

Внутренний курс

Основы Кибербезопасность

Матвеева Анастасия Сергеевна

Содержание

1 Доклад	6
Список литературы	54

Список иллюстраций

1.1	6
1.2	7
1.3	8
1.4	10
1.5	11
1.6	12
1.7	13
1.8	15
1.9	16
1.10	17
1.11	19
1.12	19
1.13	20
1.14	21
1.15	21
1.16	22
1.17	23
1.18	24
1.19	25
1.20	26
1.21	27
1.22	28
1.23	30
1.24	31
1.25	32
1.26	33
1.27	34
1.28	35
1.29	35
1.30	36
1.31	36
1.32	37
1.33	37
1.34	38
1.35	38
1.36	39
1.37	40

1.38	40
1.39	41
1.40	42
1.41	43
1.42	43
1.43	44
1.44	45
1.45	46
1.46	47
1.47	48
1.48	48
1.49	49
1.50	50
1.51	51
1.52	52
1.53	53

List of Tables

1 Доклад

2.1.1

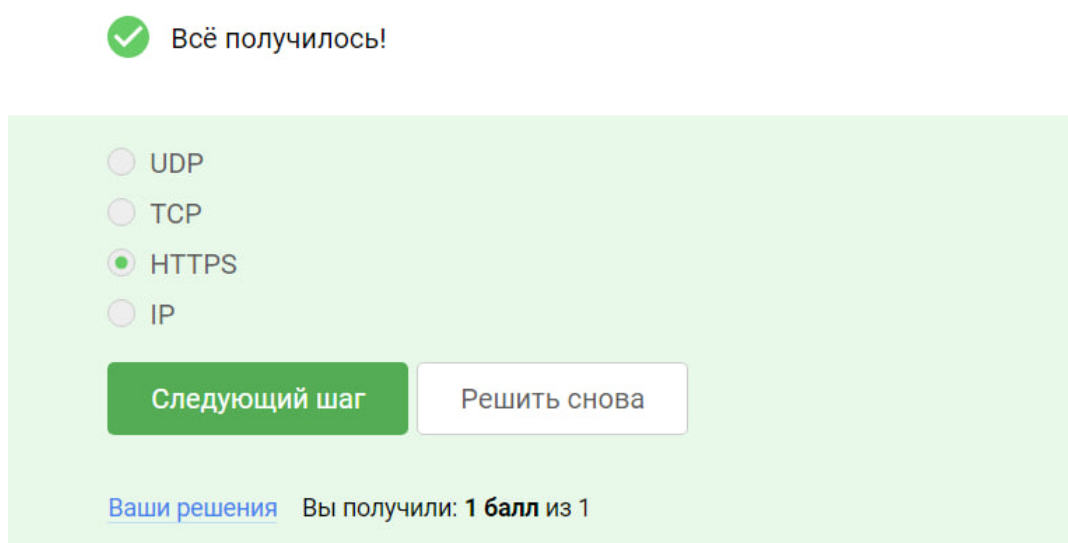


Рис. 1.1: .

HTTPS (Hypertext Transfer Protocol Secure) является протоколом прикладного уровня по следующим причинам:

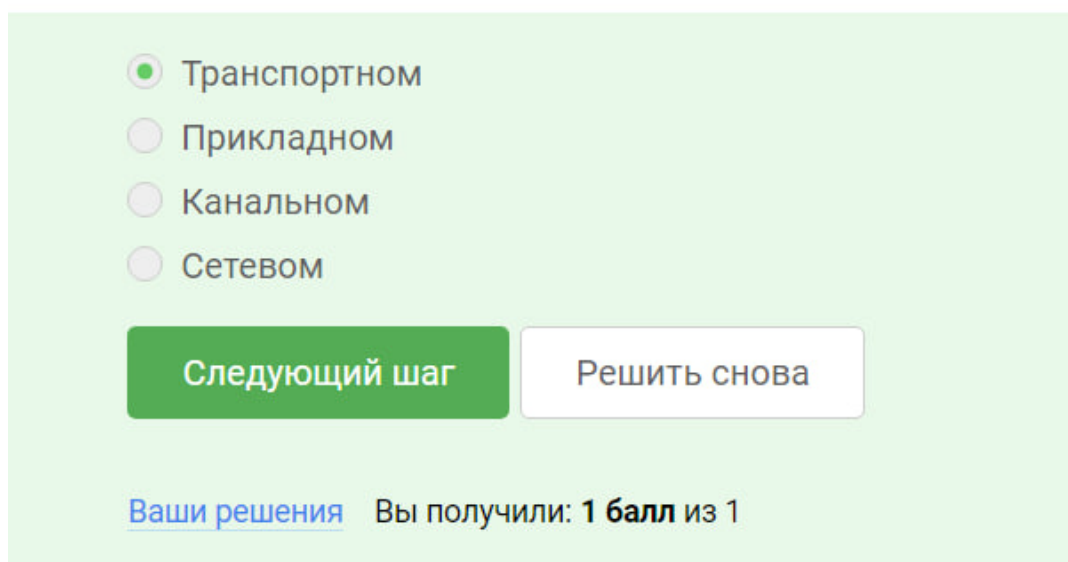
Занимается коммуникацией между приложениями: HTTPS устанавливает зашифрованный канал связи между веб-браузером (клиентское приложение) и веб-сервером (серверное приложение). Обеспечивает специальные функции приложения: HTTPS предоставляет специфичные для приложений функции, такие как шифрование и аутентификация, которые не обрабатываются более низкоуровневыми протоколами. Располагается поверх транспортного уровня: HTTPS работает поверх транспортного уровня, такого как TCP (Transmission

Control Protocol), который отвечает за передачу данных. Модель OSI: В модели взаимодействия открытых систем (OSI) HTTPS относится к прикладному уровню (уровень 7), который отвечает за представление данных для конкретных приложений.

Таким образом, HTTPS классифицируется как протокол прикладного уровня, поскольку он работает на самом высоком уровне стека протоколов для обработки коммуникации между веб-браузером и веб-сервером

2.1.2

✓ Абсолютно точно.



☒ Транспортном

☐ Прикладном

☐ Канальном

☐ Сетевом

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 1.2: .

TCP работает на транспортном уровне модели OSI по нескольким причинам:

Установление соединения: TCP устанавливает надежное соединение между двумя хостами, гарантируя упорядоченную и безошибочную доставку данных. Это важно для приложений, требующих надежной и гарантированной доставки, таких как передача файлов или веб-просмотр. Контроль потока: TCP внедряет контроль потока для обеспечения плавного и эффективного обмена данными. Он регулирует скорость передачи данных, чтобы избежать перегрузки сети и

потери пакетов.

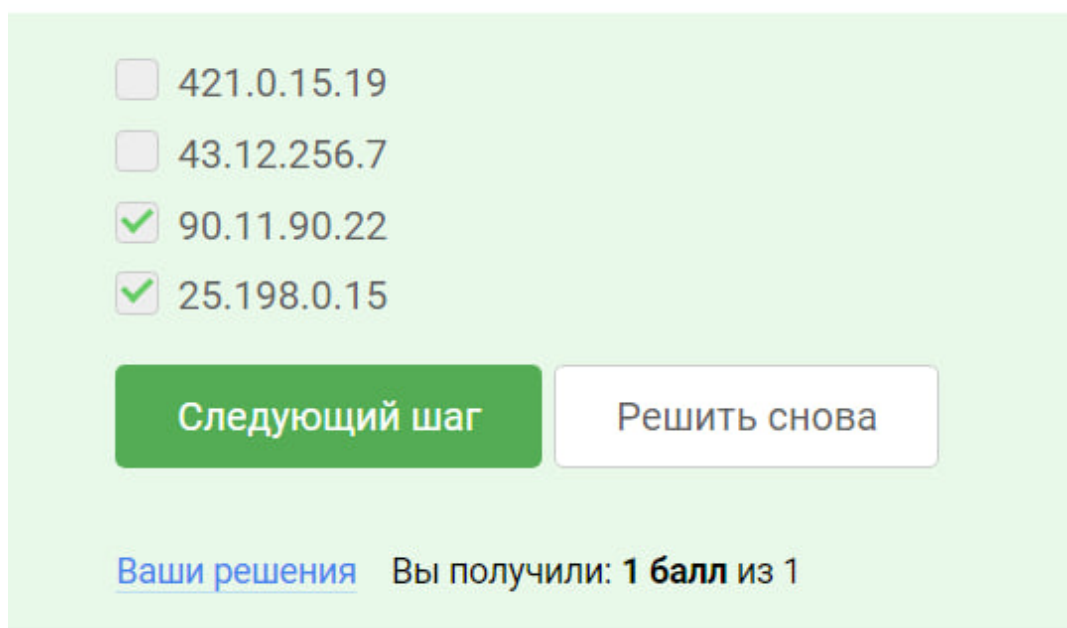
Управление перегрузкой: ТСП реализует механизмы управления перегрузкой для предотвращения перегрузок в сети. Он распознает признаки перегрузки, такие как потеря пакетов, и соответственно корректирует скорость передачи.

Мультиплексирование: ТСП позволяет нескольким приложениям совместно использовать одну физическую сеть, передавая данные между ними. Это достигается путем назначения уникальных портов каждому приложению.

Краткий заголовок: Заголовок ТСП имеет относительно небольшой размер, что снижает накладные расходы и оптимизирует использование полосы пропускания. Это особенно важно в сетях с высокой задержкой и низкой пропускной способностью.

Эти функции делают ТСП идеальным протоколом для транспортного уровня, обеспечивая надежную и эффективную передачу данных между приложениями в сети.

2.1.3



☐ 421.0.15.19

☐ 43.12.256.7

☒ 90.11.90.22

☒ 25.198.0.15

Следующий шаг **Решить снова**

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.3: .

Адрес IPv4 (Интернет-протокол версии 4) представляет собой 32-битный чис-

ловой адрес, который однозначно идентифицирует устройство в сети. Он обычно записывается в точечно-десятичной нотации, состоящей из четырех чисел, разделенных точками:

x.y.z.w

Где:

x, y, z и w - целые числа от 0 до 255

Например, допустимый адрес IPv4:

192.168.1.1

Почему адрес IPv4 имеет такой формат

Формат адреса IPv4 был разработан в 1980-х годах, когда сети были намного меньше, чем сегодня. IPv4 был разработан с использованием 32-битного адреса, потому что это было достаточно большим числом, чтобы идентифицировать устройства в сети того времени.

Точечно-десятичный формат удобен для людей, поскольку он позволяет легко записывать и читать адреса IPv4. Каждое число в точечно-десятичной нотации представляет собой 8-битный октет, который является наименьшей адресуемой единицей в сети.

Примеры адресов IPv4

0.0.0.0 (адрес сети по умолчанию) 127.0.0.1 (резервный адрес для локального узла) 192.168.1.1 (частный адрес, обычно используемый в домашних или офисных сетях) 255.255.255.255 (адрес широковещательного сообщения, который рассылается всем устройствам в сети)

2.1.4

☒ сопоставляет IP адреса доменным именам

☐ сегментирует данные на транспортном уровне

☐ выбирает маршрут пакета в сети

☐ выполняет адресацию на хосте

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.4: .

DNS-сервер (доменная система имен) сопоставляет доменные имена (например, `www.google.com`) с их соответствующими IP-адресами (например, `172.217.13.238`).

Когда пользователь вводит доменное имя в свой браузер, происходит следующий процесс:

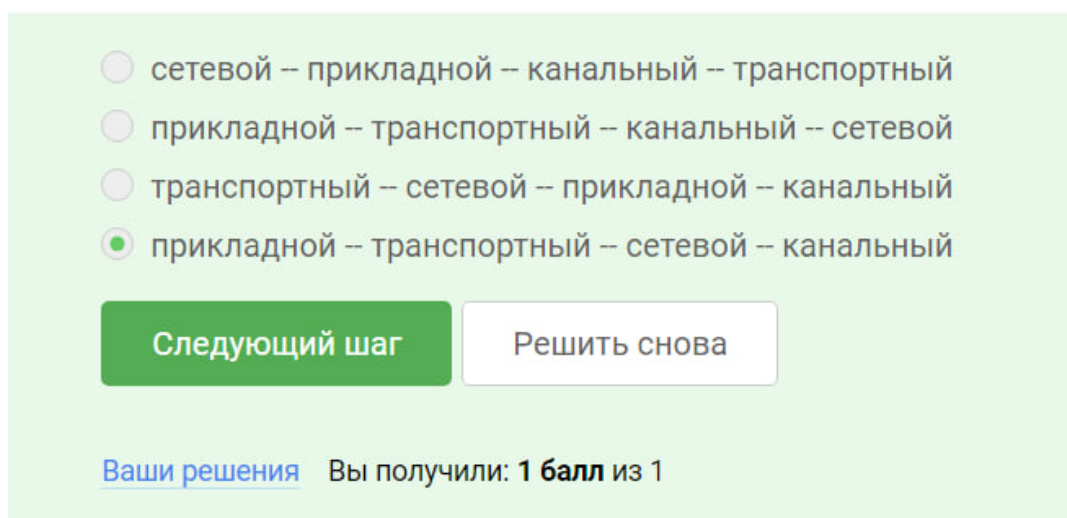
1. Запрос к DNS-серверу: Браузер отправляет запрос к DNS-серверу, указанному в настройках сети устройства.
2. Поиск записи DNS: DNS-сервер ищет в своей базе данных запись, соответствующую введенному доменному имени. В этой записи хранится IP-адрес, связанный с этим доменным именем.
3. Возврат IP-адреса: DNS-сервер возвращает браузеру найденный IP-адрес.
4. Подключение к веб-серверу: Браузер использует возвращенный IP-адрес для установления соединения с веб-сервером, на котором размещается соответствующий веб-сайт.

Таким образом, DNS-сервер действует как переводчик между доменными именами, которые легко запоминаются пользователями, и IP-адресами,

которые фактически используются веб-серверами. Без DNS-серверов пользователям пришлось бы вводить IP-адреса для доступа к веб-сайтам, что было бы намного сложнее и менее удобно.

2.1.5

✓ Хорошие новости, верно!



☐ сетевой – прикладной – канальный – транспортный

☐ прикладной – транспортный – канальный – сетевой

☐ транспортный – сетевой – прикладной – канальный

☒ прикладной – транспортный – сетевой – канальный

Следующий шаг **Решить снова**

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.5: .

