

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-308Б-22

Студентка: К. А. Былькова

Преподаватель: А. В. Борисов

Оценка:

Дата: 25.05.2025

Москва, 2025

ОГЛАВЛЕНИЕ

1	Тема	3
2	Задание	3
3	Теория	4
4	Ход лабораторной работы.....	5
5	Выводы.....	12
6	Список используемой литературы	13

1 Тема

Аутентификация с асимметричными алгоритмами шифрования в сети Интернет

2 Задание

1. Выбрать не менее 3-ёх web-серверов сети Интернет различной организационной и государственной принадлежности.
2. Запустить Wireshark/tcpdump в режиме записи.
3. Используя Firefox/Chrome/Safari/ИнойБраузер установить https соединение с выбранным сервером и убедиться в установке соединения.
4. Сохранить данные необходимы для последующего сравнительного анализа:
 - * Имя сервера, его характеристики.
 - * Версия TLS.
 - * Выбранные алгоритмы шифрования.
 - * Полученный сертификат: версия, действителен ли сертификат, правильность 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 / 1.3 принудительно изменить параметры TLS (для соединения в Firefox на TLS 1.0 / 1.1 в браузере перейти по адресу “about:config” и изменить раздел SSL\TLS, security.tls.version.enable-deprecated) 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

TLS (Transport Layer Security — Протокол защиты транспортного уровня) — криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети Интернет. TLS использует асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

Данный протокол широко используется в приложениях, работающих с сетью Интернет, таких как веб-браузеры, работа с электронной почтой, обмен мгновенными сообщениями и IP-телефония (VoIP).

TLS даёт возможность клиент-серверным приложениям осуществлять связь в сети таким образом, что при возможном прослушивании пакетов нельзя осуществить несанкционированный доступ.

Основные шаги процедуры создания защищённого сеанса связи:

- клиент подключается к серверу, поддерживающему TLS, и запрашивает защищённое соединение;
- клиент предоставляет список поддерживаемых алгоритмов шифрования и хеш-функций;
- сервер выбирает из списка, предоставленного клиентом, наиболее надёжные алгоритмы среди тех, которые поддерживаются сервером, и сообщает о своём выборе клиенту;
- сервер отправляет клиенту цифровой сертификат для собственной аутентификации. Обычно цифровой сертификат содержит имя сервера, имя удостоверяющего центра сертификации и открытый ключ сервера;
- клиент, до начала передачи данных, проверяет валидность (аутентичность) полученного серверного сертификата относительно имеющихся у клиента корневых сертификатов удостоверяющих центров;
- для шифрования сессии используется сеансовый ключ. Получение общего секретного сеансового ключа клиентом и сервером проводится по протоколу Диффи-Хеллмана. Существует исторический метод передачи сгенерированного клиентом секрета на сервер при помощи шифрования асимметричной криптосистемой RSA (используется ключ из сертификата сервера).

4 Ход лабораторной работы

Для выполнения данной лабораторной работы я выбрала следующие веб-сервера:

- mainfo.ru — сайт с лекциями, лабораторными работами и другими материалами по курсу “Численные методы”
- faq8.ru — форум восьмого факультета МАИ
- yandex.ru — поисковая система Яндекс
- gosuslugi.ru — портал Госуслуг

Рассмотрим подробно первый веб-сервер:

mainfo.ru (84.252.139.64):

1. Открываем сервер в браузере
2. Находим пакеты нашего сервера (можно использовать фильтр: `tls.handshake.type == 1 || tls.handshake.type == 2 || tls.handshake.type == 11`, благодаря которому будут отображаться только Client Hello, Server Hello и Certificate соответственно). Также можно воспользоваться фильтром: `ip.addr == <IP_адрес> && tls`, указав IP-адрес нужного веб-сервера, и получить все TLS-сообщения от конкретного сервера.

