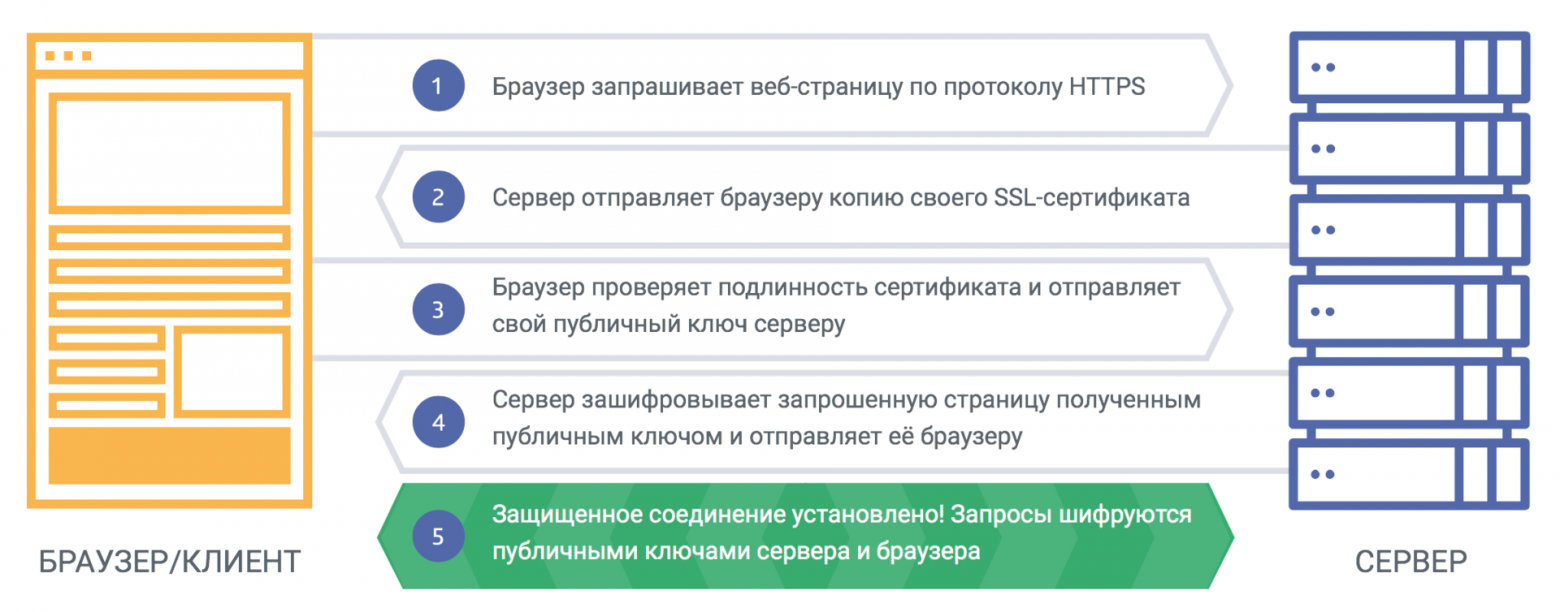
SSL (secure sockets layer — уровень защищённых cокетов) представляет собой криптографический протокол для безопасной связи. С версии 3.0 SSL заменили на TLS (transport layer security — безопасность транспортного уровня), но название предыдущей версии прижилось, поэтому сегодня под SSL чаще всего подразумевают TLS.  
  
Цель протокола — обеспечить защищенную передачу данных. При этом для аутентификации используются асимметричные алгоритмы шифрования (пара открытый — закрытый ключ), а для сохранения конфиденциальности — симметричные (секретный ключ). Первый тип шифрования более ресурсоемкий, поэтому его комбинация с симметричным алгоритмом помогает сохранить высокую скорость обработки данных.

## Рукопожатие

Когда пользователь заходит на веб-сайт, браузер запрашивает информацию о сертификате у сервера, который высылает копию SSL-сертификата с открытым ключом. Далее, браузер проверяет сертификат, название которого должно совпадать с именем веб-сайта.  
  
Кроме того, проверяется дата действия сертификата и наличие корневого сертификата, выданного надежным центром сертификации. Если браузер доверяет сертификату, то он генерирует предварительный секрет (pre-master secret) сессии на основе открытого ключа, используя максимально высокий уровень шифрования, который поддерживают обе стороны.  
  
