**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии   
и прикладная математика»   
Кафедра: 806 «Вычислительная математика   
и программирование»**

Курсовой проект  
по курсу «Криптография»

Группа: М8О-306Б-21

Студент: E. В. Черевичин

Преподаватель: А. В. Борисов

Оценка:

Дата: 11.05.2024

Москва, 2024

**ОГЛАВЛЕНИЕ**

[1 Тема 3](#_Toc162017708)

[2 Задание 3](#_Toc162017709)

[3 Теория 4](#_Toc162017710)

[4 Ход лабораторной работы 5](#_Toc162017711)

[5 Выводы 10](#_Toc162017712)

# **Тема**

Аутентификация с асимметричными алгоритмами шифрования.

# **Задание**

1. Выбрать не менее 2-ух web-серверов сети Интернет различной организационной и государственной принадлежности.

2. Запустить Wireshark и используя Firefox установить https соединение с выбранным сервером.

3. Провести анализ соединения.

4. Сохранить данные необходимые для последующего сравнительного анализа: Имя сервера, его характеристики. Версия TLS. Выбранные алгоритмы шифрования. Полученный сертификат: версия. Валидность сертификата, валидность ключа, удостоверяющий центр. Время установки соединения (от ClientHello до Finished)

5. Если список исследуемых серверов не исчерпан выбрать другой сервер и повторить соединение.

6. Если браузер поддерживал соединение TLS 1.2 принудительно изменить параметры TLS соединения в Firefox на TLS 1.0 (в браузере перейти по адресу “about:config” и изменить раздел SSL\TLS) и провести попытки соединения с выбранными серверами).

7. Провести сравнительный анализ полученной информации.

8. В качестве отчета представить результаты сравнительного анализа, выводы в отношении безопасности и корректности настройки веб-серверов с учетом их организационной и государственной принадлежности

**3 Теория**

**Wireshark** — программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Имеет графический пользовательский интерфейс.

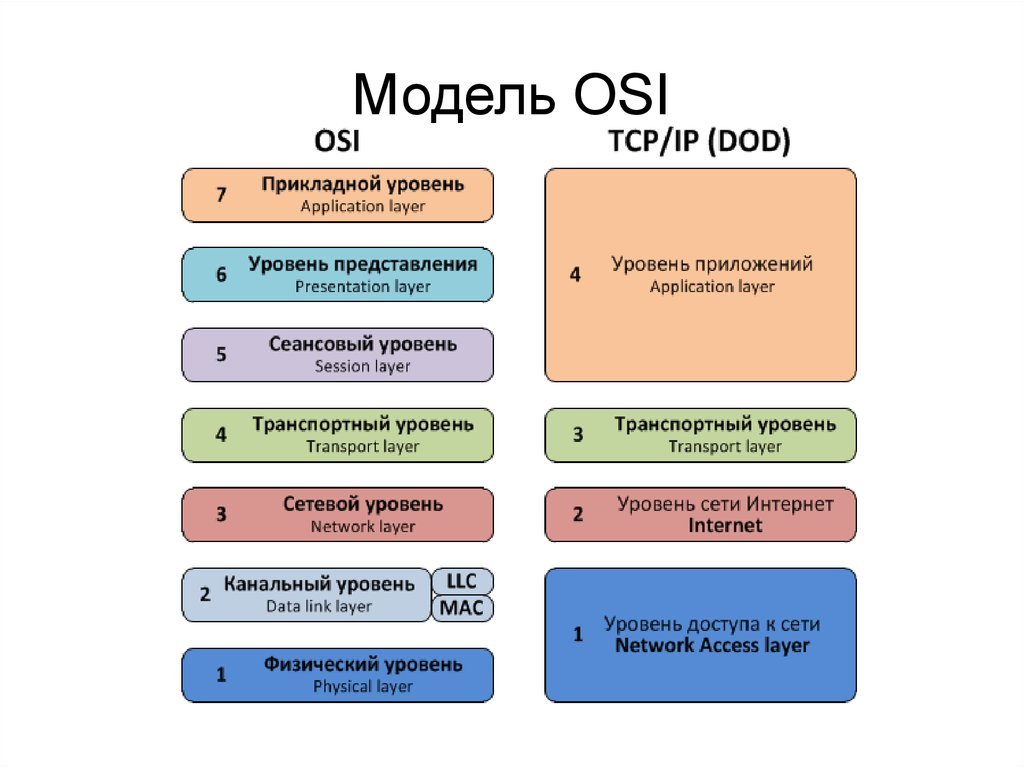
**HTTPS** – это безопасный протокол передачи данных, который поддерживает шифрование посредством криптографических протоколов SSL и TLS, и является расширенной версией протокола HTTP.

