

# Отчет о прохождении курса Stepik

---

Дашкина Анита Тагировна

17 мая, 2025, Москва, Россия

Российский Университет Дружбы Народов

## Цели и задачи

---

Ознакомиться с основами кибер безопасности по нескольким пунктам, а именно:

1. Безопасность в сети
2. Защита ПК/Телефона
3. Криптография

Ознакомиться с кибербезопасностью

# **Выполнение лабораторной работы**

---

# Первый раздел

**stepik**

Основы кибербезопасности  
Прогресс по курсу: 33/33

1 0 курсе

1.1 0 курсе

2 Безопасность в сети

2.1 Как работает интернет...

2.2 Персонализация сети

2.3 Браузер TOR. Аноним...

2.4 Беспроводные сети Wi-Fi

3 Защита ПК/телефона

3.1 Шифрование диска

3.2 Пароли

3.3 Фишинг

3.4 Вирусы. Примеры

3.5 Безопасность мессенд...

4 Криптография на практи...

4.1 Введение в криптогра...

4.2 Цифровая подпись

4.3 Электронные платежи

4.4 Блокчейн

2.1 Как работает интернет: базовые сетевые протоколы 15 из 15 шагов пройдено 9 из 9 баллов получено

Вы прошли больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

Итак, протокол TCP. Протокол TCP - это самый популярный протокол транспортного уровня, и основная его задача - обеспечить надежную передачу данных между процессами на разных машинах, то есть между моим запросом, набранным в браузере, веб-серверу, где этот веб-сайт лежит.

Для адресации моего запроса в протоколе TCP используются порты. Порт - это число от 1 до 65535, и каждое сетевое приложение имеет свой собственный порт. Например, веб-браузеры, используя на прикладном уровне протоколы HTTP или HTTPS, используют порты с номерами 80 или 443. Другие приложения могут использовать другие порты, так, например, почтовые сервисы используют порт 25 для отправки сообщений, для отправки имейлов. А, например, какие-то приложения для видеоконференций, видеосвязи используют свои порты для передачи данных. За счет чего протокол TCP обеспечивает надежную передачу данных? От протокола верхнего уровня (то есть протокола прикладного уровня) TCP получает некий поток данных. То есть мой запрос, например, набранная в браузере ссылка, преобразуется в поток данных, который получает протокол TCP.

Протокол сегментирует эти данные на какие-то кусочки фиксированной длины и затем поочередно отправляет эти сегменты данных от моей машины серверу, которому и сделала запрос. При этом он добавляет к каждому сегменту его порядковый номер. Например, первый сегмент имеет порядковый номер 001, и он отправляется кусочком целиком к серверу. И далее моя машина, мой браузер ожидает подтверждения получения этого сегмента от сервера. Как только подтверждение получено, отправляется второй сегмент, и так далее, пока все данные не будут отправлены от моей машины к получателю, к серверу. Тогда протокол TCP считается выполненным, и протокол TCP заканчивает свою работу.

Откуда протокол TCP знает, как доставить пакет и этот сегмент от моей машины к серверу? Здесь принимает участие уже протокол сетевого уровня, протокол IP. Его задача - обеспечить корректную маршрутизацию, то есть доставку этого сегмента данных по нужному адресу. Что такое адрес? Мы помним, что протокол TCP работает по портам, но это еще не все. Порт - это адрес приложения, то есть то, по какому адресу работает конкретное приложение. А вот адрес моей машины, ну или адрес моего роутера, который обеспечивает мне интернет - это уже адресация протокола IP. Вообще, адрес устройства в интернете очень похож на обычный адрес дома, на который мы получаем письма.

Существуют две версии адресации в протоколе IP. Популярный на сегодняшний день - это версия 4 адресации (IPv4), и этот адрес состоит из большого набора чисел, нежели порт в TCP протоколе, а именно это 4 числа от 0 до 255. Например, адрес IPv4 может выглядеть вот так: 192.168.1.4. Первые три числа - это номер сети. Если продолжать сравнение с почтовым адресом дома, то это по сути индекс и название улицы. Последняя цифра 4 - это номер хоста или номер дома. Хост - это, например, то устройство, которое раздает мне интернет, то есть мой роутер. Не всегда номер хоста - это последняя цифра из четырех. Иногда, как правило, в больших сетях, корпоративных сетях, большие машины, большие компьютеры подключены к сети, нежели 255, тогда номер хоста в этой сети занимает еще и вторую справа цифру, то есть в нашем случае единицу. Определяет, где у нас в адресе есть номер сети, а где у нас номер хоста, маска сети. Это еще один номер, который добавляется к адресу IPv4, это просто указатель того, где происходит разделение между номером сети и номером хоста.

