**Принцип работы по HTTPS**

У сайта в IIS в настройках биндинга задан SSL сертификат. Клиент обращаясь к сайту по https протоколу получает публичный ключ и другие параметры этого SSL сертификата (домен, юр адрес, срок годности) и эти данные подписаны СА сертификатом доверенного центра.

Если клиент у себя на компьютере не находит данный СА сертификат или обнаруживает несоответствие доменного имени, обращение по ip, а не доменному имени, истекший сертификат и др, выдается сообщение о невозможности доверия данному сертификату. Но при согласии пользователя работа продолжается. Броузер генерирует сессионный ключ и шифрует его публичным и передает их серверу, который может расшифровать его с помощью приватного ключа. Дальнейший обмен данных шифруется с помощью сессионного ключа.

**Fibertest по HTTPS**

Fibertest может работать как по http, так и по https. Для работы *Fibertest* через *https* необходим сертификат. Если использовать IIS-совский self-signed, то возникает проблема, когда браузер предупреждает пользователей о невозможности доверия данному сертификату. Проблема также усугубляется необходимостью дважды подтверждать исключения - сперва для сайта (https://domain:443), а затем для *api* (<https://domain:11080>), причем вторая ошибка видна только в инструментах разработчика (требуется скопировать адрес в отдельную вкладку и тогда подтвердить исключение).

Для *IE* и *Chrome* данная проблема может быть решена добавлением *self-signed*  сертификата в *Trusted Root CA (Certification Authorities)*. Эти браузеры используют системную базу сертификатов, соответственно если добавить сертификат через *Windows* они найдут их и будут с ними работать ( возможно потребуется рестарт браузера).

*Firefox* имеет свою собственную базу данных с сертификатами. На момент написания статьи *Firefox(v.33.0.2)* всегда показывает предупреждения для self-signed сертификатов.

Более удобный и рабочий вариант это создание своего *CA*, добавление этого *CA* в *Trusted Root CA* на **всех клиентах** и на **IIS сервере**. Затем необходимо создать сертификат для  *IIS* сервера, подписать его созданным *CA* и установить для *api* и *cloud* веб сайтов. Все клиенты доверяющие *CA* будут доверять любому сертификату подписанному им. Удобный потому как единожды устанавливаем клиентам *CA*, а затем можно сколько угодно сертификатов создать.

**Инструмент для создания ключей/сертификатов**

1. Загружаем **openssl** <http://slproweb.com/products/Win64OpenSSL.html>

*Win64 OpenSSL v1.1.0L* (данная версия - последняя на данный момент)

1. Устанавливаем в папку “*c:/openssl-win64”*. При установке выбираем для *“Copy openssl DLLs to”* *“The OpenSSL binaries (/bin) directory”*

Запускаем коммандную строку из “*c:/openssl-win64/bin”*

1. **Создание *CA* сертификата**

a. Создаем приватный ключ для *iitCA*

**openssl genrsa -out iitCA2021.key 2048**

Внимание: ключ *приватный*, никому давать нельзя.

b. Из этого ключа создаем *CA* сертификат на 1024 дня.

**openssl req -x509 -new -nodes -key iitCA2021.key -days 1024 -out iitCA2021.pem -config c:/openssl-win64/bin/cnf/openssl.cnf**

Во время создания будет спрашивать различную информацию, некоторые поля можно не заполнять. У меня получилось так:

*Country Name (2 letter code) [AU]:BY*

*State or Province Name (full name) [Some-State]:*

*Locality Name (eg, city) []:*

*Organization Name (eg, company) [Internet Widgits Pty Ltd]: IIT*

*Organizational Unit Name (eg, section) []:*

*Common Name (e.g. server FQDN or YOUR name) []: DATA CENTER IIT*

*Email Address []: info@beliit.com*

1. **Создание сертификата для Fibertest’a**
   1. Генерируем ключ для fibertest:

**openssl genrsa -out iit-fibertest.key 2048**

Внимание: ключ *приватный*, никому давать нельзя.

* 1. Из этого ключа создаем запрос сертификата:

**openssl req -new -key iit-fibertest.key -out iit-fibertest.csr**

*A challenge password []:* ***AchallengePassw0rd!***

Обратите внимание на *Common Name*, там должно быть имя по которому клиенты будут стучаться к серверу либо его ip, если по ip. Если имя не будет совпадать в некоторых браузерах будете получать предупреждение.

