**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»**

Лабораторная работа № 4   
по курсу «Криптография»

Группа: М8О-307Б-21

Студент: Дубровин Д.К.

Преподаватель: А. В. Борисов

Оценка:

Дата: 16.09.2024

Москва, 2024

**ОГЛАВЛЕНИЕ**

[1 Тема 3](#_Toc158983147)

[2 Задание 3](#_Toc158983148)

[3 Теория 4](#_Toc158983149)

[4 Ход лабораторной работы 7](#_Toc158983150)

[5 Выводы 15](#_Toc158983151)

# **Тема**

Аутентификация с асимметричными алгоритмами шифрования

# **Задание**

Порядок выполнения лабораторной работы:

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 необходимы 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.

# **Теория**

В основе безопасной передачи данных лежит аутентификация, которая является процессом верификации удостоверения личности или других атрибутов субъекта. Это важно для установления доверия между двумя сторонами, обменивающимися информацией. Параллельно с аутентификацией работает авторизация, которая определяет, какие ресурсы или действия доступны аутентифицированному пользователю, основываясь на его правах и привилегиях.

Переходя к основам передачи данных, нельзя не упомянуть **HTTPS** (Hypertext Transfer Protocol Secure), который является зашифрованной версией протокола **HTTP**. HTTPS обеспечивает защиту передаваемой информации между клиентом и сервером, используя шифрование, которое реализуется на основе протокола **TLS** (Transport Layer Security), ранее известного как **SSL** (Secure Sockets Layer).

TLS использует комбинацию асимметричного и симметричного шифрования для обеспечения конфиденциальности, целостности и аутентификации данных. В асимметричном шифровании используются пары ключей: публичный ключ, который может быть широко распространён и использован для шифрования данных, и приватный ключ, который держится в секрете и используется для расшифровки данных.

**Этапы TLS-рукопожатия:**

1. **Начало обмена (ClientHello и ServerHello):**

Процесс начинается с того, что клиент отправляет сообщение **ClientHello** серверу, указывая список поддерживаемых версий TLS, предложенные наборы шифров (**cipher suites**), методы сжатия и случайно сгенерированное число (**client random**).

Сервер отвечает сообщением **ServerHello**, выбирая версию протокола и набор шифров из предложенных клиентом, а также отправляет своё случайно сгенерированное число (**server random**).

1. **Предоставление сертификата и аутентификация сервера:**

Сервер отправляет свой цифровой сертификат клиенту в сообщении **Certificate**. Этот сертификат обычно подписан надёжным центром сертификации (**CA**) и содержит открытый ключ сервера.

В зависимости от выбранного набора шифров сервер может также отправить сообщение **ServerKeyExchange**, которое необходимо для обмена ключами.

Сертификаты, играющие ключевую роль в аутентификации сервера во время TLS-рукопожатия, обычно следуют стандарту **X.509**. Этот стандарт определяет формат сертификатов, которые связывают публичный ключ с идентификатором его владельца и подписываются доверенным центром сертификации (CA). Сертификаты X.509 обеспечивают проверку подлинности, позволяя клиентам убедиться, что они соединяются с верным сервером, а не с злоумышленником, пытающимся перехватить или подменить трафик.

1. **Подтверждение клиента и предоставление pre-master secret:**

Если сервер требует аутентификации клиента, он может запросить цифровой сертификат клиента с помощью сообщения **CertificateRequest**.

Независимо от того, запрашивал ли сервер аутентификацию клиента, клиент отправляет **ClientKeyExchange** сообщение, содержащее **pre-master secret**, зашифрованное открытым ключом сервера, извлечённым из его сертификата.

1. **Завершение рукопожатия:**

Обе стороны генерируют сессионные ключи из **pre-master secret** и обмениваются сообщениями **ChangeCipherSpec**, сигнализирующими о готовности перейти к зашифрованной коммуникации.

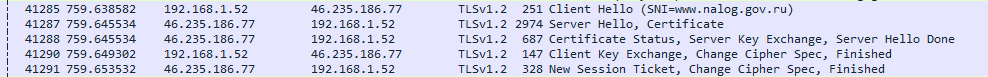
Рукопожатие завершается отправкой сообщений **Finished**, зашифрованных уже с использованием сессионных ключей, чтобы подтвердить, что процесс рукопожатия прошёл успешно и целостность канала связи не нарушена.

Весь этот процесс, начиная от аутентификации до конечного шифрованного обмена данными, позволяет обеспечить безопасную и защищённую среду в небезопасном мире интернет-коммуникаций, гарантируя, что конфиденциальная информация остаётся приватной и защищённой от неправомерного доступа.

