Верификация туннельных методов протокола аутентификации EAP

А.В. Никешин, В.З. Шнитман

Институт системного программирования Российской академии наук

Аннотация. Одним из основных механизмов защиты информации в современных сетях является предварительное создание защищенного канала, внутри которого происходит передача данных. Данная тенденция не обошла и различные механизмы аутентификации. Протокол аутентификации EAP определяет для этих целей так называемые туннельные методы, предполагающие создание защищенного туннеля, внутри которого применяются другие методы аутентификации. К туннельным методам можно отнести такие методы EAP как PEAP, EAP-TTLS, EAP-FAST, TEAP. Все они используют протокол TLS для создания защищенного канала. В данной работе представлен опыт верификации туннельных методов протокола EAP с использованием технологии UniTESK и наработок коллектива в тестировании сетевых протоколов. Технология UniTESK позволяет автоматизировать процесс верификации сетевых протоколов на основе их формальных моделей.

Ключевые слова: безопасность, аутентификация, EAP, методы EAP, протоколы, тестирование, верификация, оценка устойчивости, Интернет, стандарты, формальные методы спецификации

The verification of tunnel methods of the Extensible Authentication Protocol (EAP)

A.V. Nikeshin, V.Z. Shnitman

Ivannikov Institute for System Programming of the Russian Academy of Sciences

Abstract. In modern networks one of the main mechanisms of information security is the establishing of a secure tunnel and transmitting data under the protection of that tunnel. The same mechanism is used in some authentication protocols. Extensible Authentication Protocol (EAP) provide an effective flexible authentication mechanism, that can be easily expanded with new authentication methods. A tunnel-based methods of EAP executes various EAP methods under the protection of the secure tunnel. There are several tunnel-based EAP methods: Protected EAP (PEAP), EAP Tunneled TLS Authenticated Protocol (EAP-TTLS), EAP Flexible Authentication via Secure Tunneling (EAP-FAST), Tunnel EAP (TEAP). They all use Transport Layer Security protocol (TLS) to establish the secure

tunnel. This paper presents approachs and results of verification of some tunnel-based methods of EAP Protocol using UniTESK technology and our team developments in testing network protocols. UniTESK technology allows to automate the process of test construction and evaluate whether an implementation conforms to the specification. Additional methods allow to send incorrect packets in each meaningful states of the protocol.

Keywords: security, authentication, EAP, EAP methods, protocols, testing, verification, evaluate robustness, Internet, standards, formal specifications

1. Введение

В современных информационных сетях обеспечение безопасности передачи данных стало одной из основных задач, что вызвано рядом объективных причин: широким проникновением информационных технологий в различные сферы деятельности, возрастающей сложностью информационных систем и информационных связей между ними, а также значительно выросшей производительностью вычислительных ресурсов, позволяющих более результативно проводить различные виды атак. Одним из основных механизмов защиты данных является предварительное создание защищенного канала, внутри которого происходит основной обмен. Данная схема применяется и в различных механизмах аутентификации, в том числе в протоколе EAP [1].

Важной особенностью протокола EAP является его гибкость. Сам протокол определяет лишь общую схему аутентификации. Конкретные механизмы аутентификации и криптографические схемы определяются в расширениях, так называемых методах EAP. Выбор необходимого метода происходит в процессе согласования сторон. Это позволяет довольно просто добавлять и использовать новые методы аутентификации.

Изначально разработанный для простой аутентификации пользователей через проводную телефонную сеть общего пользования (с использованием протокола точка-точка, Point to Point Protocol) [2], EAP нашел применение в самых разных средах и устройствах передачи данных: в проводных и беспроводных сетях, выделенных каналах и коммутируемых устройствах. На данный момент зарегистрировано несколько десятков расширений протокола [3]. Часть ранних методов EAP, обеспечивавших достаточную защиту для своего времени, на данный момент являются небезопасными. Тем не менее, их можно использовать внутри защищенного канала.

В туннельных методах EAP один или несколько методов аутентификации выполняются в защитном канале. Туннельные методы решают две важные задачи: во-первых, обеспечивают защиту открытых данных (например, обмен идентификаторами партнеров), во-вторых, позволяют использовать слабые методы аутентификации (до сих пор одним из популярных способов аутентификации являются пользовательские пароли).