Узнать свой IP-адрес может любой человек, подключенный к сети Интернет. В интернете много различных сервисов, которые предоставляют эту услугу. Ну, например, вот этот сервис мне расскажет, какой у меня IP-адрес [на слайде - 95.221.26.172], состоящий опять же из 4 чисел от 0 до 255. Что мне может рассказать мой IP-адрес? Поскольку мой IP-адрес мне выдается моим локальным провайдером, то, как правило, у провайдера на один большой регион одинаковый номер сети, то, что называется начальным адресом, то есть номер, индексация и, скажем так, город.

Рис. 1: Ознакомление с Интернет-безопасностью

# Тесты первого раздела

The screenshot shows a test interface for the topic "Как работает интернет: базовые сетевые протоколы". At the top, there is a progress bar with 15 steps, where the 15th step is highlighted. The current question is "2.1 Как работает интернет: базовые сетевые протоколы". The progress indicates "15 из 15 шагов пройдено" and "9 из 9 баллов получено". Below the progress bar, there is a feedback section with the text "Вы прошли больше 80% курса, оставьте отзыв" and buttons "Оставить отзыв" and "Нет, спасибо". The question text is "В фазе "рукопожатия" протокола TLS не предусмотрено". Below the question, there is a prompt "Выберите один вариант из списка". The options are: "Хорошая работа." (selected), "формирование общего секретного ключа между клиентом и сервером", "аутентификация (как минимум одной из сторон)", "выбираются алгоритмы шифрования/аутентификации", and "шифрование данных". To the right of the options, there is a statistics box showing "Верно решил 931 учащийся" and "Из всех попыток 44% верных". Below the options, there are two buttons: "Следующий шаг" and "Решить снова". At the bottom, there is a section "Ваши решения" showing "Вы получили: 1 балл". The footer shows a thumbs up icon with "107", a thumbs down icon with "14", and "Шаг 15". There is also a "Следующий шаг" button with a right arrow.

2.1 Как работает интернет: базовые сетевые протоколы 15 из 15 шагов пройдено 9 из 9 баллов получено

Вы прошли больше 80% курса, оставьте отзыв [Оставить отзыв](#) [Нет, спасибо](#)

В фазе "рукопожатия" протокола TLS не предусмотрено

Выберите один вариант из списка

☒ Хорошая работа.

Верно решил 931 учащийся  
Из всех попыток 44% верных

☐ формирование общего секретного ключа между клиентом и сервером

☐ аутентификация (как минимум одной из сторон)

☐ выбираются алгоритмы шифрования/аутентификации

☒ шифрование данных

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: 1 балл

👍 107 👎 14 Шаг 15 [Следующий шаг](#)

Рис. 2: Прохождение тестов

## Раздел второй

### 3.1 Шифрование диска 3 из 5 шагов пройдено 3 из 3 файлов получено

Выполнили больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

В этой лекции мы с вами поговорим о том, как шифруются носители информации и зачем это нужно. Зачем же шифровать жесткий диск, спросите вы. И ответ на этот вопрос довольно прост - шифровать жесткий диск нужно для того, чтобы избежать утечки наших персональных данных в случае утери или кражи нашего ноутбука. По статистике, это одна из основных причин утечки каких-то конфиденциальных данных типа наших паролей и каких-то документов, и шифрование жесткого диска позволяет избежать этих утечек. То есть, если злоумышленник получит физический доступ к вашему компьютеру или ноутбуку, не зная пароля, то есть не зная ключа шифрования, он не сможет получить доступ к вашим файлам. Вообще, эта политика шифрования жесткого диска часто является обязательной для таких серьезных компаний, как Яндекс, Google или в некоторых университетах. Сотрудники обязаны на своем корпоративном ноутбуке шифровать жесткий диск.