Пример:

*Country Name (2 letter code) [AU]:BY*

*State or Province Name (full name) [Some-State]:*

*Locality Name (eg, city) []: MINSK*

*Organization Name (eg, company) [Internet Widgits Pty Ltd]: IIT*

*Organizational Unit Name (eg, section) []:*

*Common Name (e. g. server FQDN or YOUR name) []: iit-fibertest*

*Email Address []:* [*info@beliit.com*](mailto:info@beliit.com)

*Когда Common Name был* ***fibertest*** *и в хостах есть запись для* ***fibertest*** *и обращался к сайту как* [*https://fibertest*](https://fibertest)*:.... ругалось на несоответствие* ***iit-fibertest*** *и* ***fibertest****. Сделал чтобы совпадало с именем файла. Возможно можно было наоборот файлы все именовать* ***fibertest.XXX***

* 1. В текстовом редакторе создаеь iit-fibertest.ext

*authorityKeyIdentifier=keyid,issuer*

*basicConstraints=CA:FALSE*

*keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment*

*subjectAltName = @alt\_names*

*[alt\_names]*

*DNS.1 = iit-fibertest*

* 1. Подписывая запрос из 2b сертификатом, созданным в пункте 1b. cоздаем SSL сертификат на 1024 дня

**openssl x509 -req -in iit-fibertest.csr -CA iitCA2021.pem -CAkey iitCA2021.key -CAcreateserial -out iit-fibertest.crt -days 2048 -sha256 -extfile iit-fibertest.ext**

* 1. Экспортируем сертификат в pfx для дальнейшей установки в IIS или использования Kestrel’ом

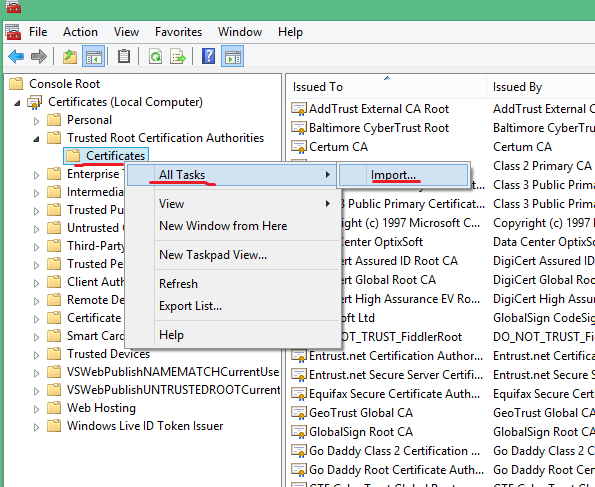
**openssl pkcs12 -export -out iit-fibertest.pfx -inkey iit-fibertest.key -in iit-fibertest.crt -certfile iitCA2021.pem**

*export password:* ***AchallengePassw0rd!***

Установка сертификатов

В винде

1. Открыть Win+R -> mmc
2. File->Add/Remove Snap-In… (Ctrl+M). Выбрать Certificates. Computer account, LocalComputer.
3. Правой кнопкой по Trusted Root CA. All Task->Import. Во время поиска указать путь к iitCA.pem (прежде выбрать AllFiles (\*.\*))



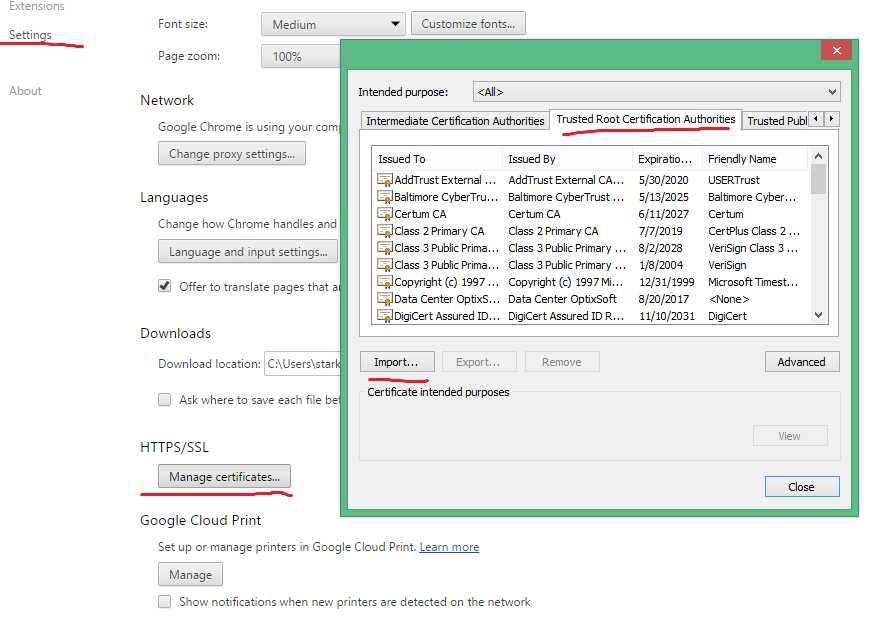
На IIS сервере

1. Открыть IIS. Server Sertificates.
2. Правой кнопкой Import.. Выбрать **iit-fibertest**.pfx, ввести пароль, нажать ok.

