Презетация по прохождению дополнительного курса

Основы информационной безопасноти

Соловьев Богдан НКАбд-04-23

Содержание

Вторая глава: Безопасность в сети

А происходит между этими двумя событиями работа так называемых сетевых протоколов. Сетевой протокол - это некая последовательность правил, по которым, во-первых, устанавливается соединение между устройствами сети, то есть между вашим роутером, который, скорее всего, стоит у вас дома, и другими устройствами сети. И во-вторых, когда соединение установлено, начинается обмен данных, то есть с вашей стороны идет запрос в Сеть на открытие страницы поисковика, а к вам из Сети приходит страница этого поисковика.

Современные сетевые протоколы удобно описывать в виде модели протоколов, и современный Интернет работает в так называемой модели TCP/IP. Название TCP/IP состоит из двух самых популярных сетевых протоколов: это протокол TCP и протокол IP. Мы более подробно изучим эти протоколы далее в этой лекции. Сейчас важно понимать следующее: протокол TCP, если переводить его с английского, означает протокол управления передачей, и этот протокол отвечает за формирование пакетов данных. Все данные, которые передаются по сети, сформированы в некие пакеты, то есть в кусочки данных, в сегменты. И все данные, которые мы отправляем или получаем, мы получаем сегментировано по пакетам.

Второй протокол - протокол IP, ответственный за передачу этих пакетов от одной машины к другой машине. Иными словами, он ответственен за корректную адресацию пакетов в Сети.

В модели TCP/IP существует несколько уровней, а именно 4. И сейчас мы рассмотрим последовательно все четыре уровня модели TCP/IP. На самом верхнем уровне, прикладном работают пользовательские программы, и задача прикладного уровня - обеспечить доступ для этих пользовательских программ к услугам Интернет.

Мы с вами пользуемся достаточно большим спектром программ в интернете, и каждая программа использует свой протокол. Например, браузеры и веб-страницы используют протокол HTTP или его современную версию HTTPS. Ни для кого не секрет, что URL странички начинается с HTTP или HTTPS. S означает, что мы общаемся с веб-страницей по зашифрованному каналу. И более подробно мы рассмотрим протокол HTTPS в следующей лекции. Вообще, протокол HTTP(S) является примером протокола прикладного уровня, по которому передаются веб-страницы. Кроме того, мы с вами можем скачивать или загружать какие-то файлы: для этого часто используется протокол FTP.

Кроме того, мы с вами пользуемся почтой, и для доставки и отправки имейлов существуют другие протоколы - протокол SMTP или протокол POP3. И в зависимости от того, что мы делаем интернете, работает тот или иной протокол прикладного уровня.

Все эти протоколы прикладного уровня работают над транспортным уровнем, это следующий уровень в модели TCP/IP. Транспортный уровень обеспечивает передачу данных между процессами на одной машине или хосте (host). Хост - это устройство, которое подключено к интернету. Это может быть компьютер, смартфон. И транспортный уровень отвечает за корректное распределение пакетов между программами. То есть он знает, что какие-то данные пришли нам в Skype, какие-то данные пришли нам на почту, какие-то данные должны прийти в

Так, например, почту или веб-страницы мы получаем по протоколу TCP. Мы знаем, что, если письмо пришло, то оно пришло, а если оно не пришло, если что-то произошло в сети, мы получаем уведомление, что письмо наше не пришло. TCP/IP как раз таки отвечает за надёжную доставку и отправку пакетов. Второй пример протокола транспортного уровня – это протокол UDP. В отличие от протокола TCP, он не обеспечивает надежную передачу данных, однако он обеспечивает скорость передачи данных, в то время как протокол TCP работает медленнее. Вы спросите: зачем нам нужно ненадёжное соединение? Как правило, протокол UDP используется, когда нам нужно передать быстро какие-то данные. Например, по протоколу UDP работает видеосвязь или голосовая связь, то есть Skype,

Zoom или какие-то другие программы, для которых важно обеспечить скорость соединения, но необязательно надежность. Мы знаем, что при звонке мы можем зависнуть, изображение может заморозиться, и потом оно отвиснет, и мы сможем продолжить наше общение, но какие-то данные будут утеряны. Но, с другой стороны, мы передаем достаточно большое количество информации (а видеопоток - это большое количество информации) довольно быстро благодаря протоколу UDP.

Транспортный уровень работает над следующим уровнем модели TCP, так называемым сетевым или межсетевым уровнем. В этом уровне принимает участие не только программа, не только наша машина, но уже наш роутер, то есть то устройство, к которому мы подключаемся, чтобы получить интернет.

Сетевой уровень ответственен за передачу данных между различными физическими сетями; так, например, мы можем подключиться с компьютера, подключаясь через WiFi-сеть, к другому компьютеру, который, быть может, подключен через проводной интернет. Мы знаем, что такое возможно, хотя эти две машины работают в разных сетях. Кроме того, на сетевом уровне работает протокол IP версии 4 (IPv4) или IP версии 6 (IPv6) или IPSec - это безопасная версия протоколов IP. И протокол IP нам обеспечивает маршрутизацию, то есть доставку пакета или данных от одной машины к другой машине.

