Обзор расширяемого протокола аутентификации и его методов

[1] *А.В. Никешин <alexn@ispras.ru>*
[1,2] *В.З. Шнитман <vzs@ispras.ru>*
[1] *Институт системного программирования РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.
[2] *Московский физико-технический институт,*
141700, Московская область, г. Долгопрудный, Институтский пер., 9

Аннотация. Данная статья представляет собой обзор расширяемого протокола аутентификации (Extensible Authentication Protocol, EAP), специфицированного комитетом Internet Engineering Task Force, IETF, и предоставляющего эффективный механизм встраивания в него различных методов аутентификации, а также обзор собственно методов аутентификации EAP, часть из которых была стандартизована в спецификациях IETF. Показано разнообразие механизмов, используемых для реализации сервиса аутентификации. Работа выполнялась при поддержке РФФИ, проект № 16-07-00603 «Верификация функций безопасности и оценка устойчивости к атакам реализаций протокола аутентификации EAP».

Ключевые слова: безопасность; аутентификация; контроль доступа; EAP, методы EAP

DOI: 10.15514/ISPRAS-2018-30(2)-7

Для цитирования: Никешин А.В., Шнитман В.З. Обзор расширяемого протокола аутентификации и его методов. Труды ИСП РАН, том 30, вып. 2, 2018 г., стр. 113-148. DOI: 10.15514/ISPRAS-2018-30(2)-7

1. Введение

Аутентификация является наиболее важным сервисом безопасности, поскольку все другие сервисы безопасности зависят от него. Этот сервис направлен против угрозы маскарада – угрозы, которая может сделать возможной реализацию множества угроз. Аутентификация дает гарантию подлинности, т.е. представляет собой средство получения уверенности в том, что люди или сущности являются именно теми, за кого или за что они себя выдают. Легитимный владелец некоторых идентификационных данных называется принципалом. Может потребоваться аутентифицировать принципалов различных физических видов, например, людей, части оборудования, или работающие в вычислительной системе приложения.

Существует огромное число различных методов аутентификации. Аутентификация связана со сценарием, в котором некоторая сторона (претендент) представила идентификационные данные (identity) принципала и заявляет о том, что является этим принципалом. Аутентификация позволяет некоторой другой стороне (верификатору) убедиться в том, что это заявление является легитимным. Аутентификация широко применяется в системах контроля доступа к сетям и ресурсам вычислительных систем. В этом контексте значительный интерес представляет расширяемый протокол аутентификации (Extensible Authentication Protocol, EAP), специфицированный IETF в RFC 3748 [1], обеспечивающий эффективный механизм встраивания в него различных методов аутентификации, а также собственно методы аутентификации EAP, часть из которых была стандартизована в спецификациях IETF. В данной статье в разделах 2 и 3 представлены обзоры протокола EAP и методов EAP, соответственно.

Работа выполнялась при поддержке РФФИ, проект № 16-07-00603 «Верификация функций безопасности и оценка устойчивости к атакам реализаций протокола аутентификации EAP».

2. Основные особенности EAP

Протокол EAP относится к протоколам аутентификации, поддерживающим расширения своей функциональности за счет добавления новых методов [1]. EAP обычно работает непосредственно на базе протоколов канального уровня типа PPP [2] или IEEE 802 [3], не требуя использования протокола IP [4]. EAP может применяться на выделенных и коммутируемых каналах, как в проводных, так и в беспроводных сетях. Данный протокол использует пошаговую схему обработки сообщений: новый запрос отправляется только после получения ответа на предыдущий. Поэтому данный протокол не предназначен для передачи больших объемов данных. EAP обеспечивает собственную поддержку повторной передачи сообщений и избавления от дубликатов, но она основана на гарантированном порядке доставки сообщений протоколом нижележащего уровня. Соответственно, обработка пакетов, доставленных с нарушением порядка, не поддерживается.

Архитектура EAP использует следующие роли для сетевых узлов:

- аутентификатор (authenticator) –сетевой узел, начинающий процесс аутентификации;
- партнер (peer) – сетевой узел, отвечающий на запросы аутентификатора;
- внутренний сервер аутентификации (backend authentication server) – выделенный сервер, предоставляющий услуги аутентификации (выполняет методы EAP) для аутентификатора;
- сервер EAP – объект, реализующий методы EAP; он либо размещается на сервере аутентификации, либо является частью

аутентификатора.

Одним из преимуществ архитектуры EAP является ее гибкость. Выбор конкретного механизма аутентификации происходит после того, как аутентифицирующая сторона (аутентификатор) получит дополнительную информацию от партнерского узла. Вместо необходимости каждый раз обновлять аутентификатор для поддержки нового метода аутентификации архитектура EAP позволяет использовать выделенный сервер, который поддерживает различные методы аутентификации, а сам аутентификатор просто пересылает сообщения от других узлов такому серверу.

Общая схема работы протокола выглядит следующим образом (рис. 1). Клиент (партнер) запрашивает доступ к некоторому ресурсу, обращаясь к системе доступа (аутентификатору). Аутентификатор передает запрос с данными клиента серверу EAP. Сервер EAP запрашивает дополнительные данные у клиента. Обмен сообщениями между клиентом и сервером EAP продолжается до тех пор, пока выбранный метод аутентификации не завершится успешно или с ошибкой. Аутентификатор на основании результата аутентификации, полученному от сервера EAP, принимает решение о предоставлении клиенту доступа к запрошенному ресурсу. Таким образом, аутентификатор исполняет роль посредника между клиентом и сервером EAP.

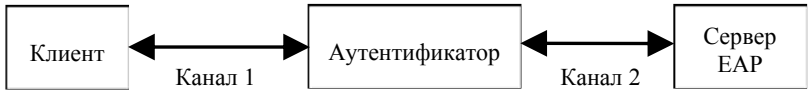


Рис. 1. Общая схема работы EAP
Fig. 1. General flow chart of EAP

Первый коммуникационный канал между клиентом и аутентификатором может быть как проводным, так и беспроводным. Довольно часто в роли клиента выступает мобильное устройство, а в роли аутентификатора – точка доступа. Пересылка пакетов EAP между клиентом и аутентификатором осуществляется посредством инкапсуляции пакетов EAP в протокол нижележащего уровня, например, в протокол PPP при использовании каналов точка-точка, или в протокол EAPOL (EAP over LAN) в сетях IEEE 802 [5]. Второй канал между аутентификатором и сервером EAP, как правило, является проводным, однако в некоторых случаях (например, в сценариях роуминга), между ними могут размещаться дополнительные объекты пересылки сообщений.

Указанная схема допускает различные вариации. Например, сам сервер EAP может располагаться как на аутентификаторе, так и на выделенном узле. Второй вариант позволяет упростить управление доступом к нескольким разделяемым ресурсам, что очень важно в целом ряде случаев. Многие сетевые устройства обычно имеют довольно ограниченные аппаратные ресурсы и, например, не могут хранить в памяти информацию о большом числе пользователей. Кроме того, число пользователей может регулярно меняться, и

возникает необходимость в единой базе данных. В такой системе от аутентификатора конкретного ресурса требуется только поддержка базовой функциональности протокола EAP. А реализация всех методов аутентификации и база данных с информацией обо всех пользователях и их правах доступа находятся в единой подсистеме – сервере аутентификации.

Пересылка пакетов EAP между внутренним сервером аутентификации и клиентом осуществляется посредством инкапсуляции пакетов EAP в протокол аутентификации, авторизации и учета (Authentication, Authorization, and Accounting, AAA), который выполняется между аутентификатором и внутренним сервером аутентификации. В качестве протокола AAA обычно применяются протоколы RADIUS [6] и Diameter [7]. При этом, однако, увеличивается объем сетевого трафика.

Возможна и комбинированная схема, в которой аутентификатор поддерживает некоторые методы аутентификации, а для других методов используется выделенный сервер. Также сервер EAP может не хранить данные для аутентификации клиентов и обращаться за ними к внешней базе данных. Таким образом, одним из важных свойств EAP является независимость от режима работы аутентификатора, т.е. любой метод EAP работает одинаково во всех аспектах независимо от того, работает ли аутентификатор в режиме ретрансляции или нет.

Другими важными свойствами протокола EAP являются независимость от среды, независимость от метода и независимость от набора шифров.

Протокол EAP первоначально разрабатывался для использования с протоколом точка-точка (Point-to-Point Protocol, PPP) [2]. Впоследствии его начали использовать для аутентификации доступа в проводных сетях IEEE 802 [5] и в беспроводных сетях IEEE-802.11 [8] и IEEE-802.16e [9]. Он применяется также в качестве одного из способов аутентификации в протоколе управления ключами в Интернет (Internet Key Exchange Protocol version 2, IKEv2) [10]. Таким образом, одной из целей создания EAP являлось обеспечение функционирования его методов поверх любого нижележащего уровня, т.е. методы EAP в процессе своего выполнения не должны пользоваться информацией конкретного нижележащего уровня, например, MAC-адресами.

Независимость от метода означает, что благодаря обеспечению режима ретрансляции аутентификатор может поддерживать любой метод, реализованный на партнере и сервере, а не только локально реализованные методы.

По существу, независимость от набора шифров обеспечивает независимость от среды. Поскольку наборы шифров разных нижележащих уровней отличаются друг от друга, для обеспечения независимости от среды требуется, чтобы экспортируемый ключевой материал имел достаточную длину и энтропию для работы с любым набором шифров.

При аутентификации по протоколу EAP могут одновременно использоваться разные среды передачи данных и, как следствие, EAP будет выполняться через

разные стеки коммуникационных протоколов. Ниже на рис. 2 показан пример стеков протоколов при использовании EAP поверх беспроводного канала (WLAN) между клиентом и аутентификатором (точка доступа) и проводного канала (LAN) между аутентификатором и сервером EAP. При этом, если аутентификатор работает в режиме ретрансляции, то ему не требуется поддерживать сами методы аутентификации, поэтому у него нет уровня методов EAP.

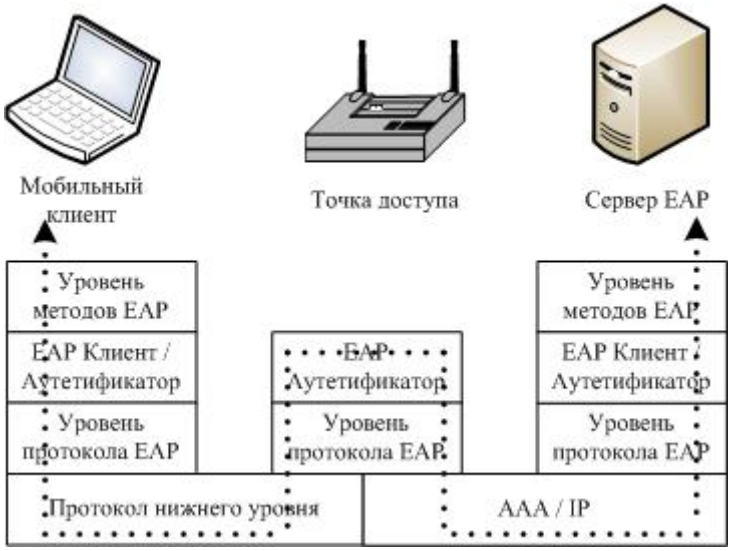


Рис. 2. Стеки протоколов EAP
Fig. 2. EAP Protocol Stacks.