# **Ход лабораторной работы**

С помощью интернета я нашел как сделать расшифровку с помощью **pre-master secret key** (см. список использованной литературы [1]). Подробности описывать не буду, просто в переменные среды добавил **SSLKEYLOGFILE** и потом в настройках **Wireshark** в пункте TLS добавил путь к логам, которые помогают расшифровывать TLS. Далее я начал тыкать такие сайты как **nalog.gov.ru** (гос. сайт) и **market.yandex.ru** (сайт организации Yandex).

1. **nalog.gov.ru (TLS 1.2)**



**ClientHello:**

* Время начала: 16 сентября 2024 года, 14:05:17.
* Исходный IP и порт: 192.168.1.52:55478.
* Целевой IP и порт: 46.235.186.77:443.
* Версия TLS: TLS 1.2.
* Предложенные шифры включают такой шифр как TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256.
* Другие расширения: SNI для www.nalog.gov.ru, поддержка расширенного мастер-секрета и другие расширения.

**ServerHello:**

* Версия TLS: TLS 1.2.
* Выбранный шифр: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256.
* Есть поддержка ALPN и форматов точек эллиптических кривых, есть сессионный билет.

**Сертификат:**

* Версия: v3, подтверждает использование самой последней версии X.509 сертификата.
* Алгоритм подписи: sha256WithRSAEncryption, соответствует текущим стандартам безопасности.
* Издатель: Let's Encrypt, известный удостоверяющий центр.
* Срок действия: с 7 августа 2024 г. по 5 ноября 2024 г., указывает на текущую валидность.
* Включают ключевое использование (цифровая подпись и шифрование ключей), расширенное использование ключа (аутентификация клиента и сервера), указатель ключа субъекта и издателя для связи с предыдущими и последующими сертификатами в цепочке, а также информацию о доступе к сертификату.

**Certificate Status, Server Key Exchange, Server Hello Done:**

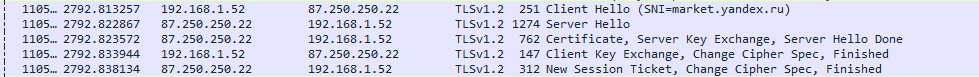
* Ответ OCSP подтверждает статус отзыва сертификата.
* Использует ECDHE для безопасного обмена ключами.
* Сервер подтверждает окончание процесса.

**Client Key Exchange, Change Cipher Spec, Finished:**

* Завершение обмена ключами клиентом.
* Переключение на защищённое шифрование.
* Завершение рукопожатия: подтверждение безопасного соединения.

От `**ClientHello**` до `**Finished**` заняло приблизительно 0.067 секунды, что свидетельствует о быстром процессе установления защищённого соединения.

1. **market.yandex.ru (TLS 1.2)**



**ClientHello:**

* Время начала: 16 сентября 2024 года, 15:10:08.
* Исходный IP и порт: 192.168.1.52:51553.
* Целевой IP и порт: 87.250.250.22:443.
* Версия TLS: TLS 1.2.
* Предложенные шифры включают такой шифр как TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256.
* Расширения: SNI с именем сервера market.yandex.ru, поддержка ALPN с протоколами h2 и http/1.1.

**ServerHello:**

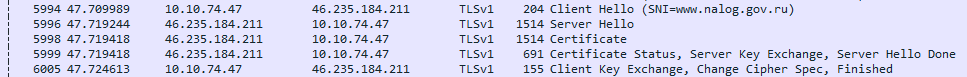
* Выбранный шифр: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256.
* Сессионный билет отсутствует, что может указывать на одноразовую сессию без возможности возобновления.

**Сертификат:**

* Версия: v3, подтверждает использование самой последней версии X.509 сертификата.
* Выдан для: market.yandex.com с дополнительными записями SAN для различных доменов Yandex, что позволяет использовать сертификат для защиты нескольких связанных доменов.
* Издатель: GlobalSign, уважаемый удостоверяющий центр, который является одним из ведущих мировых поставщиков цифровых сертификатов.
* Срок действия: Сертификат действителен с 8 апреля 2024 года по 8 ноября 2024 года, что подтверждает его актуальность и валидность на момент анализа.
* Алгоритм подписи: Использование алгоритма sha256WithRSAEncryption обеспечивает высокий уровень безопасности, используя современную хэш-функцию SHA-256 в сочетании с асимметричным шифрованием RSA для гарантии целостности данных и подтверждения подлинности.
* Другие важные расширения:
  + Basic Constraints: устанавливает, может ли сертификат быть использован как CA сертификат.
  + CRL Distribution Points (точки распределения списков отозванных сертификатов): адреса, по которым можно получить информацию об отзыве сертификата.
  + Authority Information Access (доступ к информации об удостоверяющем центре): содержит ссылки на службу OCSP и сертификаты CA, что позволяет автоматически проверять статус сертификата.