**SSL**, или secure sockets layer, — это протокол, который шифрует и защищает

данные во время их передачи по интернету. Для этого он использует специальные криптографические ключи, изменяющие данные до неузнаваемости.

**TLS**, или transport layer security, — это протокол, который защищает данные во время их передачи по Сети. Он работает на четвёртом, транспортном, уровне сетевой модели OSI, где отвечает за создание безопасных сессий обмена данными между браузером и сервером.

**Модель OSI:**



**TLS-рукопожатия** – это серия датаграмм, или сообщений, которыми обмениваются клиент и сервер. TLS-рукопожатие включает в себя несколько этапов, в ходе которых клиент и сервер обмениваются информацией, необходимой для завершения рукопожатия и обеспечения возможности дальнейшего общения.

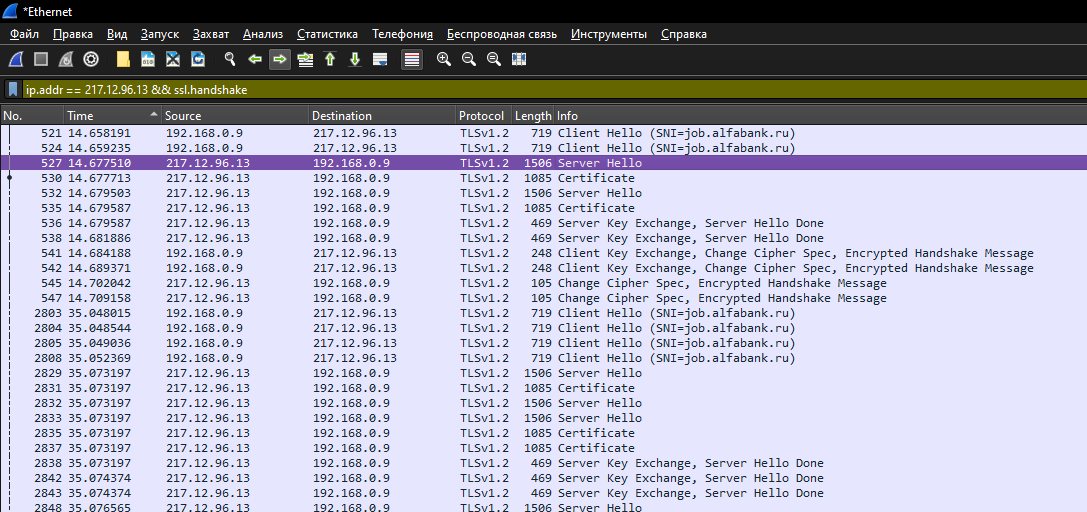
1. Сообщение "hello" клиента: Клиент инициирует рукопожатие, посылая серверу сообщение "hello". В сообщении будет указано, какую версию TLS поддерживает клиент, поддерживаемые наборы шифров и строка случайных байтов, известная как "random клиента".
2. Сообщение "hello" сервера: В ответ на сообщение "hello" клиента сервер отправляет сообщение, содержащее SSL-сертификат сервера, выбранный сервером набор шифров и "random сервера" – еще одну строку случайных байтов, генерируемую сервером.
3. Аутентификация: Клиент проверяет SSL-сертификат сервера в центре сертификации, который его выдал. Он подтверждает, что сервер является тем, за кого себя выдает, и что клиент взаимодействует с реальным владельцем домена.
4. Premaster secret: Клиент посылает еще одну строку случайных байтов, которая называется "premaster secret". Premaster secret шифруется открытым ключом и может быть расшифрован сервером только с помощью закрытого ключа. (Клиент берет открытый ключ из SSL-сертификата сервера).
5. Используемый закрытый ключ: Сервер расшифровывает "premaster secret".
6. Создание сеансовых ключей: Клиент и сервер генерируют сеансовые ключи из random'а клиента, random'а сервера и premaster secret. Они должны прийти к одинаковым результатам.
7. Клиент готов: Клиент отправляет сообщение "готово", зашифрованное сеансовым ключом.
8. Сервер готов: Сервер отправляет сообщение "готово", зашифрованное сеансовым ключом.
9. Безопасность симметричного шифрования достигнута: Рукопожатие завершено, и связь продолжается с использованием сеансовых ключей.

