УДК 004.04

РЕАЛИЗАЦИЯ ЗАЩИЩЕННОЙ СЕССИИ МЕЖДУ НЕСКОЛЬКИМИ УЧАСТНИКАМИ НА ОСНОВЕ TLS В XMPP С ПОМОЩЬЮ СЕТИ ПЕТРИ

Р.Н. Ахмадуллин
Научный руководитель З.Я. Якупов
Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, г. Казань, Российская Федерация

В данной статье рассматривается пример реализации XMPP (eXtensible Messaging and Presence Protocol «расширяемый протокол обмена сообщениями о присутствии») [1] протокола на основе инструмента сетей Петри в ПО PIPE (Platform Independent Petri net Editor). Данная модель должна решать конкретные задачи и отвечать заданным требованиям.

Актуальность темы следует из необходимости обязательной реализации защищенной связи в следующих сферах: социальные сети, мессенджеры, облачные вычисления.

Цель: реализация данного протокола на примере открытия сессии между несколькими участниками, установления TLS-канала и дальнейшего взаимодействия участников по защищённому каналу.

Задачи:

1) изучить описание XMPP протокола.
2) изучить использование TLS с целью организации защищённого межсетевого взаимодействия.
3) реализовать модель XMPP протокола.

Простая Jabber (XMPP) сессия представляет из себя следующую последовательность операций [2].

XMPP включает в себя метод обеспечения безопасности потока от фальсификации и подслушивания. Базовые требования защищённого соединения должны быть следующими [3].

В данной статье выделяются следующие этапы для реализации модели протокола:

1. Открытие сессии между клиентом и сервером.
2. Аутентификация на основе TLS.

3. Создание защищённой сессии обмена сообщениями между двумя участниками.

Последовательность шагов, которая применяется для построения модели TLS с помощью сетей Петри, следующая:

1) P0 – открытие сессии; T0 – клиент формирует поток к серверу по открытой сессии;
2) P1 – получение потока на стороне сервера; T1 – сервер откликается посылкой тэга потока клиенту;
3) P2 –создание сессии; T2 – сервер предлагает клиенту расширение STARTTLS и данные о механизме аутентификации с другими особенностями потока. В противном случае клиенту будет отказано в предоставлении данных – позиция P4. Тогда на основе перехода T3 возвращаемся в прежнее состояние P0;
4) P3 – клиент получил данные STARTTLS от сервера; T4 – клиент посылает серверу ответ STARTTLS;
5) P5 – сервер получил ответ от клиента (server_STARTTLS_reply); T5 – сервер информирует клиента о том, что он может продолжить работу. В другом случае сервер информирует клиента, что согласование TLS не состоялось и следует прервать поток и разорвать TCP соединение (позиция P7, переход T6). Тогда на основе перехода T13 возвращаемся в прежнее состояние P3;
6) P6 – согласование TLS прошло успешно; T7 – клиент формирует новый поток к серверу;
7) P8 – сервер получает новый поток от клиента, сервер реагирует посылкой клиенту заголовка потока и любых характеристик потока – переход T9. Если согласование TLS не получилось, сервер закрывает TCP соединение – переход T8 (позиция P9 «close_session»);
8) P10 – клиент получил посылку; T10 – клиент устанавливает защищённое соединение с сервером по TLS;
9) P11 – сервер получил данные от клиента; T11 – сервер формирует сессию в виде ответа клиенту;
10) P12 – сессия открыта между клиентом и сервером;
11) P13 – в другом случае сессия будет закрыта (переход T12).

Последовательность этапов, используемая для создания модели TLS с помощью сетей Петри, показана на рис. 1.

