МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Чернігівська політехніка»

Навчально-науковий інститут електроних та інформаційних технологій

Кафедра кібербезпеки та математичного моделювання

**ЗВІТ**

про виконання лабораторної роботи

з дисципліни «Основи криптографічного захисту інформації»

Виконав: здобувач 3 курсу, групи КБ-221 Регент Анастасія

Перевірив: Шелест Михайло Євгенович

НУ «Чернігівська політехніка» 2025

Лабораторна робота № 9

**Public Key Infrastructure / TLS**

**Мета роботи:** ознайомлення з процедурою створення ключів та сертифікатів для них.

В ході виконання завдань ви:

згенеруєте вашу ключову пару (приватний та відкритий ключі);

створите самопідписаний сертифікат з цієї пари;

Створите запит CSR (Certificate Signing Request).

В завданнях надані команди для виконання OpenSSL в командному рядку. Проте ті самі дії можна виконати програмно, використовуючи бібліотеку можливості роботи з X.509 в бібліотеці cryptography. Інструкції з використання cryptography для роботи з сертифікатами наведені [в документації](https://cryptography.io/en/latest/x509/tutorial/) бібліотеки.

Використовувати OpenSSL в командному рядку, або програмну бібліотеку cryptoraphy – залишається на ваш вибір.

**Завдання № 1 Створення ключової пари**

Необхідно створити ключову пару RSA.

Алгоритм RSA є повільним порівняно з сучаснішими альтернативами, проте раніше мав ширшу підтримку в програмному забезпеченні. Зараз це поволі змінюється і алгоритми ECC стають все більш поширеними. Проте скоро знову доведеться мати справу з проблемами сумісності через перехід на алгоритми стійкі до квантових атак (про це детальніше в майбутніх лекціях).

Для створення ключа RSA можна скористатись командою OpenSSL:

openssl genpkey -algorithm RSA -out server.key -pkeyopt rsa\_keygen\_bits:2048.

Згенерований ключ буде збережено в форматі PEM. Дослідити вміст ключа можна за допомогою команди:

openssl rsa -text -in server.key.