**Каким образом будет происходить анализ:**

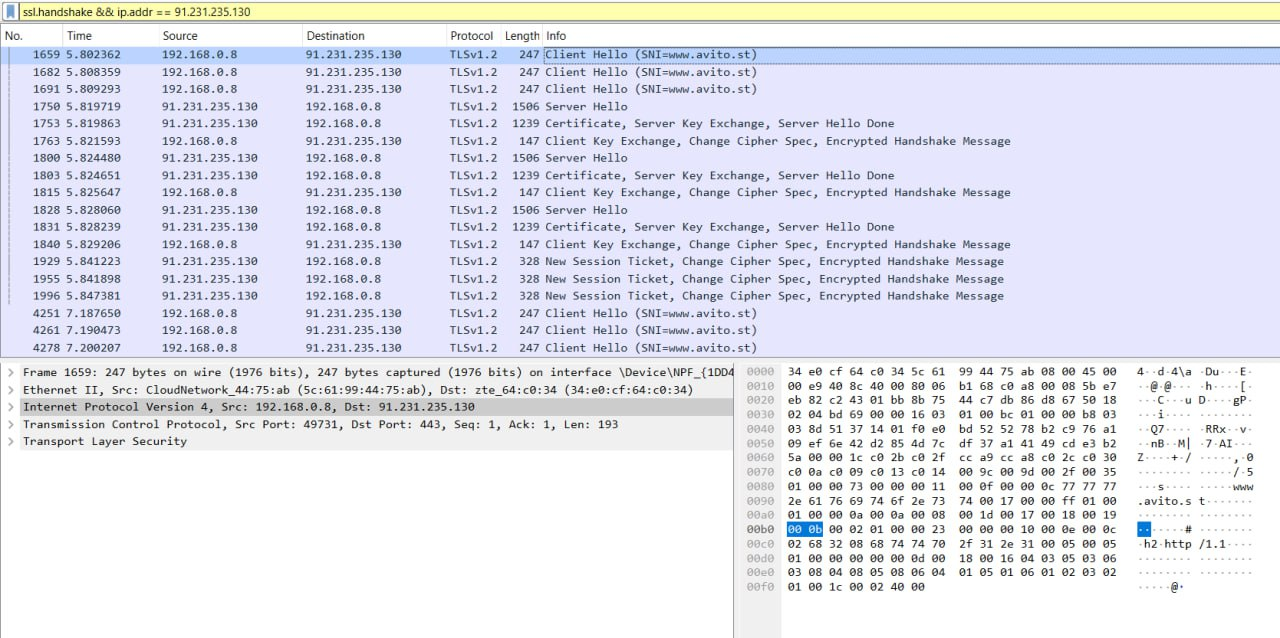
* **Имя сервера** из сообщения **Client Hello** поля SNI (Server Name Indication).
* **Характеристики сервера** попробуем извлечь из какого-нибудь пакета, передаваемого протоколом http/http2.
* **Используемую версию TLS** получим из сообщения **Server Hello** поля “version”
* **Алгоритм шифрования** (обмена ключами) получим из сообщения **Server Hello** поля “Cipher Suite”
* **Версию сертификата** получим из сообщения **Certificate** поля “signedCertificate->version”
* **Валидность сертификата** из сообщения **Certificate** поля “signedCertificate->validity->notBefore/notAfter.”
* **Информацию об удостоверяющем центре** получим из сообщения **Certificate** поля signedCertificate->issuer->rdnSequence->OrganizationName
* **Время установки** соединения получим, посчитав сколько времени прошло между отправкой сообщения **Client Hello** и сообщения **Finished.**

# **4 Ход лабораторной работы**

1. Для проведения лабораторной работы мною было выбрано два web-серверов сети Интернет: https://www.avito.ru/, https://group-kremny.ru/
2. Запустим Wiresharck и при помощи Web-браузера установим https соединение с выбранными серверами.
3. Попробуем перехватить защищённую информацию в программе Wireshark. Для начала находим рукопожатие с помощью фильтра, введя «ip.addr == (ip адрес) && ssl.handshake», и проверяем сообщение сервера.



Теперь можно приступать к анализу нашего соединения (согласно плану указанному в топике **теория**).

1. **https://www.avito.ru/ (91.231.235.130)** 

Начнем анализировать пакеты avito. Сразу видно TLS рукопожатие: Client Hello (SNI [www.avito.st](http://www.avito.st)), Server Hello, Certificate …

Далее приведу листинг информации необходимой для анализа:

* Имя сервера: [www.avito.st](http://www.avito.st)
* Версия TLS: TLS 1.2 (0x0301)
* Выбранный алгоритм шифрования: Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0xc02f)
* Версия сертификата: version: v3
* Видимость сертификата Validity:

notBefore: utcTime (0)

utcTime: 2018-11-21 00:00:00 (UTC)

notAfter: utcTime (0)

utcTime: 2028-11-21 00:00:00 (UTC)

* Удостоверяющий центр: GlobalSign
* Время установки соединения: 3.841223 - 3.813901 = 0.027322 (27 мс)

# **Выводы**

В ходе выполнения работы я сравнил 5 типов текстов и проанализировал процентное соотношение совпавших букв. Самый высокий процент совпадений у текстов на естественных языках, что так же мною было доказано в рассуждениях. Далее по частоте идут тексты из случайных слов. Самый низкий процент имеют тексты из случайных букв.