Последовательность протоколов

При отправке пакета данных устройства следуют следующей общей последовательности протоколов:

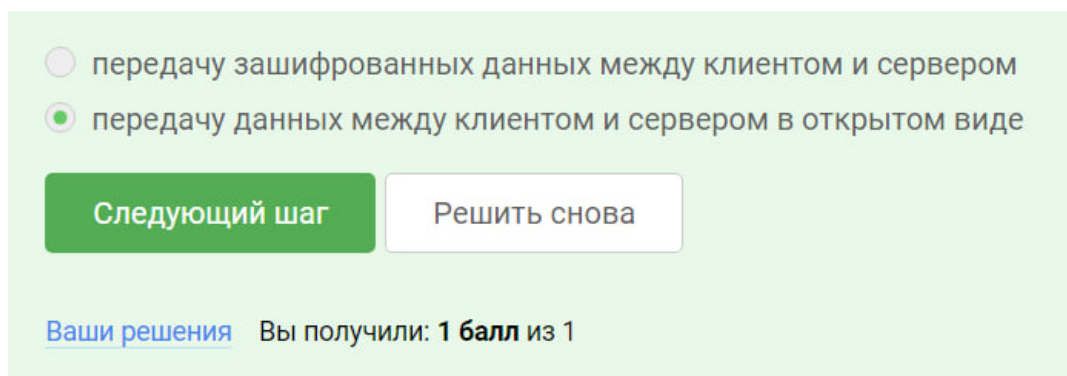
1. Прикладной уровень: Приложение создает данные.
2. Транспортный уровень: TCP или UDP инкапсулирует данные в сегменты и добавляет заголовки, такие как порты для идентификации конечных точек.
3. Сетевой уровень: IP добавляет заголовок, содержащий адрес источника и назначения.
4. Уровень сетевого доступа: Данные передаются через сетевой интерфейс, такой как сетевая карта.

При получении пакета данных устройства выполняют следующую последовательность протоколов в обратном порядке:

1. Уровень сетевого доступа: Данные принимаются через сетевой интерфейс.
2. Сетевой уровень: IP удаляет заголовок и направляет данные на правильный узел.
3. Транспортный уровень: TCP или UDP удаляет заголовок сегмента и извлекает данные.
4. Прикладной уровень: Приложение получает данные.

2.1.6

✓ Хорошая работа.



☐ передачу зашифрованных данных между клиентом и сервером

☒ передачу данных между клиентом и сервером в открытом виде

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.6: .

Протокол HTTP (Hypertext Transfer Protocol) является популярным протоколом передачи данных в сети Интернет, но он имеет ряд недостатков, связанных с безопасностью передачи данных.

Основная причина, по которой данные передаются в открытом виде в HTTP, заключается в том, что этот протокол был изначально разработан для обмена гипертекстовыми документами, а не для передачи конфиденциальной информации.

Вот несколько основных причин, почему HTTP передает данные в открытом виде:

1. Простота реализации: Протокол HTTP был разработан для быстрой и простой передачи веб-страниц, поэтому его разработчики не фокусировались на безопасности передачи данных.
2. Отсутствие шифрования: HTTP не включает в себя механизмы шифрования данных, поэтому информация передается в открытом виде, без защиты от перехвата или модификации.
3. Обратная совместимость: Для обеспечения обратной совместимости с более ранними версиями веб-браузеров и серверов, HTTP продолжает использоваться без шифрования.
4. Производительность: Шифрование данных требует дополнительных вычислительных ресурсов, что может снизить производительность системы.

Для решения этой проблемы безопасности был разработан протокол HTTPS (HTTP Secure), который использует SSL/TLS-шифрование для защиты передаваемой информации. HTTPS обеспечивает конфиденциальность, целостность и аутентификацию данных, делая их менее уязвимыми для перехвата или модификации.

2.1.7

The image shows a quiz interface with four radio button options. The second option, 'двух фаз: рукопожатия и передачи данных' (two phases: handshake and data transfer), is selected with a green dot. Below the options are two buttons: 'Следующий шаг' (Next step) in green and 'Решить снова' (Solve again) in white. At the bottom, it says 'Ваши решения' (Your solutions) and 'Вы получили: 1 балл из 1' (You received: 1 point out of 1).

- ☐ одной фазы аутентификации сервера
- ☒ двух фаз: рукопожатия и передачи данных
- ☐ двух фаз: аутентификация клиента и сервера и шифрования данных
- ☐ трех фаз: аутентификации клиента, аутентификация сервера, генерация общего ключа

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 1.7: .

Протокол HTTPS (HTTP Secure) состоит из двух основных фаз: фаза рукопожатия (handshake) и фаза передачи данных.

1. Фаза рукопожатия (Handshake):

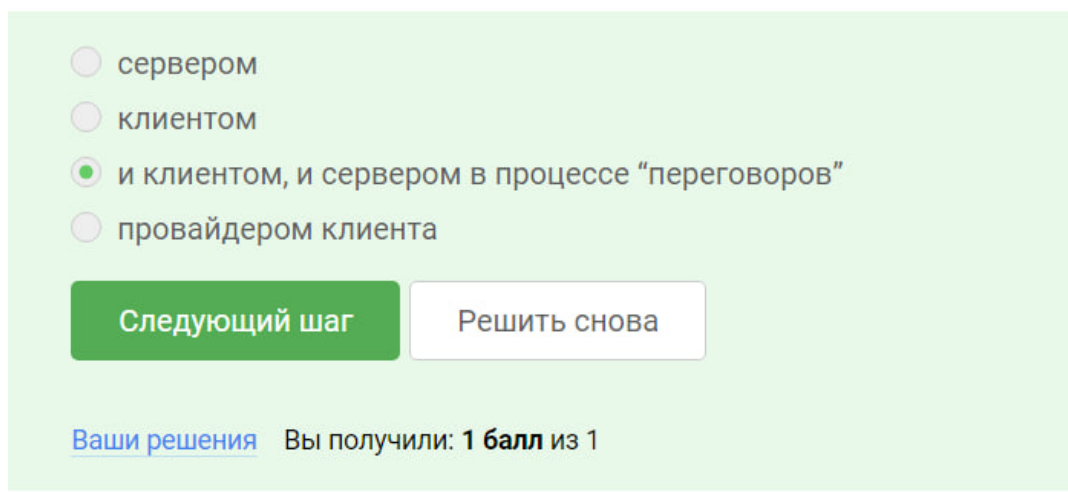
- Клиент (например, веб-браузер) инициирует соединение с сервером, отправляя запрос на установление защищенного HTTPS-соединения.
- Сервер отвечает, отправляя свой SSL/TLS сертификат, который содержит информацию о сервере, включая его открытый криптографический ключ.
- Клиент проверяет подлинность сертификата сервера, используя предварительно установленные корневые сертификаты доверенных центров сертификации.
- Клиент генерирует случайный ключ сеанса (session key) и шифрует его с помощью открытого ключа сервера, полученного из сертификата. Затем клиент отправляет зашифрованный ключ сеанса серверу.
- Сервер расшифровывает ключ сеанса с помощью своего закрытого ключа и подтверждает успешное установление соединения.

2. Фаза передачи данных:

- После завершения фазы рукопожатия, клиент и сервер используют ключ сеанса для шифрования и дешифрования всех последующих данных, передаваемых между ними.
- Этот ключ сеанса используется для симметричного шифрования, что обеспечивает высокую скорость шифрования/дешифрования в сравнении с асимметричной криптографией, использованной в фазе рукопожатия.
- Шифрование данных на этой стадии гарантирует их конфиденциальность, целостность и аутентичность в процессе передачи между клиентом и сервером.

Таким образом, фаза рукопожатия устанавливает защищенное SSL/TLS-соединение, а фаза передачи данных использует этот канал для безопасной передачи информации между клиентом и сервером.

2.1.8



The image shows a quiz interface with a light green background. It contains four radio button options: "сервером", "клиентом", "и клиентом, и сервером в процессе “переговоров”", and "провайдером клиента". The third option is selected. Below the options are two buttons: "Следующий шаг" (green) and "Решить снова" (white). At the bottom, it says "Ваши решения" followed by "Вы получили: 1 балл из 1".

Рис. 1.8: .

Вы правы, версия протокола TLS в HTTPS-соединении определяется в результате “переговоров” между клиентом и сервером.

Процесс определения версии TLS происходит следующим образом:

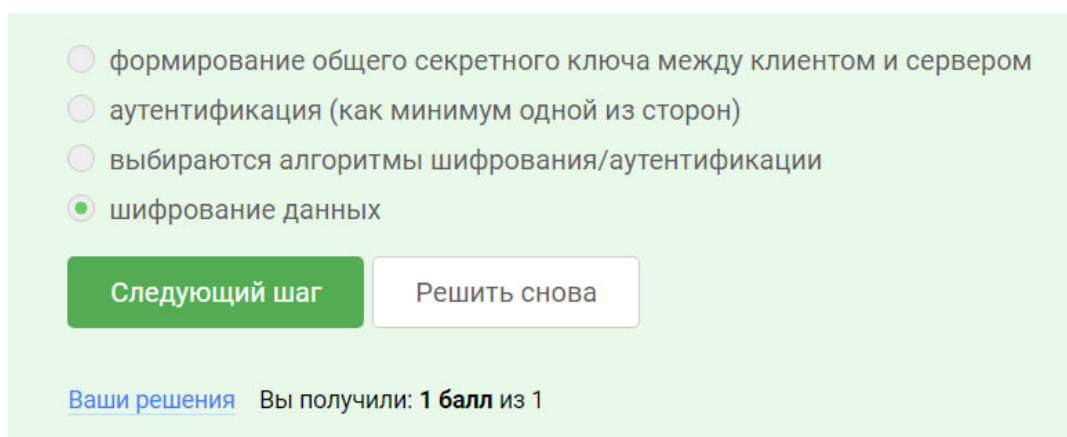
1. Клиент инициирует соединение и предлагает список версий TLS, которые он поддерживает (например, TLS 1.2, TLS 1.3).
2. Сервер анализирует список версий, предложенных клиентом, и выбирает наиболее подходящую версию TLS, которую он также поддерживает.
3. Сервер отправляет выбранную версию TLS в ответ клиенту, сообщая, какую версию он будет использовать для дальнейшего соединения.
4. Клиент подтверждает использование выбранной версии TLS, и дальнейший обмен данными происходит с использованием этой версии протокола.

Этот процесс “переговоров” позволяет клиенту и серверу согласовать единую версию TLS, которая будет применяться для защищенной передачи данных. Он

важен, так как разные версии TLS могут иметь различные алгоритмы шифрования, возможности и уровни безопасности.

Таким образом, окончательная версия TLS определяется в результате совместного решения клиента и сервера, основанного на поддерживаемых ими версиях протокола. Это обеспечивает совместимость и безопасность HTTPS-соединения.

2.1.9



☐ 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** из 1

Рис. 1.9: .

Вы правы, в фазе “рукопожатия” (handshake) протокола TLS данные не шифруются. Это связано с особенностями этой начальной фазы установления защищенного соединения.

Во время фазы “рукопожатия” происходит следующее:

1. Клиент отправляет серверу сообщение “Client Hello”, в котором указывает поддерживаемые им версии TLS, алгоритмы шифрования и хеширования.
2. Сервер отвечает сообщением “Server Hello”, в котором выбирает версию TLS и алгоритмы, которые будут использоваться.
3. Сервер также отправляет свой цифровой сертификат, который содержит открытый ключ сервера.

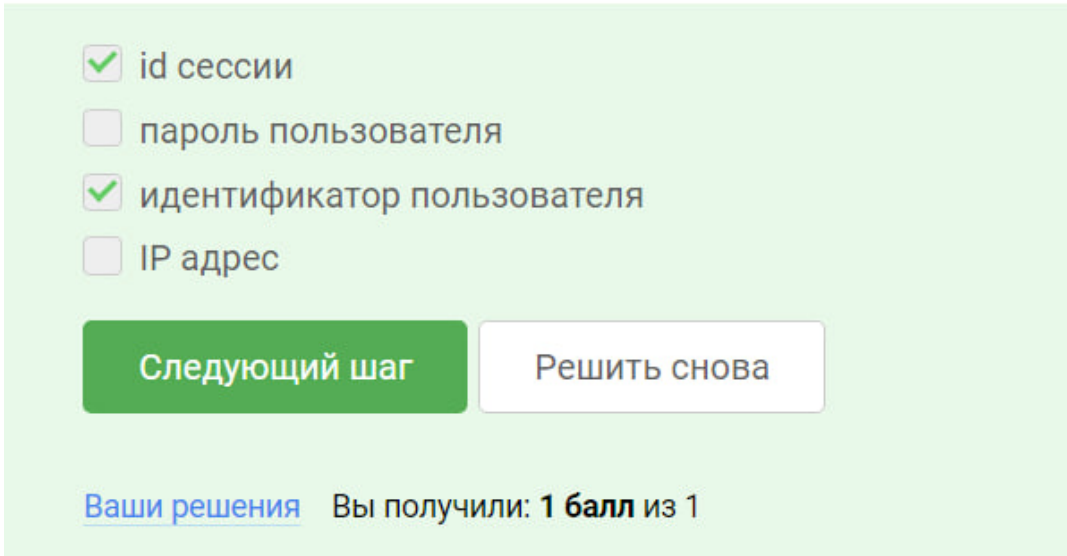
4. Клиент проверяет сертификат сервера и, если он валидный, генерирует секретный ключ сеанса (pre-master secret).
5. Клиент шифрует pre-master secret открытым ключом сервера, полученным из сертификата, и отправляет его серверу.
6. Сервер расшифровывает pre-master secret своим закрытым ключом.

На этом этапе клиент и сервер вычисляют общий секретный ключ сеанса, который будет использоваться для шифрования данных в дальнейшем.

Важно отметить, что до момента генерации общего секретного ключа сеанса данные, передаваемые между клиентом и сервером, не шифруются. Это необходимо для того, чтобы успешно провести процедуру “рукопожатия” и согласовать ключ.

Только после завершения фазы “рукопожатия” соединение переходит к шифрованию данных с применением согласованных алгоритмов и ключа сеанса.

2.2.1



The image shows a web form with a light green background. It contains four checkboxes with labels in Russian: 'id сессии' (checked), 'пароль пользователя' (unchecked), 'идентификатор пользователя' (checked), and 'IP адрес' (unchecked). Below the checkboxes are two buttons: a green 'Следующий шаг' (Next step) button and a white 'Решить снова' (Solve again) button. At the bottom, there is a blue link 'Ваши решения' (Your solutions) followed by the text 'Вы получили: 1 балл из 1' (You received: 1 point out of 1).