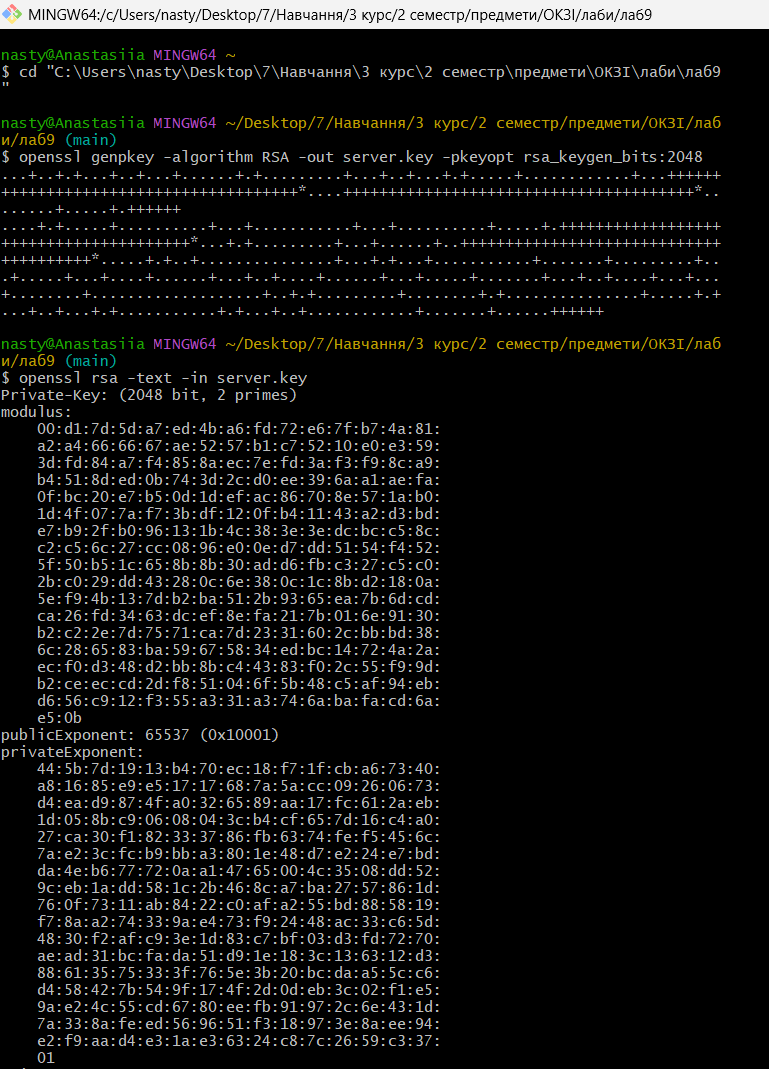


Рисунок 1 – Створення приватного ключа та перегляд вмісту у зрозумілому вигляді

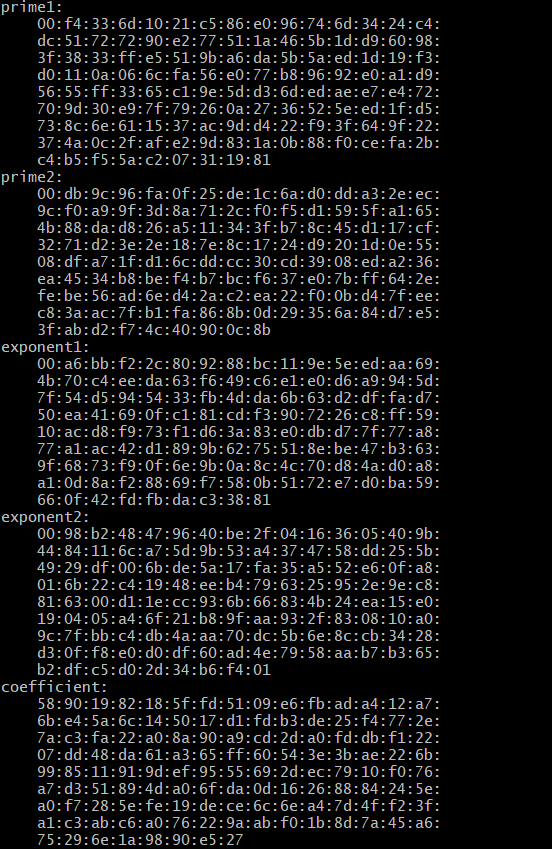


Рисунок 2 – Перегляд вмісту у зрозумілому вигляді 1.2

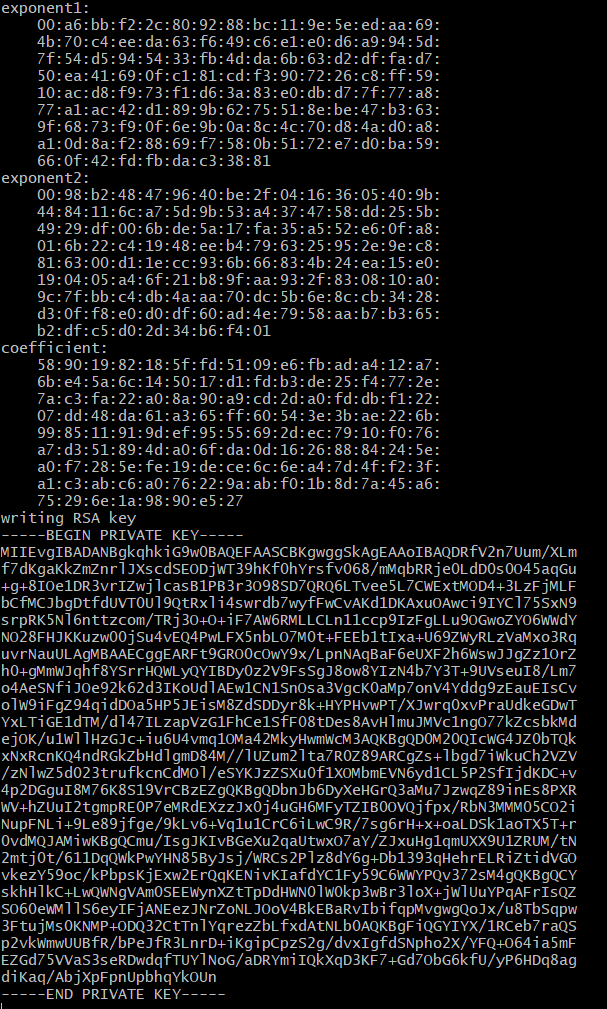


Рисунок 3 – Перегляд вмісту у зрозумілому вигляді 1.3

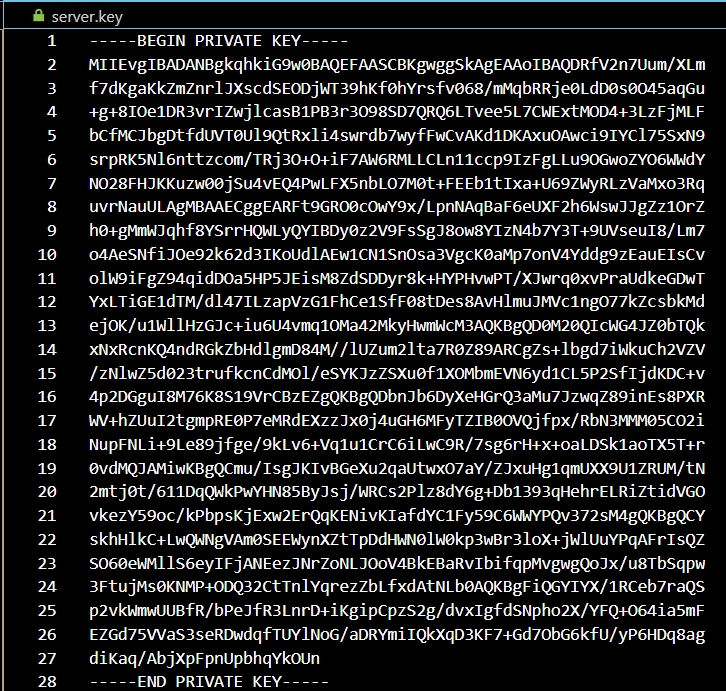


Рисунок 4 – Приватний ключ у PEM

**Завдання № 2 Створення запиту на формування сертифікату**

Маючи ключову пару (файл з приватним ключем, в якому зберігається вся необхідна інформація, включаючи відповідний відкритий ключ), необхідно згенерувати запит на формування сертифікату для цього ключа. CSR є формальним запитом до центру сертифікації (CA) на створення сертифікату на ваш ключ. CSR міститиме ваш відкритий ключ та відкриті параметри, а також мета-інформацію про власника ключа. Тож по суті, CSR є лише контейнером з мета-інформацією про ключ та його власника.

openssl req -new -keyout server.key -out server.csr \ -subj "/C=UA/L=Kyiv/O=Robot\_dreams/CN=www.robotdreams.cc/emailAddress=youremail@example.com" \ -nodes.9

Головним параметром в CSR є common name (CN), який повинен відповідати доменному імені вашого сервера. Решта параметрів є опціональними.

Перевірте вміст CSR наступною командою:

openssl req -text -noout -in server.csr.