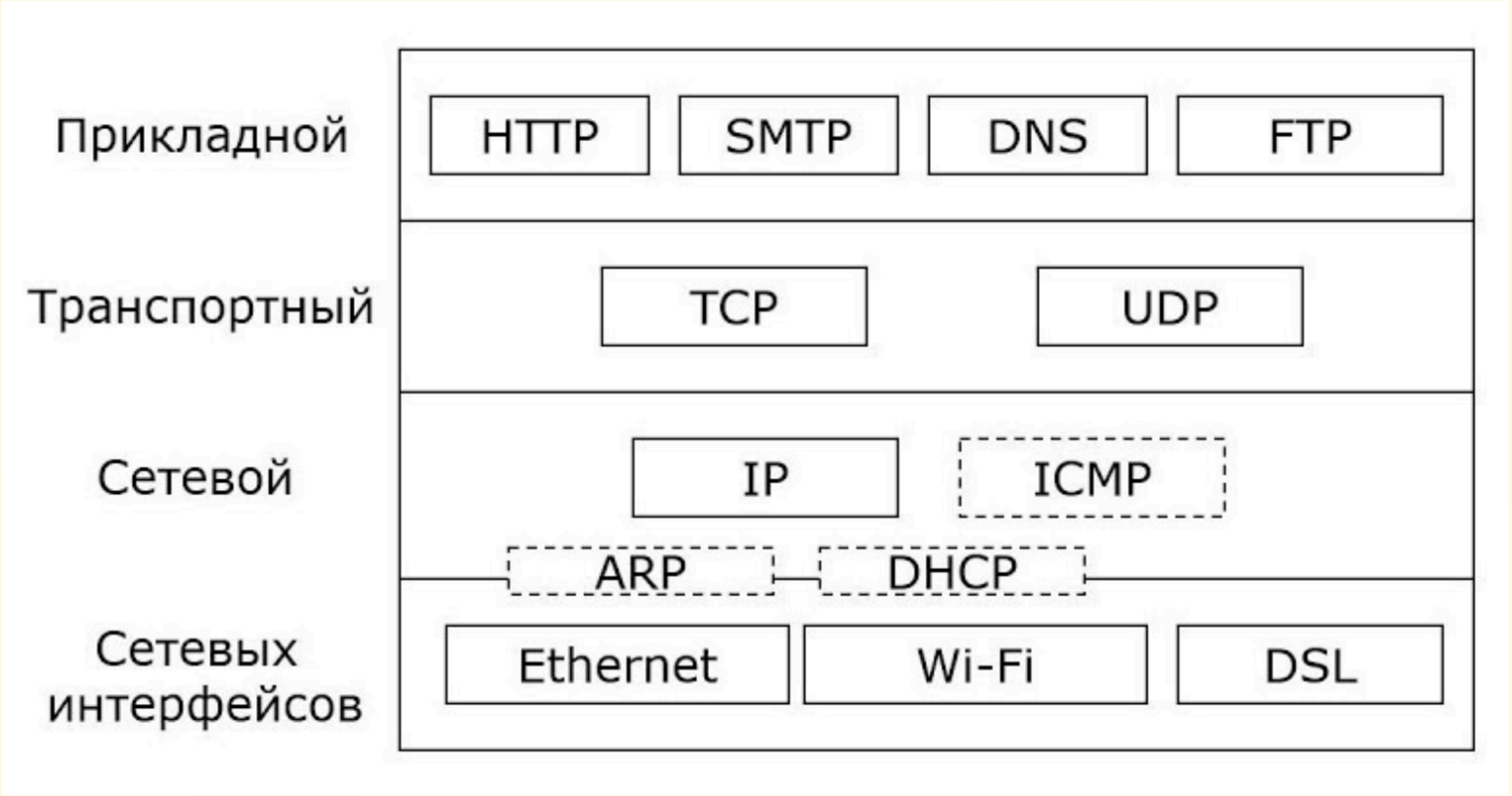
Сетевой уровень работает над так называемым канальным уровнем или самым низким уровнем модели TCP/IP. И в этой модели работает уже физика.

Здесь происходит физическая передача данных. И здесь мы уже будем говорить о пропускной способности канала, об обеспечении помехоустойчивости, и основные примеры протоколов канального уровня непосредственно связаны с тем, как мы передаем данные. Например, мы можем передавать данные по проводу, мы можем подключить свой компьютер по проводному интернету, и от нашего устройства, которое нам дает интернет, мы можем подключиться через провод к нашему компьютеру. В таком случае пример протокола канального уровня будет называться Ethernet. Наверное, более популярный на сегодняшний день пример протокола канального уровня - это WiFi или WLAN.

Но есть на самом деле еще много разных протоколов канального уровня, которые все-таки менее популярны. Основные два - это Ethernet и WLAN.

Вот эти четыре уровня модели TCP/IP и происходят между тем, как мы запросили на нашей машине открыть какую-то веб-страницу. То есть наш запрос происходит через прикладной уровень от браузера в транспортный уровень, который понимает, что это идет от браузера, в сеть, которая нам говорит, что мы хотим получить такой-то веб-сайт. И мы передаем всю эту информации в виде бит, в виде потока данных в сеть. И для того, чтобы получить назад ответ, происходит ровно тот же самый протокол, тот же самый алгоритм.

По канальному уровню наш запрос передается в сетевой уровень, затем в транспортный и прикладной, где на сервере, на котором находится наш запрос, где лежит yandex.ru, запрос обрабатывается, и мы получаем с вами на нашем компьютере и в нашем браузере страницу yandex.ru. Далее мы с вами рассмотрим подробно, как работают два основных протокола модели TCP/IP. Это протокол TCP и протокол IP.


Прикладной
HTTP
SMTP
DNS
FTP
Транспортный
TCP
UDP
Сетевой
IP
ICMP
ARP
DHCP
Сетевых интерфейсов
Ethernet
Wi-Fi
DSL

1. Протокол TCP (Transport Control Protocol)
   Задача: Обеспечить надёжную передачу данных между процессами на разных машинах (например, между браузером и веб-сервером).

Адресация: Использует порты (числа от 1 до 65535). Примеры:

HTTP/HTTPS — порты 80 и 443.

SMTP (почта) — порт 25.

Принцип работы:

Разбивает данные на сегменты фиксированной длины.

Каждому сегменту присваивает порядковый номер.

Отправляет сегменты поочерёдно и ждёт подтверждения (ACK) от получателя перед отправкой следующего.

Если подтверждение не пришло, сегмент отправляется повторно.

2. Протокол IP (Internet Protocol)
   Задача: Доставка пакетов до нужного адреса (маршрутизация).

IP-адреса:

IPv4: 4 числа от 0 до 255 (например, 192.168.1.1). Всего ~4 млрд адресов (исчерпаем).

IPv6: 8 шестнадцатеричных чисел (например, 2001:0db8:85a3::8a2e:0370:7334). Практически неисчерпаем.

Структура адреса:

Первая часть — номер сети (как "индекс и улица").

Вторая часть — номер хоста (как "номер дома").

Разделение определяется маской сети.

3. Как работает передача данных?
Пользователь вводит URL (например, yandex.ru) в браузер.

DNS-сервер преобразует доменное имя в IP-адрес (например, 77.88.55.77).

TCP разбивает запрос на сегменты, IP маршрутизирует их через транзитные узлы (маршрутизаторы).

Пакеты проходят несколько "прыжков" (можно отследить через traceroute).

Сервер отправляет ответ по IP и порту клиента.

4. Дополнительно

   География влияет на скорость: Чем дальше сервер, тем больше "прыжков" и выше задержка.

Анонимность: IP-адрес можно скрыть (VPN, Tor).

Traceroute: Утилита для отслеживания пути пакета (например, traceroute yandex.ru).