Можно отметить следующие преимущества протокола.

- Протокол EAP может поддерживать множество механизмов аутентификации без предварительного согласования используемого метода.
- Устройства сетевого доступа не обязаны понимать каждый метод аутентификации и могут действовать как посредники для внутреннего сервера аутентификации. Аутентификатор, например, может поддерживать часть методов EAP или самостоятельно обеспечивать аутентификацию только локальных сетевых узлов, и, в то же время, работать в режиме посредника для внешних узлов и непонятных методов аутентификации.
- Разделение аутентификатора и внутреннего сервера аутентификации упрощает управление доступом к сети и политиками безопасности.

3. Классификация и обзор методов EAP

В настоящее время протокол EAP широко используется в самых разных сетевых средах. В IANA (Internet Assigned Numbers Authority) зарегистрировано несколько десятков расширений данного протокола, реализующих широкий набор методов аутентификации и использующих различные средства аутентификации (пароли, сертификаты, токены и др.) [11].

В данном обзоре рассматриваются только методы EAP, спецификации которых зафиксированы в документах IETF RFC. Другие методы EAP, указанные в [11], либо остались на уровне черновых документов IETF (ietf-drafts), не получив статуса IETF RFC, либо вообще не документированы.

Спецификация EAP [1] определяет четыре основных типа сообщений: *Request* (запрос), *Response* (ответ), *Success* (успех), *Failure* (неудача)

Сообщения *Request* и *Response* содержат однобайтовое поле *Type* предназначенное для согласования метода аутентификации. Несколько значений поля *Type* зарезервированы для служебных сообщений:

1 Identity

Используется для запроса идентификатора клиента. Спецификация требует использовать данный тип запроса исключительно для целей маршрутизации и выбора подходящего метода EAP. Как правило, это первое сообщение от аутентификатора клиенту.

2 Notification

Необязательный тип сообщений. Используется для передачи аутентификатором сообщений, выводимых на экран клиента. Может передаваться в любое время во время обмена EAP.

3 Nak (Response only)

Используется в ответных сообщениях партнера, когда предложенный аутентификатором метод EAP не поддерживается. Партнер может указать в ответном сообщении желаемые методы EAP.

254 Expanded Types

Используется для расширения множества используемых методов EAP (когда множество значений однобайтового поля *Type* исчерпано). Также может использоваться для согласования специальных методов EAP, не предназначенных для общего использования (например, методов, ориентированных на конкретное оборудование).

255 Experimental use

Данный тип сообщений используется для экспериментов и тестирования (например, разработчиками нового метода EAP). Спецификация никак не ограничивает формат таких сообщений.

Остальные значения поля *Type* определяют конкретный метод EAP. Таким образом, стандарт EAP определяет единый механизм встраивания новых методов EAP в общую архитектуру протокола.

На рис.3 представлен типовой обмен сообщениями EAP (предполагается, что аутентификатор работает в режиме ретрансляции сообщений, т.е. все сообщения запроса, за исключением начального запроса идентификационных данных, и сообщение Success/Failure посылаются сервером EAP, а все ответные сообщения от партнера пересылаются серверу EAP). В первом сообщении аутентификатор (A) запрашивает идентификатор партнера (клиента, C), который в ответном сообщении указывает свои идентификационные данные. После получения этого ответа сервер (S) выбирает метод EAP и посылает первое сообщение соответствующего типа.

Если клиент поддерживает и принимает предложенный сервером метод EAP, он отвечает соответствующим сообщением того же типа. В противном случае клиент посылает сообщение Nak, и сервер EAP либо выбирает другой метод, либо прекращает выполнение EAP сообщением Failure. Количество парных сообщений запрос/ответ, пересылаемых между сервером EAP и клиентом, определяется выбранным методом EAP. Последнее сообщение является индикацией успешного или неудачного результата аутентификации.

C←*A* :	EAP-Request / Identity
C→*S* :	EAP-Response / Identity (id)
C←*S* :	EAP-Request / EAP-method
C→*S* :	EAP-Response / EAP-method
C←*S* :	EAP-Request / EAP-method
C→*S* :	EAP-Response / EAP-method
...	
C←*S* :	EAP- Success/Failure

Рис. 3. Типовой обмен сообщениями EAP
Fig. 3. Typical Message Flow of an EAP Execution

Для методов EAP, обеспечивающих установление ключей, спецификация определяет несколько типов ключевого материала. Все методы EAP, обеспечивающие вычисление ключей, создают главный сеансовый ключ (Master Session Key, MSK) и расширенный главный сеансовый ключ (Extended Master Session Key, EMSK). Дополнительно могут создаваться временные ключи EAP (Transient EAP Keys, TEKs), которые используются для защиты текущего обмена EAP. Ключи MSK, EMSK предназначены для использования за рамками метода EAP (например, другими приложениями).

Ключ MSK экспортируется нижележащим уровням и может транспортироваться аутентификатору для последующего вычисления дополнительных ключей, обеспечивающих защиту целостности и шифрование канала передачи данных между клиентом и аутентификатором. Ключ EMSK остается на сервере и резервируется для будущего использования. Ключи MSK,

EMSK не должны использоваться непосредственно для защиты данных, а должны применяться только для вычисления других временных ключей.

Временные ключи EAP (TEKs) используются исключительно внутри конкретного метода EAP. Их количество, иерархию, вычисление и использование определяет спецификация данного метода. Например, метод EAP может использовать отдельные ключи для взаимной аутентификации и дополнительные ключи для создания криптографического канала. Для вычисления ключей обычно используется заранее распределенный ключ или методы Диффи-Хеллмана.

Представленные ниже методы можно разделить на четыре группы.

- *Методы, использующие схему «запрос-отклик» (challenge-response)*. Сервер отправляет клиенту запрос с некоторыми данными. Клиент на основе этих данных и собственных заранее полученных удостоверяющих данных (пароль, секретный ключ, токен и др.) формирует уникальный ответ, корректность которого позволяет серверу аутентифицировать клиента.
- *Методы, использующие заранее распределенный секретный ключ (shared secret key)*, на основе которого затем создается общий ключ безопасности (например, используя обмен Диффи-Хэллмана), уникальный для данного сеанса, который, в свою очередь, используется (непосредственно или через дополнительные криптографические ключи) для защиты сетевого обмена. Некоторые методы из других групп также можно отнести к данной группе (например, EAP-AKA использует как механизм challenge-response, так и симметричную криптографию).
- *Методы, использующие для целей аутентификации, как основного назначения методов EAP, механизмы создания защищенного канала*. Протоколы TLS и IKE предназначены, в первую очередь, для безопасной передачи данных [12], [10]. Основной функциональностью этих протоколов является создание защищенного канала между узлами сети, в процессе которого осуществляется взаимная аутентификация партнеров. Эту особенность и используют соответствующие методы EAP: начальная фаза создания защищенного канала используется исключительно для аутентификации партнеров и создания криптографических ключей, предусмотренных спецификацией EAP. Использование самого защищенного канала для передачи данных находится за рамками спецификации рассматриваемых методов EAP.
- *Туннельные методы, создающие сначала криптографически защищенный туннель, внутри которого используются другие методы аутентификации*. На туннельные методы не распространяется запрет (по причине безопасности) спецификации RFC 3748 на использование нескольких методов аутентификации в процессе одного выполнения

EAP. Туннельные методы предназначены для защиты от атак всех внутренних методов (в частности, позволяют использовать внутри туннеля небезопасные парольные методы аутентификации), а также полезны при необходимости многоуровневой аутентификации партнера (например, при последовательной аутентификации устройства, а затем – пользователя).

К сожалению, предложенная классификация не является очень строгой, поскольку, как уже было отмечено, некоторые методы можно отнести к разным группам.

3.1 Методы, использующие схему «запрос-отклик»

3.1.1 MD5-Challenge

Тип 4. RFC 3748.

Аналог протокола PPP CHAP (RFC1994) с алгоритмом MD5 [13]. Аутентификатор отправляет клиенту произвольный набор байтов. Клиент вычисляет из них контрольную сумму, используя заранее полученный секретный ключ и алгоритм MD5, и отправляет результат обратно. В настоящее время EAP-MD5 считается запрещенным: он не обеспечивает ни взаимную аутентификацию, ни вычисление ключей. Он обладает высокой уязвимостью к активным атакам прямого подбора и атакам по словарю.

3.1.2 One-Time Password (OTP)

Тип 5. RFC 3748.

Данный метод использует механизм одноразовых паролей (RFC2289) [14]. Аутентификатор отправляет клиенту некоторый набор байтов и дополнительные данные, используемые для выбора соответствующего алгоритма и дальнейших вычислений. Клиент, используя полученные данные и свой ключ безопасности, вычисляет контрольную сумму, применяя несколько раз заданную хэш-функцию.

3.1.3 Generic Token Card (GTC)

Тип 6. RFC 3748.

Данный метод использует механизм одноразовых паролей (RFC2289) [14]. Аутентификатор отправляет клиенту некоторый набор байтов и дополнительные данные, используемые для выбора соответствующего алгоритма и дальнейших вычислений. Клиент, используя полученные данные и свой ключ безопасности, вычисляет контрольную сумму, применяя несколько раз заданную хэш-функцию.

3.1.4 EAP-POTP (EAP Protected One-Time Password Protocol)

Тип 32. RFC 4793 [15].

Данный метод использует генераторы одноразовых паролей. Это расширенный вариант метода EAP-GTC (Generic Token Card) [1]. В качестве генератора

паролей может выступать как физическое устройство, так и программное обеспечение. Метод не зависит от применяемого генераторами паролей алгоритма. В отличие от EAP-GTC позволяет проводить взаимную аутентификацию и создавать криптографические ключи. Метод EAP-POTP не следует путать с EAP-OTP [1], который описывает конкретный алгоритм создания одноразовых паролей.

Базовый режим данного метода обеспечивает только аутентификацию клиента и обычно используется внутри защищенного канала. Расширенный режим применяется для взаимной аутентификации клиента и сервера, криптографической защиты сообщений и создания криптографических ключей. Также существует возможность быстрого возобновления сеанса (session resumption).