Рис. 1.10: .

Вы правы, куки (cookies) могут хранить различную информацию, в том числе идентификатор сессии и идентификатор пользователя. Рассмотрим подробнее,

какую роль они играют:

1. Идентификатор сессии (session ID):

- Когда пользователь входит в систему, сервер генерирует уникальный идентификатор сессии.
- Этот идентификатор сохраняется в куке на клиентской стороне.
- При каждом последующем запросе от пользователя, браузер отправляет этот идентификатор сессии на сервер.
- Сервер использует этот идентификатор, чтобы связать текущий запрос с сессией пользователя и, таким образом, поддерживать состояние между запросами.
- Это позволяет серверу помнить, что данный пользователь уже авторизован и имеет определенные права доступа.

2. Идентификатор пользователя (user ID):

- Когда пользователь авторизуется в системе, сервер может сохранять его уникальный идентификатор (user ID) в куке.
- Это позволяет серверу определять, кто именно выполняет текущий запрос, даже если сессия была утеряна или истекла.
- Сохранение user ID в куке обеспечивает возможность персонализации контента и настроек для конкретного пользователя.

Использование куков для хранения идентификаторов сессии и пользователя является распространенной практикой в веб-приложениях. Это позволяет поддерживать состояние между запросами и обеспечивать персонализированный опыт для каждого пользователя.

Важно отметить, что куки могут содержать и другую информацию, например, предпочтения пользователя, корзину покупок и т.д. Однако идентификатор сессии и идентификатор пользователя являются наиболее распространенными данными, хранящимися в куках.

2.2.2

☐ аутентификации пользователя

☐ персонализации веб-страниц

☐ отслеживания информации о пользователе

☐ сборе статистики посещаемости сайта

☒ улучшения надежности соединения

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.11: .

Потому что куки не связаны напрямую с надежностью соединения.

2.2.3

☒ сервером

☐ клиентом

Следующий шаг Решить снова

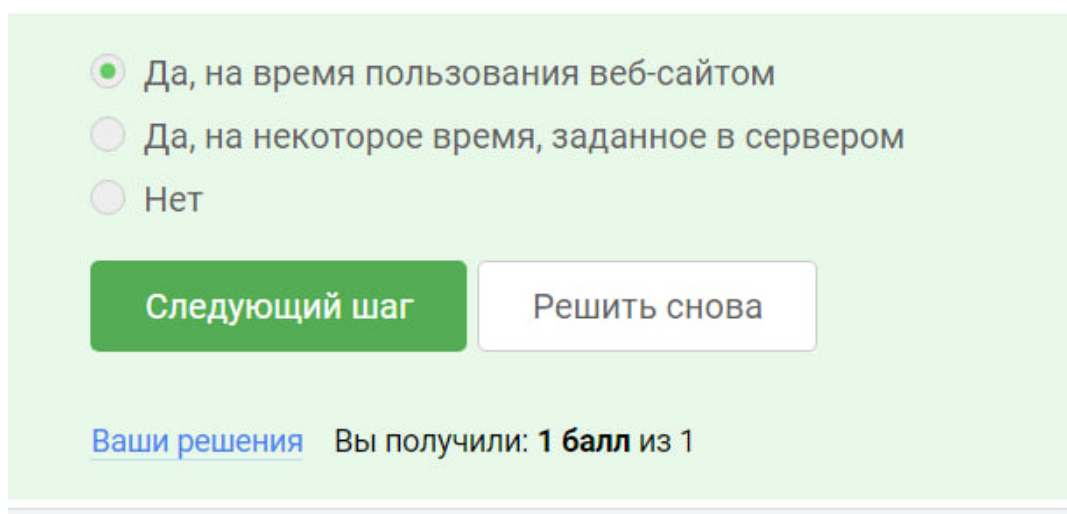
[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.12: .

Куки генерируются сервером, потому что:

Защищенная среда: Сервер — это более безопасная среда, чем браузер пользователя, что снижает риск атак и злонамеренной деятельности. Централизованный контроль: Сервер имеет централизованный контроль над созданием и управлением куки, что гарантирует согласованность и соблюдение политик. Эффективность: Сервер может генерировать куки более эффективно, чем браузер, особенно в случае больших и сложных куки. Безопасность: Сервер может применять механизмы безопасности, такие как шифрование и аутентификация, для защиты данных куки. Управление сессиями: Сервер использует куки для управления сеансами пользователей и отслеживает их деятельность на сайте. Персонализация: Сервер может использовать куки для хранения и управления пользовательскими предпочтениями, такими как тема, язык и настройки контента. Анализ и отслеживание: Куки помогают серверу отслеживать активность пользователей на сайте и собирать аналитические данные.

2.2.4



The image shows a survey interface with a light green background. It contains three radio button options for the question: "Да, на время пользования веб-сайтом" (selected), "Да, на некоторое время, заданное в сервером", and "Нет". Below the options are two buttons: "Следующий шаг" (green) and "Решить снова" (white with a grey border). At the bottom, it says "Ваши решения" followed by "Вы получили: 1 балл из 1".

☒ Да, на время пользования веб-сайтом

☐ Да, на некоторое время, заданное в сервером

☐ Нет

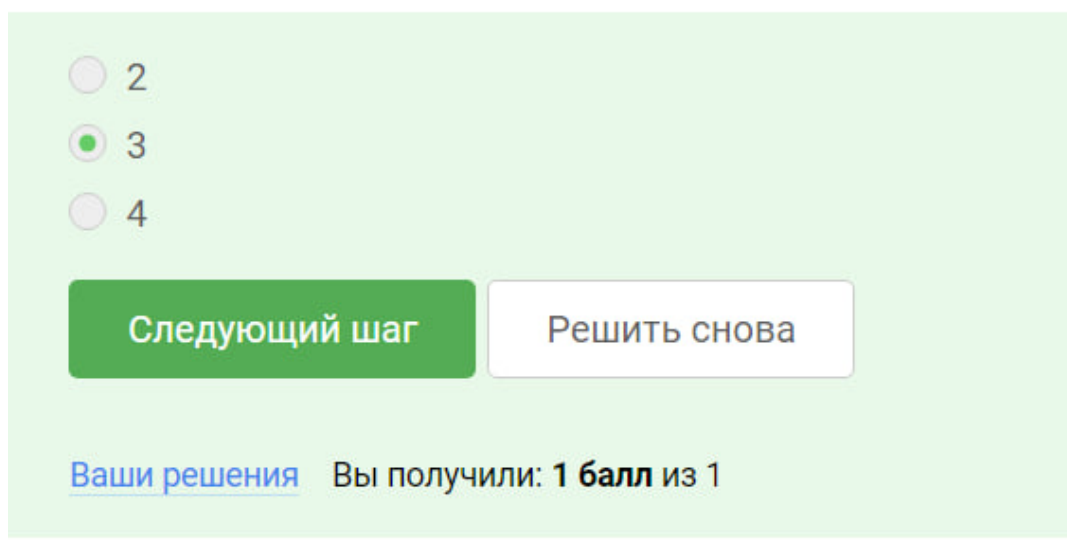
Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.13: .

сессионные куки хранятся в браузере на время пользования веб-сайтом. Они удаляются, когда пользователь закрывает браузер. Сессионные куки используются для временного хранения данных, связанных с текущим сеансом пользователя.

2.3.1



☐ 2

☒ 3

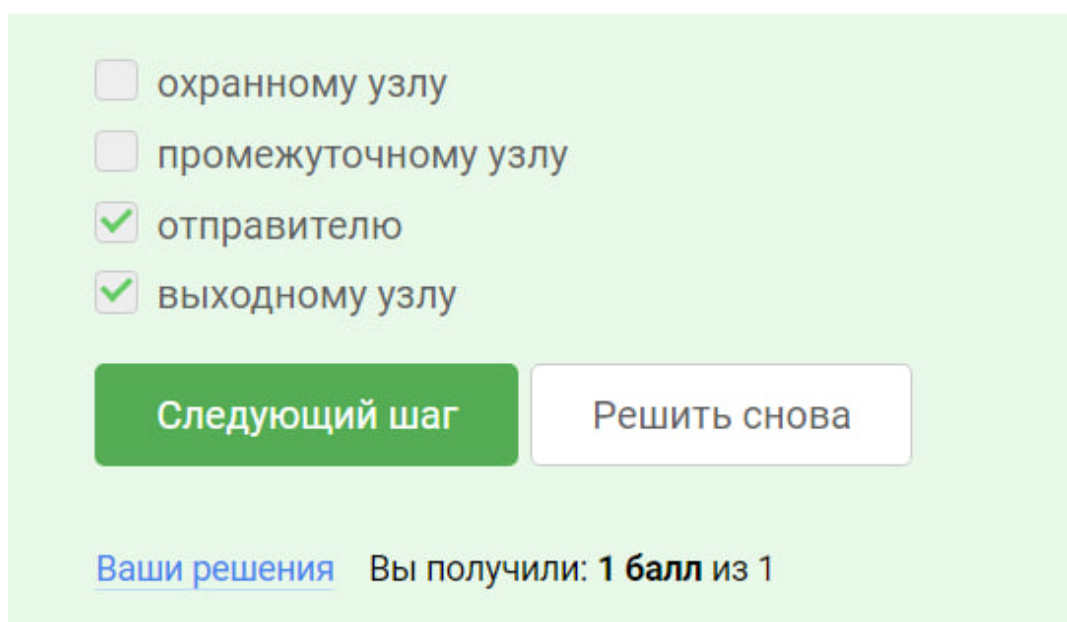
☐ 4

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.14: .

В луковой сети TOR передаваемые данные шифруются трижды и последовательно передаются через три промежуточных узла, прежде чем достигнут конечного получателя.

2.3.2



☐ охранному узлу

☐ промежуточному узлу

☒ отправителю

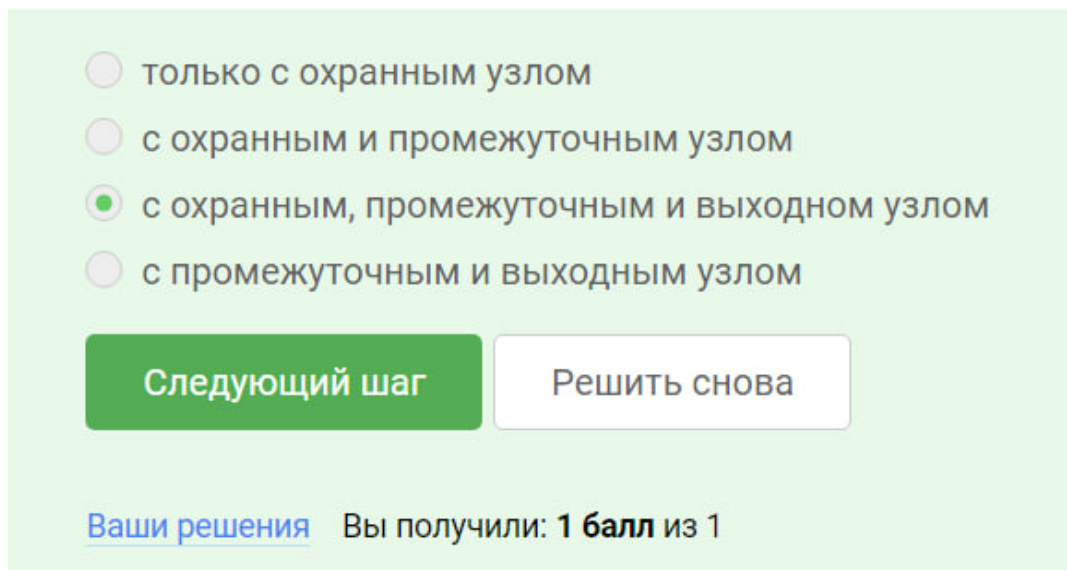
☒ выходному узлу

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.15: .

Отправителю: IP-адрес получателя известен отправителю, так как он включен в заголовок пакета, отправляемого получателю. Выходному узлу: IP-адрес получателя известен выходному узлу, так как выходной узел передает пакет от отправителя к получателю.

2.3.3



The image shows a quiz interface on a light green background. It contains four radio button options for selecting a path through nodes:

- ☐ только с охранным узлом
- ☐ с охранным и промежуточным узлом
- ☒ с охранным, промежуточным и выходным узлом
- ☐ с промежуточным и выходным узлом

Below the options are two buttons: a green button labeled "Следующий шаг" (Next step) and a white button with a grey border labeled "Решить снова" (Solve again).

At the bottom, there is a link "Ваши решения" (Your solutions) and a score display: "Вы получили: 1 балл из 1" (You received: 1 point out of 1).

Рис. 1.16: .

Генерация общего секретного ключа отправляющей стороной

Отправитель генерирует общий секретный ключ, который будет использоваться для защищенного обмена сообщениями с охранным, промежуточным и выходным узлами в сети Tor. Этот ключ включает в себя:

С охранным узлом:

Краткосрочный идентификатор (Ephemeral Identity, EI): Уникальный одноразовый ключ, используемый для аутентификации с охранным узлом. Долговременный ключ (Long-Term Key, LTK): Ключ для долгосрочной аутентификации сохранен на охранном узле.

С промежуточным узлом:

Скрытый сервис (Onion Service): Скрытый адрес, используемый для отправки сообщений промежуточному узлу.

С выходным узлом:

Общий секретный ключ (Shared Secret Key, SSHK): Ключ, совместно используемый отправителем и выходным узлом, сгенерированный во время процедуры установления соединения.