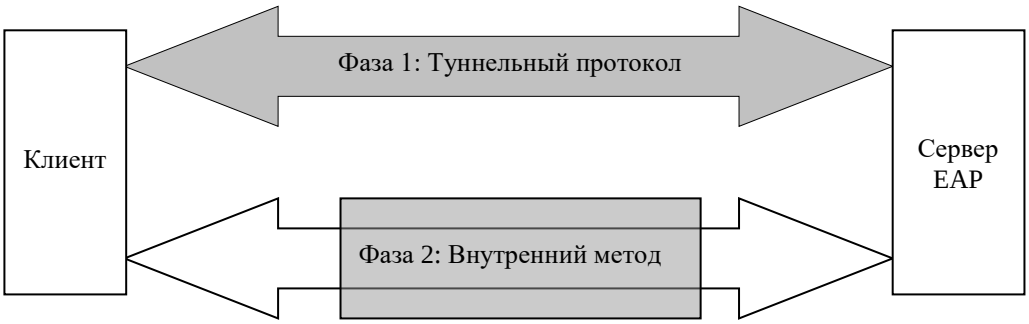

Рис. 1. Общая схема туннельного метода EAP

Туннельные методы включают две фазы. В первой фазе партнер и сервер устанавливают защищенный канал с помощью туннельного протокола. Во второй фазе, внутри этого туннеля выполняются методы аутентификации. При этом криптографические ключи, созданные при установлении туннеля, могут использоваться для связывания туннеля и внутренних методов, что обеспечит дополнительную защиту данных. Внутренние методы аутентификации могут быть как методами EAP, так и другими методами аутентификации. Стоит отметить, что спецификация EAP запрещает использование нескольких методов аутентификации в процессе одного выполнения EAP из-за уязвимости к некоторым видам атак, однако этот запрет не распространяется на туннельные методы, которые рассматриваются как единый метод аутентификации. Таким образом, в защищенном туннеле могут выполняться последовательно несколько методов аутентификации (например, сначала для аутентификации устройства, а затем для аутентификации пользователя).

Inner EAP	Other TLV
Type-Length-Value objects (TLVs)	
TLS	
EAP tunnel method	
EAP	
Carrier Protocol (EAP over LAN,	

Рис. 2. Модель уровневой организации туннельного метода

Ранее были зарегистрированы несколько туннельных методов EAP: PEAP (Protected EAP), EAP-TTLS (EAP tunneled TLS), EAP-FAST (EAP Flexible Authentication via Secure Tunneling) [4, 5, 6]. Все они для создания туннеля используют протокол TLS [7].

Ниже представлена типовая схема сетевого обмена при успешной аутентификации (EAP/MD5-Challenge) на примере EAP-TTLS.

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

Однако ни один из этих методов не имеет статуса «Стандарта Интернет». Такой статус получил созданный позднее метод TEAP (Tunnel Extensible Authentication Protocol Version 1) [8]. Кроме того, в документе RFC 6678 были сформулированы требования к методам EAP, использующим защищенный канал для последующей аутентификации (tunnel-based EAP method) [9].

Метод TEAP использует протокол TLS для создания защищенного канала с 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 EAP аутентификацию. В качестве основы TEAP 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 [7];

– криптографические ключи создаются в соответствии с требованиями RFC 5705, а соответствующие криптографические функции согласовываются через обмен TLS [10];

– TEAP полностью совместим с требованиями RFC 5077 (расширение TLS для быстрой переустановки сессии с хранением состояния сессии на стороне клиента) [11];

– добавлены дополнительные атрибуты для передачи метаданных и связывания каналов (channel binding);

– добавлена поддержка простого пароля в качестве одного из внутренних методов аутентификации

2. Используемые методы верификации

Тестирование реализаций сетевых протоколов направлено на решение двух важных задач: проверки совместимости различных реализаций и проверки их корректности и надежности.

В наших проектах мы используем наработанные нами методики по тестированию сетевых протоколов: автоматизированное тестирование на соответствие формальным спецификациям и методы мутации данных.

В текущих экспериментах используется разработанная нами на основе спецификаций RFC модель протокола EAP и его методов, описывающая сложную схему функционирования протокола.

Тестирование реализаций на соответствие формальным спецификациям проводится с использованием технологии UniTESK, предоставляющей средства автоматизации тестирования на основе использования конечных автоматов[12]. Состояния тестируемой системы определяют состояния автомата, а тестовые воздействия – переходы этого автомата. При выполнении перехода заданное воздействие передается на тестируемую реализацию, после чего регистрируются реакции реализации и автоматически выносится вердикт о соответствии наблюдаемого поведения спецификации. В UniTESK алгоритм обхода конечного автомата реализован как внутренний компонент и не зависит от протокола и тестируемой системы.

