключей, поддерживающим генерацию ключей методам EAP **следует** обеспечивать возможность генерации ключей MSK и EMSK с эффективной стойкостью не менее 128 битов.

Методы, поддерживающие генерацию ключей, **должны** демонстрировать криптографическое разделение ветвей MSK и EMSK в иерархии ключей EAP. Атакующий, который получил ключ MSK или EMSK, **не должен** получить возможность восстановления других данных с существенно меньшей затратой усилий, чем при тупом переборе, без нарушения фундаментальных криптографических допущений (таких, как необратимость односторонней функции).

Не перекрывающиеся подстроки MSK **должны** быть криптографически отделены одна от другой, как определено в параграфе 7.2.1. Т. е. знание одной подстроки **не должно** помогать в раскрытии других подстрок без нарушения фундаментальных криптографических допущений. Это требование обусловлено тем, что некоторые шифры создают ключи TSK простым разбиением ключа AAA на части подходящего размера. Подобно этому, не перекрывающиеся подстроки EMSK **должны** быть криптографически отделены одна от другой, как и подстроки MSK.

Ключи EMSK зарезервированы для использования в будущем и **должны** сохраняться на сервере и проверяемой стороне EAP, где ключ был создан, эти ключи **недопустимо** передавать или использовать совместно с другими, а также применять для генерации других ключей[2].

[1]Transient Session Key.
[2]Это ограничение будет смягчено в новых документах, определяющих использование EMSK.

Поскольку EAP не поддерживает явного согласования срока действия ключей, проверяющие стороны, серверы и партнеры EAP **должны** быть готовы к ситуациям, когда одна из сторон сбрасывает состояние ключа, который остается пригодным для другой стороны.

В данной спецификации не содержится подробного руководства по созданию методами EAP ключей MSK и EMSK, генерации AAA-Key из MSK и/или EMSK и генерации TSK из AAA-Key.

Разработка и проверка алгоритмов генерации ключей сложна, поэтому методам EAP **следует** пользоваться хорошо известными и проверенными механизмами создания ключей (например, указанными в спецификациях IKE [RFC2409] или TLS [RFC2246]) вместо разработки новых. Методам EAP также **следует** использовать хорошо известные и проверенные механизмы генерации ключей MSK и EMSK. Дополнительные сведения о генерации ключей EAP приведены в работе [KEYFRAME].

7.11. Слабость шифров

Если после начальной аутентификации EAP пакеты данных передаются без аутентификации, защиты целостности и предотвращения повторного использования на уровне отдельных пакетов, атакующий, у которого есть доступ к среде, может вставлять пакеты, менять биты в существующих пакетах, повторно использовать пакеты и даже полностью захватить сессию. Без обеспечения конфиденциальности на уровне пакетов можно «перлюстрировать» пакеты данных.

Для защиты от изменения данных, подмены и перлюстрации рекомендуется использовать методы EAP, поддерживающие взаимную аутентификацию и создание ключей (как определено в параграфе 7.2.1) в комбинации с нижележащим уровнем, который обеспечивает на уровне отдельных пакетов защиту конфиденциальности и целостности, а также предотвращение повторного использования пакетов.

В дополнение к этому нижележащему уровню, который поддерживает согласование шифров, следует понимать, что EAP сам по себе не обеспечивает защиты целостности такого согласования. Поэтому для предотвращения атак с целью снижения криптографического уровня клиентов, реализующих согласование шифронабора на нижележащем уровне, **следует** защищать от атак с целью снижения криптографического уровня.

Такая защита может быть реализована за счет предоставления пользователям возможности выбора подходящих шифронаборов в политике безопасности или за счет **возможности** аутентификации согласования шифров с использованием ключевого материала, полученного от аутентификации EAP и использования алгоритма MIC, согласованного ранее партнерами нижележащего уровня.

7.12. Канальный уровень

Существует ряд вопросов надежности и безопасности индикации нижележащего уровня в PPP, ЛВС IEEE 802 и беспроводных ЛВС IEEE 802.11:

[a] **PPP**. Индикация канального уровня типа LCP-Terminate (индикация отказа в канале) и NCP (индикация успешного создания канала) не аутентифицируются и не защищаются в плане целостности. Следовательно, атакующий с доступом к каналу может использовать обманную индикацию.