Спецификация метода определяет более десятка атрибутов, используемых в сообщениях для согласования различных параметров аутентификации: версия метода, параметры аутентификатора, алгоритм для генерации паролей, идентификатор клиента, идентификатор ключа генератора паролей (Token Key Identifier), криптографические алгоритмы и др.

Данный метод определяет пять криптографических ключей:

Ключ K_MAC используется для взаимной аутентификации партнеров и защиты целостности сообщений.

Ключ K_ENC используется для криптографической защиты некоторых данных.

Ключ SRK используется для быстрого возобновления сеанса.

Ключи MSK, EMSK предназначены для использования за рамками данного метода EAP (см. RFC 3748 [1]).

Обмен сообщениями при аутентификации клиента (рекомендуется использовать внутри защищенного канала). Как показано на рис. 4, клиент отправляет данные для аутентификации в блоке User Identifier TLV.

C←S : EAP-Request / Type=Identity
C→S : EAP-Response / Type=Identity

C←S : EAP-Request / Type=OTP-X
Version TLV: Highest=0,Lowest=0
OTP TLV: P=0,C=0,N=0,T=0,E=0,R=0
C→S : EAP-Response / Type=OTP-X
Version TLV: Highest=0
OTP TLV: P=0,C=0,N=0,T=0,E=0,R=0
Authentication Data=V1
User Identifier TLV: User Identifier=V2

C←S : EAP- Success

Рис. 4. Основной режим, односторонняя аутентификация
Fig. 4. Basic Mode, Unilateral Authentication

На рис.5 представлен обмен сообщениями при взаимной аутентификации.

C←S : EAP-Request / Type=Identity
C→S : EAP-Response / Type=Identity

C←S : EAP-Request / Type=OTP-X
Version TLV: Highest=0,Lowest=0
Server-Info TLV: N=0, Session Identifier=V1,
Server Identifier=V2, Nonce=V3
OTP TLV: P=1,C=0,N=0,T=0,E=0,R=0
Pepper Length=0, Iteration Count=V4
C→S : EAP-Response / Type=OTP-X
Version TLV: Highest=0
OTP TLV: P=1,C=0,N=0,T=0,E=0,R=0, Pepper Length=0,
Iteration Count=V4, Authentication Data=V5
User Identifier TLV: User Identifier=V6
Token Key Identifier TLV: Token Key Identifier=V7

C←S : EAP-Request/Type=OTP-X
Confirm TLV: C=0, Authentication Data=V8
Pepper Identifier=V9, Encrypted Pepper=V10
C→S : EAP-Response / Type=OTP-X
Confirm TLV: (no data)

C←S : EAP- Success

Рис. 5. Взаимная аутентификация без возобновления сеанса
Fig. 5. Mutual Authentication without Session Resumption

3.1.5 EAP-SIM (EAP GSM Subscriber Identity Modules)

Тип 18. RFC 4186 [16].

Метод использует параметры и алгоритмы SIM-карты для аутентификации и создания криптографических ключей. Данный метод основан на механизмах аутентификации GSM. GSM – стандарт мобильных сетей второго поколения. Алгоритмы A3/A8, используемые SIM-картой и GSM оператором, принимают 128-битное случайное число RAND и секретный ключ с SIM-карты Ki в качестве входных данных и выдают 32-битное хэш-значение SRES и 64-битный ключ Kc, используемые для аутентификации и шифрования данных [17].

В отличие от стандарта GSM, данный метод EAP-SIM не использует ключ Kc непосредственно для шифрования данных из-за его слабой криптографической стойкости. Вместо этого несколько значений RAND используются для генерации нескольких ключей Kc, которые затем объединяются для создания более сильных ключей безопасности. EAP-SIM обеспечивает взаимную аутентификацию партнеров, защиту целостности сообщений, криптографическую защиту некоторых данных, а также механизм быстрой повторной аутентификации.

Использование метода EAP-SIM требует наличия на стороне клиента специализированного устройства для работы с SIM-картами. При этом процесс

аутентификации проходит прозрачно для клиента, ему не требуется вводить какие-либо данные.

На рис.6 представлена процедура полной аутентификации:

C←S : EAP-Request / Identity
C→S : EAP-Response / Identity

C←S : EAP-Request / SIM/Start (AT_VERSION_LIST)
C→S : EAP-Response / SIM/Start
(AT_NONCE_MT, AT_SELECTED_VERSION)

C←S : EAP-Request / SIM/Challenge
(AT_RAND, AT_MAC)

Партнер выполняет алгоритмы GSM, проверяет
AT_MAC и вычисляет сеансовые ключи

C→S : EAP-Response / SIM/ Challenge (AT_MAC)

C←S : EAP- Success

Рис. 6. Полная процедура аутентификации EAP-SIM
Fig. 6. EAP-SIM full authentication procedure

Все параметры сеанса передаются в сообщениях в виде атрибутов. Спецификация определяет два десятка атрибутов, такие как версия метода EAP-SIM, запрос идентификатора клиента, значения RAND, контрольная сумма сообщения, параметры алгоритма шифрования и сами зашифрованные данные, информационные сообщения, сообщения об ошибке и др.

Сообщение EAP-Response/Identity обычно содержит IMSI (International Mobile Subscriber Identity) SIM-карты клиента [18]. Получив от клиента это сообщение, сервер EAP посылает партнеру пакет EAP-Request / SIM/Start, содержащий в атрибуте AT_VERSION_LIST список версий EAP-SIM, которые он поддерживает. Партнер в ответном сообщении указывает выбранную им версию протокола в атрибуте AT_SELECTED_VERSION, а также выбранное им случайное число в атрибуте AT_NONCE_MT.

Затем, получив от клиента сообщение EAP-Response/SIM/Start, сервер запрашивает несколько GSM триплетов (RAND, SRES, Kc), как правило, в центре аутентификации оператора сети. Из триплетов создаются ключи безопасности, а значения RAND отправляются клиенту в атрибуте AT_RAND в следующем сообщении, которое включает также код аутентификации в атрибуте AT_MAC.

Приняв это сообщение партнер выполняет алгоритмы GSM, вычисляет копию MAC и проверяет ее на совпадение со значением, присланным сервером, а также вычисляет сеансовые ключи. В случае несовпадения кодов MAC партнер посылает серверу сообщение об ошибке. При совпадении кодов MAC посылка следующего сообщения серверу означает, что партнер успешно аутентифицировал сервер и что обмен EAP соответствует локальной политике

партнера. В это сообщение включается код MAC, покрывающий содержимое пакета в конкатенации со значениями SRES партнера. Сервер EAP проверяет корректность MAC и посылает сообщение EAP-Success, подтверждающее успешное завершение аутентификации.

Механизм быстрой повторной аутентификации не задействует алгоритмы A3/A8 и инфраструктуру GSM, используя ключи, созданные во время полной аутентификации, что позволяет экономить вычислительные ресурсы.

3.1.6 EAP-AKA (EAP Method for 3rd Generation Authentication and Key Agreement)

Тип 23. RFC 4187 [19].

Метод использует параметры и алгоритмы SIM-карты для аутентификации и создания криптографических ключей. Данный метод основан на механизмах аутентификации и согласования ключей AKA, используемых в мобильных сетях третьего поколения (3G) UMTS и CDMA2000 [20],[21]. Механизмы AKA, основанные на алгоритмах с симметричными ключами, отличаются от используемых в сетях второго поколения GSM и основанном на них методе EAP-SIM (в частности, использованием более стойких ключей безопасности). AKA используется в SIM-картах типов USIM (UMTS Subscriber Identity Module) и (R)UIM ((Removable) User Identity Module).

Процесс аутентификации выглядит следующим образом (рис. 7). Сервер запрашивает идентификатор клиента, и затем, используя заранее распределенный секретных ключ и счетчик сообщений, создает криптографический набор, называемый вектором аутентификации (authentication vector): случайное значение RAND, хэш-значение AUTH (используется клиентом для аутентификации сервера), ожидаемое хэш-значение клиента XRES, два 128-битных ключа IK/CK. RAND и AUTH передаются клиенту. Клиент таким же образом создает свой криптографический набор, проверяет значение AUTH и отправляет серверу свое хэш-значение RES. Если взаимная аутентификация прошла успешно, ключи IK/CK используются для создания ключей безопасности данного сеанса.

C←S : EAP-Request/Identity

C→S : EAP-Response / Identity (включает NAI пользователя)

Сервер выполняет алгоритмы AKA,
генерирует RAND и AUTN.

C←S : EAP-Request / AKA-Challenge
(AT_RAND, AT_AUTN, AT_MAC)

Партнер выполняет алгоритмы AKA, проверяет
AUTN и MAC, вычисляет RES и сеансовые ключи.

C→S : EAP-Response / AKA-Challenge
(AT_RES, AT_MAC)

Сервер проверяет данные RES

и MAC и убеждается в их правильности.

C←S : EAP- Success

Рис. 7. Полная процедура аутентификации EAP-AKA
Fig. 7. EAP- AKA full authentication procedure.

Все параметры сеанса передаются в сообщениях в виде атрибутов. Спецификация определяет два десятка атрибутов, такие как запрос идентификатора клиента, значения RAND и AUTH, контрольная сумма сообщения, параметры алгоритма шифрования и сами зашифрованные данные, информационные сообщения, сообщения об ошибке и др.

Сообщение клиента EAP-Response/Identity обычно содержит IMSI (International Mobile Subscriber Identity) SIM-карты клиента для радиосетей или NAI (Network Access Identifier) для сервисов потоковых данных (IP multimedia service) [22].

Вектор аутентификации может быть получен сервером из центра аутентификации оператора сети.

Механизм быстрой повторной аутентификации не задействует алгоритмы AKA и инфраструктуру, используя ключи, созданные во время полной аутентификации, что позволяет экономить вычислительные ресурсы.

3.1.7 EAP-AKA' (Improved EAP-AKA)

Тип 50. RFC 5448 [23].

Данный метод является небольшой модификацией метода EAP-AKA. Он использует новый механизм создания криптографических ключей, связывая создаваемые внутри метода ключи и имя сети доступа (access network). Также заменена базовая хэш-функция: SHA-256 вместо SHA-1.

В спецификации RFC 5448 описан также способ противодействия атакам понижения уровня, которые может проводить злоумышленник (человек посередине) при согласовании оконечными точками методов EAP-AKA и EAP-AKA', чтобы вынудить их использовать менее стойкий метод, по сравнению с методом, которым они могут обе воспользоваться. В частности, для EAP-AKA определяется новый механизм, позволяющий оконечным точкам выяснить возможности друг друга. Таким образом, предполагается, что метод EAP-AKA' всегда является предпочтительным.

3.2 Методы с общим секретным ключом

3.2.1 EAP-PSK (A Pre-Shared Key EAP Method)

Тип 47. RFC 4764 [24].

