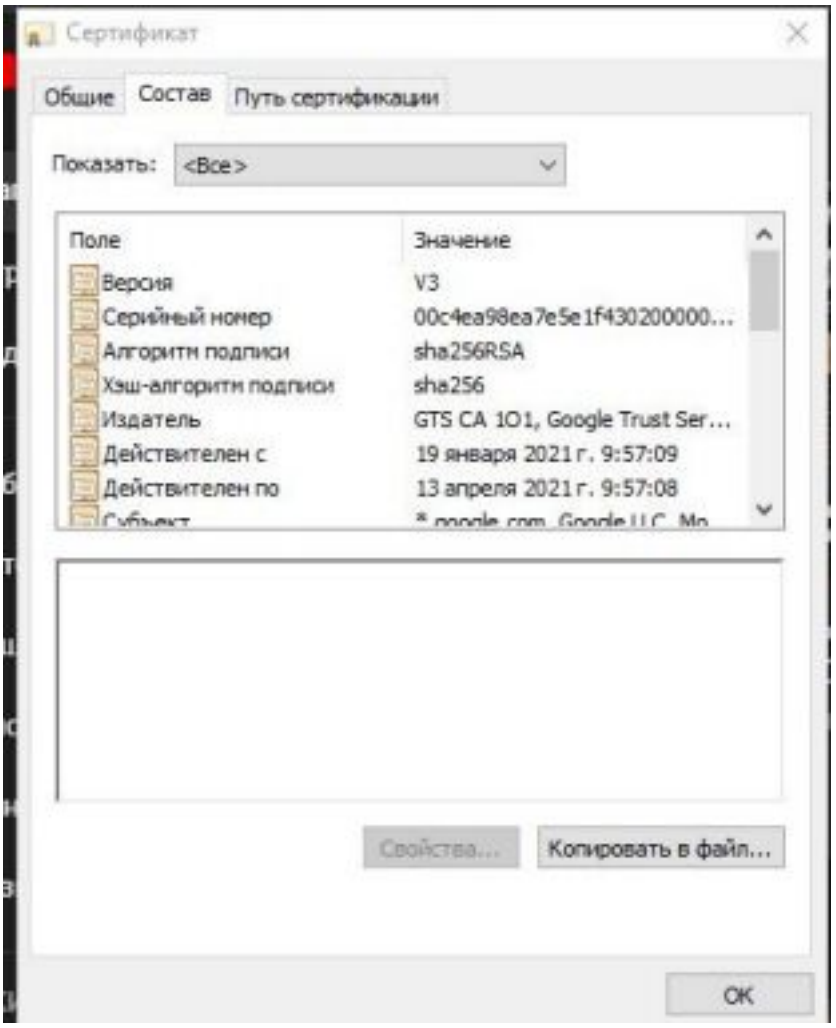
Lab 7 Report

Алгоритми генерації ключа було вибрано з сертифікат youtube.com.



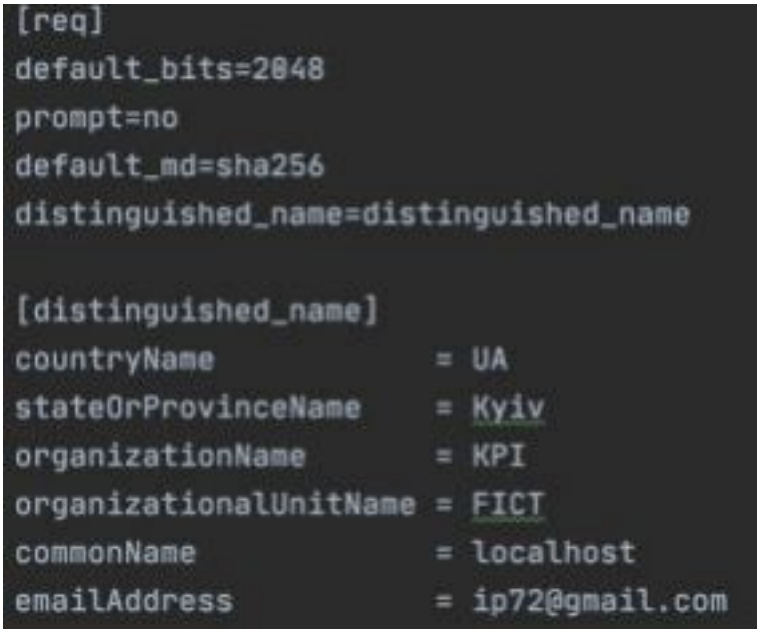
Для генерації пари ключів в ньому використовується алгоритм на основі elliptic curves ***openssl ecparam -genkey -name prime256v1 -out key.pem***

Специфікація openssl - "***prime256v1***" це аліас до "secp256r1".