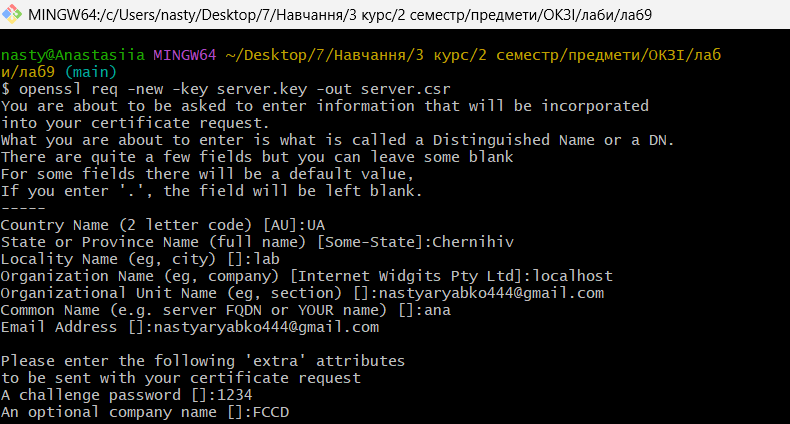


Рисунок 5 – Введення даних вручну

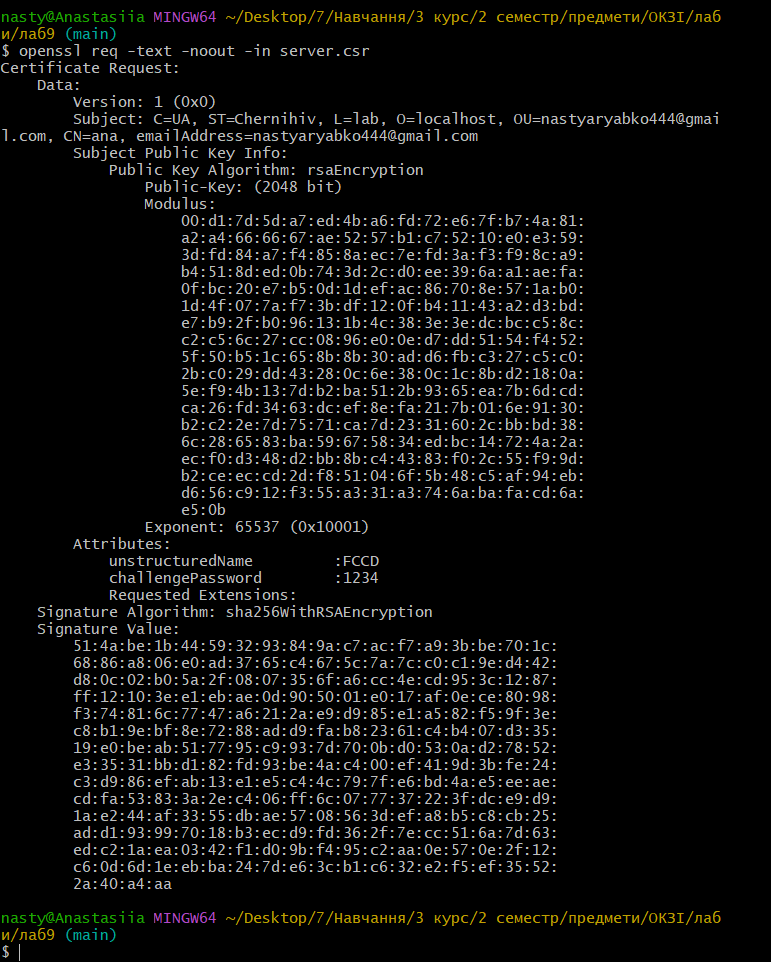


Рисунок 6 – Перевірка вмістуЯк CSR



Рисунок 7 – Запит на сертифікат

**Завдання № 3 Створити самопідписаний сертифікат**

Створіть самопідписаний сертифікат (тобто сертифікат, підписаний тим же ключем, який він завіряє). Це часто стає в нагоді для локального тестування та розробки:

openssl x509 -req -sha256 -in server.csr -signkey server.key -out server.crt.

Отриманий сертифікат server.crt можна додати до списку довірених кореневих сертифікатів на вашій системі – тоді система сприйматиме його як повноцінний сертифікат.

Дослідіть вміст створеного сертифікату командою:

openssl x509 -text -noout -in server.crt.

Також ви можете переглянути структуру ASN.1 цього сертифікату:

openssl asn1parse -in server.crt.