Метод EAP, основанный на заранее распределенных ключах безопасности. Обеспечивает взаимную аутентификацию партнеров и создание сеансовых криптографических ключей.

При его разработке ставились следующие цели:

- простота настройки и использования: метод использует единственный криптографический алгоритм AES-128 и фиксированный формат сообщений (отсутствуют поля формата Тип-Длина-Значение / Type-Length-Value);
- широкое применение: метод может использоваться в любых типах сетей, включая беспроводные.
- Защищенность: метод разработан для использования в незащищенных сетях.
- Расширяемость: в процессе аутентификации создается защищенный канал, который может использоваться для расширения функциональности метода.

На рис.8 представлен стандартный обмен сообщениями.

C←*S* : EAP-Request / EAP-PSK (Flags||RAND_S||ID_S)
C→*S* : EAP-Response / EAP-PSK (Flags||RAND_S||RAND_P||MAC_P||ID_P)

C←*S* : EAP-Request / EAP-PSK (Flags||RAND_S||MAC_S||PCHANNEL_S_0)
C→*S* : EAP-Response / EAP-PSK (Flags||RAND_S||PCHANNEL_P_1)

Рис. 8. Стандартная аутентификация EAP-PSK
Fig. 8. EAP-PSK Standard Authentication

Все сообщения EAP-PSK включают некоторый вид заголовка, состоящего из поля Flags и 16-байтного случайного числа RAND_S, которое посылается сервером и служит в качестве идентификатора сеанса. В первом сообщении сервер посылает RAND_S и указывает свой идентификатор ID_S. Во втором сообщении партнер посылает свое 16-байтное случайное число RAND_P, свой идентификатор ID_P и аутентифицирует себя путем демонстрации своей возможности вычислить правильный код аутентификации сообщения MAC_P, который зависит от ключа аутентификации AK, идентификаторов партнера и сервера, а также от RAND_S и RAND_P.

Третье сообщение служит для обеспечения аутентификации сервера партнером, в котором сервер демонстрирует свою возможность вычислить правильный код аутентификации сообщения MAC_S, зависящий от ключа AK, идентификатора сервера и RAND_P, а также устанавливает защищенный канал (поле PCHANNEL_S_0), подтверждая, что он вычислил сеансовые ключи, и индицируя защищенный результат аутентификации. Четвертое сообщение, посылаемое партнером серверу, завершает установку защищенного канала (поле PCHANNEL_P_1), подтверждает вычисление сеансовых ключей на стороне партнера и обеспечивает индикацию защищенного результата аутентификации.

Спецификация позволяет при необходимости продолжить обмен сообщениями в рамках созданного защищенного канала.

На рис. 9 представлена иерархия ключей, используемых данным методом.

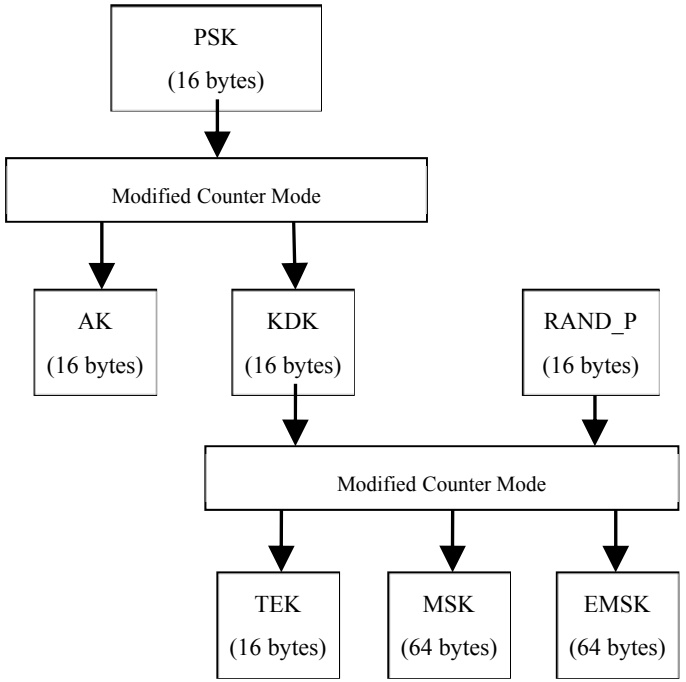


Рис. 9. Иерархия ключей EAP-PSK
Fig. 9. EAP-PSK Key Hierarchy

В EAP-PSK используется единственный криптографический примитив – блочный шифр AES-128. На его основе реализуется алгоритм вычисления кода аутентификации сообщений (MAC).

Ключ AK (Authentication Key) используется для взаимной аутентификации партнеров.

Ключ KDK (Key-Derivation Key) используется для создания сеансовых ключей (TEK, MSK, EMSK).

Ключ TEK (Transient EAP Key) используется для создания защищенного канала.

Ключи MSK, EMSK предназначены для использования за рамками данного метода EAP (например, другими приложениями) в соответствии с требованиями спецификации EAP (RFC 3748).

3.2.2 EAP-SAKE (EAP Method for Shared-secret Authentication and Key Establishment)

Тип 48. RFC 4763 [25].

Метод EAP, основанный на заранее распределенных ключах безопасности. В отличие от метода EAP-PSK, позволяет использовать различные

криптографические алгоритмы. Основан на протоколе взаимной аутентификации, предложенном Михиром Белуаром и Филиппом Рогвеем [26-27].

На рис.10 представлен стандартный обмен сообщениями.

C←S : EAP-Request/ SAKE/Challenge
(AT_RAND_S, AT_SERVERID)

C→S : EAP-Response / SAKE/Challenge
(AT_RAND_P, AT_PEERID, AT_SPI_P, AT_MIC_P)

C←S : EAP-Request / SAKE/Confirm
(AT_SPI_S, AT_ENCR_DATA, AT_MIC_S)

C→S : EAP-Response / SAKE/ Confirm
(AT_MIC_P)

C←S : EAP- Success

Рис. 10. Процедура аутентификации EAP-SAKE (с согласованием набора шифров)
Fig. 10. EAP-SAKE Authentication Procedure (with ciphersuite negotiation).

AT_MIC_P, AT_MIC_S – контрольные суммы сообщений, обеспечивающие целостность сообщений и взаимную аутентификацию партнеров.

AT_SPI_P, AT_SPI_S – атрибуты, предназначенные для согласования криптографического набора.

AT_ENCR_DATA – зашифрованные данные.

Атрибуты AT_SPI_P, AT_SPI_S, AT_ENCR_DATA являются необязательными и включаются, если требуется обеспечить конфиденциальность данных.

Спецификация определяет несколько необязательных дополнительных атрибутов, расширяющих функциональность метода.

На рис.11 представлена иерархия ключей, используемых данным методом.

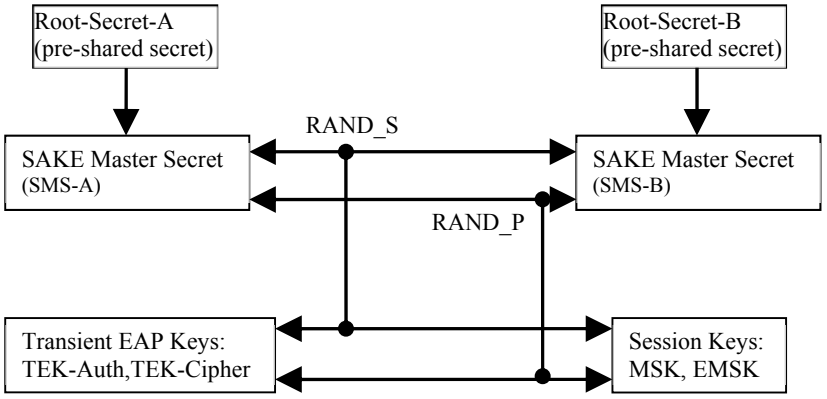


Рис. 11. Иерархия ключей метода EAP-SAKE
Fig. 11. EAP-SAKE Key Hierarchy

Общий ключ (pre-shared secret) делится на две части: Root-Secret-A и Root-Secret-B.

Ключи SMS-A, SMS-B используются для создания сеансовых ключей - TEK, MSK, EMSK.

Ключи TEK используются только внутри данного метода EAP для вычисления контрольных сумм и криптографической защиты.

Ключи MSK, EMSK предназначены для использования за рамками данного метода EAP (например, другими приложениями).

3.2.3 EAP-GPSK (EAP Generalized Pre-Shared Key Method)

Тип 51. RFC 5433 [28].

Простой метод EAP, основанный на заранее распределенных секретных ключах и обеспечивающий взаимную аутентификацию и создание криптографических ключей. К особенностям метода можно отнести простоту реализации, упрощенные криптографические вычисления на основе симметричных ключей, минимальное количество сообщений и поддержку нескольких криптографических алгоритмов (в спецификации метода определены два набора шифров). На рис. 12 представлен обмен сообщениями в случае успешной взаимной аутентификации участников.

C←S : EAP-Request/Identity
C→S : EAP-Response/Identity

C←S : EAP-Request
(ID_Server, RAND_Server, CSuite_List)
C→S : EAP-Response
SEC_SK(ID_Peer, ID_Server, RAND_Peer, RAND_Server,
CSuite_List, CSuite_Sel, [ENC_PK(PD_Payload_Block)])

C←S : EAP-Request
SEC_SK(RAND_Peer, RAND_Server, ID_Server, CSuite_Sel,
[ENC_PK(PD_Payload_Block)])
C→S : EAP-Response
SEC_SK([ENC_PK(PD_Payload_Block)])

C←S : EAP- Success

Рис. 12. Успешный обмен EAP-GPSK
Fig. 12. EAP-GPSK Successful Exchange

Во время выполнения EAP-GPSK партнер и сервер обмениваются неповторяющимися значениями (нонсами RAND_Peer и RAND_Server), которые используются вместе с заранее распределенным ключом для получения иерархии ключей EAP. Поэтому безопасность установления ключей (рис. 13) зависит от используемой функции вычисления ключей (key derivation function, KDF), и случайности указанных неповторяющихся значений.