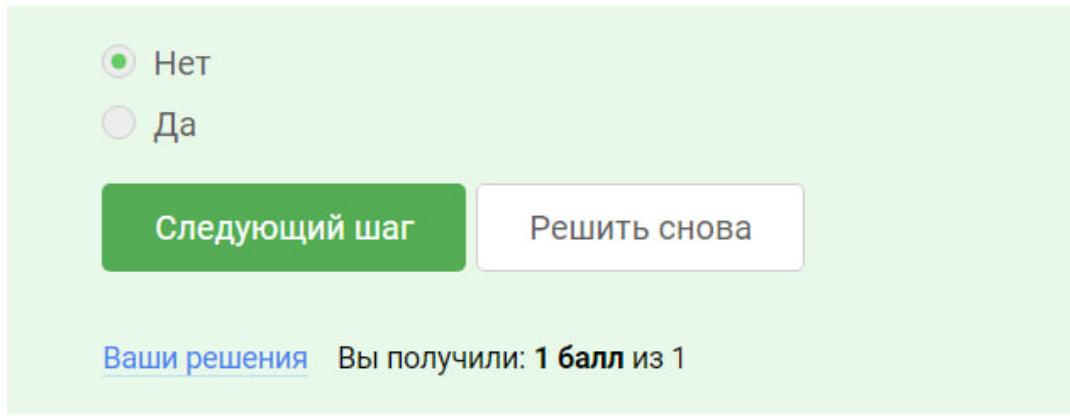
Процесс генерации

Отправитель выполняет следующие шаги для генерации общего секретного ключа:

1. Генерирует EI и LTK.
2. Отправляет EI на охранный узел.
3. Получает LTK от охранного узла.
4. Создает скрытый сервис и предоставляет его промежуточному узлу.
5. Устанавливает прямое соединение с выходным узлом и генерирует SSHK.

Этот процесс гарантирует, что только отправитель и соответствующие узлы могут расшифровать сообщения, передаваемые через сеть Tor.

2.3.4

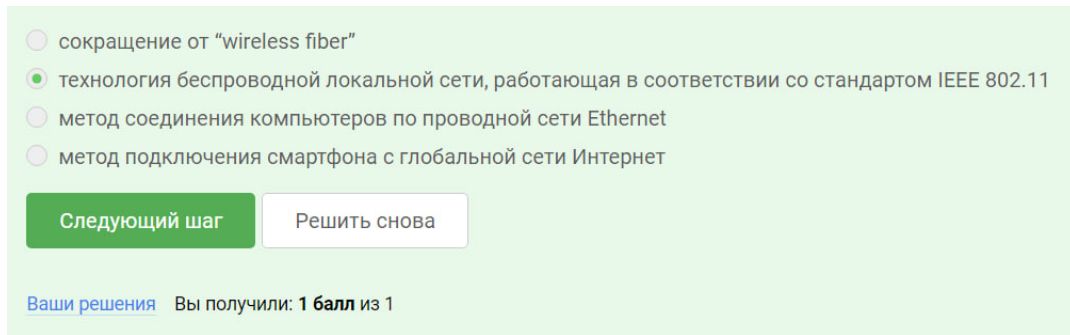


The image shows a confirmation dialog box with a light green background. At the top, there are two radio buttons: the first is selected and labeled 'Нет' (No), and the second is unselected and labeled 'Да' (Yes). Below the buttons are two rectangular buttons: a green one labeled 'Следующий шаг' (Next step) and a white one with a grey border labeled 'Решить снова' (Solve again). At the bottom, there is a link 'Ваши решения' (Your solutions) followed by the text 'Вы получили: 1 балл из 1' (You received: 1 point out of 1).

Рис. 1.17: .

получатель не обязан использовать браузер Tor для успешного получения пакетов. Браузер Tor основан на луковой маршрутизации, которая шифрует и анонимизирует трафик, но эта функция не требуется для получения пакетов. Для получения пакетов получателю нужен только обычный веб-браузер.

2.4.1



☐ сокращение от "wireless fiber"

☒ технология беспроводной локальной сети, работающая в соответствии со стандартом IEEE 802.11

☐ метод соединения компьютеров по проводной сети Ethernet

☐ метод подключения смартфона с глобальной сети Интернет

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.18: .

Wi-Fi (Wireless Fidelity) - это беспроводная технология локальной сети, которая позволяет устройствам подключаться к интернету и друг к другу без использования проводов. Она работает в соответствии со стандартом IEEE 802.11, который определяет технические характеристики и протоколы связи для беспроводных сетей.

Стандарт IEEE 802.11 включает несколько различных вариантов, каждый из которых имеет свои возможности и характеристики. Наиболее распространенными вариантами являются:

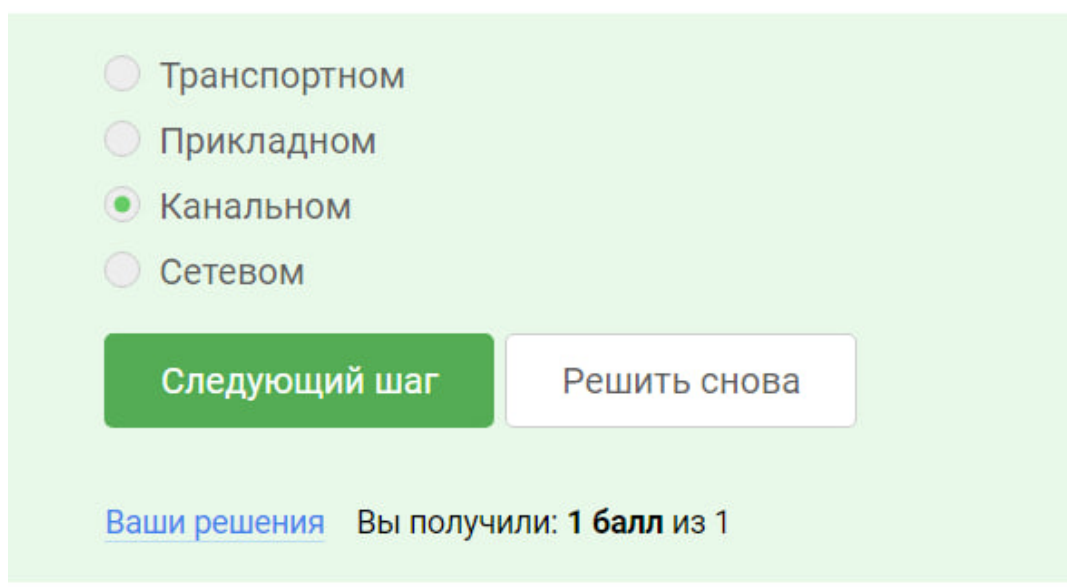
802.11a: Работает в диапазоне 5 ГГц, обеспечивая высокую скорость передачи данных, но с меньшей дальностью действия. 802.11b: Работает в диапазоне 2,4 ГГц, обеспечивая более низкую скорость передачи данных, но с большей дальностью действия. 802.11g: Работает в диапазоне 2,4 ГГц, обеспечивая среднюю скорость передачи данных между 802.11a и 802.11b. 802.11n: Работает как в диапазоне 2,4 ГГц, так и в диапазоне 5 ГГц, обеспечивая более высокую скорость передачи данных и расширенный радиус действия. 802.11ac: Работает только в диапазоне 5 ГГц, обеспечивая еще более высокую скорость передачи данных и меньшую задержку.

Ключевые характеристики Wi-Fi

Беспроводное соединение: Позволяет устройствам подключаться к сети без проводов, обеспечивая свободу перемещения. Диапазон: Дальность действия

Wi-Fi-сигнала зависит от стандарта 802.11 и препятствий в среде. Скорость передачи данных: Скорость передачи данных варьируется в зависимости от стандарта 802.11, частотного диапазона и других факторов. Безопасность: Wi-Fi-сети можно защитить с помощью различных методов, таких как пароли, шифрование и брандмауэры. Удобство использования: Wi-Fi-сети просты в настройке и использовании, что позволяет легко подключать устройства к интернету и обмениваться данными.

2.4.2



☐ Транспортном

☐ Прикладном

☒ Канальном

☐ Сетевом

Следующий шаг **Решить снова**

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.19: .

Потому что протокол WiFi работает на Канальном уровне модели OSI, которая отвечает за физическую передачу данных и управление доступом к общей среде передачи.

2.4.3

The image shows a quiz interface with a light green background. At the top, there are four radio button options: WPA, WEP, WPA2, and WPA3. The WEP option is selected, indicated by a green dot. Below the options are two buttons: a green button labeled 'Следующий шаг' (Next step) and a white button with a grey border labeled 'Решить снова' (Solve again). At the bottom, there is a link 'Ваши решения' (Your solutions) and a score display 'Вы получили: 1 балл из 1' (You received: 1 point out of 1).

Рис. 1.20: .

WEP (Wired Equivalent Privacy), эквивалент проводного шифрования, является небезопасным методом обеспечения шифрования и аутентификации в сети Wi-Fi из-за следующих причин:

Слабые ключи шифрования: WEP использует 128-битные (или 64-битные) ключи шифрования, которые легко взломать с помощью атак грубой силы или атак повторного воспроизведения. Уязвимость IV: WEP использует инициализирующий вектор (IV) для шифрования каждого пакета, что позволяет злоумышленникам восстанавливать IV и расшифровывать сообщения. Атака повторного воспроизведения: Злоумышленники могут повторно перехватывать и проигрывать пакеты, зашифрованные WEP, что позволяет им обходить защиту шифрования. Атака воровского узла: Злоумышленники могут присоединиться к сети Wi-Fi и внедрять воровской узел, который перехватывает и модифицирует трафик, проходящий через сеть. Устаревшая технология: WEP была разработана в 1999 году и устарела более современными и безопасными протоколами, такими как WPA2 и WPA3.

2.4.4

☒ передаются в зашифрованном виде после аутентификации устройств

☐ передаются в открытом виде после аутентификации устройств

☐ передаются в зашифрованном виде

☐ передаются в открытом виде

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.21: .

После аутентификации устройств для подключения к сети данные между хостом (компьютером или смартфоном) и роутером передаются в зашифрованном виде для обеспечения безопасности и конфиденциальности.

Этот процесс обычно включает следующие шаги:

1. Аутентификация:

- Устройство клиента (хост) передает свои учетные данные (например, имя пользователя и пароль) роутеру для проверки полномочий.
- Роутер проверяет учетные данные и разрешает или отклоняет доступ в соответствии с настроенными правилами.

2. Настройка шифрования:

- После успешной аутентификации устанавливается защищенное соединение между хостом и роутером.
- Выбирается протокол шифрования, например WPA2 или WPA3, и генерируется уникальный ключ шифрования.

3. Шифрование данных:

- Перед передачей все данные между хостом и роутером шифруются с использованием согласованного ключа шифрования.

- Это гарантирует, что данные остаются конфиденциальными и защищены от перехвата или взлома.

4. Расшифровка данных:

- Когда данные достигают пункта назначения (хоста или роутера), они расшифровываются с использованием того же ключа шифрования.
- После расшифровки данные становятся доступными и читаемыми для авторизованного устройства.

Шифрование данных между хостом и роутером выполняет следующие функции:

Конфиденциальность: Защищает данные от несанкционированного доступа и прослушивания. Целостность данных: Обеспечивает, что данные не были изменены или повреждены во время передачи. Аутентичность: Подтверждает подлинность источника данных и предотвращает спуфинг или подделку.

2.4.5

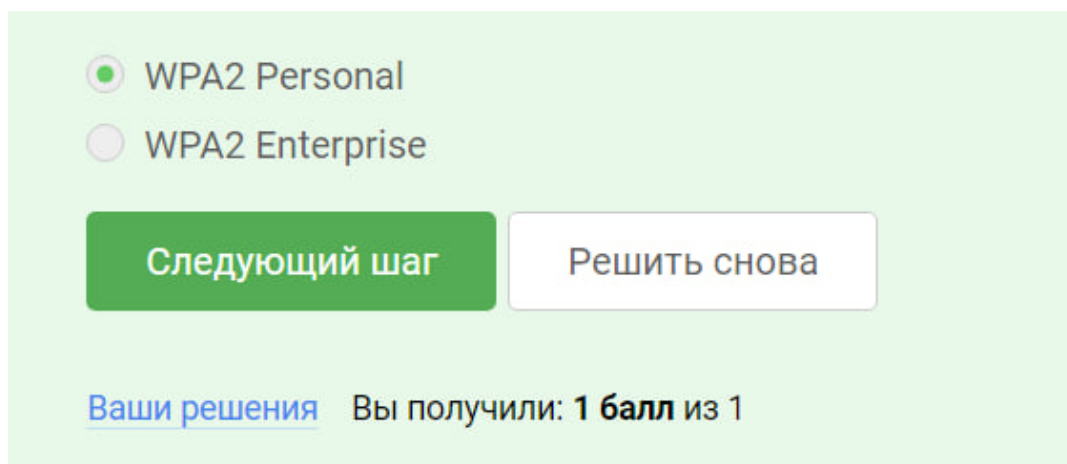


Рис. 1.22: .

WPA2 Personal используется в качестве метода аутентификации для домашних сетей по нескольким причинам:

Безопасность: WPA2 Personal обеспечивает более высокий уровень безопасности по сравнению с предыдущими методами аутентификации, такими как WEP,

за счет использования протокола Advanced Encryption Standard (AES). AES известен своей криптографической стойкостью и защищает передаваемые данные от несанкционированного доступа.

Легкость использования: WPA2 Personal прост в настройке и не требует использования сертификатов или сервера аутентификации, как в случае WPA2 Enterprise. Это делает его идеальным выбором для домашних пользователей, которые не обладают техническими знаниями.

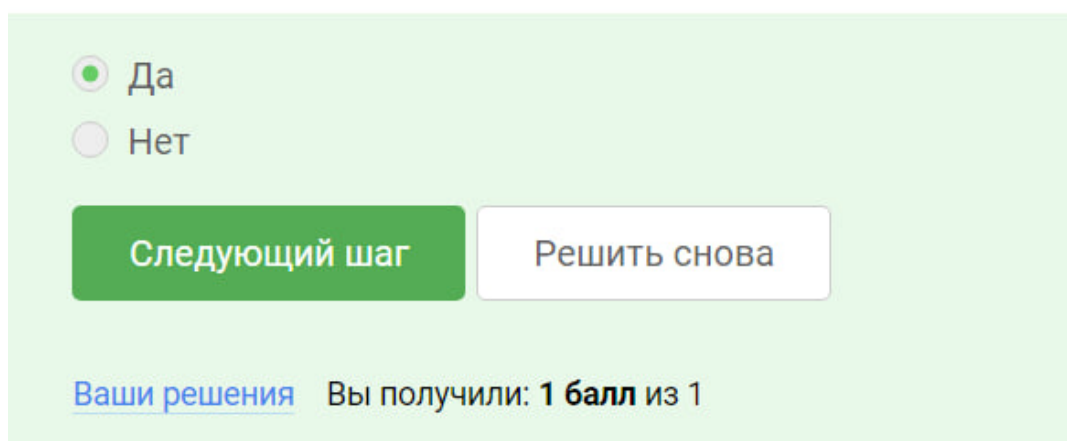
Распространенность: WPA2 Personal широко поддерживается клиентами беспроводной связи и маршрутизаторами. Это гарантирует совместимость с большинством устройств, позволяя пользователям легко подключаться к домашним сетям.