622	41.795685	192.168.50.254	84.252.139.64	TLSv1.2	248	Client Hello (SNI=mainfo.ru)
626	41.822104	84.252.139.64	192.168.50.254	TLSv1.2	1514	Server Hello
627	41.822183	84.252.139.64	192.168.50.254	TLSv1.2	1514	Certificate

3. Откроем пакет Client Hello: здесь находится имя сервера – **mainfo.ru**

```
▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 189
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 185
    Version: TLS 1.2 (0x0303)
    Random: b52e976afe027c501f2251d2b1fbb586a83ddb4a0f5c8db0f141b4ce50b15e8
    Session ID Length: 0
    Cipher Suites Length: 28
    Cipher Suites (14 suites)
    Compression Methods Length: 1
    Compression Methods (1 method)
    Extensions Length: 116
    ▼ Extension: server_name (len=14) name=mainfo.ru
      Type: server_name (0)
      Length: 14
      ▼ Server Name Indication extension
        Server Name list length: 12
        Server Name Type: host_name (0)
        Server Name length: 9
        Server Name: mainfo.ru
```

4. Далее откроем Server Hello: здесь находим
- Версию протокола TLS, которая была использована при установке соединения: **TLS 1.2**. Можно заметить, что клиент предлагал версию ниже, а именно TLS 1.0;

- Сгенерированное значение Random, необходимое для генерации разделяемого ключа, используемого в алгоритме Диффи-Хеллмана;
- Набор шифров Cipher Suite:

TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, что говорит о том, что здесь используется алгоритм **Elliptic Curve Diffie-Hellman Ephemeral** (Диффи-Хеллмана на эллиптических кривых ephemeral). При использовании данного алгоритма новые значения Random будут генерироваться у сервера и клиента заново при каждой новой сессии, если прошло достаточно времени после предыдущей, потому что в противном случае будет использован протокол восстановления сессии). Для аутентификации сервера (подписи сертификата) используется **RSA**. Используется потоковый алгоритм шифрования данных **CHACHA20**. Алгоритм аутентификации сообщений **POLY1305**. Используемая хэш-функция – **SHA256**.

```

▼ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 110
  ▼ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 106
    Version: TLS 1.2 (0x0303)
    ▶ Random: 4a87b0f1842d813bf9f1ed0702b543c365b954b1cc7405f6c76ad0437d96989e
    Session ID Length: 32
    Session ID: e223e51c9cd53a4a710ecbe50b0b15c8748f3a9a1b3b3250ab70612f5b15650a
    Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc8)
    Compression Method: null (0)
    Extensions Length: 34
    ▼ Extension: renegotiation_info (len=1)
      Type: renegotiation_info (65281)
      Length: 1
      ▶ Renegotiation Info extension
    ▼ Extension: server_name (len=0)
      Type: server_name (0)
      Length: 0
    ▼ Extension: ec_point_formats (len=4)

```

5. Откроем Certificate

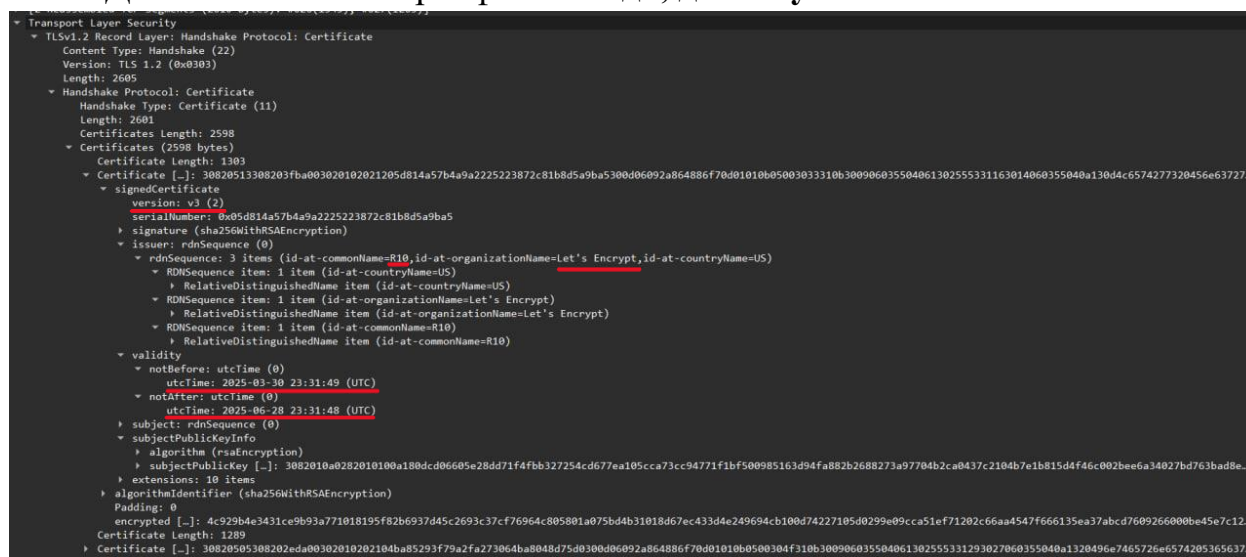
```

▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2605
    ▼ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2601
      Certificates Length: 2598
      ▼ Certificates (2598 bytes)
        Certificate Length: 1303
        ▶ Certificate [...]: 30820513308203fba003020102021205d814a57b4a9a225223872c81b8d5a9ba5300d06
          Certificate Length: 1289
        ▶ Certificate [...]: 30820505308202eda00302010202104ba85293f79a2fa273064ba8048d75d0300d06092a