Рассмотрим, как это работает. Если не углубляться в детали, то шифрование жесткого диска происходит в три этапа. На самом первом этапе пользователь с помощью какой-то из утилит (про утилиты мы поговорим позднее) генерирует ключ для шифрования. Конечно, генерирует не вы сами, за вас это делает программа. Для того, чтобы этот ключ получить потом при дешифровании, вы должны запомнить пароль. Пароль - позволяет потом разблокировать этот ключ шифрования и дешифрования. Иногда вас спросят запомнить довольно длинный пароль. Вот, например, у меня на картинке пример пароля, предложенного утилитой шифрования в MacOS, и этот пароль вам нужно сохранить (да и как его хранить, мы сейчас поговорим подробнее). Сейчас стоит запомнить, что на первом шаге мы генерируем ключ. Что мы с ним потом делаем? Потом мы с этим ключом шифруем. Программа берет этот ключ, берет наши данные, будь то весь жесткий диск или какой-то его сегмент или может быть даже загрузочный сегмент, и шифрует данные с помощью ключа. На выходе мы с вами получаем данные в зашифрованном виде. Когда мы с вами хотим получить доступ к нашим зашифрованным данным, жесткому диску или загрузочному сектору, мы дешифруем, при этом работает программа, алгоритм дешифрования берет то же самый ключ, зашифрованные данные и выдает нам дешифрованные данные: либо наши файлы в открытом виде, либо, если мы шифровали загрузочный сектор операционной системы, нам загружает эту операционную систему.

Поговорим немного о деталях, о том, как происходит шифрование. Шифрование больших объемов данных, например, жесткого диска или сегмента жесткого диска или какой-то большой флешки, осуществляется с помощью симметричного шифрования, как правило, алгоритма AES. Это американский стандарт симметричного шифрования, он также используется для конфиденциальной передачи данных по сети. Это эффективный алгоритм, который реализован в процессоре быстро, то есть на аппаратном уровне. Благодаря тому, что это хороший алгоритм, пользователь практически не наблюдает задержек в работе, то есть данные шифруются/дешифруются быстро. Как правило, это происходит на заднем фоне, мы можем при этом работать на компьютере, будут происходить какие-то параллельные операции на шифрование и дешифрование.

На современных процессорах семейства Intel встроен TPM модуль или TPM криптопроцессор. TPM - это аббревиатура от Trusted Platform Module, и этот криптопроцессор позволяет, во-первых, эффективно шифровать, то есть на нем аппаратно реализован алгоритм шифрования AES, а во-вторых, позволяет хранить этот самый ключ, с помощью которого мы шифовали и дешифровали, на специальном сегменте блока данных, который защищен от злоумышленников.

Как я уже сказал, шифровать можно не только жесткий диск, где мы храним файлы, можно шифовать и загрузочный сектор диска. Это тот сектор, который включается сразу после того, как мы ставим компьютер. Компьютер считывает данные в сегмент, который хранит у себя загрузочные файлы операционной системы, и мы получаем либо свой рабочий стол, либо запрос на авторизацию для доступа к нашему рабочему столу. Вот для того,

Рис. 3: Знакомство с шифрованием



# Раздел второй

## 3. Защита ПК/телефона

15/15



3.1 Шифрование диска

3 / 3



3.2 Пароли

6 / 6



3.3 Фишинг

2 / 2



3.4 Вирусы. Примеры

2 / 2



3.5 Безопасность мессенджеров

2 / 2

**Рис. 4:** Успешное прохождение  
всех тестов

# Раздел третий

## 4.1 Введение в криптографию 7 из 7 часов пройдено 5 из 5 баллов получено

Вы прошли больше 95% курс, оставьте отзыв

Оставить отзыв Нет, спасибо

Довольно часто люди, даже те, которые работают в IT-сфере, путают основные криптографические понятия. Они иногда не отличают цифровую подпись от шифрования, от аутентификации, от хэш-функции. Моя сегодняшняя цель – это научить вас отличать основные криптографические протоколы, их еще называют примитивами, а именно – симметричное шифрование, аутентификация, цифровая подпись и хэширование.

Для того, чтобы мы с вами говорили на одном языке и чтобы вы не путались в этих понятиях, имеет смысл структурировать их и определить, зачем они нужны и какую функцию они несут. Эти данные протоколы – подписка, симметричное шифрование, аутентификация – имеют равную роль, и поэтому они такие по-разному строятся. Как они строятся, мы рассказывать в этом курсе не будем, это довольно сложные математические объекты, и аппарат, который нужно иметь, чтобы понимать, как они устроены, очень сложный. Но зачем эти примитивы созданы и какую цель они преследуют, мы сейчас с вами разберем.

Вообще, если начинать категоризировать протоколы в криптографии, то мы с вами встретим два раздела: первый раздел – это симметричная криптография, второй раздел – это асимметричная криптография. Специфическое свойство симметричной криптографии состоит в том, что она включает шифр протоколы, где две или более стороны имеют общие секретные ключи, поэтому она и называется симметричной. К таким протоколам относятся симметричное шифрование и некоторые протоколы аутентификации. Часто симметричный протокол довольно сложно построить, сложно установить потенциальный канал связи, исключительно основываясь на симметричных протоколах, поэтому мы можем стандартизировать общий секретный ключ, то есть, либо как-то физически встретиться с другим человеком и с другим устройством, либо что-то такое сделать, чтобы мы стандартизировали общий секрет. И легитимные решения этого вопроса являются протоколами асимметричной криптографии.