Итог

TCP отвечает за надёжность (порты, сегменты, подтверждения).

IP отвечает за доставку (адресация, маршрутизация).

DNS связывает доменные имена с IP-адресами.

Следующая тема: безопасность на прикладном уровне (HTTP/HTTPS).

1. Что такое Tor?
   Tor (The Onion Router) — сеть для анонимного обмена данными и браузер, использующий луковую маршрутизацию.

Цели:

Анонимность пользователя.

Конфиденциальность передаваемых данных.

2. Отличие от классической маршрутизации (TCP/IP)
В обычном интернете:

Каждый промежуточный узел (маршрутизатор) знает IP отправителя и получателя.

В Tor:

Только охранный узел знает отправителя, а выходной узел — получателя.

Промежуточные узлы не знают ни источник, ни назначение трафика.

3. Как работает луковая маршрутизация?
Три типа узлов:

Охранный (Guard) — знает IP отправителя, но не знает содержимого данных.

Промежуточный (Middle) — не знает ни отправителя, ни получателя.

Выходной (Exit) — знает получателя, но не знает отправителя.

Фиксированное число узлов: Всегда 3 (меньше — недостаточно анонимности, больше — не повышает безопасность).

4. Механизм шифрования

Генерация ключей:

Отправитель создает общие ключи с каждым узлом (A, B, C) через криптографические алгоритмы (например, из TLS).

Многослойное шифрование:

Данные шифруются трижды (как слои лука):

Сначала для выходного узла (C), затем для промежуточного (B), потом для охранного (A).

Передача и дешифровка:

Охранный узел (A) снимает свой слой шифра, видит адрес следующего узла (B) и передаёт данные.

Промежуточный узел (B) дешифрует свой слой и передаёт пакет выходному узлу (C).

Выходной узел (C) расшифровывает финальный слой и отправляет данные получателю.

5. Особенности Tor

Анонимность:

Реальный IP-адрес пользователя скрыт (сервисы видят только IP выходного узла).

Выходные узлы могут находиться в любой стране (например, Швейцария, США, Эстония).

Ограничения:

Скорость: Медленнее обычного интернета из-за многократного шифрования.

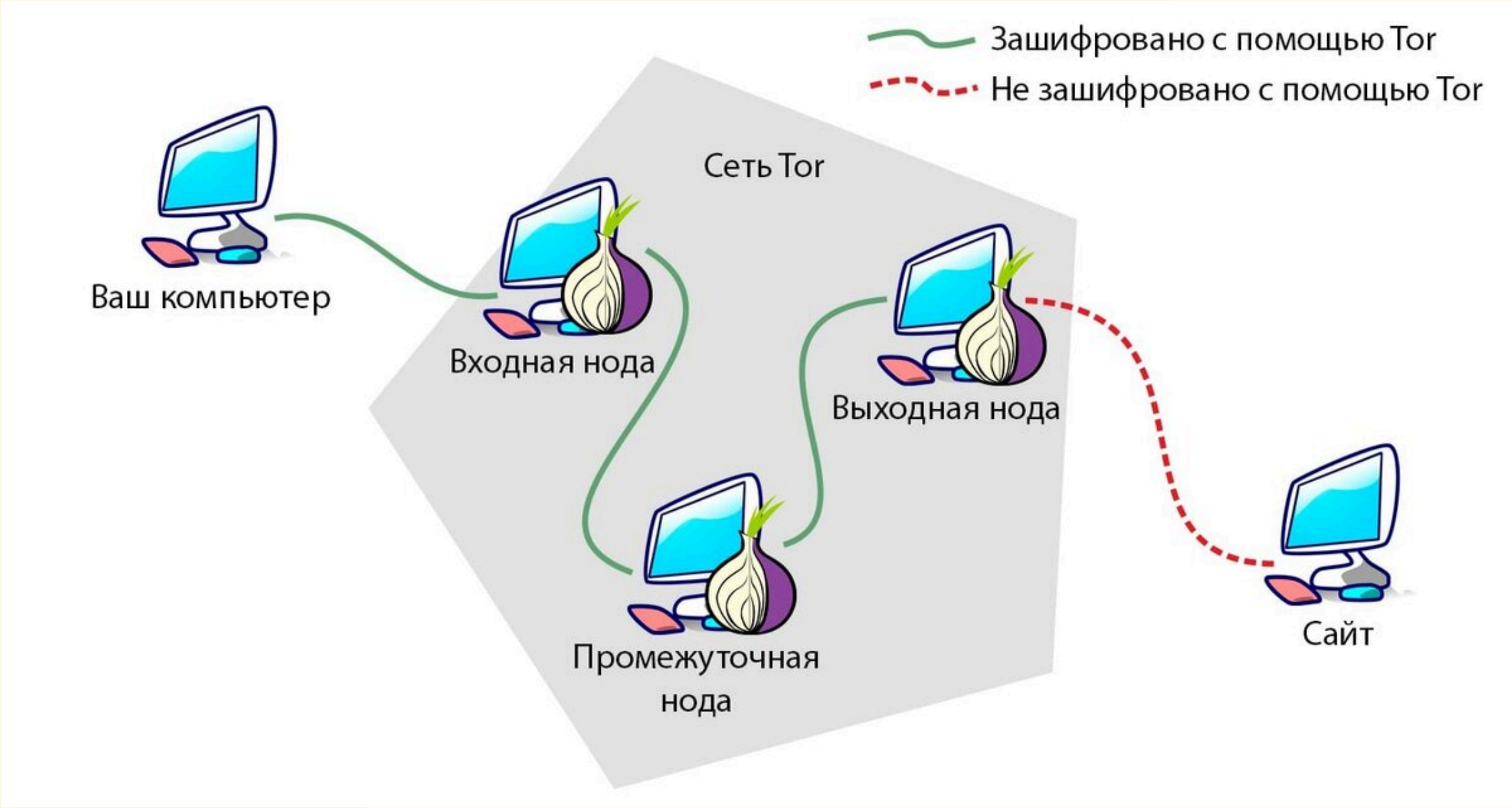
Запреты: В некоторых странах (включая РФ) запрещено быть узлом Tor, но можно использовать браузер.

Уязвимости: Существуют атаки на Tor (например, анализ трафика), но полной деанонимизации сложно достичь.

6. Пример использования
   Браузер Tor выглядит как Firefox или Chrome.

При запросе сайта (например, google.com) язык и контент могут меняться в зависимости от страны выходного узла.

Сервисы для проверки IP показывают адрес выходного узла, а не пользователя.