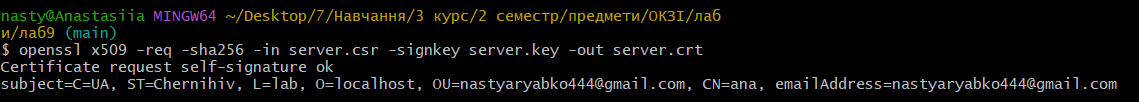
****

Рисунок 8 – Створення самопідписаного сертифіката

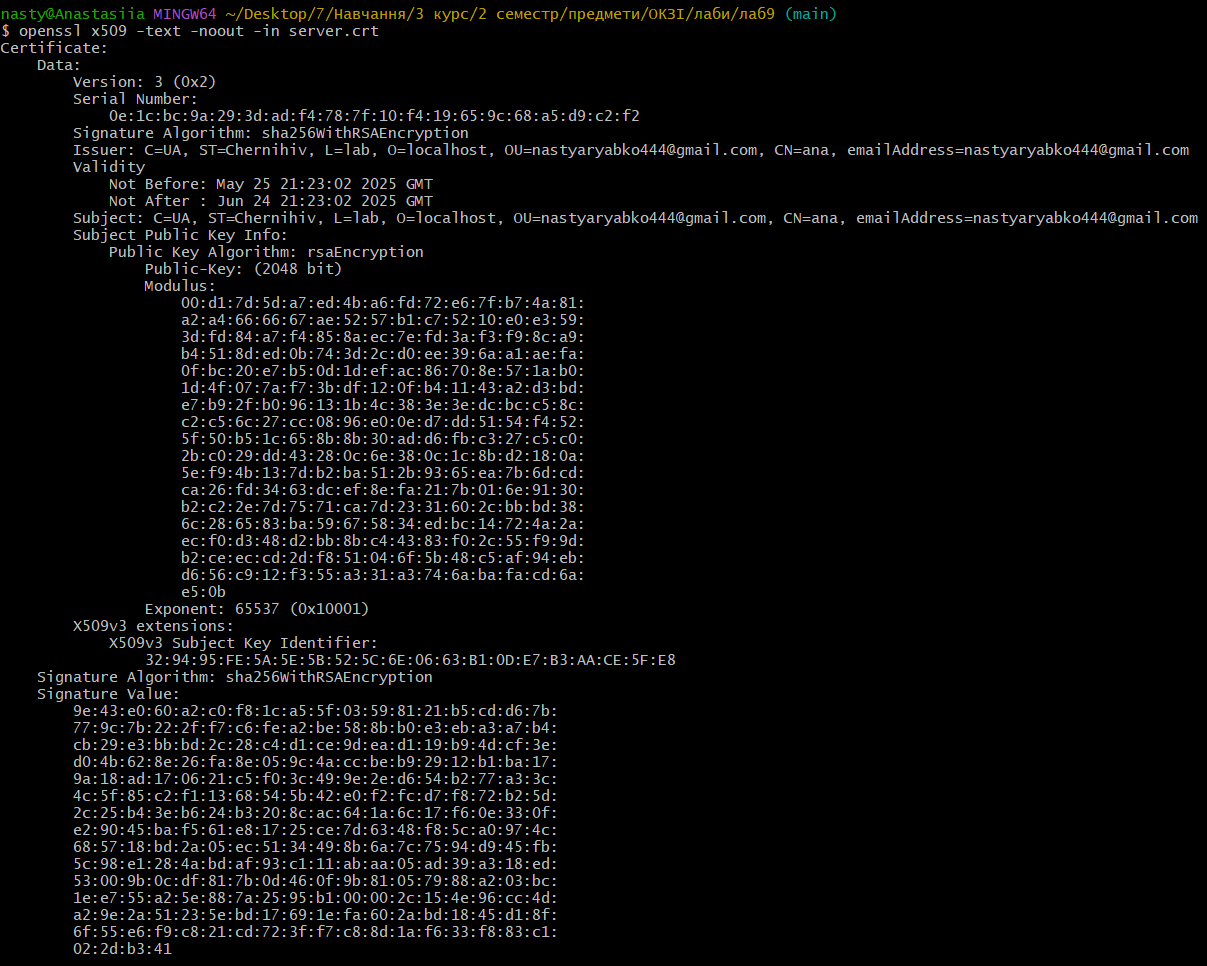
****

Рисунок 9 – Перегляд вмісту сертифіката

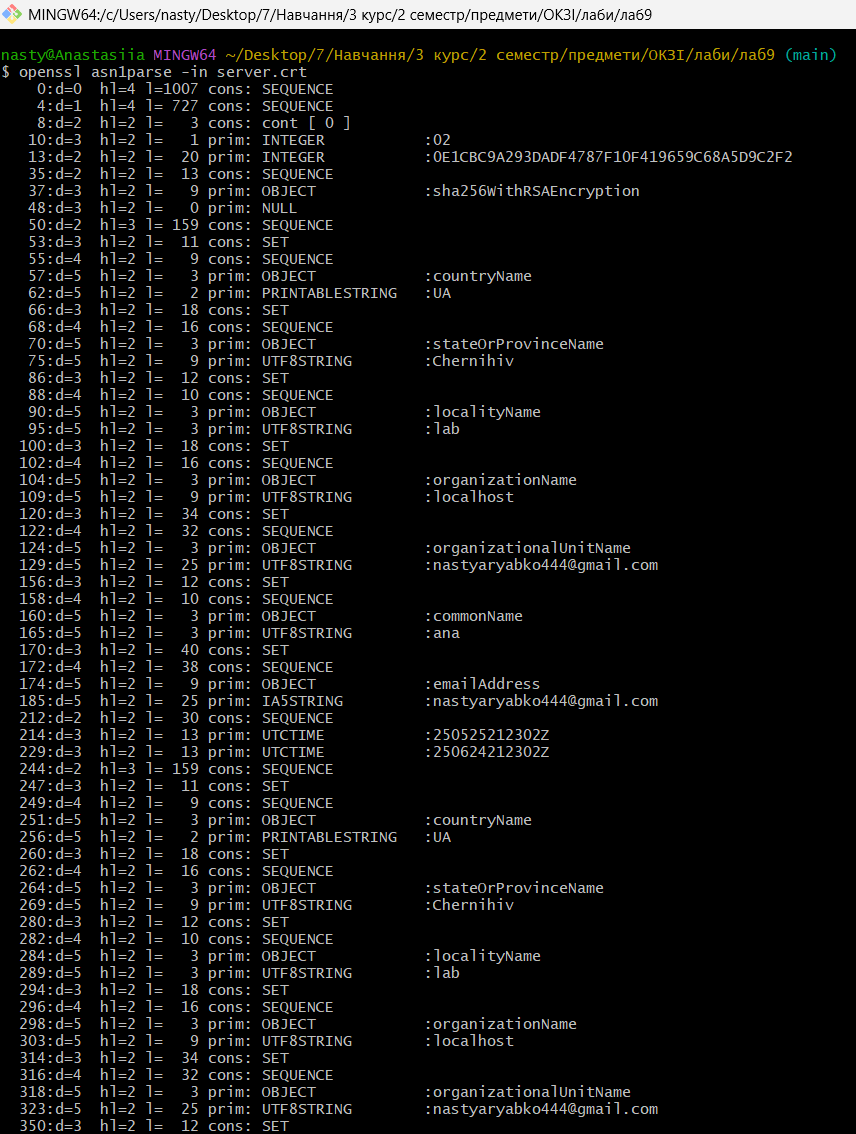
****

Рисунок 10 – Перегляд ASN.1 структури сертифіката

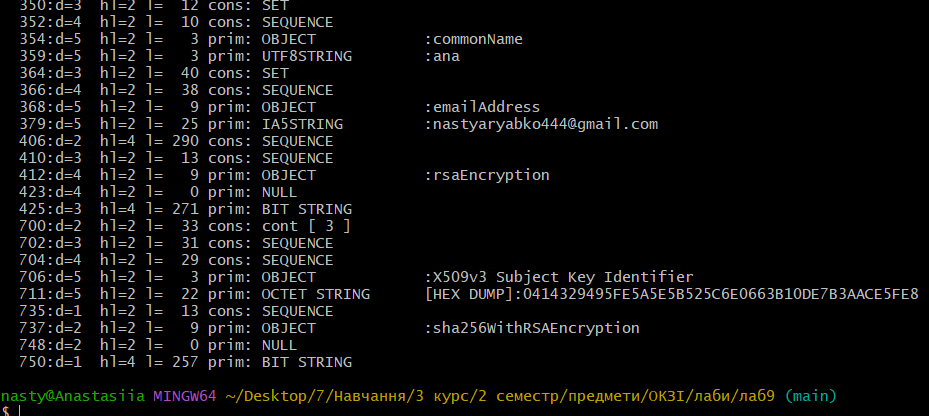
****

Рисунок 11 – Перегляд ASN.1 структури сертифіката 1.2

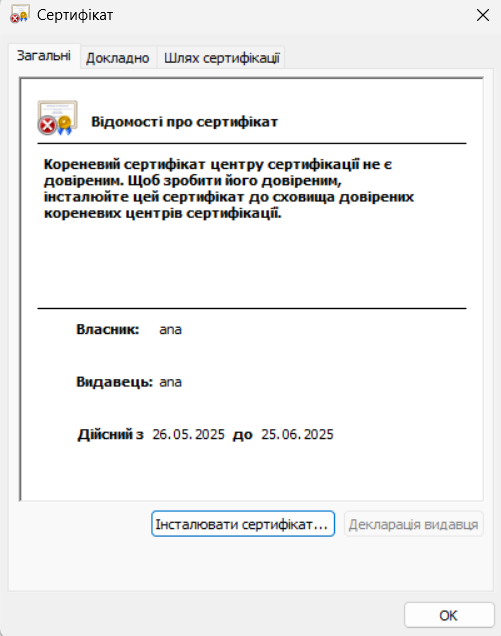


Рисунок 12 – Сертифікат

**Завдання № 4 Дослідження публічного сертифікату**

Створений нами самопідписаний сертифікат має найпростішу можливу структуру з мінімум необхідної інформації. Справжні публічні сертифікати містять значно більше інформації.

Ви можете отримати повний ланцюжок сертифікатів для будь-якого публічного вебсайту що використовує TLS за допомогою OpenSSL.

Отримайте ланцюжок сертифікатів для сайту robotdreams.cc наступною командою: openssl s\_client -showcerts -connect robotdreams.cc:443

Для виконання цього завдання необхідно дати відповіді на наступні питання, досліджуючи отримані сертифікати:

Якою є назва організації що видала чинний сертифікат для сайту robotdreams.cc?

Який сертифікат з ланцюга має найдовший термін дії?

Дослідивши детальніше сертифікат виданий robotdreams.cc можна знайти деталі про ключ, який використовується сервером robotdreams.cc для автентифікації сервера. Який розмір ключа (в бітах) та яка еліптична крива була використана для формування ключа robotdreams.cc?

**Додаткова інформація**

Детальніше дослідити кожен сертифікат з отриманого ланцюжка можна скопіювавши PEM кодування сертифікату в окремий текстовий файл (скажімо, test.crt) та продивитися його командою:

openssl x509 -text -noout -in test.crt.

Відповіді на питання можна зберегти у текстовий файл та завантажити до репозиторію з домашніми завданнями.