Время установления соединения составило примерно 0.042 секунды, что свидетельствует о хорошей производительности и быстром установлении защищённого соединения.

Давайте попробуем теперь TLS 1.0



1. **nalog.gov.ru (TLS 1.0)**

**ClientHello:**

* Время начала: 16 сентября 2024 года, 15:20:08.
* Версия TLS: TLS 1.0, что указывает на использование старой версии протокола, у которой известны уязвимости.
* Предложенные шифры включают TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA, TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA и другие, показывая предпочтение к шифрам ECDHE для обмена ключами.
* Расширения включают server\_name (SNI для www.nalog.gov.ru), extended\_master\_secret и другие.

**ServerHello:**

* Версия TLS остается TLS 1.0, сервер соглашается на устаревшую версию протокола.
* Выбранный шифр: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA, что соответствует сильному шифру в рамках используемой версии TLS.

**Сертификат:**

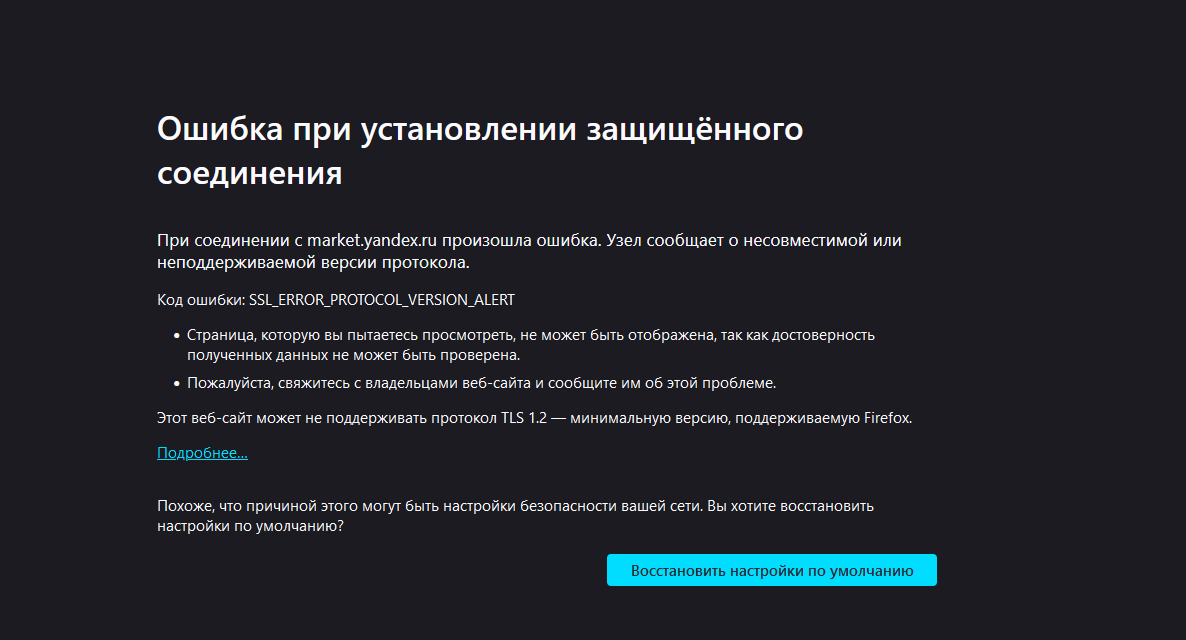
* Версия: v3.
* Алгоритм подписи: sha256WithRSAEncryption.
* Издатель: Let's Encrypt,
* Срок действия: Сертификат валиден с 7 августа 2024 года по 5 ноября 2024 года.
* Применяются расширения, такие как Key Usage и Extended Key Usage, подтверждающие назначение сертификата для аутентификации сервера.