[b] **IEEE 802**. Кадры IEEE 802.1X EAPOL-Start и EAPOL-Logoff не аутентифицируются и целостность их не защищена. Следовательно, атакующий с доступом к каналу может использовать обманную индикацию.

[c] **IEEE 802.11**. В IEEE 802.11 индикация канального уровня включает кадры Disassociate и Deauthenticate (индикация отказов в канале), а также первое сообщение 4-этапного согласования (успешная организация канала). Для этих сообщений не обеспечивается аутентификация и защита целостности и хотя они не являются пересылаемыми, атакующий может использовать обманную индикацию, находясь в зоне доступа.

В IEEE 802.11 кадры данных IEEE 802.1X могут передаваться как кадры класса 3 с индивидуальной адресацией, и поэтому такие кадры можно пересылать. Это ведет к тому, что кадры EAPOL-Start и EAPOL-Logoff могут быть подменены аутентифицированным атакующим при включенном режиме предварительной аутентификации, несмотря на использование аутентификации и защиты целостности.

В IEEE 802.11 индикация «падения канала» является ненадежной индикацией наличия проблем в канале, поскольку уровень сигнала может меняться сам по себе и может находится под влиянием интерференции радиоволн, создаваемой атакующим. Для предотвращения ненужных сбросов разумно подавлять такую индикацию вместо ее передачи непосредственно в EAP. Поскольку EAP поддерживает повторную передачу пакетов, этого достаточно для устойчивости к временной потере связи.

7.13. Разделение проверяющей стороны и внутреннего сервера

Для сервера EAP и партнера возможна взаимная аутентификация и создание ключа AAA для шифра, который будет служить для защиты последующего трафика. Это не составляет проблемы для проверяемой стороны, поскольку партнер и клиент EAP находятся на одной машине. Все, что требуется от клиента - это создание ключа AAA из ключей MSK и EMSK, экспортируемых методом EAP и последующая передача сеансового ключа TSK модулю шифра.

Однако при размещении проверяющей стороны и сервера аутентификации в разных местах возникает ряд вопросов.

[a] Аутентификация может происходить между сервером и партнером, но не между проверяющей стороной и партнером. Это означает, что партнер не может проверить идентификационные данные проверяющей стороны, используя только EAP.

[b] Как обсуждалось в [RFC3579], проверяющая сторона зависит от протокола AAA в плане получения информации о результатах аутентификационной транзакции и не видит инкапсулированный пакет EAP (если он присутствует) для определения результата. На практике это означает, что протокол AAA, используемый для обмена между проверяющей стороной и сервером, **должен** поддерживать аутентификацию, защиту целостности и предотвращение повторного использования на уровне отдельных пакетов.

[c] После завершения транзакции EAP при включенных функциях защиты на нижележащем уровне (таких, как конфиденциальность, целостность и предотвращение повторного использования на уровне пакетов) протоколу

защищенной связи **следует** работать между проверяющей стороной и партнером для обеспечения взаимной аутентификации сторон, гарантии жизненности сеансовых ключей, предоставления защищенного согласования шифра и возможностей для последующей передачи данных, а также синхронизации использования ключа.

[d] Ключ AAA созданный на основе MSK и/или EMSK, согласованный между сервером аутентификации и партнером, **может** быть передан проверяющей стороне. Поэтому требуется механизм передачи ключа AAA от сервера проверяющей стороне, которой этот ключ нужен. Спецификация механизмов создания, транспортировки и «заворачивания[1]» ключа AAA-key выходит за пределы этого документа. Информация о создании AAA-Key имеется в документе [KEYFRAME].

7.14. Открытые пароли

Данная спецификация не определяет механизм аутентификации открытых (не зашифрованных) паролей. Это сделано преднамеренно. Использование открытых паролей позволит атакующим перехватывать их при наличии доступа к каналу, через который передаются пакеты EAP.

Поскольку инкапсулирующие EAP протоколы (типа RADIUS [RFC3579]) могут не обеспечивать конфиденциальности, пакеты EAP могут быть перехвачены при передаче информации через Internet.

В результате открытые пароли невозможно безопасно использовать в EAP за исключением случаев инкапсуляции в защищенный туннель с аутентификацией сервера. Некоторые из этих рисков относятся и к методам EAP, не обеспечивающим устойчивость к атакам по словарю, как определено в параграфе 7.2.1. Дополнительная информация содержится в параграфе 7.6.