```

Откроем первый сертификат:

- Версия — version: **v3 (2)**
- Номер — serialNumber: 0x05d814a57b4a9a2225223872c81b8d5a9ba5
- Алгоритм записи RSA — Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
- Удостоверяющий центр, выдавший сертификат — rdnSequence: 3 items (id-at-commonName=**R10**,id-at-organizationName=**Let's Encrypt**,id-at-countryName=US)
- Действителен ли сертификат — **да, действует**



6. Далее необходимо рассчитать время установки соединения:

От пакета ClientHello следуем по протоколу TCP до Finished и находим разницу во времени: $41.836665 - 41.795685 = 0.04098$

Для остальных веб-серверов я кратко покажу необходимую информацию.

faq8.ru (202.61.198.136):

- Имя сервера, его характеристики

```
▼ Server Name Indication extension
  Server Name list length: 10
  Server Name Type: host_name (0)
  Server Name length: 7
  Server Name: faq8.ru
```

- Версия TLS

```
TLSv1.2 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 81
```

- Выбранные алгоритмы шифрования

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

- Полученный сертификат: версия, действителен ли сертификат, правильность ключа, удостоверяющий центр

```
Certificate [...]: 3082051b30820403a003020102021205e14855ab9f4bec0216809050512d77b25d300d06092a864886f70d01010b05003033310b3009060355040613025553311630
▼ signedCertificate
  version: v3 (2)
  serialNumber: 0x05e14855ab9f4bec0216809050512d77b25d
  ▶ signature (sha256WithRSAEncryption)
  ▼ issuer: rdnSequence (0)
    ▼ rdnSequence: 3 items (id-at-commonName=R11,id-at-organizationName=Let's Encrypt,id-at-countryName=US)
      ▶ RDNSequence item: 1 item (id-at-countryName=US)
      ▶ RDNSequence item: 1 item (id-at-organizationName=Let's Encrypt)
      ▶ RDNSequence item: 1 item (id-at-commonName=R11)
  ▼ validity
    ▼ notBefore: utcTime (0)
      utcTime: 2025-04-16 17:26:40 (UTC)
    ▼ notAfter: utcTime (0)
      utcTime: 2025-07-15 17:26:39 (UTC)
  ▶ subject: rdnSequence (0)
  ▶ subjectPublicKeyInfo
  ▶ extensions: 10 items
  ▶ algorithmIdentifier (sha256WithRSAEncryption)
  Padding: 0
  encrypted [...]: 3c2dac077e8119c46606da33fb87ab292ba66697cf65c2f7c249f8aed30b22fdb59e46fd162eb77f6750639be4ad3e09fa26675e6d9d8d16680891b5ea6201f5450
Certificate Length: 1290
```

- Время установки соединения (от ClientHello до Finished)

$12.964248 - 12.803805 = \mathbf{0.160443}$

yandex.ru (77.88.44.55):

- Имя сервера, его характеристики

```
Server Name Indication extension
Server Name list length: 12
Server Name Type: host_name (0)
Server Name length: 9
Server Name: yandex.ru
```

- Версия TLS

```
TLSv1.2 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 72
```

- Выбранные алгоритмы шифрования

TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0хсса9)