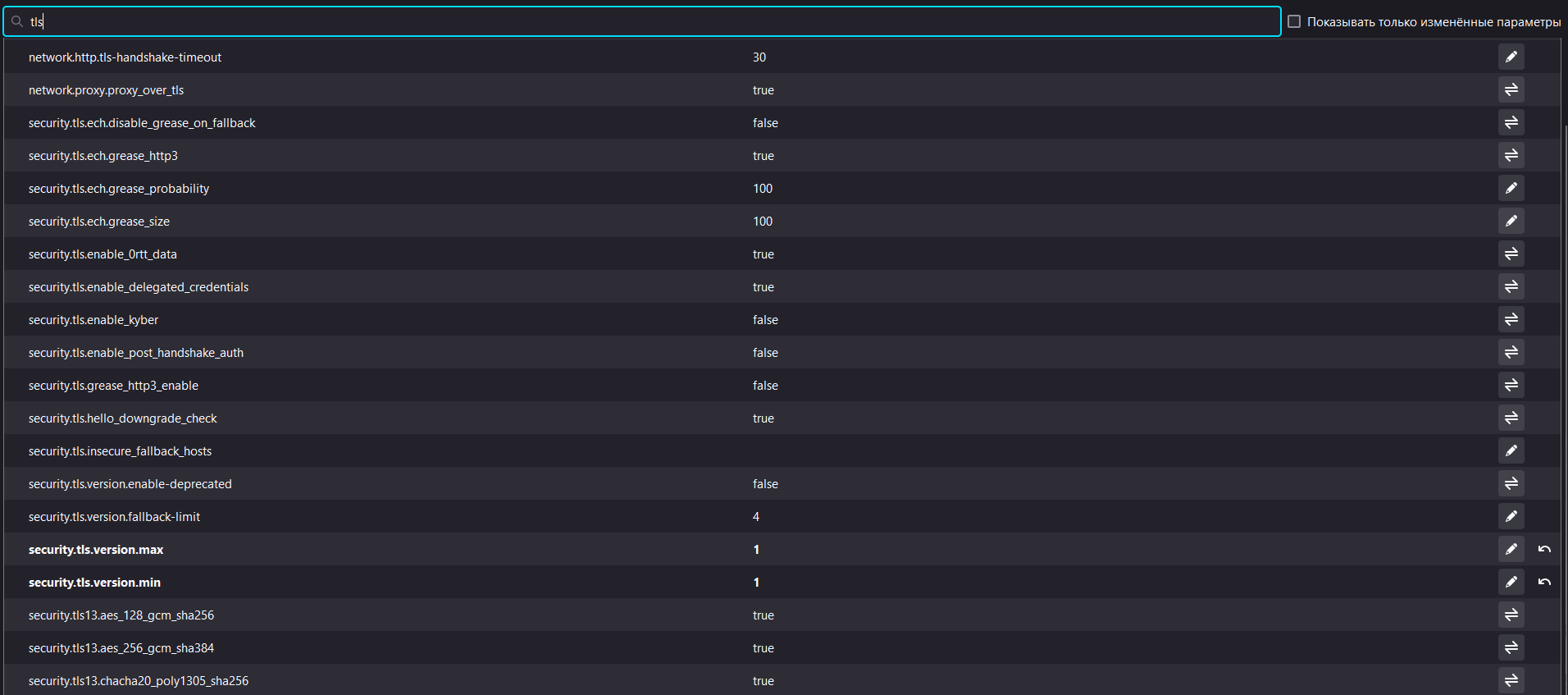
**Серверный ключ и завершение:**

Использование ECDHE подтверждает стремление к безопасному обмену ключами даже в рамках устаревшей версии TLS.

Время установления соединения составило примерно 0.0146 секунды, что свидетельствует о хорошей производительности и быстром установлении защищённого соединения.

1. **market.yandex.ru (TLS 1.0)**





Как мы видим, сайт **market.yandex.ru** не работает при TLS 1.0.

**Анализ**

И **nalog.gov.ru**, и **market.yandex.ru** используют TLS 1.2, что является хорошей практикой. Однако поддержка **nalog.gov.ru** соединений через TLS 1.0 может представлять собой риск безопасности. **market.yandex.ru** показывает более строгий подход к безопасности, не разрешая использование устаревших версий TLS.

Я думаю, что **nalog.gov.ru** рекомендуется отключить поддержку TLS 1.0 и, возможно, TLS 1.1, чтобы исключить потенциальные уязвимости и соответствовать лучшим практикам безопасности.

# **Выводы**

Это была очень интересная лабораторная работа, я познакомился с TLS, потыкал зашифрованные пакеты и проанализировал сайты государственной и организационной принадлежности

# **Список используемой литературы**

1. <https://www.comparitech.com/net-admin/decrypt-ssl-with-wireshark/>