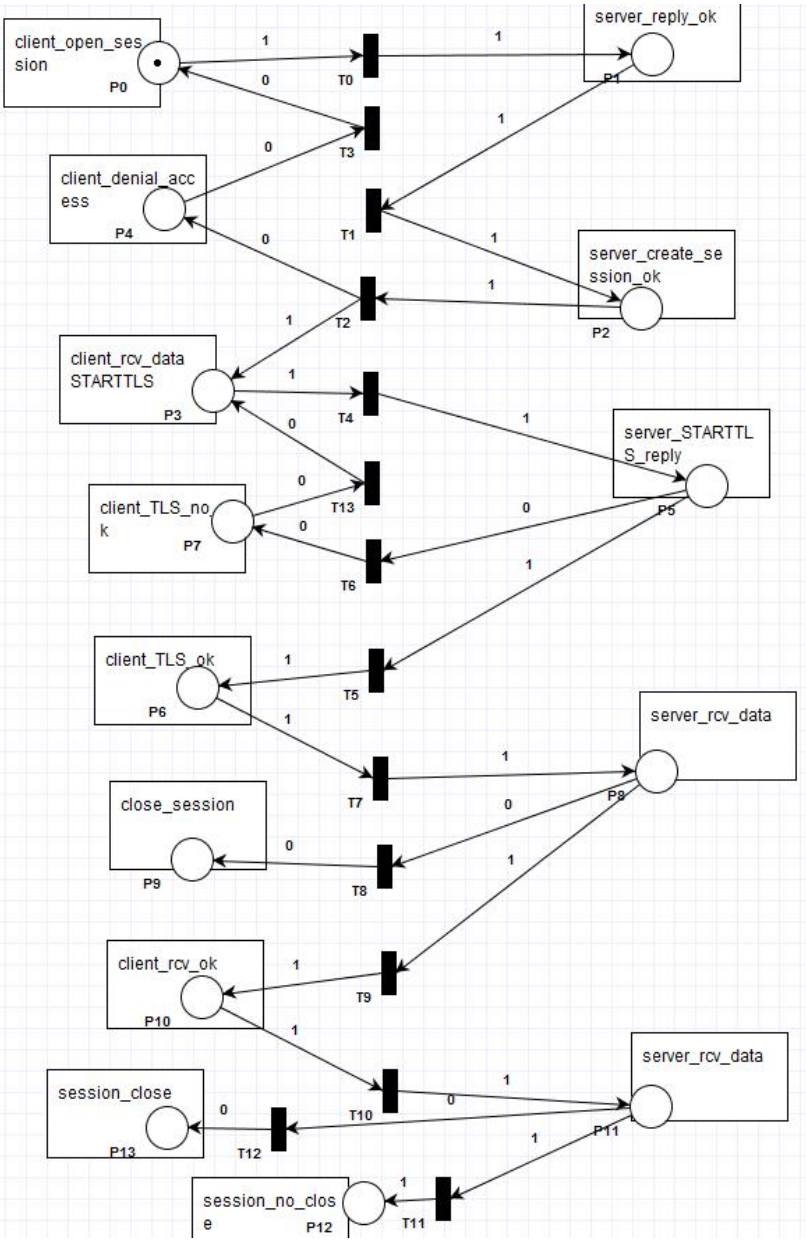

Рис. 1. Использование TLS на примере клиент-серверного взаимодействия

Для анализа взаимодействия нескольких участников в канале рассматриваются следующие ключевые события:

1. SendRequest (Reply) – отправка запроса (ответа) к получателю.
2. Connection – соединение.
3. Processing – обработка сообщения.
4. Delivery – доставка сообщения.
5. ReplyOk (NotOk) – положительный отклик (отрицательный отклик).
6. RepeatSession – повтор соединения.

Последовательность шагов, используемая для создания модели TLS с помощью сетей Петри, при применении расширенной схемы клиент-серверного взаимодействия показана на рис. 2.