- Полученный сертификат: версия, действителен ли сертификат, правильность ключа, удостоверяющий центр

```
Certificate [...] 308207a73082072da003020102020c6ba036a4f414f04a19bac93b300a06082a8648ce3d0403033050310b300906035504061302424531193017060355040a1310476c6f6261
  signedCertificate
    version: v3 (2)
    serialNumber: 0x6ba036a4f414f04a19bac93b
    signature (ecdsa-with-SHA384)
      Algorithm Id: 1.2.840.10045.4.3.3 (ecdsa-with-SHA384)
    issuer: rdnSequence (0)
      rdnSequence: 3 items (id-at-commonName=GlobalSign ECC OV SSL CA 2018,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)
        RDNSequence item: 1 item (id-at-countryName=BE)
          RelativeDistinguishedName item (id-at-countryName=BE)
        RDNSequence item: 1 item (id-at-organizationName=GlobalSign nv-sa)
          RelativeDistinguishedName item (id-at-organizationName=GlobalSign nv-sa)
        RDNSequence item: 1 item (id-at-commonName=GlobalSign ECC OV SSL CA 2018)
          RelativeDistinguishedName item (id-at-commonName=GlobalSign ECC OV SSL CA 2018)
    validity
      notBefore: utcTime (0)
        utcTime: 2025-04-16 08:08:59 (UTC)
      notAfter: utcTime (0)
        utcTime: 2025-10-14 20:59:59 (UTC)
    subject: rdnSequence (0)
    subjectPublicKeyInfo
      algorithm (id-ecPublicKey)
        Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
        ECParameters: namedCurve (1)
        Padding: 0
        subjectPublicKey: 04677f007c5be50f0662aee64db92ae15adef3d9c674a3776e041c2d4783228a7749b52dcb227b87ef8df0b698eeaa4e7d94dd08b89bdee3bd4307b5d1534e102
    extensions: 10 items
    algorithmIdentifier (ecdsa-with-SHA384)
    Padding: 0
    encrypted: 3065023100958e7807197df1b9d86237cf064d3da6e4a889cb1ae2568df999993d77308dc4c910a0e4e74bb6ebc121483abc26902302be9d5797290d39836bb6479a6671a6b5
```

- Время установки соединения (от ClientHello до Finished)

$5917.758812 - 5917.744802 = \mathbf{0.01401}$

gosuslugi.ru (109.207.1.118):

- Имя сервера, его характеристики

```
Server Name Indication extension
Server Name list length: 19
Server Name Type: host_name (0)
Server Name length: 16
Server Name: www.gosuslugi.ru
```

- Версия TLS

```
TLSv1.2 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 100
```

- Выбранные алгоритмы шифрования

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

- Полученный сертификат: версия, действителен ли сертификат, правильность ключа, удостоверяющий центр

```
▼ Certificate [...]: 3082065130820539a003020102020c131747f16fdd092dfc416a44300d06092a864886f70d01010b05003053310b300906035504061302424531193017060355040a1310476c6f62616c5369676e206c
  ▼ signedCertificate
    version: v3 (2)
    serialNumber: 0x131747f16fdd092dfc416a44
    signature (sha256WithRSAEncryption)
    issuer: rdnSequence (0)
      ▼ rdnSequence: 3 items (id-at-commonName=GlobalSign GCC R3 DV TLS CA 2020,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)
        ▼ RDNSequence item: 1 item (id-at-countryName=BE)
          RelativeDistinguishedName item (id-at-countryName=BE)
        ▼ RDNSequence item: 1 item (id-at-organizationName=GlobalSign nv-sa)
          RelativeDistinguishedName item (id-at-organizationName=GlobalSign nv-sa)
        ▼ RDNSequence item: 1 item (id-at-commonName=GlobalSign GCC R3 DV TLS CA 2020)
          RelativeDistinguishedName item (id-at-commonName=GlobalSign GCC R3 DV TLS CA 2020)
      ▼ validity
        notBefore: utcTime (0)
          utcTime: 2024-10-22 10:37:08 (UTC)
        notAfter: utcTime (0)
          utcTime: 2025-11-23 09:08:08 (UTC)
    subject: rdnSequence (0)
    subjectPublicKeyInfo
      algorithm (rsaEncryption)
      subjectPublicKey [...]: 3082010a0282010100c4fdc6d3811dd41124f70c603311480c710d98d548e305fc32f8fd25b7e83da21849c0f00621fb51e0f76699e74913c8832ef363cd698ef60865bd712bebf0
      extensions: 10 items
    algorithmIdentifier (sha256WithRSAEncryption)
    Padding: 0
    encrypted [...]: 790b107afe83911d9f538c2eb1070826b4a0333b9e577add5b2ce64ae0844f456c05ee0f3a8bd80fd98327df39a5857459c1c2af4a9fa03e75137e0bec41da94193c11c07f9f8c1a6aa5a170e12eace
    Certificate Length: 1204
```