7.15. Связывание каналов

Возможна передача непригодной информации от скомпрометированной или некачественно реализованной проверяющей стороны EAP серверу и/или партнеру EAP. Это позволяет проверяющей стороне представить себя другим проверяющим узлом или передавать некорректную информацию с использованием других протоколов (таких, как AAA или протокол нижележащего уровня).

При использовании EAP в проходном режиме проверяемый узел обычно не проверяет отождествление проходной проверяющей стороны, ограничиваясь проверкой доверия к ней со стороны сервера EAP. Это создает потенциальную уязвимость защиты.

В параграфе 4.3.7 документа [RFC3579] описано, как может быть обнаружена проверяющая сторона EAP в проходном режиме, которая, действуя в качестве клиента AAA, пытается представить себя другим проверяющим узлом (например, путем передач некорректных атрибутов NAS-Identifier [RFC2865], NAS-IP-Address [RFC2865] или NAS-IPv6-Address [RFC3162] через протокол AAA). Однако проходная проверяющая сторона, действуя в качестве клиента AAA, может может предоставлять корректную информацию серверу AAA и в то же время передавать искаженные данные партнеру EAP по протоколу нижележащего уровня.

Например, скомпрометированная проверяющая сторона может использовать значения Called-Station-Id или NAS-Identifier другой проверяющей стороны при обмене с партнером EAP по протоколу нижележащего уровня или проходной проверяющий узел, действующий в качестве клиента AAA, может передать некорректное значение Calling-Station-Id партнера [RFC2865][RFC3580] серверу AAA по протоколу AAA.

Для устранения этой уязвимости методы EAP могут поддерживать защищенный обмен параметрами канала типа идентификаторов конечных точек, включая (но не ограничиваясь) Called-Station-Id [RFC2865][RFC3580], Calling-Station-Id [RFC2865][RFC3580], NAS-Identifier [RFC2865], NAS-IP-Address [RFC2865], NAS-IPv6-Address [RFC3162].

Используя защищенный обмен можно сравнить параметры канала, предоставленные проверяющей стороной с использованием отдельного механизма, с параметрами, полученными от метода EAP. Обнаруженные расхождения **следует** фиксировать в системном журнале, **возможно** также выполнять иные действия (например, отказ в доступе).

7.16. Защищенная индикация результатов

В EAP пакеты Success и Failure не подтверждаются и целостность их не защищена. Индикация результатов повышает устойчивость к потере пакетов Success и Failure при работе EAP с протоколами нижележащего уровня, которые не поддерживают повторной передачи или синхронизации состояния аутентификации. В средах типа IEEE 802.11, которые поддерживают повтор передачи и синхронизацию состояния аутентификации за счет 4-этапного согласования, определенного в стандарте [IEEE-802.11i], дополнительная устойчивость обычно не дает заметного преимущества.

В зависимости от метода и обстоятельств аутентификация результатов может подменяться атакующим. Метод считается защищенным в части индикации результатов, если он поддерживает такую индикацию, наряду с защитой целостности и предотвращением повторов. Метод, поддерживающий защищенную индикацию результатов, **должен** показывать, защищена ли эта индикация на самом деле.

Защищенная индикация результатов не требуется для защиты от поддельных проверяющих узлов. В методах с взаимной аутентификацией требование аутентификации сервера со стороны партнера до восприятия последним пакета Success не позволяет атакующему прикинуться проверяющей стороной.

Однако атакующий может подделать пакет Success после того, как сервер был аутентифицирован со стороны партнера, но до аутентификации партнера сервером. Если партнер примет обманный пакет Success и попытается получить доступ в сеть до завершения его аутентификации сервером, по отношению к этому партнеру может быть организована атака на отказ служб. После такой атаки, если нижележащий уровень поддерживает индикацию отказов, проверяющая сторона может синхронизировать состояние с партнером, предоставляя индикацию отказа от нижележащего уровня. Дополнительная информация содержится в параграфе 7.12.

Когда сервер аутентифицировал партнера и передал пакет Success до проверки аутентификации этого партнера проверяющей стороной, может возникать простой, если партнер еще не аутентифицировал проверяющую сторону. При поддержке нижележащего уровня, проверяющая сторона, чувствуя отсутствие партнера, может высвободить ресурсы.

[1]Wrapping.

В поддерживающих индикацию результата методах партнер, аутентифицированный сервером, не считает эту аутентификацию успешной, пока не получит индикацию того, что сервер его аутентифицировал. Аналогично, сервер, аутентифицированный партнером, не будет считать аутентификацию успешной, пока не получит от партнера индикацию того, что последний аутентифицировал сервер.

