1. **Web security**……………………………………………………………………………….
   1. **Цифровые сертификаты**……………………………………………………………
   2. **CORS**…………………………………………………………………………………..
   3. **Цифровые сертификаты**

[Цифровой сертификат](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D1%81%D0%B5%D1%80%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82) — файл, который уникальным образом идентифицирует серверы, короче говоря – цифровая подпись сайта, подтверждающая его подлинность. Сертификат используется для шифрования данных и идентификации сайта при установлении защищённого соединения HTTPS. Цифровой сертификат подписывается и заверяется центрами сертификации или их партнёрами. Работает этот сертификат поверх SSL (Secure Socket Layer) — протокол шифрования данных, которыми обмениваются клиент и сервер. Безопасный обмен обеспечивается за счёт шифрования и аутентификации цифрового сертификата. SSL-сертификат даёт возможность владельцу применить к своему сайту технологию SSL-шифрования.

Таким образом, назначение SSL-сертификата — обеспечить безопасное соединение между сервером и браузером пользователя, надёжно защитить данные от перехвата и подмены. Информация передаётся в зашифрованном виде, и расшифровать её можно только с помощью специального ключа, являющегося частью сертификата. Тем самым гарантируется сохранность данных. Посетители сайта вправе ожидать, что защита их информации, если она важная, будет обеспечена с помощью SSL-сертификата. Они могут покинуть ваш сайт, если видят, что он не защищён. Если сайт имеет SSL-сертификат, то в строке состояния браузера отображается значок в виде замка.

**Как работает HTTPS (схема действия SSL):**



— Пользователь заходит на защищённый сайт;  
— Выполняется проверка DNS и определение IP-адреса хоста веб-сайта;  
— Запись веб-сайта найдена, переход на веб-сервер хоста;  
— Запрос безопасного SSL-соединения с хоста веб-сайта;  
— Хост отвечает валидным SSL-сертификатом;  
— Устанавливается защищённое соединение, передаваемые данные шифруются.

**Защита для бизнеса и клиентов**

• принимаемые и отправляемые данные при передаче шифруются, применяется процедура аутентификации;

• снижение вероятности различных атак, например, значительное затруднение использование фишинговых систем;

• стандартный http в адресной строке заменячется на https (говорит о том, что соединение между клиентом и сервером использует SSL); адресная строка имеет зеленый фон, если SSL является EV (extended validation); если нет фона или замочка, то значит соединение не использует SSL.

**SEO**

 Cайты, подтверждённые сертификатами, занимают более высокие позиции в результатах выдачи поисковых систем по сравнению с конкурентами без SSL.

**Удостоверяющие центры (УЦ)**

• GlobalSign (партнёр: REG.RU)

• Symantec

• GoDaddy

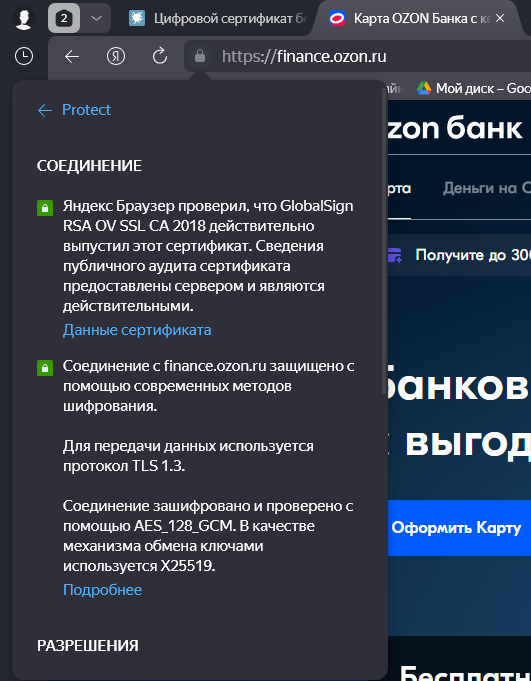
• Comodo

**Где купить SSL-сертификат?**

Приобретают сертификаты обычно не напрямую у удостоверяющего центра, а через партнёров.