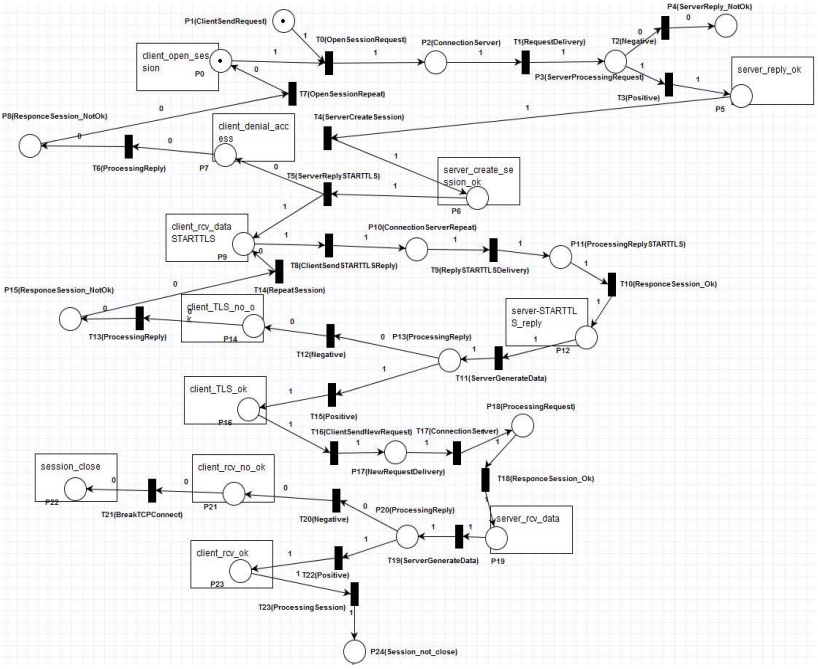

Рис. 2. Расширенный пример использования TLS на примере клиент-серверного взаимодействия

На рис. 3 продемонстрирована модель обмена сообщениями между двумя участниками по защищённому соединению после аутентификации TLS. В данной модели рассмотрены положительные и негативные исходы на стороне сервера.

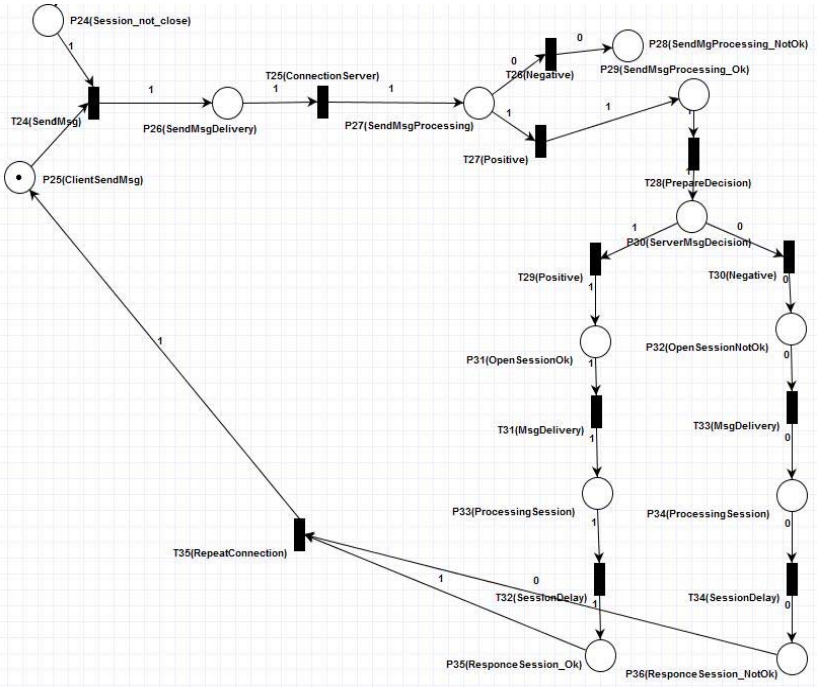

Рис. 3. Пример обмена сообщениями между несколькими участниками по защищенному каналу

В данной статье разработанная модель показывает обмен данными между клиентом и сервером благодаря обеспечению безопасности потока посредством механизма STARTTLS (TLS). С помощью сетей Петри показана реализация защищённого соединения по TLS между несколькими участниками.

Литература

1. Знакомство с протоколом XMPP // IBM URL: https://www.ibm.com/developerworks/ru/library/x-xmppintro/ (дата обращения: 29.01.2020).
2. Протоколы прикладного уровня: Jabber/XMPP часть1 // PS URL: http://seriyps.ru/blog/2009/02/15/protokoly-prikladnogo-urovnya-jabberxmpp-chast1/ (дата обращения: 29.01.2020).
3. Расширяемый протокол обмена сообщениями и данными о присутствии (XMPP): Ядро // ИТЭФ-МФТИ URL: http://book.itep.ru/4/45/xmpp.htm#6 (дата обращения: 29.01.2020).