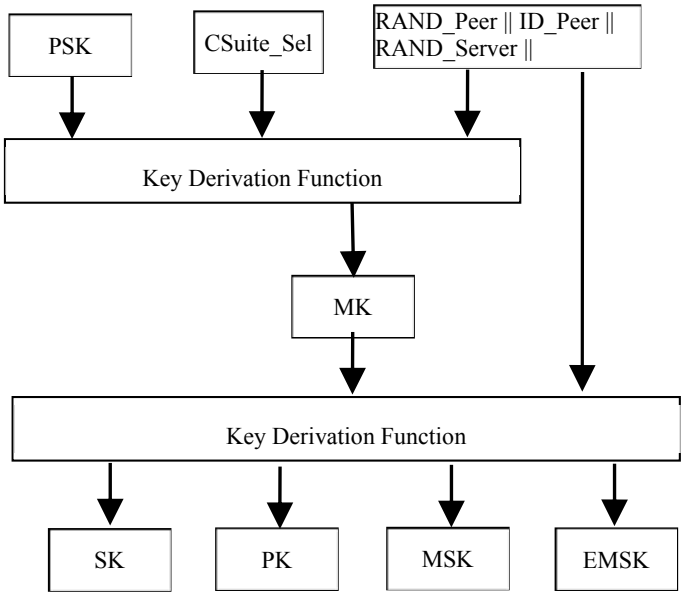


Рис. 13. Вычисление ключей в EAP-GPSK
Fig. 13. EAP-GPSK Key Derivation

Сервер предлагает список поддерживаемых алгоритмов через атрибут CSuite_List.

Атрибут SEC_SK обеспечивает защиту целостности сообщения.

Необязательный атрибут ENC_PK обеспечивает криптографическую защиту некоторых данных.

Метод использует несколько ключей безопасности:

Ключ PSK – заранее распределенный секретный ключ.

Ключ MK - используется для создания других ключей сеанса.

Ключ SK - используется для защиты целостности сообщений.

Ключ PK - используется для шифрования данных.

Ключи MSK, EMSK предназначены для использования за рамками данного метода EAP (см. RFC 3748 [1]).

3.2.4 EAP-pwd (EAP Authentication Using Only a Password)

Тип 52. RFC 5931 [29].

Использование пароля до сих пор является одним из самых распространенных способов аутентификации в Интернет с присущими ему многочисленными проблемами безопасности. Метод EAP-pwd использует в качестве входных данных пароль клиента. На его основе, используя криптографию дискретных

логарифмов (discrete logarithm cryptography), создаются и согласовываются ключи безопасности с заданной криптографической стойкостью, которые в дальнейшем используются для защиты сеанса и взаимной аутентификации клиента и сервера [30]. Данный механизм аутентификации обеспечивает защиту от различных типов атак, включая атаки по словарю, поскольку исходный пароль не передается по сети.

Метод определяет три пары сообщений: Identity, Commit, Confirm. В начале обмена исходный пароль преобразуется в бинарную строку в соответствии с заданными в EAP-pwd правилами, что должно обеспечить идентичные исходные данные на стороне клиента и сервера.

C←S : EAP-pwd-ID/Request
C→S : EAP-pwd-ID/Response

C←S : EAP-pwd-Commit/Request
C→S : EAP-pwd-Commit/Response

C←S : EAP-pwd-Confirm/Request
C→S : EAP-pwd-Confirm/Response

C←S : EAP- Success

Рис. 14. Успешный обмен EAP-pwd
Fig. 14. A Successful EAP-pwd Exchange

Успешный обмен EAP-pwd показан на рис. 14. Сообщения Identity используются для обмена идентификаторами и согласования криптографического набора. Также сервер может использовать эти сообщения, чтобы сообщить клиенту о необходимости предварительных преобразований пароля. В сообщениях Commit передаются данные для создания общего криптографического ключа. Третья пара сообщений Confirm используется для верификации созданных ключей и, как следствие, взаимной аутентификации сторон. Сообщения содержат проверочные блоки данных, защищенные с помощью согласованных криптографических алгоритмов и созданных ключей безопасности.

3.2.5 EAP-EKE (EAP Authentication Method Based on the Encrypted Key Exchange Protocol)

Тип 53. RFC 6124 [31].

Еще один метод, использующий в качестве входных данных обычный пароль клиента. Во многом повторяет EAP-pwd, из отличий: другой способ генерации общего ключа безопасности (используется механизм защищенного согласования ключей (Encrypted Key Exchange (EKE) на основе обмена Диффи-Хеллмана) и отсутствие требований предварительного преобразования исходного пароля к общему виду [32].

На основе исходного пароля создаются и согласовываются ключи безопасности с заданной криптографической стойкостью, которые в дальнейшем используются для защиты сеанса и взаимной аутентификации клиента и

сервера. Также стоит отметить, что исходный пароль не передается по сети, а созданный из пароля общий ключ сеанса не используется напрямую – из него вычисляются другие ключи, которые и используются для защиты сообщений

Метод определяет три пары сообщений: Identity, Commit, Confirm. Сообщения Identity используются для обмена идентификаторами и согласования криптографического набора. В сообщениях Commit передаются данные для создания общего криптографического ключа. Третья пара сообщений Confirm используется для верификации созданных ключей и, как следствие, взаимной аутентификации сторон. Сообщения содержат проверочные блоки данных, защищенные с помощью согласованных криптографических алгоритмов и созданных ключей безопасности. Успешный обмен EAP-EKE показан на рис. 14.

C←S : EAP-pwd-ID/Request
ID_S, CryptoProposals
C→S : EAP-pwd-ID/Response
ID_P, CryptoSelection

C←S : EAP-pwd-Commit/Request
Encr(Password, y_s)
C→S : EAP-pwd-Commit/Response
Encr(Password, y_p), Prot(Ke, Ki, Nonce_P)

C←S : EAP-pwd-Confirm/Request
Prot(Ke, Ki, Nonce_S | Nonce_P), Auth_S
C→S : EAP-pwd-Confirm/Response
Prot(Ke, Ki, Nonce_S), Auth_P

C←S : EAP- Success

Рис. 15. Успешный обмен EAP-EKE
Fig. 15. A Successful EAP-EKE Exchange

3.3 Методы, использующие механизмы создания защищенного канала для целей аутентификации

3.3.1 EAP-TLS (EAP TLS Authentication Protocol)

Тип 13. RFC 5216 [33].

Метод EAP-TLS использует фазу рукопожатия протокола TLSv1.1 (TLS handshake) для взаимной аутентификации клиента и сервера на основе сертификатов, а также для согласования дополнительных ключей безопасности [34]. Метод поддерживает стандартный механизм возобновления сеанса TLS. На рис.16 представлен успешный обмен EAP-TLS.

C←S : EAP-Request/Identity
C→S : EAP-Response/Identity (MyID)

C←S : EAP-Request/EAP-TLS (TSL start)
C→S : EAP-Response/EAP-TLS (ClientHello)

C←S : EAP-Request/EAP-TLS
(ServerHello, Certificate, [ServerKeyExchange],
CertificateRequest, ServerHelloDone)
C→S : EAP-Response/EAP-TLS
(Certificate, ClientKeyExchange, CertificateVerify,
ChangeCipherSpec, Finished)

C←S : EAP-Request/EAP-TLS
(ChangeCipherSpec, Finished)
C→S : EAP-Response/EAP-TLS

C←S : EAP- Success

Рис. 16. Успешный обмен EAP-TLS
Fig. 16. A Successful EAP-TLS Exchange

Получив идентификатор партнера, сервер EAP должен ответить пакетом EAP-TLS с установленным флагом S (Start), начинающим обмен рукопожатия TLS. В своем сообщении ClientHello партнер посылает случайное число ClientHello.random, максимальный номер версии TLS, которую он поддерживает, предлагаемый набор криптографических алгоритмов и метод сжатия, а также идентификатор сеанса, Session ID, (в случае попытки продолжения сеанса TLS).

Сервер отвечает пакетом, инкапсулирующим одну или несколько записей TLS. Сообщение рукопожатия ServerHello, содержит выбранные из предложенных клиентом версию TLS, идентификатор сеанса (Session ID), криптографические алгоритмы и метод сжатия, а также свое случайное число ServerHello.random. Выбранная версия TLS должна быть максимальной из поддерживаемых. Если сервер находит идентификатор сеанса в своем кэше, то это означает возобновление ранее установленного сеанса TLS. В противном случае сервер выбирает значение Session ID для установления нового сеанса TLS.

Сообщение Certificate содержит цепочку сертификатов либо для открытого ключа обмена ключами (RSA или ДиффиХеллмана), либо открытого ключа подписи (RSA или DSS). В последнем случае в этот обмен должно быть включено сообщение рукопожатия ServerKeyExchange. Если сервер аутентифицирован, он может запросить сертификат клиента (CertificateRequest). Кроме того, сервер посылает сообщение ServerHelloDone, указывающее, что фаза приветствия завершена. После чего ждет ответа клиента.

Если был соответствующий запрос сервера, клиент посылает свой сертификат. Затем следует сообщение ClientKeyExchange, содержимое которого зависит от выбранного в сообщениях ClientHello и ServerHello алгоритма с открытым ключом. Если сертификат клиента используется для цифровой подписи, то посылается подписанное сообщение CertificateVerify для явной проверки подлинности сертификата. Клиент посылает сообщение об изменении состояния ChangeCipherSpec и заменяет текущее состояние ожидаемым. В

завершение клиент посылает сообщение Finished, защищенное только что согласованными алгоритмами и содержащее код аутентификации (MAC), вычисленный над всеми предыдущими сообщениями обмена.

Сервер расшифровывает полученное сообщение Finished, проверяет правильность MAC (в случае ошибки соединение разрывается). Сервер отвечает своим сообщением изменения состояния ChangeCipherSpec, заменяет текущее состояние ожидаемым и посылает заключительное сообщение Finished, применяя согласованные алгоритмы и ключи.

Клиент также расшифровывает и проверяет полученное сообщение, о чем сообщает серверу в предпоследнем сообщении.

Следует отметить, что EAP-TLS в течение достаточно длительного времени был единственным понятным и надежным механизмом, обеспечивающим взаимную аутентификацию между клиентом и сервером EAP. Однако широкое его распространение оказалось ограниченным в связи со сложностью обеспечения клиентов сертификатами. Создание и поддержка инфраструктуры открытых ключей оказалось для многих организаций исключительно сложной задачей.

3.3.2 EAP-IKEv2

Тип 49. RFC 5106 [35].

Данный метод основан на протоколе обмена ключами IKEv2 (Internet Key Exchange Protocol version 2) и обеспечивает взаимную аутентификацию партнеров и согласование ключей безопасности на основе различных аутентификационных данных: паролей, заранее распределенных ключей или сертификатов. Протокол IKEv2 обеспечивает согласование криптографического набора, который затем используется для защиты сетевого обмена: шифрования данных, защиты целостности сообщений, генерации различных криптографических ключей [10]. Формат сообщений метода EAP аналогичен сообщениям протокола IKEv2. Также поддерживается быстрое возобновление сеанса. Метод EAP-IKEv2 не поддерживает инкапсуляцию других методов EAP.