Для предотвращения проблем с синхронизацией перед передачей индикации успеха отправителю желательно проверить наличие полномочий для предоставления доступа, хотя, как отмечено ниже, такое возможно не всегда.

Хотя индикация результата может позволять синхронизацию результата аутентификации между сервером и партнером, это не гарантирует того, что проверяющая сторона и партнер также будут синхронизированы, поскольку могут возникать вопросы с полномочиями и тайм-ауты. Например, сервер EAP может не знать о решении вопроса предоставления доступа AAA-прокси, сервер AAA может проверить полномочия только после успешной аутентификации и обнаружить отсутствие полномочий или сервер AAA может предоставить доступ, а проверяющая сторона не сможет его обеспечить по причине временной нехватки ресурсов. В таких случаях синхронизация может быть достигнута только с помощью индикации результатов нижележащего уровня.

Индикация успеха может быть явной или неявной. Например, если метод поддерживает сообщения об ошибках, неявная индикация успеха может быть определена, как получение конкретного сообщения без предшествующего сообщения об ошибке. Отказы обычно указываются явно. Как описано в параграфе 4.2, партнер отбрасывает без уведомления пакет Failure, полученный в тот момент, когда метод не разрешает явно делать это. Например, метод, поддерживающий свои сообщения об ошибках, может требовать от партнера получения сообщения об ошибке перед восприятием пакета Failure.

Аутентификация, защита целостности и предотвращение повторного использования на уровне пакетов для индикации результатов защищает от обманных пакетов. Поскольку защита индикации результатов требует использования ключа для аутентификации и защиты целостности на уровне пакетов, методы, поддерживающие такую защиту, **должны** также поддерживать создание ключей, взаимную аутентификацию, защиту целостности и предотвращение повторов.

Защищенная индикация результатов устраняет некоторые уязвимости к атакам на службы с использованием обманных пакетов Success и Failure. Обычно методы EAP могут обеспечивать защищенную индикацию результатов лишь в некоторых случаях. Например, ошибка может произойти до создания ключа, поэтому такую индикацию защитить не удастся. Возможны случаи, когда индикация результатов не может поддерживаться в обоих направлениях или синхронизация возможна не во всех режимах работы.

Например, в EAP-TLS [RFC2716] при согласовании аутентификации клиента сервер аутентифицирует партнера, но не получает защищенной индикации от него в части аутентификации тем сервера. Напротив, партнер аутентифицирует сервер и знает, что сервер аутентифицировал его. При согласовании восстановления сессии партнер аутентифицирует сервер, но не получает защищенной индикации того, что сервер аутентифицировал его. В этом режиме сервер аутентифицирует партнера и знает, что тот аутентифицировал его.

8. Благодарности

В этом протоколе много заимствований из документа AHA (Dave Carrel), а также из протокола PPP CHAP [RFC1994]. Значимые отклики прислали Yoshihiro Ohba из Toshiba America Research, Jari Arkko из Ericsson, Sachin Seth из Microsoft, Glen Zorn из Cisco Systems, Jesse Walker из Intel, Bill Arbaugh, Nick Petroni и Bryan Payne из университета штата Мэрилэнд, Steve Bellovin из AT&T Research, Paul Funk из Funk Software, Pasi Eronen из Nokia, Joseph Salowey из Cisco, Paul Congdon из HP, а также члены рабочей группы EAP.

Использование для методов EAP раздела «Параметры защиты», как требует параграф 7.2 и задано для каждого описанного здесь метода EAP, было предложено Glen Zorn в [EAP-EVAL].

9. Литература

9.1. Нормативные документы

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

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

[RFC2243] Metz, C., "OTP Extended Responses", RFC 2243, November 1997.

[RFC2279] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[RFC2289] Haller, N., Metz, C., Nesser, P. and M. Straw, "A One-Time Password System", RFC 2289, February 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

[IEEE-802] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture", IEEE Standard 802[1], 1990.

[IEEE-802.1X] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Port-Based Network Access Control", IEEE Standard 802.1X[1], September 2001.

9.2. Дополнительная литература

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1510] Kohl, J. and B. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[1]Этот стандарт доступен на сайте http://standards.ieee.org. *Прим. перев.*

[RFC1750] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750[1], December 1994.