Зашифровано с помощью Tor
Не зашифровано с помощью Tor
Сеть Tor
Ваш компьютер
Входная нода
Выходная нода
Промежуточная нода
Сайт

1. Основы Wi-Fi
   Wi-Fi — технология беспроводной локальной сети на основе стандарта IEEE 802.11.

Стандарты:

802.11 (1997 г.) — первый стандарт (1.2 Мбит/с, инфракрасный порт).

802.11ac (Wi-Fi 5) — современный стандарт (до 6 Гбит/с).

802.11ax (Wi-Fi 6) и 802.11be (будущий) — скорости свыше 30 Гбит/с.

Wi-Fi Alliance — организация, сертифицирующая устройства на соответствие стандартам.

2. Уровень работы Wi-Fi
   Работает на канальном уровне модели TCP/IP (аналогично Ethernet, но без проводов).

Функции:

Модуляция сигнала.

Контроль за помехами между разными Wi-Fi сетями.

3. Режимы работы Wi-Fi

Инфраструктурный режим:

Классическая схема с роутером, раздающим интернет.

Идентификатор сети — SSID (Service Set Identifier).

Ad-Hoc режим:

Прямое соединение устройств без роутера (например, раздача интернета с телефона).

4. Безопасность Wi-Fi

Устаревшие методы (небезопасны)

WEP (Wired Equivalent Privacy):

Использовал короткий ключ (40 бит).

Легко взламывается.

Современные методы
WPA/WPA2/WPA3:

Используют шифрование AES (ключ 128+ бит).

Аутентификация:

WPA Personal — по паролю (для домашних сетей).

WPA Enterprise — через базу данных пользователей (для корпоративных сетей).

Особенности WPA3:

Защита даже слабых паролей (например, "12345").

Forward Secrecy — при каждом новом подключении генерируется новый ключ (безопасность даже при утечке старого пароля).

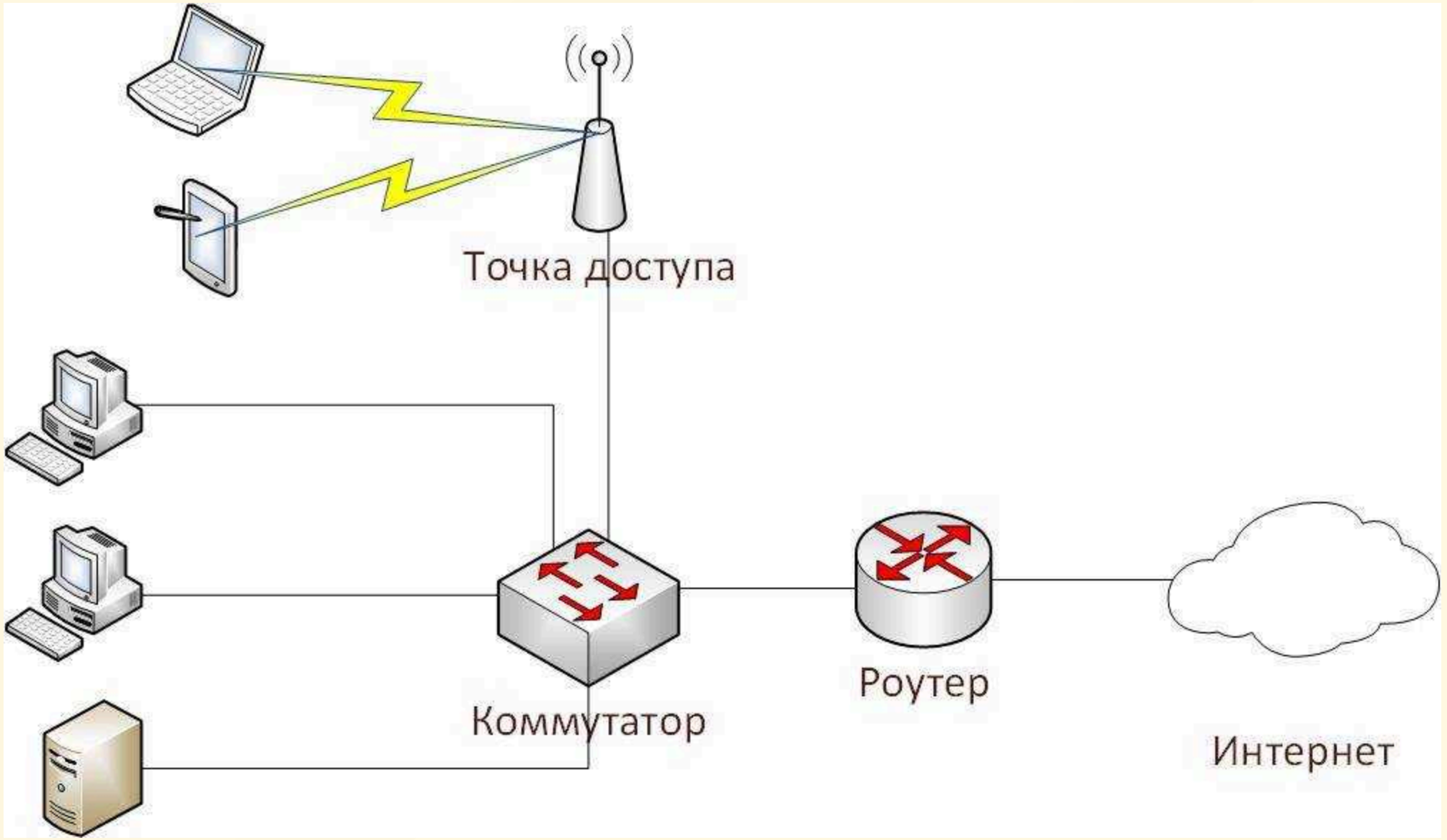
5. Рекомендации по безопасности
Используйте WPA3 (или WPA2, если WPA3 недоступен).

Избегайте WEP и открытых сетей (без пароля).

Для корпоративных сетей предпочтительна WPA Enterprise.

Итог
Wi-Fi обеспечивает удобство, но требует надежного шифрования (WPA3).

Безопасность зависит от выбора протокола и сложности пароля.


Точка доступа
Коммутатор
Роутер
Интернет

Третья глава: Защита ПК/телефона

1. Зачем шифровать жесткий диск?
   Основная цель: Защита персональных данных при утере или краже устройства.

Примеры рисков:

Утечка паролей, документов, конфиденциальной информации.