Методы мутационного тестирования используются для обнаружения неадекватного поведения тестируемой системы (завершение из-за фатальной ошибки, "подвисание", ошибки доступа к памяти). Как правило, подобные ситуации не рассматриваются в спецификациях. В сообщения, сформированные на основе разработанной модели протокола, вносятся какие-либо изменения, при этом модель протокола позволяет менять данные на любом этапе обмена, что позволяет тестовому сценарию проходить через все значимые состояния протокола и в каждом таком состоянии проводить тестирование реализации в соответствие с заданной программой.

3. Устройство тестового стенда

В данной работе мы используем классическую схему организации доступа по протоколу EAP, состоящую из трех сетевых узлов, выполняющих следующие роли:

- Клиент: Компьютер, которому требуется пройти аутентификацию. На нем исполняется основной поток управления тестовой системы под управлением UniTESK, обход тестового автомата и верификация наблюдаемых реакций. Тестовые сообщения протокола, сформированные

модельной реализацией, передаются через аутентификатор тестируемой системе, после чего регистрируются реакции тестируемого узла.

- Аутентификатор: Сетевой узел, с которым соединяется клиент. В общем случае аутентификатор используется как ретранслятор, передавая пакеты EAP между партнером и сервером EAP. Сервер EAP информирует аутентификатор о результате аутентификации. На основе этого результата аутентификатор либо предоставляет, либо запрещает доступ клиента к сети.

- Сервер EAP: Внутренний сервер с тестируемой реализацией протокола, который выполняет аутентификацию партнера и определяет, прошла ли аутентификация успешно или нет. Сервер EAP осуществляет обмен данными с аутентификатором и информирует его о результатах.

Клиент запрашивает доступ к сети, подключаясь к 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. На основании этого результата аутентификатор, принимает решение о предоставлении клиенту доступа к сети. Фактически аутентификация EAP выполняется между партнером и сервером EAP.

В качестве аутентификатора используется коммутатор Dell Networking N2048.

Протокол EAP может одновременно использоваться в разных средах передачи данных и, соответственно, выполняться через разные стеки сетевых протоколов. В нашем случае клиент и аутентификатор осуществляют обмен данными по протоколу 802.1x (EAP over LAN) [13]. Аутентификатор и сервер EAP осуществляют обмен данными через проводной канал поверх протокола AAA RADIUS [14].

В качестве первой реализации протокола EAP выбрана реализация FreeRADIUS [15]. Данная реализация заявлена как самый распространенный сервер RADIUS с открытым исходным кодом, с развитой функциональностью и поддержкой распространенных методов аутентификации EAP. Данная реализация использовалась нами на разных этапах данного проекта. Сервер установлен под ОС CentOS 7 [16]. В качестве второй реализации протокола EAP используется Windows Server 2012 [17].

Несмотря на довольно широкую распространенность, данные реализации не поддерживают метод TEAP. Основным туннельным методом FreeRADIUS является TTLSv0 [5]. Windows Server использует исключительно PEAP

собственной разработки – PEAPv0 [4]. По этим причинам данные методы были выбраны для текущих экспериментов.

4. Результаты тестирования

На текущем годовом этапе проекта в рамках технологии UniTESK выполнены следующие задачи:

- разработаны модели методов аутентификации TTLSv0, PEAP и TEAP, которые интегрированы в разработанную ранее модель базового протокола EAP,
- разработана спецификация и медиаторы для указанных методов,
- разработан набор тестов, покрывающий часть требований спецификаций,

Найдены несколько отклонений обеих реализаций от спецификаций.
Freeradius:

- В заголовке TTLSv0 поле Длина присутствует во всех фрагментах (При использовании протокола TLS большие блоки данных разбиваются на фрагменты и отправляются последовательно несколькими сообщениями EAP, при этом полная длина исходного TLS сообщения передается только с первым фрагментом и не должна присутствовать в остальных фрагментах).
- Если во входящем сообщении установлен бит M (указывает, что используется фрагментация и текущий фрагмент не последний), реализация ждет следующие фрагменты независимо от значений других полей.
- Если во входящем сообщении установлен бит M, реализация требует, чтобы в следующих входящих фрагментах в заголовке TTLSv0 присутствовало поле длины. Если поле длины отсутствует, реализация отправляет пустое сообщение и ждет следующий фрагмент.
- Значение поля Version (версия метода EAP-TTLS) не проверяется.
- В различных случаях не проверяется согласование длины пакета EAP, длины данных TLS и длины в заголовке TTLSv0.
- Также обнаружена критическая уязвимость реализации. После установления TLS-туннеля при получении внутри туннеля данных определенного формата реализация выдает ошибку "Segmentation fault" и "падает". Данная уязвимость проявляется до аутентификации клиента (т.е. до использования туннелируемых методов аутентификации) и позволяет любому клиенту (злоумышленнику достаточно знать имя клиента, для которого применяется метод EAP-TTLS) провести DoS атаку ("отказ в обслуживании"). Уязвимость наблюдается для FreeRADIUS 3.0.13 под ОС CentOS 7 (release 7.4.1708).