[RFC2246] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. and P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2284] Blunk, L. and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

[RFC2486] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

[RFC2408] Maughan, D., Schneider, M. and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409[2], November 1998.

[RFC2433] Zorn, G. and S. Cobb, "Microsoft PPP CHAP Extensions", RFC 2433, October 1998.

[RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, June 1999.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2716] Aboba, B. and D. Simon, "PPP EAP TLS Authentication Protocol", RFC 2716, October 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960[3], October 2000.

[RFC3162] Aboba, B., Zorn, G. and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[RFC3580] Congdon, P., Aboba, B., Smith, A., Zorn, G. and J. Roese, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines", RFC 3580, September 2003.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.

[DECEPTION] Slatalla, M. and J. Quittner, "Masters of Deception", Harper-Collins, New York, 1995.

[KRBATTACK] Wu, T., "A Real-World Analysis of Kerberos Password Security", Proceedings of the 1999 ISOC Network and Distributed System Security Symposium, http://www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/wu.pdf.

[KRBLIM] Bellovin, S. and M. Merrit, "Limitations of the Kerberos authentication system", Proceedings of the 1991 Winter USENIX Conference, pp. 253-267, 1991.

[KERB4WEAK] Dole, B., Lodin, S. and E. Spafford, "Misplaced trust: Kerberos 4 session keys", Proceedings of the Internet Society Network and Distributed System Security Symposium, pp. 60-70, March 1997.

[PIC] Aboba, B., Krawczyk, H. and Y. Sheffer, "PIC, A Pre-IKE Credential Provisioning Protocol", Work in Progress, October 2002.

[IKEv2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", Work in Progress[4], January 2004.

[PPTPv1] Schneier, B. and Mudge, "Cryptanalysis of Microsoft's Point-to- Point Tunneling Protocol", Proceedings of the 5th ACM Conference on Communications and Computer Security, ACM Press, November 1998.

[IEEE-802.11] Institute of Electrical and Electronics Engineers, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Standard 802.11[5], 1999.

[SILVERMAN] Silverman, Robert D., "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths", RSA Laboratories Bulletin 13, April 2000 (Revised November 2001), http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html.

[KEYFRAME] Aboba, B., "EAP Key Management Framework", Work in Progress[6], October 2003.

[SASLPREP] Zeilenga, K., "SASLprep: Stringprep profile for user names and passwords", Work in Progress[7], March 2004.

[IEEE-802.11i] Institute of Electrical and Electronics Engineers, "Unapproved Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements -

[1]Документ заменен RFC 4086. *Прим. перев.*
[2]Этот документ заменен RFC 4306. *Прим. перев.*
[3]Этот документ заменен RFC 4960. *Прим. перев.*
[4]Эта работа завершена и опубликована в RFC 4306. *Прим. перев.*
[5]Этот стандарт доступен на сайте http://standards.ieee.org. *Прим. перев.*
[6]Эта работа завершена и опубликована в RFC 5247. *Прим. перев.*
[7]Эта работа завершена и опубликована в RFC 4013. *Прим. перев.*

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security", IEEE Draft 802.11i (work in progress)[1], 2003.

[DIAM-EAP] Eronen, P., Hiller, T. and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", Work in Progress[2], February 2004.

[EAP-EVAL] Zorn, G., "Specifying Security Claims for EAP Authentication Types", Work in Progress, October 2002.

[BINDING] Puthenkulam, J., "The Compound Authentication Binding Problem", Work in Progress, October 2003.

[MITM] Asokan, N., Niemi, V. and K. Nyberg, "Man-in-the-Middle in Tunneled Authentication Protocols", IACR ePrint Archive Report 2002/163, October 2002, <http://eprint.iacr.org/2002/163>.

[IEEE-802.11i-req] Stanley, D., "EAP Method Requirements for Wireless LANs", Work in Progress[3], February 2004.

[PPTPv2] Schneier, B. and Mudge, "Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)", CQRE 99, Springer-Verlag, 1999, pp. 192-203[4].

Приложения A. Отличия от RFC 2284

В этом разделе перечислены основные различия между [RFC2284] и данным документом. Мелкие отличия, включая стили, грамматику, опечатки и редакторские правки не упомянуты в списке.