Корпоративные требования: Компании (Google, Яндекс) и университеты часто обязывают сотрудников шифровать корпоративные устройства.

2. Принцип работы шифрования
Генерация ключа:

Программа создает криптографический ключ (например, 256-битный для AES).

Пользователь задает пароль для доступа к ключу (не сам ключ!).

Шифрование данных:

Ключ применяется ко всем данным на диске/разделе (файлы, загрузочный сектор).

Данные преобразуются в нечитаемый формат.

Дешифровка:

При вводе пароля система получает ключ и расшифровывает данные "на лету".

3. Технические детали
   Алгоритм: AES (Advanced Encryption Standard) — симметричное шифрование.

Реализован на аппаратном уровне (TPM-модуль в процессорах Intel/AMD).

Минимальные задержки при работе (шифрование/дешифровка в фоновом режиме).

TPM (Trusted Platform Module):

Криптопроцессор для безопасного хранения ключей.

Защищает ключ от извлечения злоумышленником.

4. Что можно шифровать?
   Весь жесткий диск (включая систему).

Отдельные разделы/флешки.

Загрузочный сектор:

Требует ввода пароля при включении компьютера.

Пример: BitLocker (Windows), FileVault (macOS).

5. Где хранить ключ?
   Не на зашифрованном диске (иначе доступ невозможен при утере пароля).

Варианты:

Физические носители (флешка, токен).

Облачное хранилище (например, у антивирусных компаний).

Администратор сети (для корпоративных устройств).

6. Программы для шифрования
Встроенные утилиты:

Windows: BitLocker.

macOS: FileVault.

Linux: LUKS.

Сторонние решения:

VeraCrypt (open-source).

PGPDisk.

7. Важные моменты
   Длина ключа: 256 бит (AES-256) — обеспечивает высокую стойкость.

Пароль ≠ ключ: Пароль лишь разблокирует доступ к ключу.

Рекомендации:

Шифруйте не только диск, но и съемные носители (флешки).

Используйте TPM-модуль для защиты ключа.

Итог:
Шифрование носителей — обязательная мера для защиты данных.
Современные инструменты (AES, TPM) делают процесс безопасным и незаметным для пользователя.

1. Что такое пароль?
   Пароль — это средство аутентификации, используемое для:

Входа в соцсети, почту, банковские сервисы.

Разблокировки устройств (PIN, отпечаток пальца, графический ключ).

Доступа к электронным кошелькам (например, 12-словная seed-фраза в Bitcoin).

2. Критерии стойкости пароля

Основная угроза — перебор (брутфорс). Стойкость зависит от:

Длины:

Пароль из 6 цифр: 10^6 = 1 млн вариантов → взламывается за секунду.

Пароль из 8 символов (буквы + цифры): 36^8 ≈ 3 трлн вариантов → сложно подобрать.

Алфавита:

Цифры (0-9) + буквы (a-z, A-Z) + спецсимволы (!@#) резко увеличивают сложность.

Рекомендации:

Используйте пароли длиной от 12 символов.

Включайте верхний/нижний регистр, цифры, спецсимволы.

3. Типичные ошибки

Частые пароли: 12345, password, qwerty — легко взламываются.

Повторение паролей на разных сервисах → компрометация одного аккаунта угрожает остальным.

Утечки баз данных: Пароли из LinkedIn, eBay и других сервисов часто оказываются в даркнете.

4. Защита от перебора

Ограничение попыток: Например, 3 попытки для PIN-кода.

Капча: Тест для отличия человека от бота (например, выбор изображений с мостами).

Недостатки:

Сложность для людей с плохим зрением.

ИИ научился обходить некоторые капчи.

Злоумышленники нанимают людей для решения капч.

5. Хранение паролей на сервере

Пароли никогда не хранятся в открытом виде. Используется:

Хэш-функция: Преобразует пароль в строку фиксированной длины (например, SHA-256).

Свойства:

Необратимость: По хэшу нельзя восстановить пароль.

Детерминированность: Один и тот же пароль → одинаковый хэш.

Примеры: SHA-2, SHA-3, ГОСТ Р 34.11-2018.

Механизм работы:

Пользователь регистрируется с паролем 4@w&t2!.

Сервер сохраняет не пароль, а его хэш: hash(4@w&t2!).

При следующем входе сервер сверяет хэш присланного пароля с сохранённым.

Защита слабых паролей — "соль" (salt):

Сервер добавляет к паролю случайную строку (например, 12345 + s8d2f1).

Хранится хэш от "пароль + соль" и сама соль.

Зачем? Чтобы даже слабые пароли (12345) не совпадали с предвычисленными хэшами из словарей.

6. Где хранить пароли?
Менеджеры паролей:

KeePassXC (кроссплатформенный).

Keychain Access (macOS).

Мастер-пароль: Достаточно запомнить один сложный пароль для доступа ко всем остальным.

7. Рекомендации

Используйте длинные и сложные пароли.

Не повторяйте пароли на разных сервисах.

Меняйте пароли регулярно (особенно для почты и банковских аккаунтов).

Проверяйте утечки на haveibeenpwned.com.

Итог:

Стойкость пароля зависит от длины и разнообразия символов.

Серверы хранят хэши паролей с "солью" для защиты от утечек.

Менеджеры паролей — лучший способ безопасного хранения.

Ключевые термины:

Брутфорс — атака перебором.

Хэш-функция — криптографическое преобразование пароля.

Соль (salt) — случайные данные для усиления стойкости хэша.

Для углублённого изучения:

Атаки радужных таблиц (rainbow tables).

Двухфакторная аутентификация (2FA).

Биометрическая аутентификация (отпечатки, Face ID).

1. Что такое фишинг?
   Фишинг (от англ. to fish — «ловить рыбу») — это метод мошенничества, при котором злоумышленники выманивают конфиденциальные данные (логины, пароли, банковские реквизиты), маскируясь под доверенные сервисы или организации.

2. Типы фишинговых атак
   Адресный фишинг (email-фишинг):

Рассылка писем с поддельными ссылками на фальшивые сайты (например, поддельный «ВКонтакте»: vk.club5.ru).

Цель: заставить пользователя ввести свои данные на поддельной странице.

Телефонный фишинг (вишинг):

Звонки от якобы банков или служб поддержки с просьбой сообщить реквизиты карт или коды подтверждения.

Календарный фишинг:

Рассылка ссылок на «события» в календаре, которые ведут на вредоносные сайты.

Фишинг в мессенджерах:

Поддельные сообщения в WhatsApp, Telegram и др. с просьбой перейти по ссылке или скачать файл.