Защита от перехвата: WPA2 Personal использует динамический ключ, который меняется при каждом подключении. Это предотвращает перехват злоумышленниками ключей шифрования и доступ к конфиденциальным данным, передаваемым по сети.

Компромисс между безопасностью и удобством: WPA2 Personal предлагает хороший компромисс между безопасностью и удобством использования. Он обеспечивает достаточный уровень безопасности для домашних сетей, не жертвуя при этом удобством использования.

В отличие от WPA2 Enterprise, который используется в корпоративных средах, WPA2 Personal предназначен для упрощения аутентификации в домашних сетях. Он не требует сложной настройки и позволяет пользователям подключаться к сети с использованием предварительно согласованного пароля.

3.1.1



☒ Да
☐ Нет

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

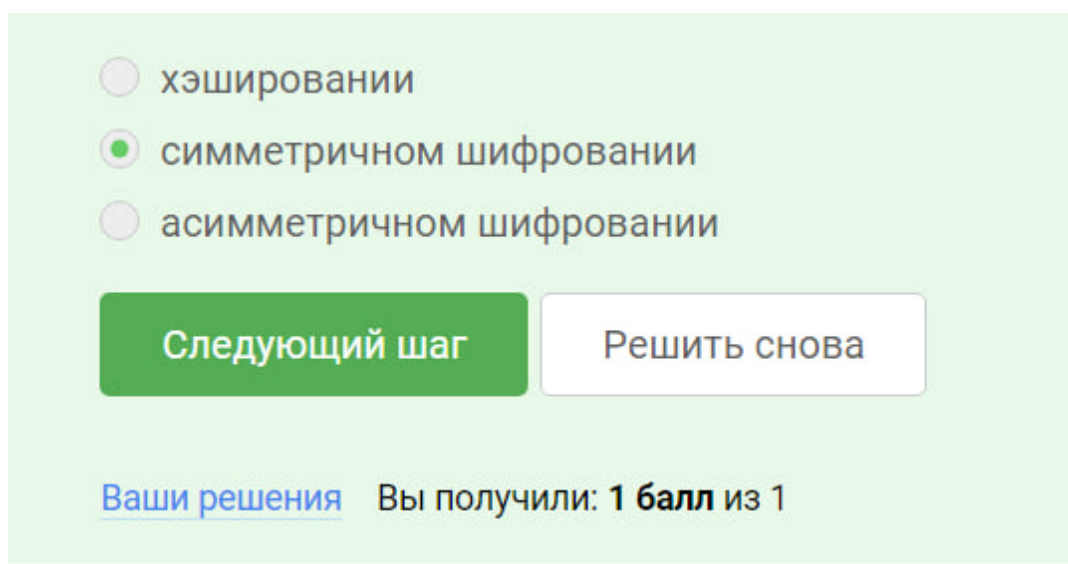
Рис. 1.23: .

Да, можно зашифровать загрузочный сектор диска.

Причины:

Предотвращение несанкционированной загрузки: Шифрование загрузочного сектора делает невозможным загрузку операционной системы без ключа шифрования. Это защищает компьютер от несанкционированного доступа и атак с использованием вредоносных программ. Защита важных данных: Загрузочный сектор содержит важные данные, такие как таблица разделов и информация о загрузке. Шифрование этих данных защищает их от модификации или уничтожения. Соблюдение нормативных требований: Некоторые отрасли и организации требуют шифрования загрузочных секторов для соответствия нормативным требованиям. Защита от атак с помощью вредоносных программ: Шифрование загрузочного сектора затрудняет для вредоносных программ установку и модификацию системных файлов.

3.1.2



☐ хэшировании

☒ симметричном шифровании

☐ асимметричном шифровании

Следующий шаг

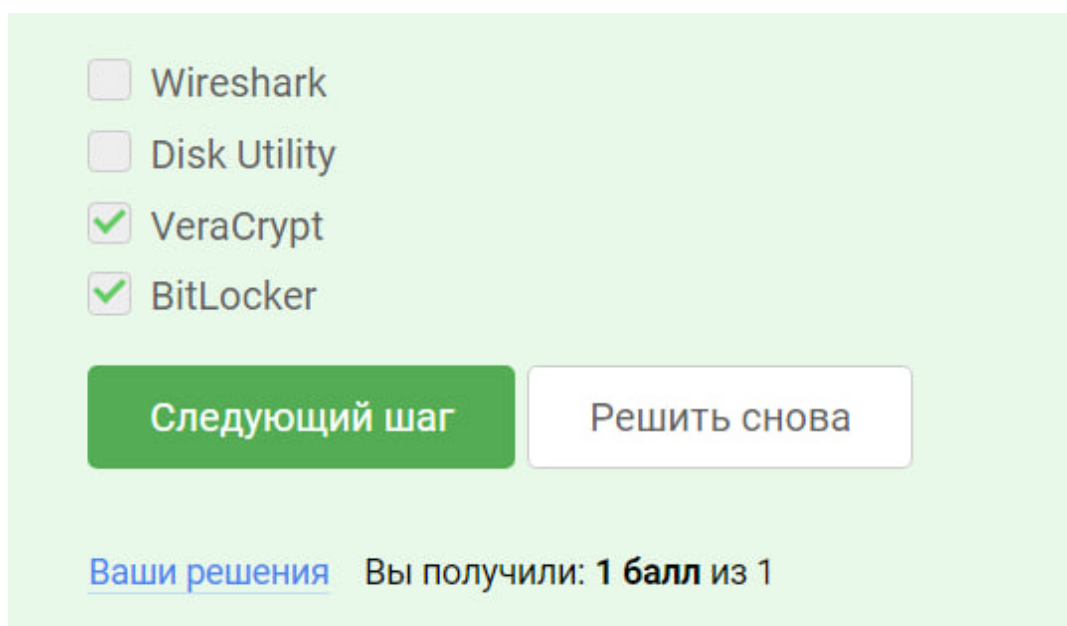
Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.24: .

Симметричное шифрование предполагает использование одного и того же ключа для шифрования и расшифровки данных. При шифровании диска применяется этот подход, поскольку ключ хранится внутри устройства и используется для защиты всех данных на диске. Симметричное шифрование обеспечивает высокую скорость шифрования и расшифровки, что делает его подходящим выбором для шифрования больших объемов данных на диске.

3.1.3



☐ Wireshark

☐ Disk Utility

☒ VeraCrypt

☒ BitLocker

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.25: .

Обе приведенные программы, VeraCrypt и BitLocker, могут быть использованы для шифрования жестких дисков. Вот их краткое сравнение:

VeraCrypt

Бесплатная и с открытым исходным кодом Доступно для Windows, macOS, Linux и Android Поддерживает несколько алгоритмов шифрования, включая AES, Serpent и Twofish Создает зашифрованные тома, которые можно монтировать как обычные диски Имеет расширенные функции безопасности, такие как скрытые тома и связанные тома

BitLocker

Встроен в Windows Доступен только для Windows Поддерживает алгоритм шифрования AES Шифрует весь жесткий диск или отдельные разделы Имеет функции самовосстановления и защиты от атак грубой силы

Почему обе программы полезны:

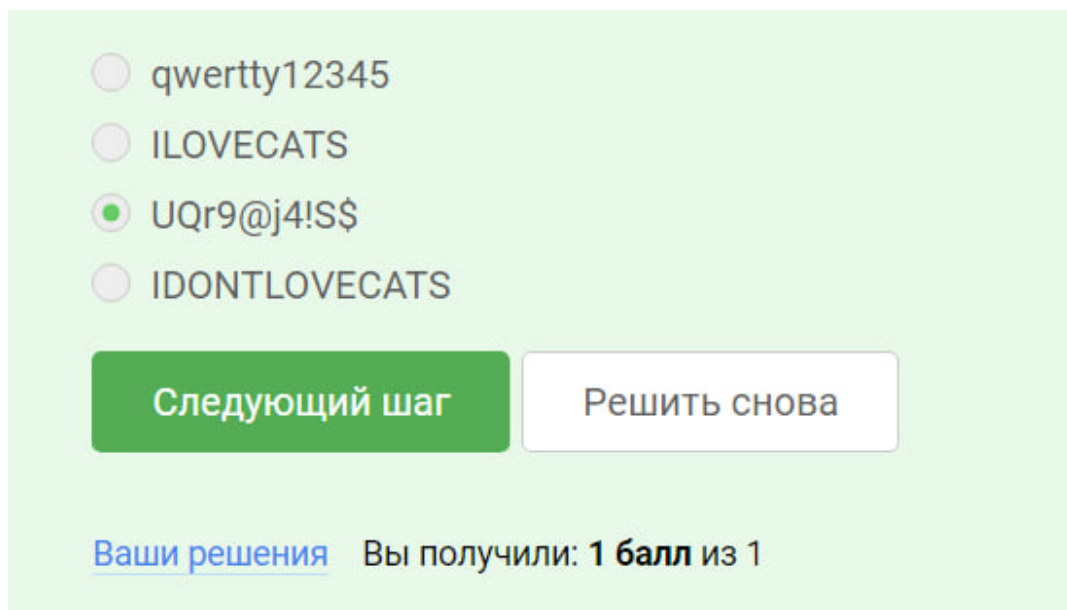
VeraCrypt является отличным выбором для тех, кто ищет бесплатное и многоплатформенное решение для шифрования дисков. Он предлагает широкий набор функций и алгоритмов, чтобы удовлетворить различные потребности в

безопасности.

BitLocker лучше подходит для пользователей Windows, ищущих удобный и встроенный способ зашифровать свой жесткий диск. Он прост в использовании и предоставляет базовый уровень безопасности.

В конечном счете, выбор лучшей программы зависит от конкретных требований и предпочтений пользователя.

3.2.1



☐ qwerty12345

☐ ILOVECATS

☒ UQr9@j4!S\$

☐ IDONTLOVECATS

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.26: .

Пароль UQr9@j4!S\$ можно отнести к стойким по следующим причинам:

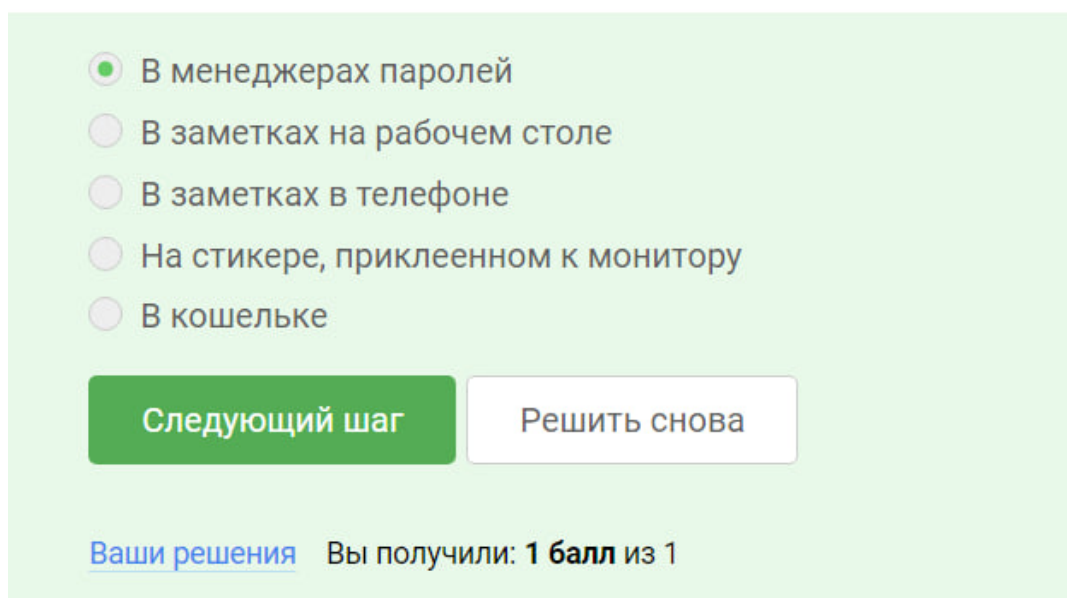
Длина: 11 символов, что превышает рекомендованный минимум в 8 символов. Разнообразие символов: Пароль включает строчные и прописные буквы, цифры и специальные символы. Отсутствие общеупотребительных слов или фраз: Пароль не содержит легко угадываемых слов или выражений. Сложность для угадывания: Сочетание длины, разнообразия символов и отсутствия общеупотребительных слов делает пароль сложным для угадывания с помощью автоматизированных инструментов.

Дополнительно, пароль также имеет следующие характеристики:

Наличие символов из разных регистров (строчные и прописные буквы): Это увеличивает сложность угадывания пароля с помощью атаки перебора грубой силой. Использование специальных символов: Специальные символы, такие как “!”, “\$” и “%”, значительно расширяют пространство возможных паролей, что затрудняет их взлом. Отсутствие последовательных символов: Пароль не содержит последовательностей символов, таких как “1234” или “abcd”, которые легко угадать.

В целом, пароль UQr9@j4!S\$ можно считать стойким, поскольку он отвечает большинству критериев, необходимых для надежного пароля.

3.2.2



☒ В менеджерах паролей

☐ В заметках на рабочем столе

☐ В заметках в телефоне

☐ На стикере, приклеенном к монитору

☐ В кошельке

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.27: .

Менеджеры паролей — это безопасный и удобный способ хранения паролей. Они используют шифрование и двухфакторную аутентификацию для защиты ваших паролей от несанкционированного доступа. Вот почему менеджеры паролей являются одним из самых надежных и рекомендуемых методов хранения паролей:

3.2.3

☒ Для защиты от автоматизированных атак, направленных на получение несанкционированного доступа
☐ Для безопасного хранения паролей на сервере
☐ Она заменяет пароли
☐ Для защиты кук пользователя

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 1.28: .

Используя капчи, веб-сайты и приложения могут различать людей и ботов, защищая себя от автоматизированных атак и обеспечивая доступность и безопасность своих услуг.