R←I : EAP-Request/Identity
R→I : EAP-Response/Identity (Id)

R←I : EAP-Request (HDR, SAi, KEi, Ni)
R→I : EAP-Response (HDR, SAr, KEr, Nr, [CERTREQ], [SK{IDr}])

R←I : EAP-Request (HDR, SK{IDi, [CERT], [CERTREQ], [NFID], AUTH})
R→I : EAP-Response (HDR, SK{IDr, [CERT], AUTH})

R←I : EAP- Success

Рис. 17. Полный успешный обмен по протоколу EAP-IKEv2
Fig. 17. EAP-IKEv2 Full Successful Protocol Run.

В EAP-IKE2 предполагается, что сервер EAP исполняет роль инициатора (I), а партнер - роль ответчика (R). Семантика и формат сообщений EAP-IKE2 подобны, хотя и не совпадают полностью с семантикой и форматом сообщений IKE2. Полное выполнение протокола EAP-IKE2 включает пару сообщений запрос/ответ для выяснения идентификатора партнера и две пары сообщений запрос/ответ, за которыми следует либо сообщение EAP-Success, либо сообщение EAP-Failure.

Первые два сообщения являются стандартными сообщениями запроса идентификатора партнера и соответствующего ответа. С третьего сообщения начинается реальный аутентификационный обмен. Оно содержит индекс параметров безопасности инициатора (Security Parameter Index, SPI) в заголовке EAP-IKE2 (HDR), набор криптографических алгоритмов, которые сервер предполагает использовать для защиты трафика EAP-IKE2 (шифрования и защиты целостности) и вычисления сеансового ключа. Этот набор алгоритмов кодируется в блоке данных (Security Association, SAi). В блоках данных KEi и Ni пересылаются открытое значение Диффи-Хеллмана и одноразовый номер (нонс) инициатора.

Получив это сообщение, партнер генерирует ненулевое значение SPI ответчика, выбирает криптографический набор из множества, предложенного инициатором, и сообщает о своём выборе четвертым сообщением в блоке SAr, завершает обмен Диффи-Хеллмана, пересылая свое открытое значение в блоке KEr, и посылает свой одноразовый номер Nr.

После получения инициатором этого сообщения каждый из участников, используя результат обмена Диффи-Хеллмана, может найти порождающий секретный ключ SKEYSEED, на основе которого вычисляются все необходимые ключи в соответствии со спецификацией IKEv2 [10]. Для всех последующих сообщений будет обеспечиваться защита целостности и шифрование всего содержимого, кроме заголовка. Для этого используются ключи SK_a (аутентификация) и SK_e (шифрование) соответственно, получаемые из SKEYSEED. Для каждого направления вычисляются свои ключи. Кроме того, вычисляется величина SK_d, на основе которой впоследствии будет генерироваться ключевой материал для CHILD_SA. Именно ключи SK_e и SK_a используются для шифрования и обеспечения целостности данных в обозначении SK{...}.

В зависимости от используемых для аутентификации удостоверяющих данных локальная политика партнера может потребовать включения в четвертое сообщение необязательного блока данных CERTREQ, который запрашивает сертификат открытого ключа сервера, а в случае использования для аутентификации симметричной криптографии как на стороне сервера, так и на стороне партнера последний должен включить в это сообщение необязательный блок данных SK{IDr}, который содержит его идентификатор EAP-IKE2, зашифрованный и защищенный целостностью в блоке данных Encrypted. Этот идентификатор партнера необходим серверу для выбора

правильного симметричного ключа или пароля для создания блока данных AUTH в пятом сообщении.

Пятое сообщение, посылаемое сервером, содержит заголовок EAP-IKE2, за которым следует единственный блок данных Encrypted. Инициатор должен в этот блок данных Encrypted встроить по крайней мере два блока данных: блок данных Identification, содержащий EAP-IKE2 идентификатор инициатора, и блок данных Authentication (AUTH), который доказывает знание секрета, связанного с этим идентификатором, и обеспечивает целостность своего предыдущего сообщения. Инициатор также, возможно, посылает свой сертификат в блоке CERT и запрашивает сертификат партнера, указывая в блоке CERTREQ список сертификационных центров, которым инициатор доверяет. Если в сообщение были включены сертификаты, то первый из них должен содержать открытый ключ для проверки AUTH. Кроме того, в этот блок данных Encrypted может быть встроен блок данных Next Fast-Reconnect Identifier (NFRI), который спецификацией предполагалось использовать для быстрого повторного соединения.

После получения пятого сообщения ответчик (партнер EAP) аутентифицирует инициатора (сервер EAP). Для этого он осуществляет необходимый контроль расшифровывая блок данных Encrypted, проверяя его целостность и выясняя, содержит ли блок данных AUTH ожидаемое значение. Если все проверки прошли успешно, то ответчик отправляет шестое сообщение, которое содержит заголовок EAP-IKE2 и блок данных Encrypted, в который в свою очередь встраиваются блоки данных, показанные на рис.17.

После получения шестого сообщения инициатор (сервер EAP) аутентифицирует ответчика (партнера EAP). Как и в предыдущем случае проверки зависят от выбранного способа аутентификации, локальной политики и должны включать расшифровывание и проверку блока данных Encrypted, а также проверку того, что блок данных AUTH содержит ожидаемое значение. Если в четвертое сообщение был включен блок данных SK{IDr}, то сервер EAP должен также гарантировать, что блок данных IDr в шестом сообщении совпадает с соответствующим блоком данных из четвертого сообщения.

Если аутентификация прошла успешно, то ответчику отправляется седьмое сообщение EAP-Success. После успешного выполнения протокола EAP-IKE2 сервер EAP и партнер EAP генерируют ключи MSK и EMSK в соответствии с разделом 5 спецификации [35].

3.4 Туннельные методы

3.4.1 EAP-TTLSv0 (EAP Tunneled TLS Authenticated Protocol Version 0)

Тип 21. RFC 5281 [36].

Метод EAP-TLS использует протокол TLS для взаимной аутентификации клиента и сервера. Метод EAP-TTLS расширяет функциональность последнего, используя защищенный канал TLS, созданный в результате обмена

рукопожатия (TSL handshake), для безопасной передачи дополнительных данных между клиентом и сервером.

Информационный обмен данного метода состоит из двух фаз: фазы рукопожатия (handshake phase) и фазы передачи данных (data phase). Первая фаза использует стандартный механизм протокола TLS для аутентификации сервера клиентом (и, при желании сервера, для взаимной аутентификации) на основе сертификатов. В результате стороны согласовывают криптографический набор и ключи безопасности для создания защищенного соединения.

Во второй фазе созданный криптографический канал используется для передачи произвольных данных, а также, при необходимости, для аутентификации клиента сервером (или взаимной аутентификации) с помощью произвольного механизма аутентификации (может использоваться как какой-либо метод EAP, так и независимый протокол: PAP, CHAP, MS-CHAP-V2). Также во второй фазе могут последовательно применяться несколько механизмов аутентификации. Кроме того, метод EAP-TTLS позволяет создавать криптографические ключи для использования за пределами данного метода (ключи MSK, EMSK). В сообщениях EAP-TTLS для передачи данных используются атрибуты AVP (attribute-value pairs) совместимые с протоколами RADIUS и DIAMETR [37, 38].

Метод определяет сетевые объекты (рис. 18), взаимодействующие между собой.

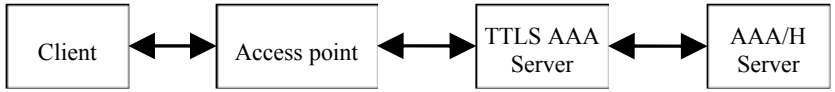
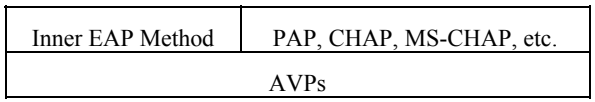


Рис. 18. Модель взаимодействия объектов метода EAP-TTLSv0
Fig. 18. EAP-TTLSv0 Architectural Model

Точка доступа (access point) и серверы TTLS и AAA/H могут располагаться как на одном сетевом узле, так и на нескольких. Как правило, между клиентом и точкой доступа отсутствуют предустановленные средства безопасности передачи данных. Ключи MSK и EMSK, создаваемые внутри метода EAP и предназначенные для использования за пределами метода (см. RFC 3748 [1]), могут быть переданы точке доступа сервером TTLS и могут в дальнейшем использоваться для криптографической защиты соединения между точкой доступа и клиентом.

На рис. 19 представлена многоуровневая модель сообщений метода EAP-TTLS, где каждый уровень сообщения инкапсулирует вышележащий уровень.



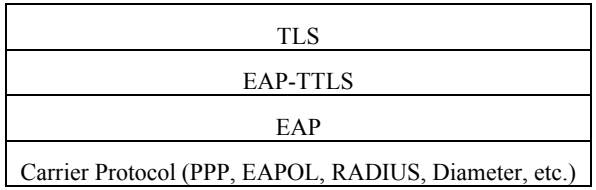


Рис. 19. Модель уровневой организации протокола
Fig. 19. Protocol Layering Model.

EAP-TTLS поддерживает стандартный механизм возобновления сеанса TLS (TLS session resumption). Однако данный механизм не должен использоваться, если клиент не смог пройти успешную аутентификацию во второй фазе метода. Несоблюдение данного требования сервером TTLS может привести к серьезному нарушению безопасности.

Схема сетевого обмена при успешной аутентификации (EAP/MD5-Challenge) показана на рис. 20.

C←S : EAP-Request/Identity
C→S : EAP-Response/Identity (Id)

C←S : EAP-Request/TTLS-start
C→S : EAP-Response/TTLS (ClientHello)

C←S : EAP-Request/TTLS (ServerHello, Certificate, ServerKeyExchange, ServerHelloDone)
C→S : EAP-Response/TTLS (ClientKeyExchange, ChangeCipherSpec, Finished)

C←S : EAP-Request/TTLS (ChangeCipherSpec, Finished)
C→S : EAP-Response/TTLS (EAP-Response/Identity)

C←S : EAP-Request/TTLS (EAP-Request/MD5-Challenge)
C→S : EAP-Response/TTLS (EAP-Response/ MD5-Challenge)

C←S : EAP- Success

Рис. 20. Успешная аутентификация посредством туннелируемого метода EAP/MD5-Challenge
Fig. 20. Successful Authentication via Tunneled EAP/MD5-Challenge

3.4.2 EAP-FAST (Flexible Authentication via Secure Tunneling EAP Method)

Тип 43. RFC 4851 [39].

Метод EAP-FAST использует протокол TLS для создания защищенного канала, внутри которого затем используются другие методы аутентификации. Для передачи данных внутри TLS-туннеля используются TLV-объекты (Type-Length-Value). Чтобы уменьшить использование вычислительных ресурсов, EAP-FAST поддерживает механизм быстрого возобновления сеанса TLS; в этом случае сервер не хранит параметры сеанса, а передает их клиенту в виде специальной структуры (ticket) [40].