Установка CA сертификата для браузеров на клиентских машинах (отдаем только optixsoftCA.pem)

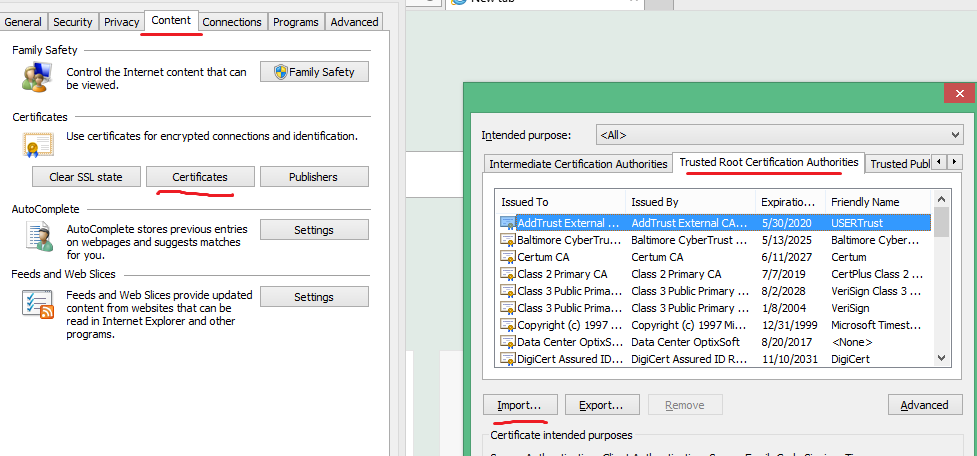
**Chrome**:

1. Settings, Show advanced settings.., Manage certificates...
2. Закладка Trusted Root CA, нажать Import, выбрать AllFiles(\*.\*), выбрать optixsoftCA.pem
3. Рестарт браузера.



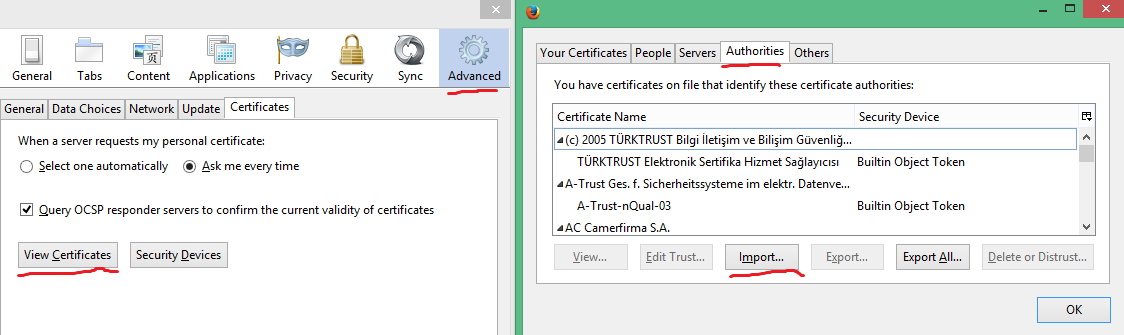
**IE**:

1. Настройки, Контент, Сертификаты
2. Закладка Trusted Root CA, нажать Import, выбрать AllFiles(\*.\*), выбрать optixsoftCA.pem
3. Рестарт браузера.



**Firefox**:

1. Настройки, Расширенные, Просмотр сертификатов.
2. Закладка Authorities, нажать Import, выбрать optixsoftCA.pem
3. Рестарт браузера.



Для IE и Chrome также можно добавить сертификат через mmc (см. добавление сертификата на IIS сервере ).

Дополнительно

**Для создания multi-domain** сертификата необходимо править (создать копию) openssl.cfg

1. В [ req ] секцию добавить req\_extensions = v3\_req
2. В [ v3\_req ] секцию добавить subjectAltName = @alt\_names
3. Создать секцию [ alt\_names ] с альтенативными доменами.
4. Команда генерации с дополнительными опциями

*openssl x509 -req -in opx-fiberizer.csr -CA optixsoftCA.pem -CAkey optixsoftCA.key -CAcreateserial -out opx-fiberizer..crt -days 1024 -extensions v3\_req -extfile openssl.cfg*

Пример для п 2. и п 3.,

*[ v3\_req ]  
  
# Extensions to add to a certificate request  
  
basicConstraints = CA:FALSE  
keyUsage = nonRepudiation, digitalSignature, keyEncipherment  
subjectAltName = @alt\_names*

*[ alt\_names ]  
DNS.1 = www.example.com  
DNS.2 = example.com*

**Для проверки certificate request** можно использоваь команду:

*openssl req -in some.csr -noout -text*