В асимметричной криптографии (еще называют криптографией с открытым ключом) у каждой из сторон есть пара ключей: открытый ключ и секретный ключ. Открытый ключ публикуется в открытом доступе, а закрытый или секретный ключ стороны хранит у себя. К протоколам асимметричной криптографии относят электронно-цифровую подпись и протокол передачи общего ключа – это этот протокол, который позволяет нам не общаться физически друг с другом, а установить и выдать общий секретный ключ.

Как правило, в секретных протоколах, в современных конфиденциальных коммуникациях используются именно симметричная криптография и асимметричная криптография. Это сделано, в частности, для того, чтобы сделать конфиденциальную коммуникацию эффективной, так как симметричные примитивы обычно являются более эффективными по времени, чем асимметричные примитивы.

Есть еще один примитив, который выходит за рамки симметричной и асимметричной криптографии, поскольку он бесключевой. Иным и, наверное, единственным примером такого криптографического примитива является криптографическая хэш-функция. Вообще и чаще есть просто хэш-функция, а есть криптографическая хэш-функция. Криптографическая хэш-функция берет на вход произвольный объем данных, то есть какие-то байты и выдает на выходе фиксированный строок, например длины  $n$ . Важно, что, как правило, функция сжимает данные: она берет большой набор данных и выдает потом маленькое фиксированное значение.

Важное свойство криптографической хэш-функции, то, что делает её криптографической – это стойкость к коллизиям. Что такое коллизия? Коллизия – это два разных входа в хэш-функцию, которые дают одинаковый выход. То есть это две разные строки например  $k$  и  $j$ , где  $k \neq j$ , такие, что значение хэш-функции на них совпадают, то есть  $h(k) = h(j)$ . Это важное свойство отличает криптографическую функцию от некриптографической. Можно доказать, что этого достичь с нами не удалось, что из этого свойства коллизии следуют другие важные свойства, а именно то, что

Рис. 4: Знакомство с криптографией

## 4. Криптография на практике

16/16



4.1 Введение в криптографию

5 / 5



4.2 Цифровая подпись

5 / 5



4.3 Электронные платежи

3 / 3



4.4 Блокчейн

3 / 3

**Рис. 5: Успешное  
прохождение всех тестов**

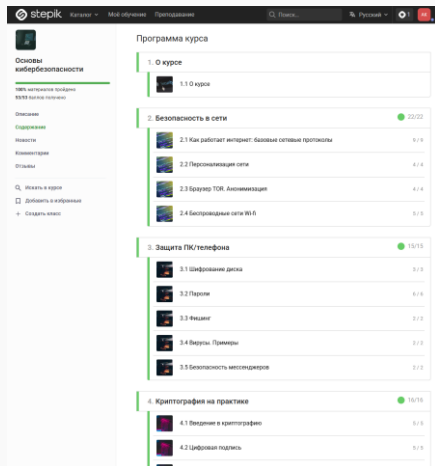
## **Выводы**

---

# Результаты выполнения лабораторной работы

В результате выполнения работы этот курс научил, как защищаться от цифровых угроз в современном мире. Я узнал, что такое кибербезопасность и почему она важна для каждого. Раздел криптографии объяснит, как работают шифрование и цифровые подписи, чтобы ваши сообщения и данные оставались конфиденциальными. В части защиты сетей расскажут, как безопасно пользоваться Wi-Fi, распознавать фишинг и защищаться от хакерских атак с помощью VPN и фаерволов. Веб-безопасность научит избегать опасных сайтов и защищать свои аккаунты от взлома. Раздел защиты ПК и телефонов даст практические советы: ставить антивирусы, вовремя обновлять программы и не качать подозрительные файлы. Вы поймете, как социальная инженерия манипулирует людьми (например, через фальшивые звонки "из банка"), и научитесь не попадаться на уловки мошенников. Если что-то пошло не так, раздел реагирования на инциденты подскажет, как действовать при утечке данных или заражении вирусом. Курс также затрагивает законы о персональных данных и этические вопросы — чтобы вы не только защищали себя, но и не нарушали чужие права. Главный вывод: безопасность начинается с простых шагов — сложных паролей, критического мышления и базовых цифровых привычек.

# Результаты выполнения лабораторной работы



**Рис. 5: Сертификатов не дают вот доказательство прохождения**