- Раздел, посвященный терминологии (параграф 1.2) был расширен, определены концепции и даны более точные определения.
- Введены и рассмотрены в документе концепции взаимной аутентификации (Mutual Authentication), создания ключей (Key Derivation) и индикации результатов (Result Indication).
- В разделе 2 явно указано что в аутентификационном обмене EAP может происходить множество обменов пакетами Request и Response. Возможности использования этого подробно описаны в параграфе 2.1.
- В разделе 2 явно приведены некоторые требования к проверяющей стороне в проходном режиме.
- Добавлена модель мультиплексирования EAP (параграф 2.2) для иллюстрации типового применения EAP. Реализации не обязаны точно следовать этой модели, достаточно совместимого с ней поведения.
- Описана работа EAP с различными протоколами нижележащего уровня в дополнение к протоколу PPP, для которого разрабатывался EAP. Добавлен раздел 3, посвященный поведению нижележащего уровня.
- При описании взаимодействия запросов и откликов EAP (параграф 4.1) более точно описано поведение в случае получения дубликатов запроса и отбрасывание пакетов без уведомления.
- В параграфе 4.2 разъяснено, что пакеты Success и Failure не должны содержать дополнительных данных и расширены примечания для разработчиков. Добавлен параграф с требованиями по обработке пакетов Success и Failure.
- В разделе 5 описаны два новых типа EAP - Expanded (параграф 5.7), который используется для расширения пространства типов, и Experimental. В пространстве Expanded Type добавлен новый тип Expanded Nak (параграф 5.3.2). Приведены дополнительные разъяснения для большинства существующих типов. Добавлены описания параметров защиты для методов аутентификации.
- В параграфах 5, 5.1 и 5.2 добавлены требования к отображаемым полям по использованию символов ISO 10646 в кодировке UTF-8.
- В параграфе 5.1 сказано, что при наличии в поле Type-Data запроса Identity символа NUL отображается только часть сообщения до этого символа. RFC 2284 запрещает использование null-символов в поле Type-Data сообщений Identity. Это правило смягчает требования к запросам Identity и поле Type-Data в них может включать null-символы.
- В параграфе 5.5 добавлена поддержка откликов OTP Extended [RFC2243] в EAP OTP.
- Добавлен раздел «6. Взаимодействие с IANA» с правилами регистрации имен для EAP.
- Раздел «7. Вопросы безопасности» был существенно расширен и включает в настоящем документе более полное описание возможных угроз и других вопросов безопасности.
- В параграф 7.5 был добавлен текст, описывающий конкретное поведение методов EAP в плане проверки целостности. При наличии возможности желательно рассчитывать специфическое для метода значение MIC с покрытием всего пакета EAP, включая заголовок уровня EAP (поля Code, Identifier, Length) и заголовок уровня метода EAP (поля Type, Type-Data).
- В параграфе 7.14 описаны риски, связанные с использованием открытых паролей в EAP.
- В параграф 7.15 добавлен текст, посвященный детектированию подставных NAS.

Адреса авторов

Bernard Aboba

Microsoft Corporation

One Microsoft Way

[1]Этот стандарт завершен и доступен на сайте http://standards.ieee.org. *Прим. перев.*
[2]Эта работа завершена и опубликована в RFC 4072. *Прим. перев.*
[3]Эта работа завершена и опубликована в RFC 4017. *Прим. перев.*
[4]Эта статья доступна на сайте http://www.schneier.com/paper-pptpv2.html. *Прим. перев.*

Redmond, WA 98052

USA

Phone: +1 425 706 6605

Fax: +1 425 936 6605

EMail: bernarda@microsoft.com

Larry J. Blunk

Merit Network, Inc

4251 Plymouth Rd., Suite 2000

Ann Arbor, MI 48105-2785

USA

Phone: +1 734-647-9563

Fax: +1 734-647-3185

EMail: ljb@merit.edu

John R. Vollbrecht

Vollbrecht Consulting LLC

9682 Alice Hill Drive

Dexter, MI 48130

USA

EMail: jrv@umich.edu

James Carlson

Sun Microsystems, Inc

1 Network Drive

Burlington, MA 01803-2757

USA

Phone: +1 781 442 2084

Fax: +1 781 442 1677

EMail: james.d.carlson@sun.com

Henrik Levkowetz

ipUnplugged AB

Arenavagen 33

Stockholm S-121 28

SWEDEN

Phone: +46 708 32 16 08

EMail: henrik@levkowetz.com

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

Полное заявление авторских прав

Интеллектуальная собственность

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.