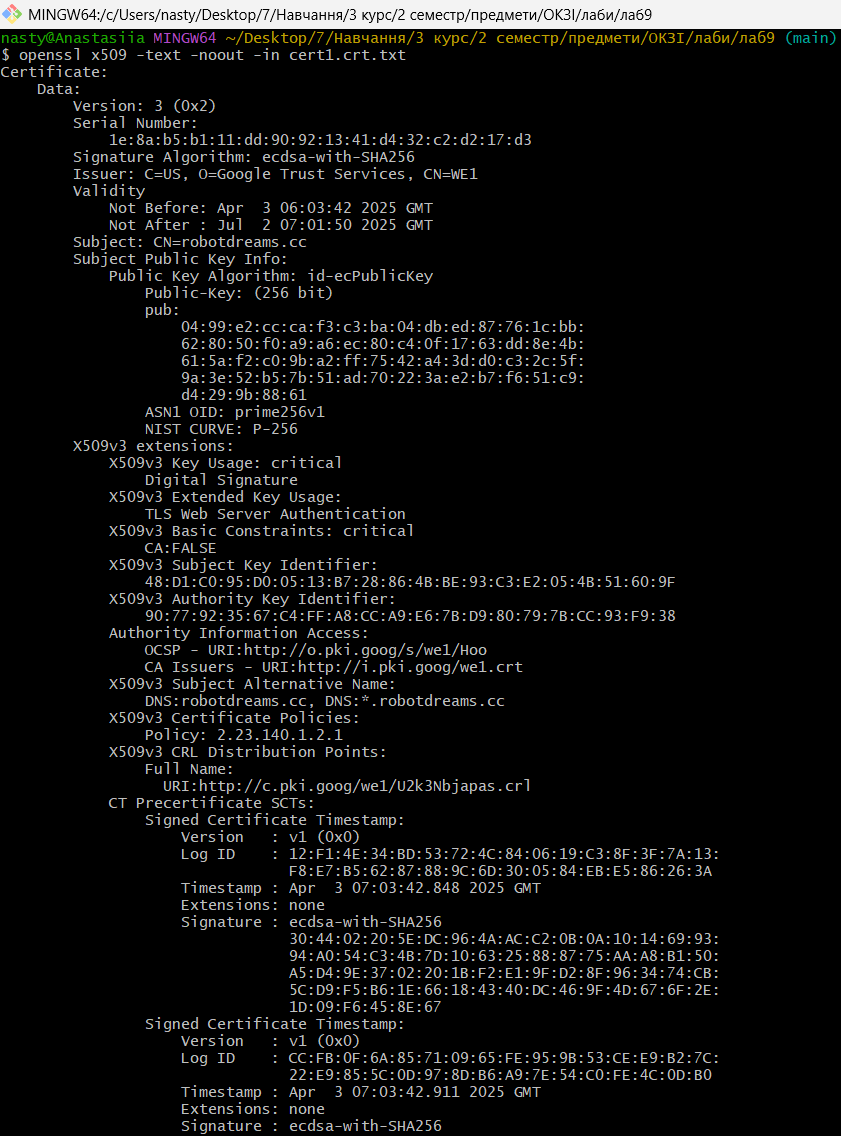


Рисунок 13 – Перший сертифікат (зазвичай сайту)

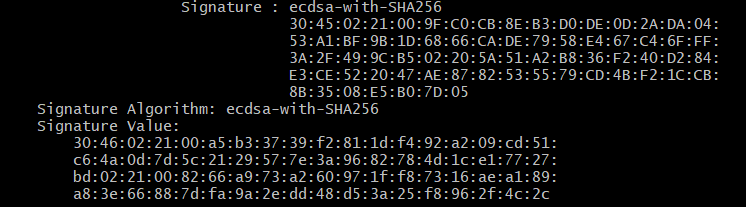


Рисунок 14 – Перший сертифікат (зазвичай сайту) 1.2

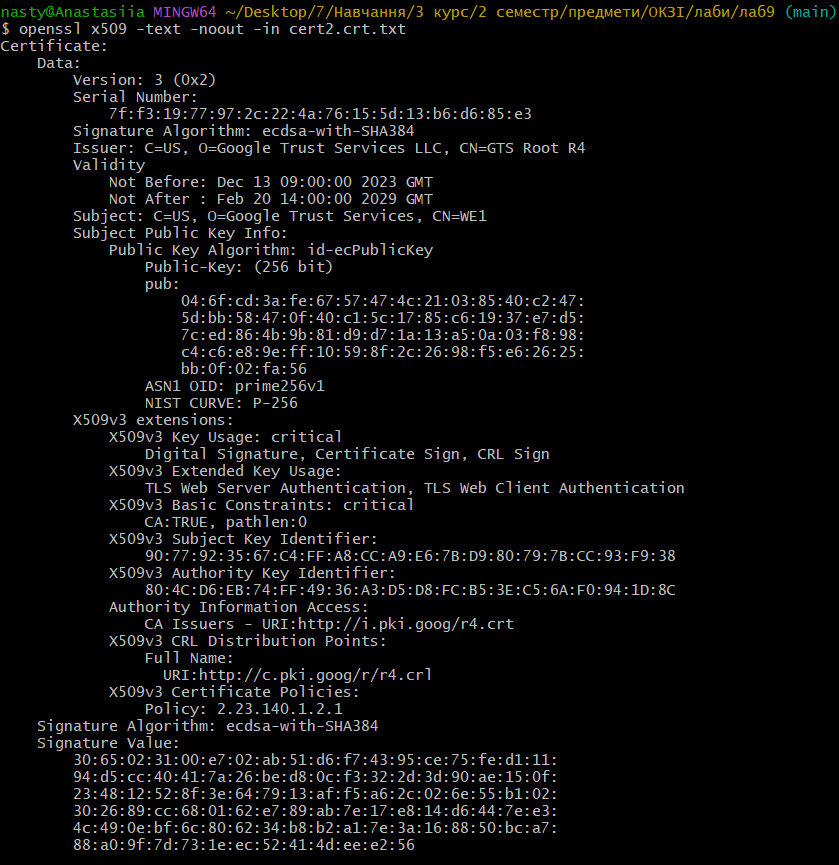


Рисунок 15 – Сертифікат проміжного центру сертифікації (CA)