  
  
Сервер расшифровывает предварительный секрет с помощью своего закрытого ключа, соглашается продолжить коммуникацию и создать общий секрет (master secret), используя определенный вид шифрования. Теперь обе стороны используют симметричный ключ, который действителен только для данной сессии. После ее завершения ключ уничтожается, а при следующем посещении сайта процесс рукопожатия запускается сначала.

## Алгоритмы шифрования

Для симметричного шифрования использовались разные алгоритмы. Первым был блочный [шифр](https://www.vocal.com/cryptography/data-encryption-standard-des/) DES, разработанный компанией IBM. В США его утвердили в качестве стандарта в 70-х годах. В основе алгоритма лежит [сеть Фейстеля](https://en.wikipedia.org/wiki/Feistel_cipher) с 16-ю циклами. Длина ключа составляет 56 бит, а блока данных — 64.  
  
Развитием DES является алгоритм 3DES. Он [создавался](https://www.vocal.com/cryptography/tdes/) с целью совершенствования короткого ключа в алгоритме-прародителе. Размер ключа и количество циклов шифрования увеличилось в три раза, что снизило скорость работы, но повысило надежность.  
  
Еще был блочный [шифр](http://www.isiloniq.com/emc-plus/rsa-labs/standards-initiatives/rc2.htm) RC2 с переменной длиной ключа, который работал быстрее DES, а его 128-битный ключ был сопоставим с 3DES по надежности. Потоковый [шифр](https://poland.emc.com/emc-plus/rsa-labs/standards-initiatives/rc4.htm) RC4 был намного быстрее блочных и строился на основе генератора псевдослучайных битов. Но сегодня все эти алгоритмы [считаются](https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/) небезопасными или устаревшими.  
  
Самым современным признан [стандарт](https://www.vocal.com/cryptography/advanced-encryption-standard-aes/) AES, который официально заменил DES в 2002 году. Он основан на блочном алгоритме Rijndael и скорость его работы в 6 раз выше по сравнению с 3DES. Размер блока здесь равен 128 битам, а размер ключа — 128/192/256 битам, а количество раундов шифрования зависит от размера ключа и может составлять 10/12/14 соответственно.  
  
Что касается асимметричного шифрования, то оно чаще всего строится на базе таких алгоритмов, как RSA, DSA или ECC. RSA (назван в честь авторов Rivest, Shamir и Adleman) используется и для шифрования, и для цифровой подписи. Алгоритм основан на сложности факторизации больших чисел и поддерживает все типы SSL-сертификатов.  
  
DSA (Digital Signature Algorithm) используется только для создания цифровой подписи и основан на вычислительной сложности взятия логарифмов в конечных полях. По безопасности и производительности полностью сопоставим с RSA.  
  
ECC (Elliptic Curve Cryptography) определяет пару ключей с помощью точек на кривой и используется только для цифровой подписи. Основным преимуществом алгоритма является более высокий уровень надежности при меньшей длине ключа (256-битный ECC-ключ [сопоставим](https://www.ssl247.com/kb/ssl-certificates/generalinformation/what-is-rsa-dsa-ecc) по надежности с 3072-битным RSA-ключом.  
  
Более короткий ключ также влияет на время обработки данных, которое заметно сокращается. Этот факт и то, что алгоритм эффективно обрабатывает большое количество подключений, сделали его удобным инструментом для работы с мобильной связью. В SSL-сертификатах можно использовать сразу несколько методов шифрования для большей защиты.

## Хеш и MAC

Цель хеш-алгоритма — [преобразовывать](https://www.symantec.com/connect/blogs/how-manage-sha-1-deprecation-ssl-encryption) все содержимое SSL-сертификата в битовую строку фиксированной длины. Для шифрования значения хеша применяется закрытый ключ центра сертификации, который включается в сертификат как подпись.  
  
Хеш-алгоритм также [использует](https://technet.microsoft.com/en-us/library/cc783349(v=ws.10).aspx) величину, необходимую для проверки целостности передаваемых данных — MAC (message authentication code). MAC использует функцию отображения, чтобы представлять данные сообщения как фиксированное значение длины, а затем хеширует сообщение.  
  
В протоколе TLS применяется HMAC (hashed message authentication code), который использует хеш-алгоритм сразу с общим секретным ключом. Здесь ключ прикрепляется к данным, и для подтверждения их подлинности обе стороны должны использовать одинаковые секретные ключи, что обеспечивает большую безопасность.  
  