3.2.4

☐ Для того, чтобы пароль не передавался в открытом виде.
☐ Для того, чтобы ускорить процесс авторизации
☒ Для того, чтобы не хранить пароли на сервере в открытом виде.
☐ Для удобства разработчиков

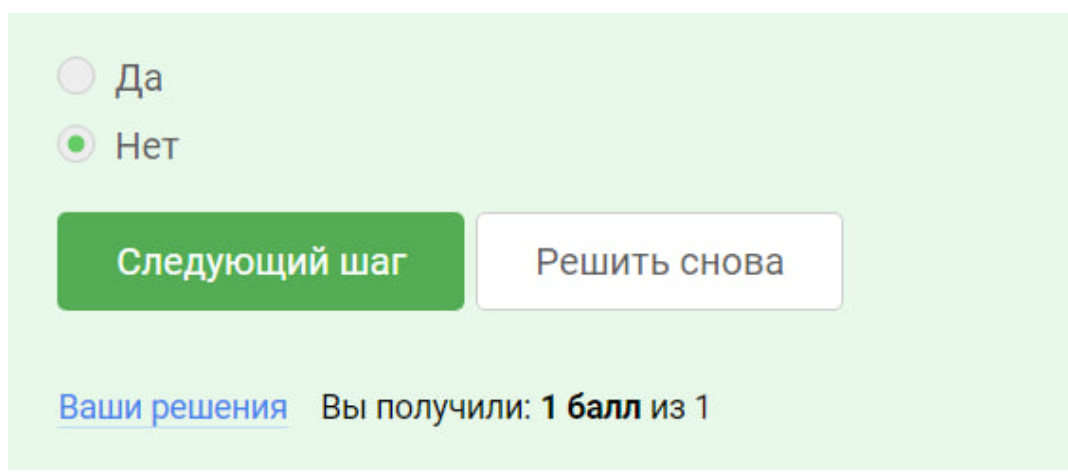
Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 1.29: .

Хэширование паролей применяется для того, чтобы не хранить пароли на сервере в открытом виде. Таким образом, даже если злоумышленник получит доступ к базе данных с паролями, он не сможет использовать их для входа в систему, так как они будут храниться в виде хэшей - односторонних функций, которые невозможно обратить.

3.2.5



☐ Да

☒ Нет

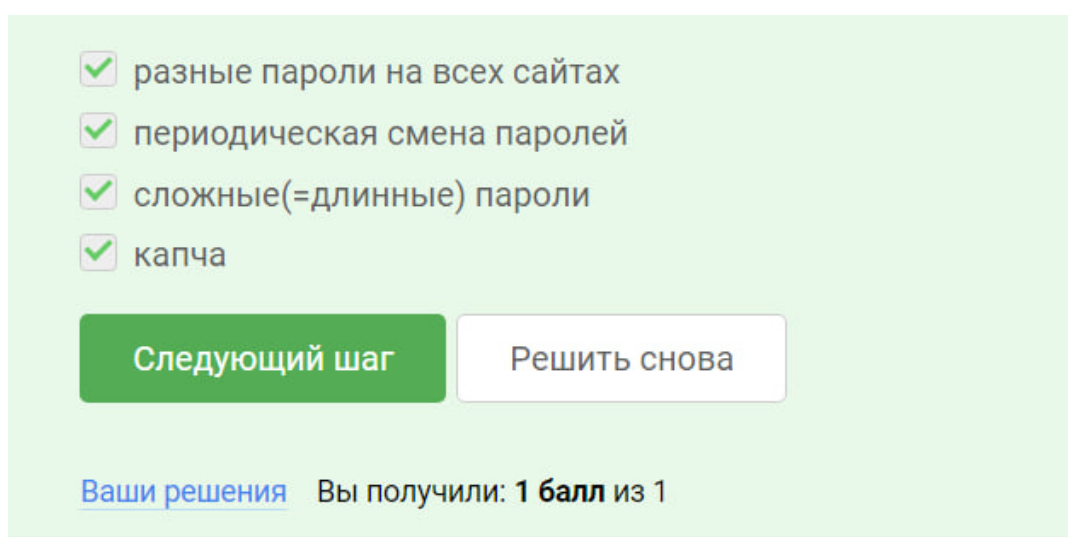
Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.30: .

Если злоумышленник получил доступ к серверу, у него будет прямой доступ к базе данных паролей, и он сможет извлечь пароли в виде хешей. Добавление соли к паролям не изменит значения хешей, поэтому злоумышленник все равно сможет совершить атаку перебором с теми же хешами.

3.2.6



☒ разные пароли на всех сайтах

☒ периодическая смена паролей

☒ сложные(=длинные) пароли

☒ капча

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.31: .

сложные(=длинные) пароли

3.3.1

☐ <https://accounts.google.com.br/signin/v2/identifier?hl=ru> (страница входа в аккаунт Google)
☒ <https://online.sberbank.wix.ru/CSAFront/index.do> (вход в Сбербанк.Онлайн)
☐ https://e.mail.ru/login?lang=ru_RU (вход в аккаунт Mail.Ru)
☒ https://passport.yandex.ucoz.ru/auth?origin=home_desktop_ru (вход в аккаунт Яндекс)

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 1.32: .

Фишинговыми являются следующие ссылки:

<https://online.sberbank.wix.ru/CSAFront/index.do> https://passport.yandex.ucoz.ru/auth?origin=home_desktop_ru

Признаки фишинга:

Доменное имя не соответствует официальному имени компании: Официальное доменное имя Сбербанка: sberbank.ru Официальное доменное имя Яндекса: yandex.ru Адрес электронной почты отправителя не связан с компанией. Наличие грамматических ошибок или опечаток в тексте ссылки.

3.3.2

☒ Да
☐ Нет

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл** из 1

Рис. 1.33: .

Да, фишинговый имейл может прийти от знакомого адреса.

Это происходит, когда злоумышленники взламывают учетные записи электронной почты проверенных пользователей и используют их для отправки фишинговых писем их контактам. Получатели могут ошибочно принять эти пись-

ма за законные, поскольку они исходят от знакомого адреса, и оказаться более восприимчивыми к фишингу.

3.4.1

A quiz interface with a light green background. It contains four radio button options: 'протокол для отправки имейлов', 'подмена адреса отправителя в имейлах' (selected), 'атака перебором паролей', and 'метод предотвращения фишинга'. Below the options are two buttons: 'Следующий шаг' (green) and 'Решить снова' (white). At the bottom, it says 'Ваши решения' with a link, followed by 'Вы получили: 1 балл из 1'.

- ☐ протокол для отправки имейлов
- ☒ подмена адреса отправителя в имейлах
- ☐ атака перебором паролей
- ☐ метод предотвращения фишинга

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.34: .

Email Спуфинг – это подмена адреса отправителя в имейлах поясни

3.4.2

A quiz interface with a light green background. It contains four radio button options: 'обязательно шифрует данные и требует ключ дешифрования', 'маскируется под легитимную программу' (selected), 'работает исключительно под ОС Windows', and 'разработан греками'. Below the options are two buttons: 'Следующий шаг' (green) and 'Решить снова' (white). At the bottom, it says 'Ваши решения' with a link, followed by 'Вы получили: 1 балл из 1'.

- ☐ обязательно шифрует данные и требует ключ дешифрования
- ☒ маскируется под легитимную программу
- ☐ работает исключительно под ОС Windows
- ☐ разработан греками

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

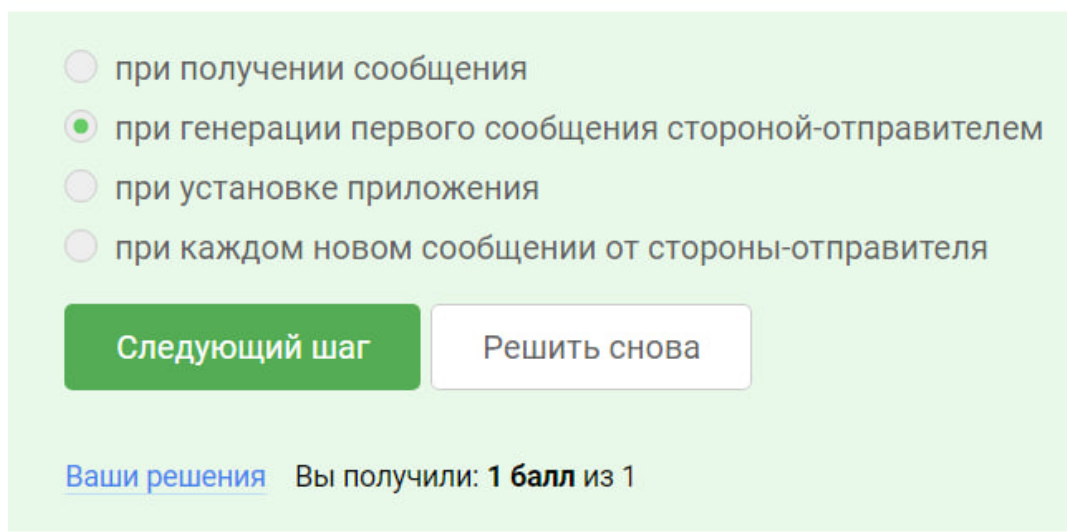
Рис. 1.35: .

Вирусы-трояны маскируются под легитимные программы, потому что это по-

могает им:

Избежать обнаружения: Легитимные программы обычно считаются безопасными, поэтому антивирусное программное обеспечение с меньшей вероятностью будет их сканировать. **Получить доверие пользователей:** Когда пользователи устанавливают легитимные программы, они доверяют им и с большей вероятностью предоставят им необходимые разрешения для работы. **Увеличить распространение:** Пользователи с большей вероятностью откроют или установят легитимные программы, что увеличивает шансы трояна на заражение компьютера. **Уклоняться от мер безопасности:** Некоторые вирусы-трояны маскируются под обновления безопасности или другие необходимые программы, что обманывает пользователей, заставляя их отключать меры безопасности и устанавливать троян. **Устанавливать долгосрочное присутствие:** Маскируясь под легитимную программу, троян может оставаться незамеченным и активным в течение более длительного периода времени.

3.5.1



☐ при получении сообщения

☒ при генерации первого сообщения стороной-отправителем

☐ при установке приложения

☐ при каждом новом сообщении от стороны-отправителя

Следующий шаг Решить снова

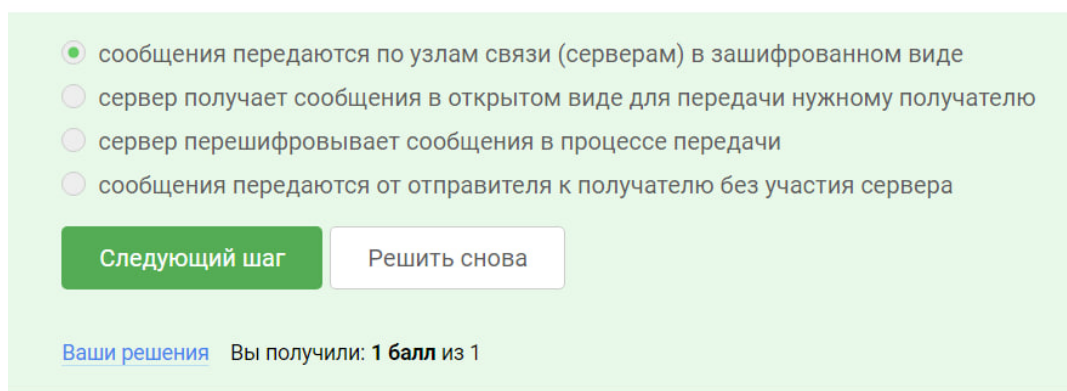
[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.36: .

Дело в том, что ключи шифрования в Signal формируются сторонами еще до отправки первого сообщения в процессе обмена открытыми ключами Diffie-Hellman. Каждый пользователь генерирует свою пару закрытого/открытого

ключей и обменивается открытым ключом с другим пользователем, используя защищенный канал. Используя этот открытый ключ и свои собственные закрытые ключи, пользователи производят общий секретный ключ, используемый для шифрования сообщений. Таким образом, ключ шифрования фактически формируется до отправки первого сообщения.

3.5.2



☒ сообщения передаются по узлам связи (серверам) в зашифрованном виде

☐ сервер получает сообщения в открытом виде для передачи нужному получателю

☐ сервер перешифровывает сообщения в процессе передачи

☐ сообщения передаются от отправителя к получателю без участия сервера

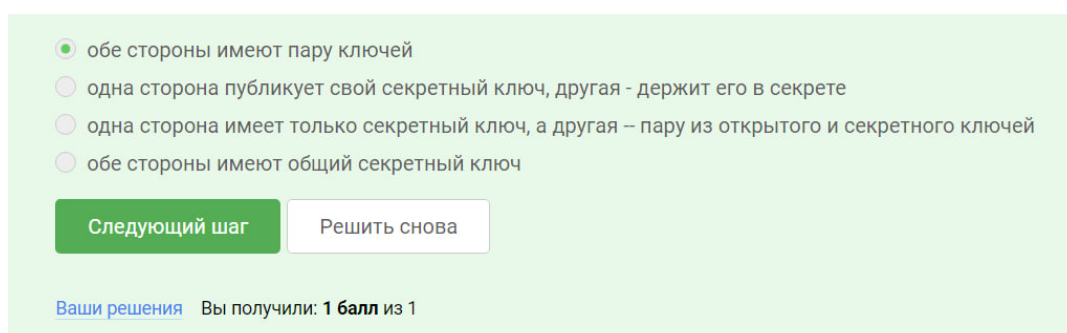
[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.37: .

Суть сквозного шифрования состоит в том, что сообщения передаются по узлам связи (серверам) в зашифрованном виде, чтобы их содержимое могли прочитать только отправитель и получатель, а не кто-либо еще, включая сами серверы.

4.1.1



☒ обе стороны имеют пару ключей

☐ одна сторона публикует свой секретный ключ, другая - держит его в секрете

☐ одна сторона имеет только секретный ключ, а другая – пару из открытого и секретного ключей

☐ обе стороны имеют общий секретный ключ

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.38: .

В асимметричных криптографических примитивах обе стороны действитель-

но имеют пару ключей по следующим причинам:

Разделение открытого и закрытого ключей: Асимметричная криптография основана на разделении открытого и закрытого ключей. Открытый ключ известен всем, а закрытый ключ хранится в секрете владельцем.