Windows Server 2012:

- Реализация игнорирует бит S ("Start TLS") во входящем сообщении от клиента (данный бит используется для инициации сервером TLS обмена и применяется только в первом пустом сообщении от сервера клиенту).
- В некоторых случаях игнорирует значение поля длины в заголовке PEAP (данное поле указывает полную длину TLS сообщения до фрагментации).
- В некоторых случаях не проверяется согласование длины пакета EAP, длины данных TLS и длины в заголовке PEAP.
- Реализация отбрасывает входящие сообщения с битом M (указывает, что используется фрагментация и текущий фрагмент не последний), если считает, что текущее сообщение не должно быть фрагментировано (например, сообщение EAP-TTLS-TLS_ClientHello).
- Если реализация ждет следующий фрагмент от клиента, а получает пустое сообщение, то отправляет в ответ также пустое сообщение (EAP-TTLS Acknowledgement packet). Такой цикл можно повторять очень долго, в реализации вероятно отсутствует ограничение количества таких сообщений (например, freeradius прерывает аутентификацию после 50 таких циклов).

Разработка тестового набора продолжается.

5. Заключение

В данной работе представлен опыт верификации туннельных методов протокола аутентификации EAP, который является завершающим этапом проекта по верификации этого протокола безопасности. Создание защищенного канала, внутри которого происходит передача других данных, является сегодня одним из основных механизмов защиты информации в современных сетях. Данная тенденция не обошла и различные механизмы аутентификации. Протокол аутентификации EAP определяет для этих целей так называемые туннельные методы, предполагающие создание защищенного туннеля, внутри которого применяются другие методы аутентификации.

В работе использовался новый тестовый набор, разработанный с использованием технологии UniTESK и наработок коллектива в тестировании сетевых протоколов. Технология UniTESK позволяет автоматизировать процесс верификации сетевых протоколов на основе их формальных моделей, методы мутационного тестирования позволяют протестировать устойчивость реализации протокола к искаженным сообщениям.

Представленный подход доказал свою эффективность в наших предыдущих проектах, обеспечив обнаружение различных отклонений от спецификации и других ошибок при тестировании сетевых протоколов [18,19].

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

Литература

1. Aboba B. et al. Extensible Authentication Protocol (EAP). June 2004. IETF RFC 3748. — URL: https://tools.ietf.org/html/rfc3748 .
2. Simpson W. The Point-to-Point Protocol (PPP). July 1994. IETF RFC 1661. — URL: https://tools.ietf.org/html/rfc1661 .
3. Extensible Authentication Protocol (EAP) Registry. — URL: http://www.iana.org/assignments/eap-numbers/eap-numbers.xhtml .
4. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. — URL: https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018 .
5. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. IETF RFC 5281. — URL: https://tools.ietf.org/html/rfc5281 .
6. Cam-Winget et al. The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). May 2007. IETF RFC 4851. — URL: https://tools.ietf.org/html/rfc4851 .
7. Dierks T. and Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2. August 2008. IETF RFC 5246. — URL: https://tools.ietf.org/html/rfc5246 .
8. Zhou et al. Tunnel Extensible Authentication Protocol (TEAP) Version 1. May 2014. IETF RFC 7170. — URL: https://tools.ietf.org/html/rfc7170 .
9. Hoeper K., Hanna S., Zhou H., and Salowey J. Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method. July 2012. IETF RFC 6678. — URL: https://tools.ietf.org/html/rfc6678 .
10. Rescorla E. Keying Material Exporters for Transport Layer Security (TLS). March 2010. IETF RFC 5705. — URL: https://tools.ietf.org/html/rfc5705 .
11. Salowey J., Zhou H., Eronen P., and Tschofenig H. Transport Layer Security (TLS) Session Resumption without Server-Side State. January 2008. IETF RFC 5077. — URL: https://tools.ietf.org/html/rfc5077 .
12. Bourdonov I., Kossatchev A., Kuliamin V., and Petrenko A. UniTesK Test Suite Architecture // Proceedings of FME 2002. LNCS 2391, pp. 77–88, Springer-Verlag, 2002 .
13. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control, 2010.
14. Aboba B. and Calhoun P. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. IETF RFC 3579. — URL: https://tools.ietf.org/html/rfc3579 .