Все алгоритмы шифрования сегодня поддерживают алгоритм хеширования SHA2, чаще всего именно SHA-256. SHA-512 имеет [похожую](https://en.wikipedia.org/wiki/SHA-2) структуру, но в нем длина слова равна 64 бита (вместо 32), количество раундов в цикле равно 80 (вместо 64), а сообщение разбивается на блоки по 1024 бита (вместо 512 бит). Раньше для тех же целей применялся алгоритм SHA1 и MD5, но сегодня они считаются [уязвимыми](https://support.comodo.com/index.php?/comodo/Knowledgebase/Article/View/973/102/important-change-announcement---deprecation-of-sha-1).  
  
Разговоры об отказе от SHA1 велись достаточно давно, но в конце февраля алгоритм был официально [взломан](https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html). Исследователям удалось добиться коллизии хешей, то есть одинакового хеша для двух разных файлов, что доказало небезопасность использования алгоритма для цифровых подписей. Первая попытка была сделана еще в 2015, хотя тогда удалось подобрать только те сообщения, хеш которых совпадал. Сегодня же речь идет о целых документах.

## Сертификаты бывают разные

Теперь, когда мы разобрались, что представляет собой протокол SSL/TLS и как происходит соединений на его основе, можно поговорить и о [видах](https://www.ssl.com/article/dv-ov-and-ev-certificates/) сертификатов.  
  
Domain Validation, или сертификаты с проверкой домена, подходят для некоммерческих сайтов, так как они подтверждают только веб-сервер, обслуживающий определенный сайт, на который был осуществлен переход. Этот вид сертификата самый дешевый и популярный, но не может считаться полностью безопасным, так как содержит только информацию о зарегистрированном доменном имени.  
  
Organization Validation, или сертификаты с проверкой организации, являются более надежными, так как подтверждают еще регистрационные данные компании-владельца. Эту информацию юридическое лицо обязано предоставить при покупке сертификата, а удостоверяющий центр может связаться напрямую с компанией для подтверждения этой информации. Сертификат отвечает стандартам [RFC](https://ru.wikipedia.org/wiki/RFC) и содержит информацию о том, кто его подтвердил, но данные о владельце не отображаются.  
  
Extended Validation, или сертификат с расширенной проверкой, считается самым надежным. Собственно, зеленый замочек или ярлык в браузере означает как раз то, что у сайта есть именно такой сертификат. О том, как разные браузеры информируют пользователей о наличии сертификата или возникающих ошибках можно почитать [тут](https://1cloud.ru/blog/otobrazhenie-https-v-raznyh-brauzerah).  
  
Он нужен веб-сайтам, которые проводят финансовые транзакции и требуют высокий уровень конфиденциальности. Однако многие сайты предпочитают перенаправлять пользователей для совершения платежей на внешние ресурсы, подтвержденные сертификатами с расширенной проверкой, при этом используя сертификаты OV, которых вполне хватает для защиты остальных данных пользователей.  
  
Кроме того, сертификаты могут различаться в зависимости от количества доменов, на которые они были выданы. Однодоменные сертификаты (Single Certificate) привязываются к одному домену, который указывается при покупке. [Мультидоменные](https://1cloud.ru/services/ssl/san) сертификаты (типа Subject Alternative Name, Unified Communications Certificate, Multi Domain Certificate) будут действовать уже для большего числа доменных имен и серверов, которые также определяются при заказе. Однако за включение дополнительных доменов, свыше определенной нормы, потребуется платить отдельно.  
  
Еще существуют [поддоменные](https://1cloud.ru/services/ssl/wc) сертификаты (типа WildCard), которые охватывают все поддомены указанного при регистрации доменного имени. Иногда могут потребоваться сертификаты, которые будут одновременно включать не только несколько доменов, но и поддомены. В таких случаях можно приобрести сертификаты типа [Comodo PositiveSSL Multi-Domain Wildcard](https://1cloud.ru/services/ssl/order/27) и [Comodo Multi-Domain Wildcard SSL](https://1cloud.ru/services/ssl/order/46) или (лайфхак) обычный мультидоменный сертификат, где в списке доменов указать также и нужные поддоменные имена.  
  
Получить SSL-сертификат можно и самостоятельно: пара ключей для этого генерируется через любой генератор, например, бесплатный [OpenSSL](https://www.openssl.org/). И такой защищенный канал связи вполне получится использовать для внутренних целей: между устройствами своей сети или приложениями. Но вот для использования на веб-сайте сертификат необходимо приобретать официально, чтобы в цепочке подтверждения сертификатов обязательно имелся корневой сертификат, браузеры не показывали сообщений о небезопасном соединении, а пользователи были спокойны за свои данные.