**Как пользователю посмотреть информацию о сертификате**

Чтобы увидеть детальную информацию о SSL-сертификате, пользователю достаточно щёлкнуть мышью на значке замка и выбрать в меню «Просмотр сертификата». Браузеры могут различаться, но сертификат всегда содержит одну и ту же информацию.



**Уровни проверки сертификатов**

1. Domain Validation (DV) - Сертификат с проверкой доменов — самый низкий и недорогой уровень. Сертификат подойдёт для защиты персональных данных пользователей.
2. OV (organization validation) – сертификат удостоверяет не только доменное имя, но и то что сайт принадлжеит существующей организации, права на доменное имя и т.д.
3. Extended Validation – сертификат с расширенной проверкой. При выпуске EV-сертификата проводится очень тщательная проверка организации.

**Современная версия**

**TLS** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *transport layer security* — Протокол защиты транспортного уровня[[1]](https://ru.wikipedia.org/wiki/TLS#cite_note-1)), как и его предшественник [SSL](https://ru.wikipedia.org/wiki/SSL) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *secure sockets layer* — слой защищённых сокетов), — [криптографические протоколы](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), обеспечивающие защищённую передачу данных между узлами в сети [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)[[2]](https://ru.wikipedia.org/wiki/TLS#cite_note-2). TLS и SSL используют [асимметричное шифрование](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) для аутентификации, [симметричное шифрование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) для конфиденциальности и [коды аутентичности сообщений](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%BE%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0) для сохранения целостности сообщений.

Данный протокол широко используется в приложениях, работающих с сетью [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), таких как [веб-браузеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), [работа с электронной почтой](https://ru.wikipedia.org/wiki/E-mail), [обмен мгновенными сообщениями](https://ru.wikipedia.org/wiki/IM) и [IP-телефония (VoIP)](https://ru.wikipedia.org/wiki/VoIP-%D1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0%BD).

**TLS 1.3** — версия протокола защиты транспортного уровня ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Transport Layer Security*)[[1]](https://ru.wikipedia.org/wiki/TLS_1.3#cite_note-1), являющаяся седьмой итерацией протокола [TLS](https://ru.wikipedia.org/wiki/TLS) и его предшественника [SSL](https://ru.wikipedia.org/wiki/SSL) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Secure Sockets Layer*). Протокол предназначен для защиты передаваемых данных между узлами сети, а именно предоставление шифрования, аутентификации и целостности соединения[[2]](https://ru.wikipedia.org/wiki/TLS_1.3#cite_note-2).

Версия 1.3 была утверждена в качестве стандарта 9 августа 2018 года [Инженерным советом Интернета](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%81%D0%BE%D0%B2%D0%B5%D1%82_%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B0) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Internet Engineering Task Force*, IETF)[[3]](https://ru.wikipedia.org/wiki/TLS_1.3#cite_note-3)[[4]](https://ru.wikipedia.org/wiki/TLS_1.3#cite_note-4).

* 1. **CORS**

CORS (Cross-origin resource sharing) – совместное использование ресурсов разных источников. **Механизм, который контролирует запросы от одного источника к другому, в целях безопасности.** Решает проблемы, например, утечки данных, от разных неправомерных запросов.

Приложение может запросить данные откуда-то. И вот то самое откуда-то является источником. А приложение, которое запрашивает данные, может располагаться на другом хосте и также являться источником. И возникает такая ситуация, что запросы могут осуществляться между разными источниками. Получается пара источников: источник приложения (источник запроса) и источник сервера.

Например: запущенные клиент и серверные части приложения: localhost:8080 и localhost:3000. Могут возникнуть CORS ошибки из-за того, что разные источники определятся по разным параметрам. Если в адресах запроса меняются какие-то из этих параметров – однозначно определяется разность источников.



Настроить CORS можно путем, например, настройки сервера: указать заголовки с доменными именами для различных методов HTTP-запросов. Своего рода – whitelist. Например, на все ответы POST запросов устанавливаются заголовки с разрешенными доменными именами. Также помимо разрешенных доменов могут указываться разрешенные заголовки.



**Засчёт CORS можно регулировать, с каких источников будут приниматься и обрабатываться запросы на сервере.**