3. Как работает email-фишинг?
   Подделка домена: Ссылка выглядит как настоящая, но ведет на фальшивый сайт (например, vk.club5.ru вместо vk.com).

Отсутствие HTTPS: Поддельные сайты часто используют незащищенный протокол HTTP.

Email spoofing: Мошенники подменяют адрес отправителя, чтобы письмо казалось отправленным от знакомого или организации.

4. Защита от спуфинга (spoofing)
   SPF (Sender Policy Framework): Проверяет, разрешено ли отправителю использовать данный IP-адрес.

DMARC (Domain-based Message Authentication): Определяет, что делать с письмами, которые не прошли проверку (отклонить или пометить как спам).

5. Как распознать фишинг?
   Подозрительные ссылки: Наведите курсор на ссылку (не кликая!), чтобы увидеть настоящий URL.

Неожиданные письма: Даже от «знакомых» — перепроверяйте через другой канал связи.

Грамматические ошибки: Часто встречаются в фишинговых письмах.

Давление: Фразы вроде «Срочно!», «Ваш аккаунт будет заблокирован!» — признаки мошенничества.

6. Как защититься?
   Не переходите по подозрительным ссылкам и не скачивайте вложения из непроверенных писем.

Проверяйте домен сайта: Официальные сервисы используют HTTPS и корректные домены (например, yandex.ru, а не yandex.club1.com).

Включите двухфакторную аутентификацию (2FA) для важных аккаунтов.

Используйте почтовые сервисы с защитой (Gmail, Яндекс.Почта, Outlook), которые фильтруют спам.

Обучайте антиспам-фильтры: Помечайте фишинговые письма как спам.

7. Что делать, если вы попались на фишинг?
   Немедленно смените пароль на взломанном аккаунте.

Если ввели данные карты — заблокируйте ее и сообщите в банк.

Включите 2FA для всех важных сервисов.

Итог:
Фишинг остается одной из самых распространенных киберугроз. Защита строится на внимательности и использовании технических средств (SPF, DMARC, HTTPS).

Ключевые термины:

Спуфинг (spoofing) — подмена отправителя (email, IP).

SPF/DMARC — протоколы проверки подлинности писем.

2FA — двухфакторная аутентификация.

1. Требования к безопасности мессенджеров
   Целостность сообщений - гарантия, что сообщение не было изменено при передаче.

Конфиденциальность - доступ к сообщению имеют только отправитель и получатель.

Аутентичность - подтверждение личности отправителя.

Доставка сообщений - устойчивость к потерям при временном отсутствии связи.

Прямая секретность (Forward Secrecy) - защита прошлых сообщений при компрометации текущего ключа.

Пост-компрометационная безопасность - возможность восстановить конфиденциальность после утечки ключа.

2. Протокол Signal
   Разработчик: Open Whisper Systems

Используется в: WhatsApp, Facebook Messenger, Skype, собственном мессенджере Signal

Особенности:

Открытый исходный код

Регулярно обновляется

Ориентирован на максимальную конфиденциальность

3. Сквозное шифрование (E2EE)
Принцип работы:

Сервер выступает только как маршрутизатор, не имея доступа к содержимому сообщений.

Сообщения шифруются на устройстве отправителя и расшифровываются только на устройстве получателя.

Процесс обмена сообщениями:

Обмен ключами:

Каждый пользователь имеет пару ключей: открытый и закрытый

Открытые ключи публикуются на сервере

Генерация общего ключа:

Используется алгоритм Диффи-Хеллмана

Позволяет создать общий секретный ключ без его передачи по сети

Шифрование сообщений:

Для каждого сеанса связи генерируется новый ключ

Обеспечивает прямую секретность

4. Групповые чаты
Особенности реализации:

Каждый участник группы имеет свой ключ шифрования

Отправитель шифрует сообщение своим ключом

Получатели используют заранее полученные ключи отправителя для расшифровки

При добавлении нового участника требуется обмен ключами со всеми членами группы

5. Преимущества протокола Signal
Высокий уровень безопасности:

Реализует все основные требования к защите сообщений

Использует современные криптографические алгоритмы

Децентрализация:

Сервер не имеет доступа к содержимому сообщений

Гибкость:

Поддерживает как индивидуальные, так и групповые чаты

Открытость:

Возможность независимой проверки реализации

6. Рекомендации по безопасности
   Используйте мессенджеры с поддержкой протокола Signal