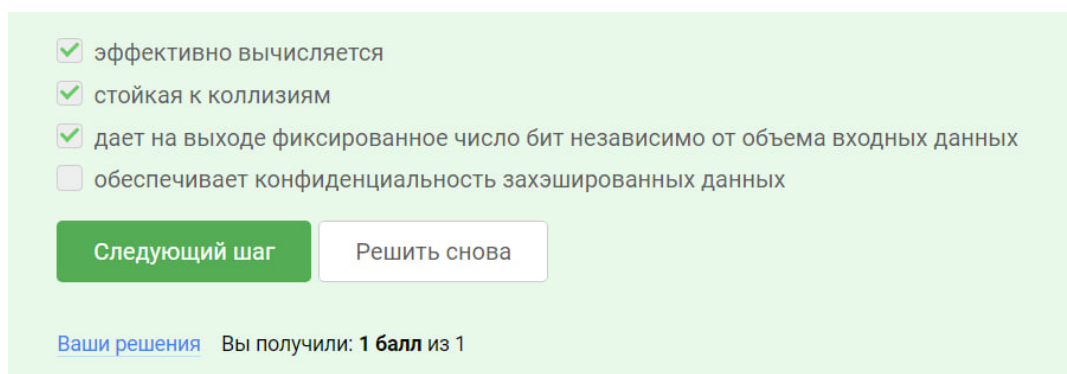
Шифрование и расшифровка: Открытый ключ используется для шифрования сообщений, которые может расшифровать только владелец соответствующего закрытого ключа. Это обеспечивает безопасность передачи сообщений через незащищенный канал.

Цифровые подписи: Закрытый ключ используется для создания цифровых подписей, которые подтверждают подлинность и целостность сообщений. Открытый ключ может использоваться для проверки этих подписей.

Удобство: Использование пары ключей удобно, поскольку устраняет необходимость обмена секретными ключами между сторонами.

Защита от компрометации: Если один из ключей пары будет скомпрометирован, другой ключ остается безопасным, обеспечивая дополнительный уровень защиты.

4.1.2



☒ эффективно вычисляется

☒ стойкая к коллизиям

☒ дает на выходе фиксированное число бит независимо от объема входных данных

☐ обеспечивает конфиденциальность зашифрованных данных

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

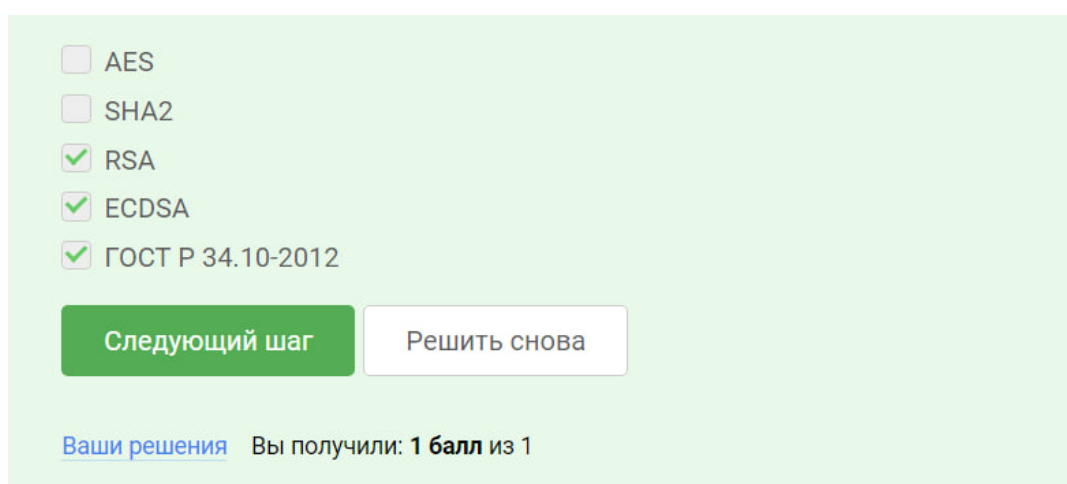
Рис. 1.39: .

Эффективно вычисляется: Криптографические хэш-функции разработаны так, чтобы их можно было быстро вычислить, что делает их практичными для реального применения.

Стойкая к коллизиям: Криптографические хэш-функции предназначены для защиты от коллизий (ситуаций, когда разные входные данные дают одинаковый хэш-результат), что делает их безопасными для использования в приложениях, где важна целостность данных.

Дает на выходе фиксированное число бит независимо от объема входных данных: Криптографические хэш-функции производят хэш-значения фиксированного размера независимо от размера исходных данных. Это упрощает их использование для сравнения и проверки целостности данных.

4.1.3



The screenshot shows a quiz interface with a light green background. On the left, there is a list of five options, each with a checkbox: AES, SHA2, RSA, ECDSA, and ГОСТ Р 34.10-2012. The checkboxes for RSA, ECDSA, and ГОСТ Р 34.10-2012 are checked and marked with a green checkmark. Below the list, there are two buttons: a green button labeled 'Следующий шаг' (Next step) and a white button with a grey border labeled 'Решить снова' (Solve again). At the bottom left, there is a link 'Ваши решения' (Your solutions) in blue. At the bottom right, it says 'Вы получили: 1 балл из 1' (You received: 1 point out of 1).

Рис. 1.40: .

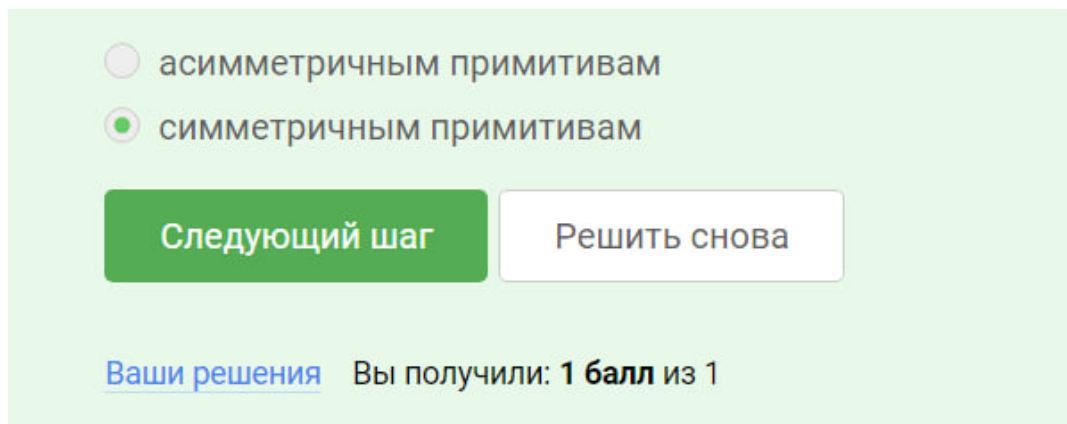
Потому что RSA, ECDSA и ГОСТ Р 34.10-2012 являются алгоритмами цифровой подписи.

Алгоритм цифровой подписи — это криптографический алгоритм, используемый для создания электронной подписи, которая обеспечивает целостность и подлинность цифровых данных.

RSA (Алгоритм Ривеста — Шамира — Адельмана) — один из первых и наиболее широко используемых алгоритмов цифровой подписи. ECDSA (Алгоритм цифровой подписи с эллиптическими кривыми) — более современный и эффективный алгоритм цифровой подписи, основанный на криптографии эллиптических кривых. ГОСТ Р 34.10-2012 — российский стандарт алгоритма цифровой

подписи, основанный на ГОСТ 28147-89.

4.1.4



☐ асимметричным примитивам

☒ симметричным примитивам

Следующий шаг Решить снова

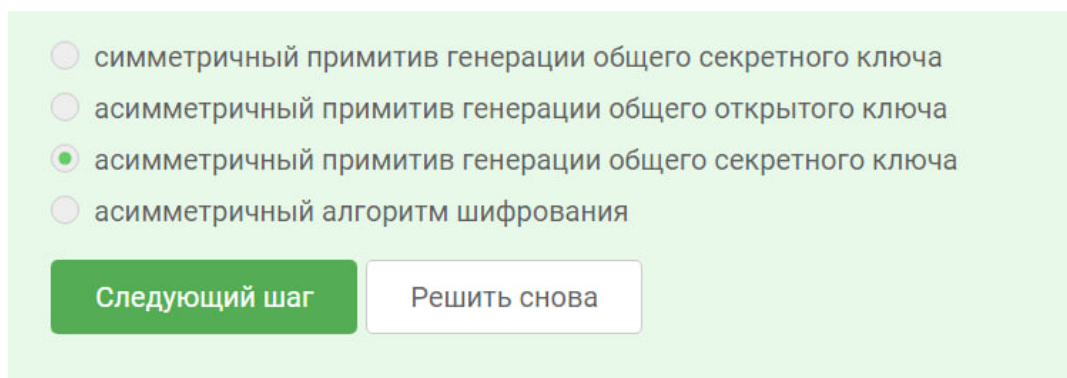
[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.41: .

Код аутентификации сообщения относится к симметричным примитивам, потому что он использует один и тот же секретный ключ как для отправителя, так и для получателя для расчета и проверки кода аутентификации сообщения.

Симметричные примитивы используют один и тот же ключ для шифрования и дешифрования, а также для расчета кодов аутентификации сообщений. Это в отличие от асимметричных примитивов, которые используют разные ключи для шифрования и дешифрования, а также для цифровой подписи и проверки.

4.1.5



☐ симметричный примитив генерации общего секретного ключа

☐ асимметричный примитив генерации общего открытого ключа

☒ асимметричный примитив генерации общего секретного ключа

☐ асимметричный алгоритм шифрования

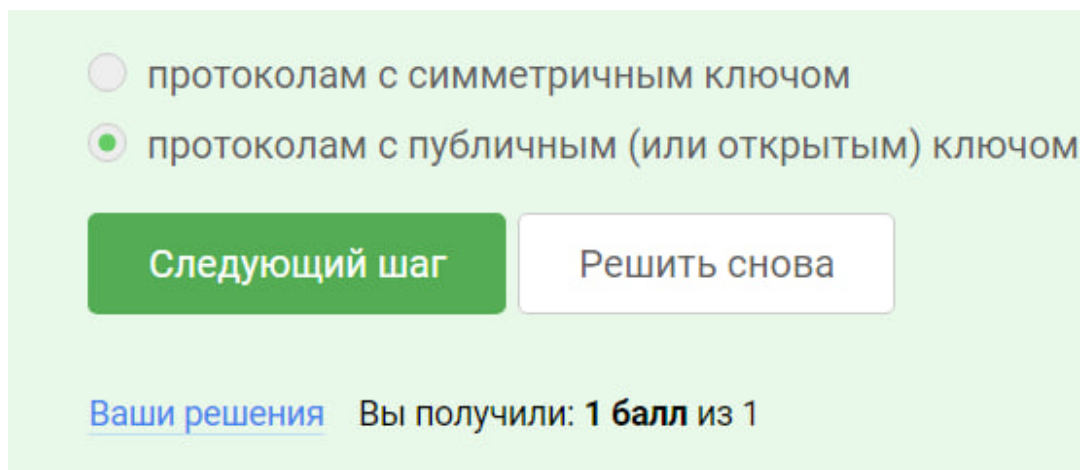
Следующий шаг Решить снова

Рис. 1.42: .

Потому что в обмене ключами Диффи-Хэллмана используются пара откры-

того и закрытого ключей, причем открытые ключи публикуются, а закрытые ключи хранятся в секрете. Это характерно для асимметричных криптосистем, в отличие от симметричных, которые используют один и тот же ключ для шифрования и дешифрования.

4.2.1



☐ протоколам с симметричным ключом

☒ протоколам с публичным (или открытым) ключом

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.43: .

Протокол электронной цифровой подписи относится к протоколам с публичным (или открытым) ключом, потому что он использует асимметричную криптографию, в которой для подписи документа используется один (частный) ключ, а для проверки подписи — другой (публичный) ключ.

В отличие от протоколов с симметричным ключом, где один и тот же ключ используется для шифрования и расшифровки, в протоколах с открытым ключом частный ключ хранится в секрете, а публичный ключ может быть общедоступным. Это позволяет подписывать документы таким образом, что их могут проверить другие стороны без необходимости обмениваться секретными ключами.

4.2.2

☒ подпись, открытый ключ, сообщение
☐ подпись, секретный ключ, сообщение
☐ подпись, секретный ключ
☐ подпись, открытый ключ

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.44: .

Для проверки электронной цифровой подписи (ЭЦП) требуется на вход подпись, открытый ключ и сообщение по следующим причинам:

Подпись: Подпись необходима для проверки ее достоверности и подтверждения подлинности подписавшего.

Открытый ключ: Открытый ключ используется для дешифрования подписи и извлечения хеша сообщения. Он обеспечивает возможность проверки того, что именно владелец закрытого ключа, соответствующего открытому ключу, подписал сообщение.

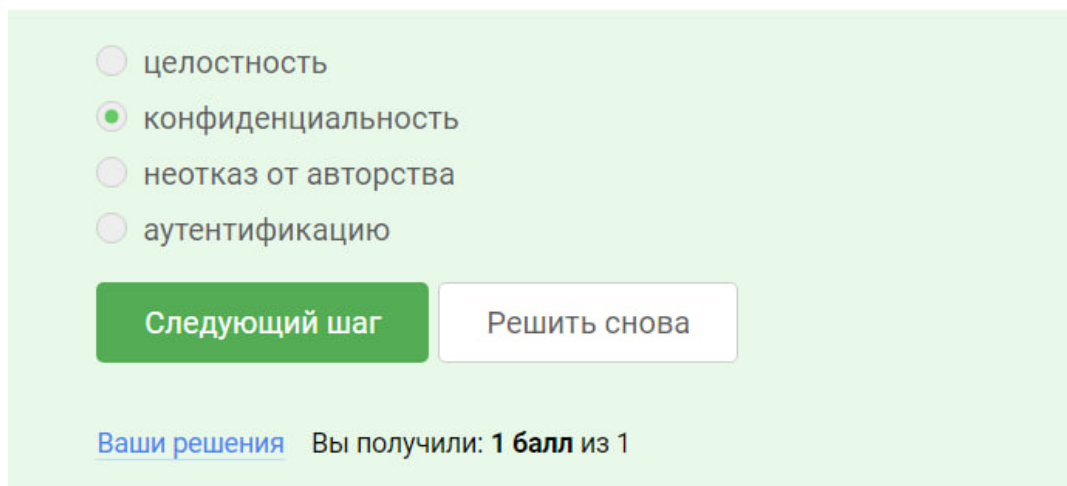
Сообщение: Сообщение необходимо для вычисления его хеш-функции. Хеш-функция используется для проверки целостности сообщения и подтверждения того, что не было никаких изменений после подписания.