Метод определяет сетевые объекты (рис. 21), взаимодействующие между собой

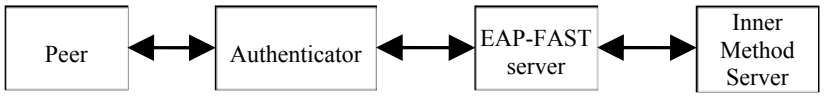


Рис. 21. Модель взаимодействия объектов метода EAP-FAST
Fig. 21. EAP-FAST Architectural Model.

При этом, объекты аутентификатор, сервер EAP-FAST и сервер внутреннего метода могут располагаться как на одном сетевом узле, так и на нескольких.

На рис. 22. представлена многоуровневая модель сообщений метода EAP-FAST, где каждый уровень сообщения инкапсулирует вышележащий уровень.

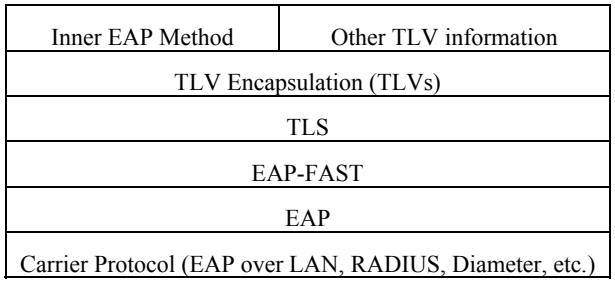


Рис. 22. Модель уровневой организации протокола EAP-FAST
Fig. 22. EAP-FAST Protocol Layering Model.

Вариант сетевого обмена с успешной аутентификацией при возобновлении сеанса показан на рис. 23.

C←S : EAP-Request/Identity
C→S : EAP-Response/Identity (MyID)

C←S : EAP-Request/EAP-FAST (S=1, A-ID)
C→S : EAP-Response/ EAP-FAST
(ClientHello with PAC-Opaque in SessionTicket extension)

C←S : EAP-Request/ EAP-FAST
(ServerHello, ChangeCipherSpec, Finished)
C→S : EAP-Response/ EAP-FAST
(ChangeCipherSpec, Finished)

Канал TLS установлен (последующие сообщения, посылаемые по каналу TLS, инкапсулируются в EAP-FAST):

C←S : EAP Payload TLV (EAP-Request/EAP-GTC(Challenge))
C→S : EAP Payload TLV (EAP-Response/EAP-GTC
(ответ содержит имя пользователя и пароль))

Необязательные дополнительные обмены (режим нового пин-кода, изменение пароля и т.п.):

C←S : Intermediate-Result TLV (Success)
Crypto-Binding TLV (Request)
C→S : Intermediate-Result TLV (Success)
Crypto-Binding TLV (Response)

C←S : Result TLV (Success)
[Optional PAC TLV]
C→S : Result TLV (Success)
[PAC TLV Acknowledgment]

Канал TLS уничтожается (сообщения посылаются в виде открытого текста)

C←S : EAP- Success

Рис. 23. Успешная аутентификация EAP-FAST
Fig. 23. EAP-FAST Successful Authentication

В тех случаях, когда внутри TLS-туннеля последовательно применяются несколько методов аутентификации, каждый такой метод заканчивается сообщением Intermediate-Result TLV/Crypto-Binding TLV, обеспечивающим индикацию результата и криптографическое связывание с последующим методом.

3.4.3 TEAP (Tunnel Extensible Authentication Protocol Version 1)

Тип 55. RFC 7170 [41].

Метод TEAP использует протокол TLS для создания защищенного канала со взаимной аутентификацией партнеров, внутри которого затем используются другие методы EAP. Для передачи данных внутри TLS-туннеля используются TLV-объекты (Type-Length-Value). TEAP может использоваться с любым транспортным протоколом, поддерживающим аутентификацию EAP.

На момент выхода данного стандарта были зарегистрированы несколько методов EAP, использующих TLS-туннель для защиты других методов аутентификации: PEAP (Protected EAP), EAP-TTLS (EAP tunneled TLS), EAP-FAST (EAP Flexible Authentication via Secure Tunneling) [42],[36],[39]. Однако, ни один из них не имеет статуса «Интернет Стандарта». Кроме того, в документе RFC 6678 сформулированы требования к методам EAP, использующим защищенный канал для последующей аутентификации (tunnel-based EAP method) [43].

Метод TEAP получил статус стандарта (PROPOSED STANDARD). В качестве его основы был выбран метод EAP-FAST, который был переработан в соответствии с требованиями RFC 6678, улучшена гибкость метода (особенно в отношении согласования криптографических алгоритмов), обновлены некоторые устаревшие спецификации, на которые ссылается EAP-FAST (в частности версия протокола TLS изменена на последнюю v1.2). Поэтому в плане архитектуры и сетевого обмена TEAP во многом повторяет EAP-FAST.

Ниже перечислены основные отличия от EAP-FAST:

- TEAP должен поддерживать последнюю версию протокола TLSv1.2 [12];
- криптографические ключи создаются в соответствии с требованиями RFC 5705, а соответствующие криптографические функции согласовываются через обмен TLS [44];
- TEAP полностью совместим с требованиями RFC 5077 (расширение TLS для быстрого возобновления сеанса с хранением состояния сеанса на стороне клиента) [45];
- добавлены дополнительные атрибуты для передачи метаданных и связывания каналов (channel binding);
- добавлена поддержка простого пароля в качестве одного из внутренних методов аутентификации.

4. Заключение

Протокол EAP в совокупности с его методами представляет собой мощное средство для обеспечения аутентификации и контроля доступа к сетям и ресурсам вычислительных систем. В данной статье рассмотрены основные особенности самого протокола и методов EAP. Показано разнообразие механизмов, используемых для реализации сервиса аутентификации. В ней не рассматриваются вопросы обеспечения безопасности EAP и методов EAP, которые с нашей точки зрения заслуживают отдельной публикации.

Список литературы

[1]. IETF RFC 3748. B. Aboba, et al. Extensible Authentication Protocol (EAP). June 2004. Доступно по ссылке: https://tools.ietf.org/html/rfc3748
[2]. IETF RFC 1661. W. Simpson. The Point-to-Point Protocol (PPP). July 1994. Доступно по ссылке: https://tools.ietf.org/html/rfc1661
[3]. IEEE Standard 802, Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture", 1990.
[4]. IETF RFC 791, Internet Protocol, September 1981. Доступно по ссылке: https://tools.ietf.org/html/rfc791
[5]. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control, 2010.
[6]. IETF RFC 3579. B. Aboba and P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. Доступно по ссылке: https://tools.ietf.org/html/rfc3579
[7]. IETF RFC 4072. Eronen, et al. Diameter Extensible Authentication Protocol (EAP) Application. August 2005. Доступно по ссылке: https://tools.ietf.org/html/rfc4072
[8]. IEEE Standard 802.11-2007, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific requirements – part 11: Wireless LAN Medium Access Control and Physical Layer specifications", 2007.
[9]. IEEE Standard 802.16e-2005, Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. December 2005.

[10]. IETF RFC 4306. Kaufman, C., Ed. Internet Key Exchange (IKEv2) Protocol. December 2005. Доступно по ссылке: https://tools.ietf.org/html/rfc4306
[11]. Extensible Authentication Protocol (EAP) Registry, Доступно по ссылке: http://www.iana.org/assignments/eap-numbers/eap-numbers.xhtml, 25.04.2018
[12]. IETF RFC 5246. Dierks, T. and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. August 2008. Доступно по ссылке: https://tools.ietf.org/html/rfc5246
[13]. IETF RFC 1994. W. Simpson. PPP Challenge Handshake Authentication Protoco. August 1996. Доступно по ссылке: https://tools.ietf.org/html/rfc1994
[14]. IETF RFC 2289. N. Haller, et al. A One-Time Password System. February 1998. Доступно по ссылке: https://tools.ietf.org/html/rfc2289
[15]. IETF RFC 4793. M. Nystroem. The EAP Protected One-Time Password Protocol (EAP-POTP). February 2007. Доступно по ссылке: https://tools.ietf.org/html/rfc4793
[16]. IETF RFC 4186. Haverinen & Salowey. Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM). January 2006. Доступно по ссылке: https://tools.ietf.org/html/rfc4186
[17]. European Telecommunications Standards Institute, "GSM Technical Specification GSM 03.20 (ETS 300 534): "Digital cellular telecommunication system (Phase 2); Security related network functions"", August 1997.
[18]. European Telecommunications Standards Institute, "GSM Technical Specification GSM 03.03 (ETS 300 523): "Digital cellular telecommunication system (Phase 2); Numbering, addressing and identification"", April 1997.
[19]. IETF RFC 4187. Arkko & Haverinen. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). January 2006. Доступно по ссылке: https://tools.ietf.org/html/rfc4187
[20]. 3rd Generation Partnership Project, "3GPP Technical Specification 3GPP TS 33.102 V5.1.0: "Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 5)"", December 2002.
[21]. 3rd Generation Partnership Project 2, "3GPP2 Enhanced Cryptographic Algorithms", September 2003.
[22]. 3rd Generation Partnership Project, "3GPP Technical Specification 3GPP TS 23.003 V6.8.0: "3rd Generation Parnership Project; Technical Specification Group Core Network; Numbering, addressing and identification (Release 6)"", December 2005.
[23]. IETF RFC 5448. Arkko, et al. Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA'). May 2009. Доступно по ссылке: https://tools.ietf.org/html/rfc5448
[24]. IETF RFC 4764. F. Bersani and H. Tschofenig. The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method. January 2007. Доступно по ссылке: https://tools.ietf.org/html/rfc4764
[25]. IETF RFC 4763. M. Vanderveen and H. Soliman. Extensible Authentication Protocol Method for Shared-secret Authentication and Key Establishment (EAP-SAKE). November 2006. Доступно по ссылке: https://tools.ietf.org/html/rfc4763
[26]. M. Bellare and P. Rogaway. Entity Authentication and key distribution. In Advances in Cryptology - Crypto 93 Proceedings, pages 232-249, 1993.
[27]. M. Bellare and P. Rogaway. Provably secure session key distribution: the three party case. In Proc. 27th Annual Symposium on the Theory of Computing, pages 57-66, 1995.
[28]. IETF RFC 5433. Clancy & Tschofenig. Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method. February 2009. Доступно по ссылке: https://tools.ietf.org/html/rfc5433