Регулярно обновляйте приложение для получения последних улучшений безопасности

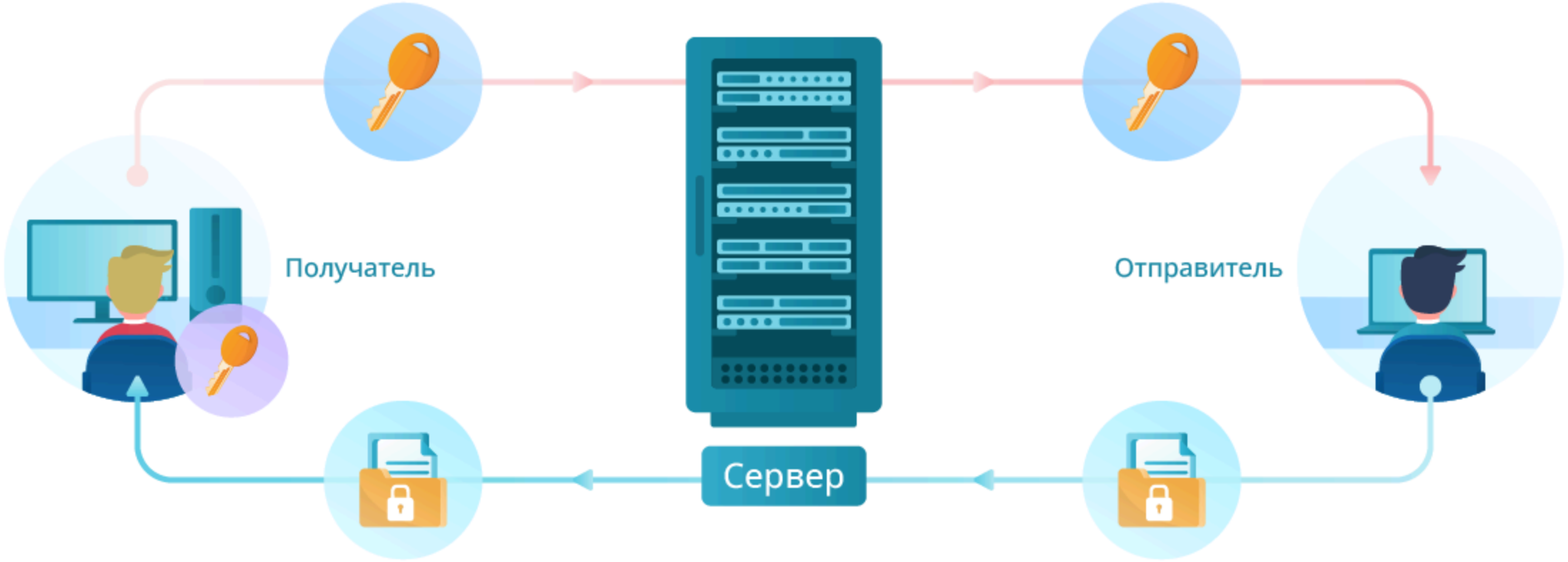
Включайте двухфакторную аутентификацию

Проверяйте ключи безопасности (Safety Numbers) при важных переписках

1 В начале, при создании сеанса связи, приложение-получатель генерирует два ключа: публичный и закрытый.

2 Публичный ключ отправляется на сервер.

3 Приложение-отправитель скачивает с сервера публичный ключ для шифрования сообщения.



5 Приложение-получатель скачивает сообщение, и расшифровывает его с помощью своего закрытого ключа.

4 Зашифрованное сообщение отправляется на сервер.

Четвертая глава: Криптография на практике

1. Классификация криптографических протоколов
   Симметричная криптография:

Обе стороны используют один общий секретный ключ

Примеры: симметричное шифрование, коды аутентификации сообщений (MAC)

Проблема: сложность безопасного обмена ключами

Асимметричная криптография:

Каждая сторона имеет пару ключей (публичный и приватный)

Примеры: цифровая подпись, протоколы обмена ключами

Решает проблему обмена ключами

Бесключевые примитивы:

Криптографические хэш-функции

2. Криптографические хэш-функции
Свойства:

Преобразуют данные произвольной длины в фиксированный хэш

Стойкость к коллизиям (невозможно найти два разных входа с одинаковым хэшем)

Необратимость (по хэшу нельзя восстановить исходные данные)

Применение:

Хранение паролей

Проверка целостности данных

Блокчейн (proof-of-work)

Примеры: SHA-2, SHA-3, ГОСТ Р 34.11-2012

3. Симметричное шифрование
Компоненты:

Генерация ключа

Функция шифрования

Функция дешифрования

Свойства:

Один ключ для шифрования и дешифрования

Высокая производительность

Обеспечивает конфиденциальность

Примеры: AES, ГОСТ 34.12-2018

4. Симметричная аутентификация (MAC)
Коды аутентификации сообщений:

Используют отдельный ключ для генерации кода

Обеспечивают целостность данных

Часто строятся на основе хэш-функций

Комбинированное использование:

Шифрование + MAC = конфиденциальность + целостность

5. Асимметричные протоколы
   Обмен ключами (Диффи-Хеллман):

Позволяет сгенерировать общий секретный ключ через открытый канал

Уязвим к MITM-атакам без дополнительной аутентификации

Цифровая подпись:

Компоненты:

Генерация ключевой пары

Подписание (с использованием приватного ключа)

Верификация (с использованием публичного ключа)

Свойства:

Аутентификация отправителя

Целостность документа

Неотказуемость (non-repudiation)

Применение:

Обновление ПО

SSL/TLS сертификаты

Примеры: RSA, ECDSA, ГОСТ Р 34.20-2012

6. Практическое применение
TLS/SSL:

Асимметричная криптография для аутентификации и обмена ключами

Симметричное шифрование для защиты данных

Мессенджеры:

Протокол Signal использует комбинацию Диффи-Хеллмана и цифровых подписей

Блокчейн:

Хэш-функции для proof-of-work

Цифровые подписи для транзакций

1. Основные этапы платежа
   Электронный платеж состоит из двух ключевых этапов:

Авторизация - технический процесс с использованием криптографии, включающий:

Проверку подлинности владельца карты

Подтверждение банком-эмитентом наличия средств и отсутствия ограничений

Транзакция - непосредственный перевод денежных средств

2. Виды платежей

Существует два основных типа платежей:

Физические платежи (Card Present)
Используются при наличии реальной карты, например в магазинах. Процесс включает:

Прикладывание карты к терминалу

Ввод PIN-кода

Создание электронной подписи на основе секретного ключа карты

Проверку подписи банком с помощью публичного ключа

Онлайн-платежи (Card Not Present)
Для интернет-покупок используется протокол 3-D Secure (версия 2.0), в котором участвуют:

Банк-эмитент (выдавший карту)

Платежная система (Visa/Mastercard/МИР)

Банк-эквайер (принимающий платеж)

3. Как работает онлайн-оплата
Покупатель вводит данные карты на сайте

Продавец обращается к платежной системе

Система определяет банк-эмитент

Банк проверяет возможность платежа

Покупатель проходит многофакторную аутентификацию (например, через SMS)

Банк подтверждает или отклоняет транзакцию

4. Методы подтверждения личности

Для безопасности используются разные факторы:

Знание (PIN, CVV-код)

Владение (телефон для SMS, приложение банка)

Биометрия (отпечаток пальца, распознавание лица)

Местоположение (геолокация)

5. Важные правила безопасности
   Все соединения защищены протоколом TLS

PIN-код используется только для платежей с физической картой

CVV-код - только для онлайн-платежей

Никогда и никому не сообщайте:

PIN-код своей карты

CVV-код (3 цифры на обороте)

Коды из SMS

1. Ключевые понятия
   Криптография (Crypto) ≠ Криптовалюта (Cryptocurrency)

Криптография — наука о защите информации

Криптовалюта — цифровые деньги на основе блокчейна

Биткоин ≠ Блокчейн

Биткоин — первая и самая популярная криптовалюта

Блокчейн — технология, лежащая в основе биткоина и других криптовалют

2. Основные свойства блокчейна
Децентрализация

Нет центрального органа контроля (например, банка)

Позволяет проводить транзакции без доверенной третьей стороны

Консенсус — согласованность данных между участниками:

Постоянство: данные нельзя удалить

Единство: все участники видят одинаковую версию (кроме последних блоков)

Живучесть: возможность добавлять новые транзакции

Открытость: любой может стать участником (в публичных блокчейнах)

3. Как устроен блокчейн?