Алгоритм верификации ЭЦП работает следующим образом:

1. Вычисляется хеш-функция сообщения.
2. Хеш используется для дешифрования подписи с использованием открытого ключа.
3. Результат дешифрования сравнивается с вычисленным хешем сообщения.

4. Если два хеша совпадают, значит, подпись является действительной и сообщение не было изменено.

4.2.3



☐ целостность

☒ конфиденциальность

☐ неотказ от авторства

☐ аутентификацию

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.45: .

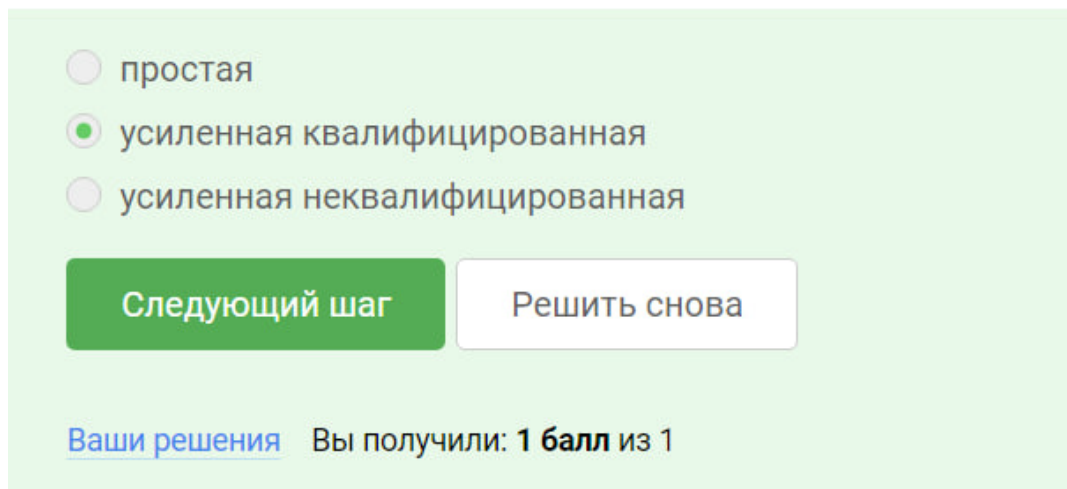
Электронная цифровая подпись (ЭЦП) не обеспечивает конфиденциальность, поскольку она не шифрует содержимое сообщения. ЭЦП предназначена для проверки подлинности и целостности сообщения, а не для его защиты от несанкционированного доступа. Шифрование является отдельным механизмом, который необходимо использовать для обеспечения конфиденциальности. Вот почему:

ЭЦП используется для аутентификации отправителя. Она подтверждает, что сообщение было отправлено определенным лицом или организацией. Она не скрывает содержимое сообщения. ЭЦП защищает от изменений. Она гарантирует, что сообщение не было изменено после его подписи. Однако она не защищает его от чтения другими лицами. Шифрование необходимо для конфиденциальности. Шифрование преобразует сообщение в нечитаемый формат, доступный только тем, кто обладает ключом или паролем.

Таким образом, ЭЦП и шифрование служат разным целям, и оба механизма необходимы для обеспечения как подлинности, так и конфиденциальности со-

общений.

4.2.4



☐ простая

☒ усиленная квалифицированная

☐ усиленная неквалифицированная

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.46: .

Усиленная квалифицированная электронная подпись (КЭП) необходима для отправки налоговой отчетности в ФНС, потому что:

Согласно статье 6 Федерального закона № 63-ФЗ “Об электронной подписи”, при представлении налоговой отчетности в электронной форме используются усиленные квалифицированные электронные подписи руководителя или уполномоченного лица. КЭП соответствует самым высоким требованиям безопасности и подтверждена аккредитованным удостоверяющим центром. Она позволяет однозначно идентифицировать отправителя налоговой отчетности и обеспечивает юридическую значимость передаваемых сведений.

4.2.5

☐ в любой организации, имеющей соответствующую лицензию ФСБ

☐ в минкомсвязи РФ

☒ в удостоверяющем (сертификационном) центре

☐ в любой организации по месту работы

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.47: .

Удостоверяющий (сертификационный) центр — это организация, которая выдает квалифицированные сертификаты ключа проверки электронной подписи. Эти сертификаты подтверждают, что электронная подпись принадлежит определенному лицу или организации и что она прошла проверку на соответствие установленным стандартам.

4.3.1

☐ BitCoin

☒ MasterCard

☐ SecurePay

☐ POS-терминал

☐ банкомат

☒ МИР

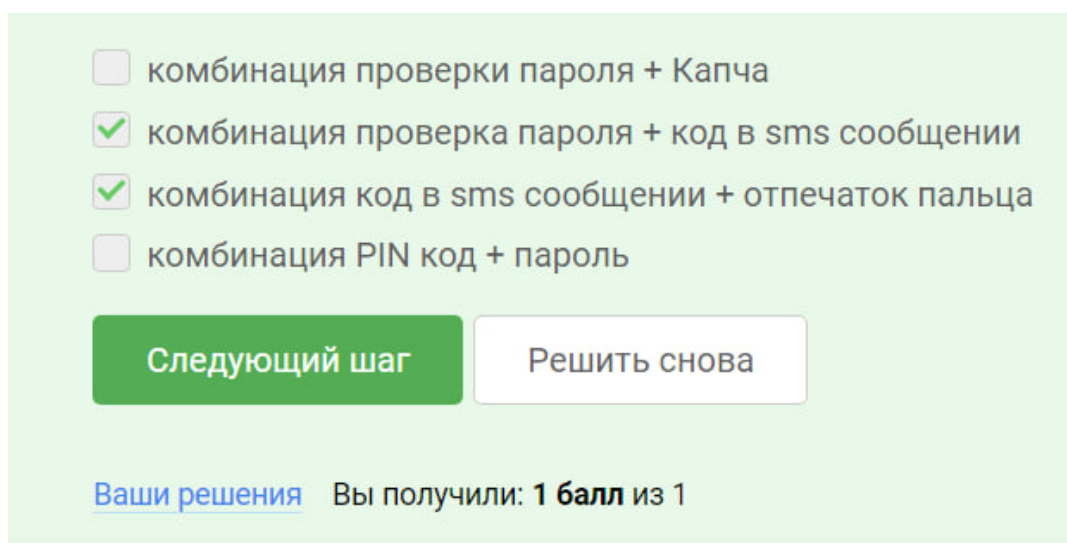
[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.48: .

потому что они являются платежными системами

4.3.2



☐ комбинация проверки пароля + Капча

☒ комбинация проверка пароля + код в sms сообщении

☒ комбинация код в sms сообщении + отпечаток пальца

☐ комбинация PIN код + пароль

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

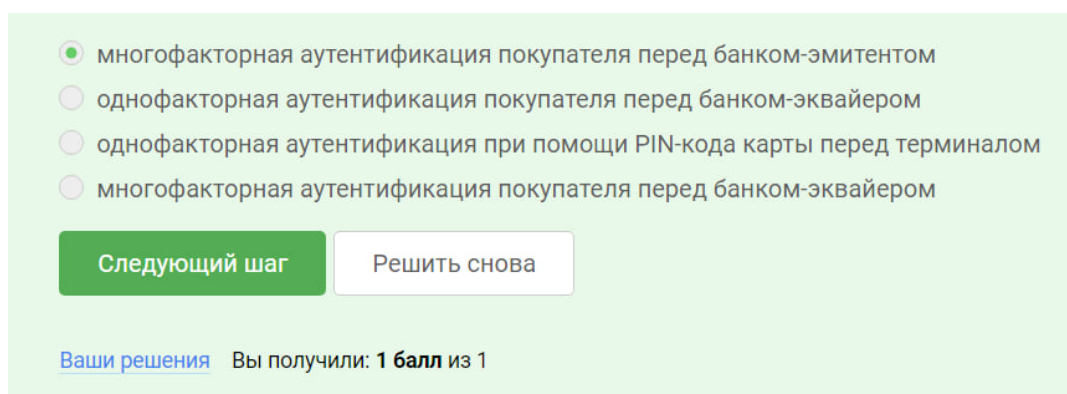
Рис. 1.49: .

Комбинация проверки пароля + код в sms сообщении является примером двухфакторной аутентификации, так как она использует два разных фактора: что-то, что вы знаете (пароль) и что-то, что у вас есть (телефон с SIM-картой, номер которой привязан к аккаунту).

Комбинация код в sms сообщении + отпечаток пальца также является примером двухфакторной аутентификации, поскольку она использует два разных фактора: что-то, что у вас есть (телефон с SIM-картой) и что-то, что является уникальным для вас (отпечаток пальца).

Многофакторная аутентификация предполагает использование трех или более различных факторов аутентификации, поэтому ни один из приведенных примеров не является примером многофакторной аутентификации.

4.3.3



☒ многофакторная аутентификация покупателя перед банком-эмитентом

☐ однофакторная аутентификация покупателя перед банком-эквайером

☐ однофакторная аутентификация при помощи PIN-кода карты перед терминалом

☐ многофакторная аутентификация покупателя перед банком-эквайером

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.50: .

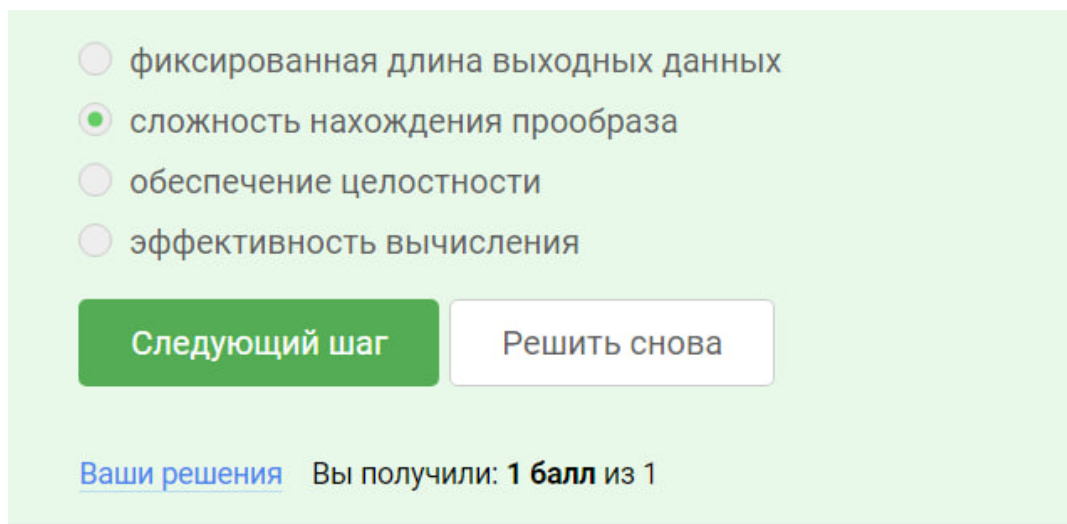
Многофакторная аутентификация используется при онлайн-платежах перед банком-эмитентом по нескольким причинам:

1. **Повышенная безопасность:** Многофакторная аутентификация добавляет дополнительный уровень защиты, требуя от пользователей предоставлять более одного способа подтверждения своей личности. Это затрудняет злоумышленникам доступ к учетным записям и совершение несанкционированных транзакций.
2. **Соблюдение нормативных требований:** Во многих странах действуют нормативные требования, которые предписывают банкам использовать многофакторную аутентификацию для защиты транзакций онлайн-банкинга. Это помогает банкам соответствовать этим требованиям и снижает риск мошенничества и штрафов.
3. **Защита от мошенничества:** Многофакторная аутентификация помогает предотвращать мошенничество, требуя от злоумышленников не только логин и пароль, но и другие факторы, такие как код подтверждения, отправленный по SMS или электронной почте, или биометрические данные.
4. **Повышение доверия клиентов:** Когда клиенты знают, что их онлайн-транзакции защищены многофакторной аутентификацией, они чувству-

ют себя более уверенно, совершая покупки в Интернете. Это повышает доверие клиентов и лояльность к банкам и поставщикам услуг.

5. Снижение потерь от мошенничества: Многофакторная аутентификация значительно снижает количество успешных мошеннических атак, что приводит к снижению потерь от мошенничества для банков и продавцов.

4.4.1



Quiz interface with four radio button options:

- ☐ фиксированная длина выходных данных
- ☒ сложность нахождения прообраза
- ☐ обеспечение целостности
- ☐ эффективность вычисления

Buttons:

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.51: .

Свойство криптографической хэш-функции, которое используется в доказательстве работы, - это сложность нахождения прообраза.

4.4.2

☒ открытость

☒ постоянства

☒ консенсус

☒ живучесть

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.52: .

Открытость: Любой может просмотреть и проверить блокчейн, чтобы убедиться в его достоверности. Постоянство: После того, как данные записаны в блокчейн, их невозможно изменить или удалить, обеспечивая постоянство записанных транзакций. Консенсус: Все участники сети должны согласиться с текущим состоянием блокчейна, прежде чем новые блоки могут быть добавлены. Живучесть: Блокчейны обычно распределены по множеству узлов, что делает их устойчивыми к сбоям отдельных узлов.

4.4.3

☐ обмен ключами

☐ шифрование

☒ цифровая подпись

☐ хэш-функция

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл** из 1

Рис. 1.53: .

потому что

Список литературы

1. Федеральный закон “О коммерческой тайне” от 29.07.2004 N 98-ФЗ [Электронный ресурс]. — Режим доступа: <http://www.consultant.ru/cons/cgi/online.cgi?req=doc&d=134&dst=1000000001,0&rnd=0.664452099339919#07903293431455025>
2. Северин В.А. Коммерческая тайна в России [Электронный ресурс]. — Режим доступа: <https://istina.msu.ru/media/publications/book/fee/4d5/2717850/Kniga.pdf>
3. Закон РСФСР от 25.12.1990 N 445-1 (ред. от 30.11.1994) «О предприятиях и предпринимательской деятельности» [Электронный ресурс]. — Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_40/
4. Закон РСФСР от 22.03.1991 N 948-1 (ред. от 26.07.2006) «О конкуренции и ограничении монополистической деятельности на товарных рынках» [Электронный ресурс]. — Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_40/
5. Указ Президента РФ от 06.03.1997 N 188 (ред. от 13.07.2015) “Об утверждении Перечня сведений конфиденциального характера» [Электронный ресурс]. — Режим доступа: <http://pravo.gov.ru/proxy/ips/?docbody=&nd=102046005::: {#refs} :::>