15. FreeRADIUS. — URL: http://freeradius.org .
16. CentOS 7. — URL: https://www.centos.org/ .
17. Windows Server 2012 R2. — URL: https://www.microsoft.com .
18. Никешин А.В., Пакулин Н.В., Шнитман В.З. Подходы к разработке тестового набора для тестирования реализаций протокола EAP и его методов // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19–24 сентября 2016 г., г. Новороссийск). — М.: ИПМ им. М.В. Келдыша, 2016. — С. 290–297. — doi:10.20948/abrau-2016-24.
19. Никешин А.В., Пакулин Н.В., Шнитман В.З. "Мутационное тестирование сетевых протоколов с использованием формальных моделей" // Научный сервис в сети Интернет: труды XVII Всероссийской научной конференции (21–26 сентября 2015 г., г. Новороссийск). – М.: ИПМ им. М.В.Келдыша, 2015. ISBN 978-5-98354-015-6. Стр. 259–266.
20. Никешин А.В., Пакулин Н.В., Шнитман В.З. "Тестирование реализаций клиента протокола TLS" // Труды Института системного программирования РАН. Том 27. Выпуск 2. 2015 г. Стр. 145–160. DOI:10.15514/ISPRAS-2015-27(2)-9.

References

1. Aboba B. et al. Extensible Authentication Protocol (EAP). June 2004. IETF RFC 3748. — URL: https://tools.ietf.org/html/rfc3748 .
2. Simpson W. The Point-to-Point Protocol (PPP). July 1994. IETF RFC 1661. — URL: https://tools.ietf.org/html/rfc1661 .
3. Extensible Authentication Protocol (EAP) Registry. — URL: http://www.iana.org/assignments/eap-numbers/eap-numbers.xhtml .
4. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. — URL: https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018 .
5. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. IETF RFC 5281. — URL: https://tools.ietf.org/html/rfc5281 .
6. Cam-Winget et al. The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). May 2007. IETF RFC 4851. — URL: https://tools.ietf.org/html/rfc4851 .
7. Dierks T. and Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2. August 2008. IETF RFC 5246. — URL: https://tools.ietf.org/html/rfc5246 .
8. Zhou et al. Tunnel Extensible Authentication Protocol (TEAP) Version 1. May 2014. IETF RFC 7170. — URL: https://tools.ietf.org/html/rfc7170 .
9. Hoeper K., Hanna S., Zhou H., and Salowey J. Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method. July 2012. IETF RFC 6678. — URL: https://tools.ietf.org/html/rfc6678 .
10. Rescorla E. Keying Material Exporters for Transport Layer Security (TLS). March 2010. IETF RFC 5705. — URL: https://tools.ietf.org/html/rfc5705 .

11. Salowey J., Zhou H., Eronen P., and Tschofenig H. Transport Layer Security (TLS) Session Resumption without Server-Side State. January 2008. IETF RFC 5077. — URL: https://tools.ietf.org/html/rfc5077 .
12. Bourdonov I., Kossatchev A., Kuliamin V., and Petrenko A. UniTesK Test Suite Architecture // Proceedings of FME 2002. LNCS 2391, pp. 77–88, Springer-Verlag, 2002 .
13. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control, 2010.
14. Aboba B. and Calhoun P. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. IETF RFC 3579. — URL: https://tools.ietf.org/html/rfc3579 .
15. FreeRADIUS. — URL: http://freeradius.org .
16. CentOS 7. — URL: https://www.centos.org/ .
17. Windows Server 2012 R2. — URL: https://www.microsoft.com .
18. Nikeshin A.V., Pakulin N.V., Shnitman V.Z. Podkhody k razrabotke testovogo nabora dlia testirovaniia realizatsii protokola EAP i ego metodov // Nauchnyi servis v seti Internet: trudy XVIII Vserossiiskoi nauchnoi konferentsii (19-24 sentiabria 2016 g., g. Novorossiisk). — M.: IPM im. M.V.Keldysha, 2016. — S. 290-297. — doi:10.20948/abrau-2016-24.
19. Nikeshin A.V., Pakulin N.V., Shnitman V.Z. Mutatsionnoe testirovanie setevykh protokolov s ispolzovaniem formalnykh modelei // Nauchnyi servis v seti Internet: trudy XVII Vserossiiskoi nauchnoi konferentsii (21-26 sentiabria 2015 g., g. Novorossiisk). — M.: IPM im. M.V.Keldysha, 2015. — S. 259-266. — ISBN 978-5-98354-015-6.
20. Nikeshin A.V., Pakulin N.V., Shnitman V.Z. TLS Clients Testing. Trudy ISP RAN /Proc. ISP RAS, vol. 27, issue 2, 2015, pp. 145-160 (in Russian). DOI:10.15514/ISPRAS-2015-27(2)-9.