[29]. IETF RFC 5931, Harkins & Zorn. Extensible Authentication Protocol (EAP) Authentication Using Only a Password. August 2010. Доступно по ссылке: https://tools.ietf.org/html/rfc5931
[30]. Barker, E., Johnson, D., and M. Smid. Recommendations for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A, March 2007.
[31]. IETF RFC 6124. Sheffer, et al. An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol. February 2011. Доступно по ссылке: https://tools.ietf.org/html/rfc6124
[32]. Bellovin, S. and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Proc. IEEE Symp. on Research in Security and Privacy , May 1992.
[33]. IETF RFC 5216. Simon, et al. The EAP-TLS Authentication Protocol. March 2008. Доступно по ссылке: https://tools.ietf.org/html/rfc5216
[34]. IETF RFC 4346. Dierks, T. and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. April 2006. Доступно по ссылке: https://tools.ietf.org/html/rfc4346
[35]. IETF RFC 5106. Tschofenig, et al. The Extensible Authentication Protocol-Internet Key Exchange Protocol version 2 (EAP-IKEv2) Method. February 2008. Доступно по ссылке: https://tools.ietf.org/html/rfc5106
[36]. IETF RFC 5281. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. Доступно по ссылке: https://tools.ietf.org/html/rfc5281
[37]. IETF RFC 2865. Rigney, C., Willens, S., Rubens, A., and W. Simpson. Remote Authentication Dial In User Service (RADIUS). June 2000. Доступно по ссылке: https://tools.ietf.org/html/rfc2865
[38]. IETF RFC 3588. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko. Diameter Base Protocol. September 2003. Доступно по ссылке: https://tools.ietf.org/html/rfc3588
[39]. IETF RFC 4851. Cam-Winget, et al. The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). May 2007. Доступно по ссылке: https://tools.ietf.org/html/rfc4851
[40]. IETF RFC 4507. Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig. Transport Layer Security (TLS) Session Resumption without Server-Side State. May 2006. Доступно по ссылке: https://tools.ietf.org/html/rfc4507
[41]. IETF RFC 7170. Zhou, et al. Tunnel Extensible Authentication Protocol (TEAP) Version 1. May 2014. Доступно по ссылке: https://tools.ietf.org/html/rfc7170
[42]. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. Доступно по ссылке: https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018
[43]. IETF RFC 6678. Hoeper, K., Hanna, S., Zhou, H., and J. Salowey. Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method. July 2012. Доступно по ссылке: https://tools.ietf.org/html/rfc6678
[44]. IETF RFC 5705. Rescorla, E. Keying Material Exporters for Transport Layer Security (TLS). March 2010. Доступно по ссылке: https://tools.ietf.org/html/rfc5705
[45]. IETF RFC 5077. Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig. Transport Layer Security (TLS) Session Resumption without Server-Side State. January 2008. Доступно по ссылке: https://tools.ietf.org/html/rfc5077

The review of Extensible Authentication Protocol and its methods

[1] *A.V. Nikeshin <alexn@ispras.ru>*
[1,2] *V.Z. Shnitman <vzs@ispras.ru>*
[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *Moscow Institute of Physics and Technology (State University), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia*

Abstract. Authentication is associated with a scenario, in which some party (the applicant) presented the identity of the principal and states that this is the principal. Authentication allows some other party (verifier) to make sure that this statement is legitimate. Authentication is widely used in access control systems to networks and resources of computing systems. In this context, of considerable interest is the Extensible Authentication Protocol (EAP), specified by the IETF in RFC 3748, which provides an effective mechanism for embedding various authentication methods into it, as well as the proper methods of EAP authentication, some of which were standardized in specifications IETF. This article is a review of Extensible Authentication Protocol (EAP) and its methods, specified by IETF. EAP provide an effective flexible authentication mechanism that can be easily expanded with new authentication methods. The variety of mechanisms used to implement the authentication service are shown. The work was performed under support of the Russian Foundation for Basic Research, research grant № 16-07-00603 "The verification of security functionality of the EAP authentication protocol and evaluation of the robustness of its implementations against attacks".

Keywords: security; authentication; access control; EAP; EAP methods.

DOI: 10.15514/ISPRAS-2018-30(2)-7

For citation: Nikeshin A.V., Shnitman V.Z. The review of Extensible Authentication Protocol and its methods. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue. 2, 2018, pp. 113-148 (in Russian). DOI: 10.15514/ISPRAS-2018-30(2)-7

References

[1]. IETF RFC 3748. B. Aboba, et al. Extensible Authentication Protocol (EAP). June 2004. Доступно по ссылке: https://tools.ietf.org/html/rfc3748
[2]. IETF RFC 1661. W. Simpson. The Point-to-Point Protocol (PPP). July 1994. Available at https://tools.ietf.org/html/rfc1661
[3]. IEEE Standard 802, Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture", 1990.
[4]. IETF RFC 791, Internet Protocol, September 1981. Available at https://tools.ietf.org/html/rfc791
[5]. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control, 2010.
[6]. IETF RFC 3579. B. Aboba and P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. Available at https://tools.ietf.org/html/rfc3579

[7]. IETF RFC 4072. Eronen, et al. Diameter Extensible Authentication Protocol (EAP) Application. August 2005. Available at https://tools.ietf.org/html/rfc4072
[8]. IEEE Standard 802.11-2007, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific requirements – part 11: Wireless LAN Medium Access Control and Physical Layer specifications", 2007.
[9]. IEEE Standard 802.16e-2005, Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. December 2005.
[10]. IETF RFC 4306. Kaufman, C., Ed. Internet Key Exchange (IKEv2) Protocol. December 2005. Available at https://tools.ietf.org/html/rfc4306
[11]. Extensible Authentication Protocol (EAP) Registry, Available at http://www.iana.org/assignments/eap-numbers/eap-numbers.xhtml, 25.04.2018
[12]. IETF RFC 5246. Dierks, T. and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. August 2008. Available at https://tools.ietf.org/html/rfc5246
[13]. IETF RFC 1994. W. Simpson. PPP Challenge Handshake Authentication Protoco. August 1996. Available at https://tools.ietf.org/html/rfc1994
[14]. IETF RFC 2289. N. Haller, et al. A One-Time Password System. February 1998. Available at https://tools.ietf.org/html/rfc2289
[15]. IETF RFC 4793. M. Nystroem. The EAP Protected One-Time Password Protocol (EAP-POTP). February 2007. Available at https://tools.ietf.org/html/rfc4793
[16]. IETF RFC 4186. Haverinen & Salowey. Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM). January 2006. Available at https://tools.ietf.org/html/rfc4186
[17]. European Telecommunications Standards Institute, "GSM Technical Specification GSM 03.20 (ETS 300 534): "Digital cellular telecommunication system (Phase 2); Security related network functions"", August 1997.
[18]. European Telecommunications Standards Institute, "GSM Technical Specification GSM 03.03 (ETS 300 523): "Digital cellular telecommunication system (Phase 2); Numbering, addressing and identification"", April 1997.
[19]. IETF RFC 4187. Arkko & Haverinen. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). January 2006. Available at https://tools.ietf.org/html/rfc4187
[20]. 3rd Generation Partnership Project, "3GPP Technical Specification 3GPP TS 33.102 V5.1.0: "Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 5)"", December 2002.
[21]. 3rd Generation Partnership Project 2, "3GPP2 Enhanced Cryptographic Algorithms", September 2003.
[22]. 3rd Generation Partnership Project, "3GPP Technical Specification 3GPP TS 23.003 V6.8.0: "3rd Generation Parnership Project; Technical Specification Group Core Network; Numbering, addressing and identification (Release 6)"", December 2005.
[23]. IETF RFC 5448. Arkko, et al. Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA'). May 2009. Available at https://tools.ietf.org/html/rfc5448
[24]. IETF RFC 4764. F. Bersani and H. Tschofenig. The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method. January 2007. Available at https://tools.ietf.org/html/rfc4764
[25]. IETF RFC 4763. M. Vanderveen and H. Soliman. Extensible Authentication Protocol Method for Shared-secret Authentication and Key Establishment (EAP-SAKE). November 2006. Available at https://tools.ietf.org/html/rfc4763

[26]. M. Bellare and P. Rogaway. Entity Authentication and key distribution. In Advances in Cryptology - Crypto 93 Proceedings, pages 232-249, 1993.
[27]. M. Bellare and P. Rogaway. Provably secure session key distribution: the three party case. In Proc. 27th Annual Symposium on the Theory of Computing, pages 57-66, 1995.
[28]. IETF RFC 5433. Clancy & Tschofenig. Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method. February 2009. Available at https://tools.ietf.org/html/rfc5433
[29]. IETF RFC 5931, Harkins & Zorn. Extensible Authentication Protocol (EAP) Authentication Using Only a Password. August 2010. Available at https://tools.ietf.org/html/rfc5931
[30]. Barker, E., Johnson, D., and M. Smid. Recommendations for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A, March 2007.
[31]. IETF RFC 6124. Sheffer, et al. An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol. February 2011. Available at https://tools.ietf.org/html/rfc6124
[32]. Bellovin, S. and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Proc. IEEE Symp. on Research in Security and Privacy , May 1992.
[33]. IETF RFC 5216. Simon, et al. The EAP-TLS Authentication Protocol. March 2008. Available at https://tools.ietf.org/html/rfc5216
[34]. IETF RFC 4346. Dierks, T. and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. April 2006. Available at https://tools.ietf.org/html/rfc4346
[35]. IETF RFC 5106. Tschofenig, et al. The Extensible Authentication Protocol-Internet Key Exchange Protocol version 2 (EAP-IKEv2) Method. February 2008. Available at https://tools.ietf.org/html/rfc5106
[36]. IETF RFC 5281. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. Available at https://tools.ietf.org/html/rfc5281
[37]. IETF RFC 2865. Rigney, C., Willens, S., Rubens, A., and W. Simpson. Remote Authentication Dial In User Service (RADIUS). June 2000. Available at https://tools.ietf.org/html/rfc2865
[38]. IETF RFC 3588. Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko. Diameter Base Protocol. September 2003. Available at https://tools.ietf.org/html/rfc3588
[39]. IETF RFC 4851. Cam-Winget, et al. The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). May 2007. Available at https://tools.ietf.org/html/rfc4851
[40]. IETF RFC 4507. Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig. Transport Layer Security (TLS) Session Resumption without Server-Side State. May 2006. Available at https://tools.ietf.org/html/rfc4507
[41]. IETF RFC 7170. Zhou, et al. Tunnel Extensible Authentication Protocol (TEAP) Version 1. May 2014. Available at https://tools.ietf.org/html/rfc7170
[42]. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. Available at https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018
[43]. IETF RFC 6678. Hoeper, K., Hanna, S., Zhou, H., and J. Salowey. Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method. July 2012. Available at https://tools.ietf.org/html/rfc6678

[44]. IETF RFC 5705. Rescorla, E. Keying Material Exporters for Transport Layer Security (TLS). March 2010. Available at https://tools.ietf.org/html/rfc5705
[45]. IETF RFC 5077. Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig. Transport Layer Security (TLS) Session Resumption without Server-Side State. January 2008. Available at https://tools.ietf.org/html/rfc5077