Генеруємо запит на підпис сертифікату за допомогою команди:

openssl req -new -sha256 -key key.pem -out csr.pem -config csr.config

Файл з конфігураціями:



Підписуємо його за допомогою раніше створеного ключа, таким чином отримуємо self-signed certificate:

openssl x509 -req -in csr.pem -signkey key.pem -out crt.pem -days 365 -sha256

Добавляємо наш ключ та сертифікат до серверу написаного на node.js.

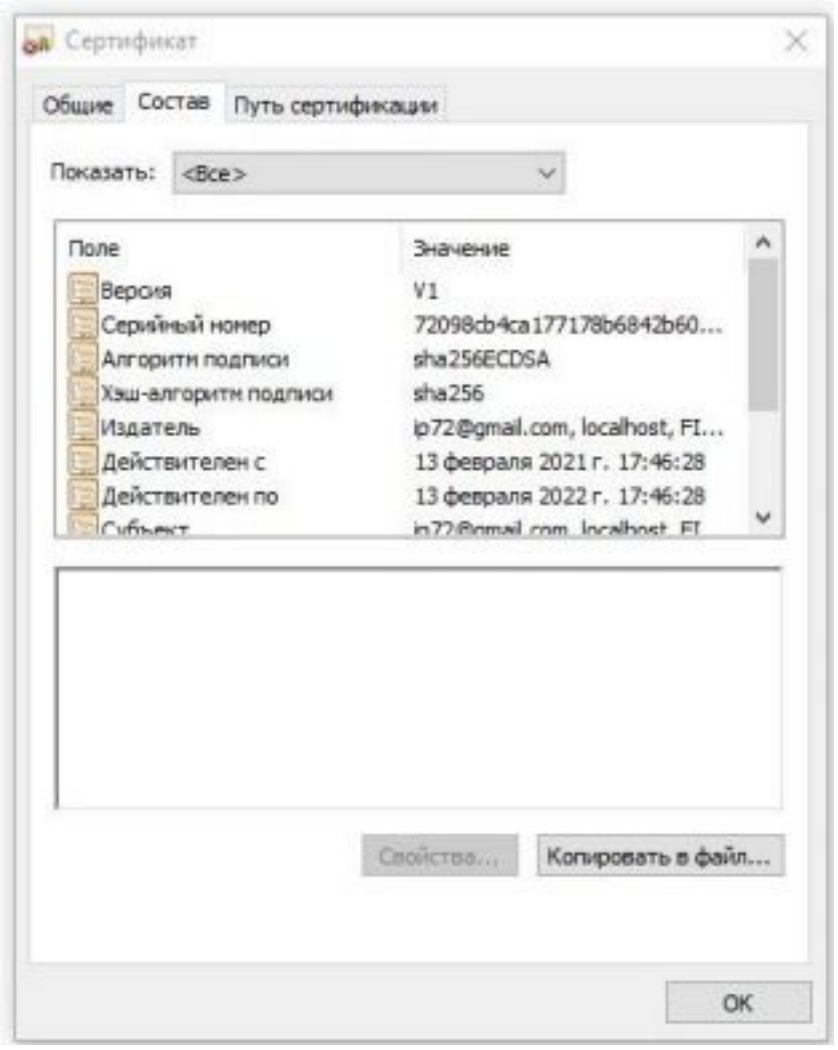
server.js

```
const https = require('https');
const fs = require('fs');
const express = require('express')

const options = {
    cert: fs.readFileSync( path: './certs/crt.pem'),
    key: fs.readFileSync( path: './certs/key.pem')
};

const app = express();

https.createServer(options, app).listen( port: 8000);
```



Для генерації ключа було використано алгоритм на базі еліптичної кривої, так як він вважається більш надійним для шифрування, ніж базовий алгоритм RSA.

Згенерований сертифікат може знаходитись поруч з кодом серверу, так як це не є секретною інформацією. Для збереження приватного ключа бажано використовувати місця з підвищеним захистом та обмеженим доступом, наприклад спеціальні програми по типу “Keychain” для того, щоб безпечно зберігати приватні ключі та надавати до них доступ тільки авторизованим користувачам чи програмам.