- Время установки соединения (от ClientHello до Finished)

7687.004361 – 7687.015208 = **0.010847**

После того, как я принудительно изменила параметры TLS в браузере (установила версию TLS на 1, что соответствует TLS 1.0), соединение со всеми выбранными веб-серверами было потеряно, и в браузере появилась ошибка **SSL_ERROR_PROTOCOL_VERSION_ALERT** — сообщение, которое выдаёт браузер при попытке установить защищённое соединение с сервером, но не удаётся согласовать версию протокола TLS/SSL.

Error code: SSL_ERROR_PROTOCOL_VERSION_ALERT

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

This website might not support the TLS 1.2 protocol, which is the minimum version supported by Firefox.

[Learn more...](#)

Таким образом, все выбранные мною сервера используют современную версию протокола TLS (TLS 1.2), которая считается безопасной и поддерживает современные криптографические алгоритмы. Отсутствие поддержки устаревших версий TLS (TLS 1.0) снижает риск атак, таких как:

- POODLE (Padding Oracle On Downgraded Legacy Encryption): атака позволяет злоумышленнику расшифровать отдельные блоки зашифрованных данных, используя уязвимость в схеме заполнения (padding) блочного шифра;
- BEAST (Browser Exploit Against SSL/TLS): атака, позволяющая расшифровать HTTPS-трафик, если используется режим шифрования CBC (Cipher Block Chaining).

Алгоритмы шифрования у серверов являются современными и безопасными, соответствуют рекомендациям по защите информации и обеспечивают высокий уровень защиты передаваемых данных.

Также все сертификаты корректно настроены, содержат все необходимые данные и подписаны надежными удостоверяющими центрами, что ещё раз гарантирует безопасность передаваемых данных.

5 Выводы

Был проведен анализ 4-х веб-серверов (mainfo.ru, faq8.ru, yandex.ru и gosuslugi.ru), используя инструмент Wireshark. Благодаря этому, был сделан вывод о том, что настройка всех веб-серверов соответствует современным требованиям безопасности. Все веб-сервера используют TLS 1.2, современные алгоритмы шифрования и надежные сертификаты, что гарантирует защиту передаваемых данных.

В ходе выполнения данного курсового проекта был освоен и применён один из самых мощных инструментов для анализа сетевого трафика — Wireshark. Благодаря его функциональным возможностям появилась возможность глубже понять процессы, происходящие при сетевом взаимодействии. Были детально изучены структура и содержание передаваемых данных. Были приобретены практические навыки захвата, фильтрации и анализа сетевых пакетов, что позволило наблюдать за реальным трафиком.

Также был изучен протокол TLS. Теперь известно, что TLS 1.0 — это устаревший протокол безопасной передачи данных по сети, и на данный момент он считается небезопасным и не рекомендован к использованию. А TLS 1.2, наоборот, является современной и безопасной версией протокола. Именно данная версия широко используется по всему миру и считается надежной.

В результате выполнения данной лабораторной работы были приобретены навыки, которые в дальнейшем будут полезны для выполнения других работ и курсовых проектов.

6 Список используемой литературы

1. Применко Э. А. Алгебраические основы криптографии — М.: Книжный дом “ЛИБРОКОМ”, 2013. — 289 с.
2. Протокол TLS: что это, зачем он нужен и как работает // SkillBox — URL: <https://skillbox.ru/media/code/protokol-tls-cto-eto-zachem-nuzhen-i-kak-rabotaet/> (дата обращения: 25.05.2025)
3. Wireshark — подробное руководство по началу использования // Хабр — URL: <https://habr.com/ru/articles/735866/> (дата обращения: 25.05.2025)
4. Методы шифрования в TLS: как обеспечивается безопасность данных в интернете // Хабр — URL: <https://habr.com/ru/companies/T1Holding/articles/893188/> (дата обращения: 25.05.2025)