Транзакции — переводы средств между участниками

Подписываются секретным ключом отправителя (электронная подпись)

Пример: "Алиса → Бобу: 10 BTC" + подпись Алисы

Блоки — наборы транзакций

Формируются майнерами

Добавляются в цепочку (blockchain)

Майнеры — участники, которые:

Собирают транзакции в блок

Решают криптографическую задачу (доказательство работы)

Получают награду за добавление блока

4. Механизмы консенсуса

Proof of Work (PoW) — доказательство работы

Майнеры ищут x, чтобы h(x) имел заданное число нулей (например, 17)

Требует больших вычислительных ресурсов (энергозатратно)

Используется в Биткоине

Proof of Stake (PoS) — доказательство доли владения

Вероятность добавления блока зависит от количества монет у майнера

Другие варианты:

Proof of Space (использование дискового пространства)

Proof of Authority (доверенные узлы)

5. Проблемы и перспективы
Энергопотребление PoW:

Криптофермы потребляют много электроэнергии

Поиск более экологичных альтернатив (например, переход на PoS)

Будущее криптовалют:

Неясно, какая валюта станет основной

Развитие новых механизмов консенсуса

6. Вывод
   Блокчейн решает сложную задачу децентрализованного учета транзакций без доверенного центра. Криптовалюты используют криптографию для защиты от подделки и двойных трат. Основные вызовы — энергоэффективность и масштабируемость.


КАК РАБОТАЕТ БЛОКЧЕЙН
Блок 1
Хэш блока 1
Хэш блока 0
Информация о транзакциях
Блок 2
Хэш блока 2
Хэш блока 1
Информация о транзакциях
Блок 3
Хэш блока 3
Хэш блока 2
Информация о транзакциях

Литература

Базовые сетевые протоколы

Более подробно об устройстве сети и сетевых протоколах можно:

Прочитать (Дуглас Э. Камер - Сети TCP IP. Принципу́, протоколу́ и структура (2003))
Посмореть «Компьютерные сети. Базовый учебный курс»
В ходе лекции были использованы:

Сайт REG.RU
Программа traceroute (в OS Windows носит название tracert)
Он-лайн версия прослеживания маршрута
Подробнее об истории создания DNS адресации можно прочитать

Как развивалась система доменных имен: эра ARPANET
История системы доменных имен: первые DNS-серверы
История DNS: когда доменные имена стали платными
Обеспечение безопасности в сети. Протокол TLS

ssllabs.com - сайт, осуществляющий проверку установленного у Вас браузера на версию и конфигурацию TLS
Иллюстрация к работе TLS v.1.2, v. 1.3
Статья "Первые несколько миллисекунд HTTPS соединения"
Описание алгоритма генерации общего секретного ключа -- так называемого протокола Диффи-Хэллмана -- можно найти (! математика внутри)

Статья с подробным описанием httmps

Беспроводные сети Wi-fi

Более подробно о технической реализации передачи данных по беспроводному типу описывается в видео-лекции
Данные, передаваемые по сети WiFi, делятся на так называемые кадры. Как именно формируются кадры подробно описано в видео
О безопасности алгоритмы WPA3 можно подробно узнать в статьях

WPA3, улучшенное открытие [Enhanced Open], простое соединение [Easy Connect]: три новых протокола от Wi-Fi Alliance
Началась сертификация устройств WPA3: слабые пароли стали более безопасными
Персонализация в сети

Cookie-стена — это вид cookie-баннера, который не даёт пользоваться веб-сайтом без предоставления согласия на размещение cookie-файлов.

О законодательном регулировании cookie-стен в ЕС можно прочитать в статье «Почему Евросоюз искореняет cookie-стены»
Помимо обычных кук, существуют еще и суперкуки. Про из особенности можно прочитать в статье «Что такое куки и супер

Что такое “отпечаток” браузера, чем он плох можно прочесть в доступной для широкого круга читателя статье (в 2х частях)

Часть 1
Часть 2
Лекция от MIT (на англ.) о приватном веб-браузинге

Private Browsing
Защита ПК/телефона

Пароли

Хит-парад паролей
Статистический анализ паролей (упоминается в лекции)
Подробнее про соль для паролей
и про перец (на англ.)
Обзор менеджеров паролей (оценки субъективны)
Шифрование диска/хранилищ

Подробно о том, как работает BitLocker (релевантно для Windows систем) можно узнать здесь
Инструкция по созданию зашифрованной флэшки здесь
О возможностях VeraCrypt
Фишинг, вирусы

TOP-10 вирусов в истории (на англ.)
Рекомендации от Gmail (для администраторов почтовых серверов организаций)
О громкой вирусе Stuxnet
Правила защиты от фишинга
Мессенджеры

Спецификация протокола Signal, упомянутая в лекции
Протокол, используемый для безопасной передачи сообщений в мессенджере Telegram называется MTProto Mobile Protocol.
Техническое описание этого протокола (на англ.)
Техническое описание MTProto на русском языке
Криптография на практике

Введение в криптографию

Доступные широкой аудитории мини-лекции от проф. Найджела Смарта о криптографии (на англ.)
Более продвинутые лекции по криптографии (для студентов-магистров, основы математики объясняются)
Для интересующихся, сайт международного криптографического сообщества IACR (International Association for Cryptologic Research)
Ютуб-канал сообщества с видеозаписями конференций и выступлений
Цифровая подпись

“Слепая” подпись -- разновидность протокола электронной цифровой подписи, в которой подписывающая сторона не знает сообщения, которое она подписывает, а подписывает шифр-текст этого сообщения. Зачем такая подпись нужна и как её построить описывается здесь
По этой же теме статья на англ.
Подробнее о типах электронных подписей с юридической точки зрения здесь
Электронные платежи

Подробно об истории и устройстве пластиковой карты, а также о современных методах электронных платежей можно узнать из видео-лекции
Подробнее о 3d-secure платежах можно узнать из видео-лекции
Доступный обзор разнообразных платежных услуг описан в статье: часть 1, часть 2
Блокчейн

Официальный сайт криптовалюты Биткоин со статьей Сатоши Накамото (рус.)
Доступное широкой аудитории объяснение логики блокчейна можно увидеть в мини-лекции (англ.)
Принципы работы криптовалюты Etherium описываются здесь
Бумажные издания, посвященные электронным деньгам

А. Антонопулос. Интернет денег. Litres, 2018
A. Antonopoulos. Mastering Bitcoin: unlocking digital crypto currencies. "O'Reilly Media, Inc. 2014