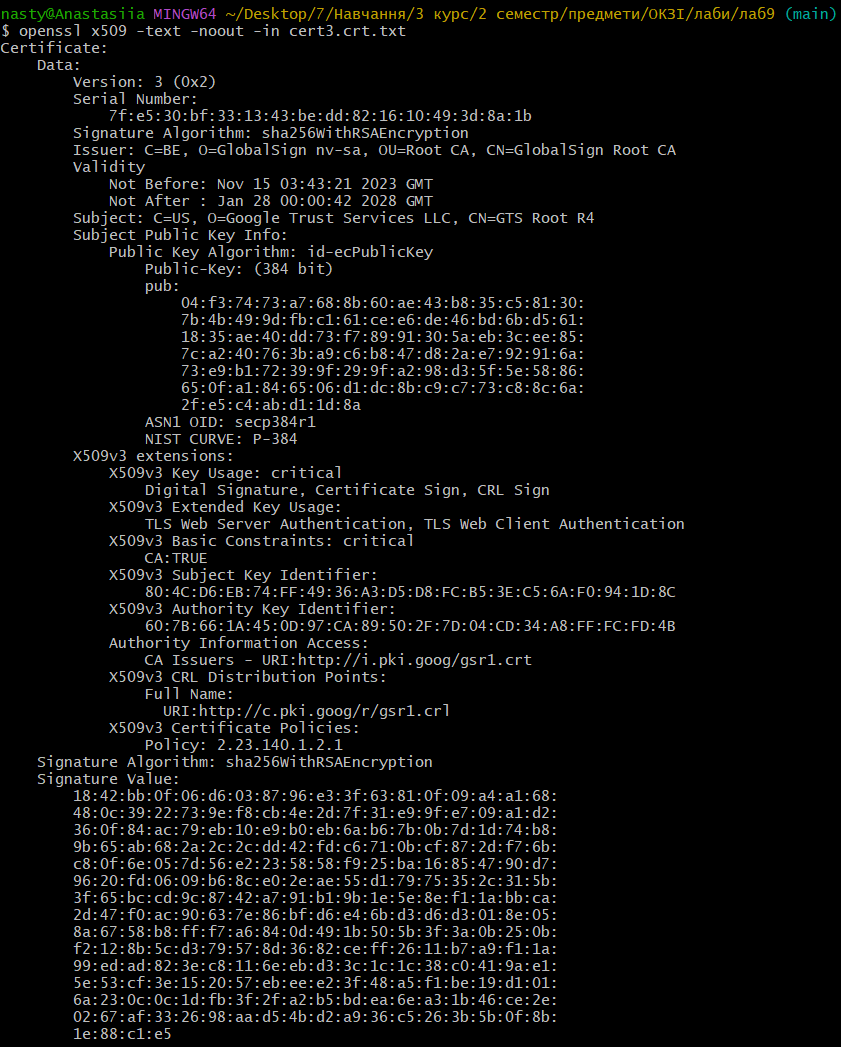


Рисунок 16 – Кореневий сертифікат (root CA)

Таблиця 1 – Дані першого сертифікату

|  |  |
| --- | --- |
| Поле | Значення |
| **Версія** | 3 (0x2) |
| **Серійний номер** | 1e:8a:b5:b1:11:dd:90:92:13:41:d4:32:c2:d2:17:d3 |
| **Алгоритм підпису** | ecdsa-with-SHA256 |
| **Видавець (Issuer)** | C=US, O=Google Trust Services, CN=WE1 |
| **Суб’єкт (Subject)** | CN=robotdreams.cc |
| **Термін дії** | Apr 3 06:03:42 2025 GMT – Jul 2 07:01:50 2025 GMT |
| **Публічний ключ** | Еліптична крива (EC), 256 біт (NIST P-256) |
| **Key Usage** | Digital Signature (критичне) |
| **Extended Key Usage** | TLS Web Server Authentication |
| **Basic Constraints** | CA:FALSE (тобто це не сертифікат центру сертифікації) |
| **SAN** | robotdreams.cc, \*.robotdreams.cc |
| **Authority Info Access** | OCSP/CA Issuers URI присутні |
| **CRL Distribution Point** | Є URI: http://c.pki.goog/we1/... |
| **SCTs** | Два SCT (Signed Certificate Timestamps) |

Таблиця 2 – Дані другого сертифікату

|  |  |
| --- | --- |
| Поле | Значення |
| **Версія** | 3 (0x2) |
| **Серійний номер** | 7f:f3:19:77:97:2c:22:4a:76:15:5d:13:b6:d6:85:e3 |
| **Алгоритм підпису** | ecdsa-with-SHA384 |
| **Видавець (Issuer)** | CN=GTS Root R4 |
| **Суб’єкт (Subject)** | CN=WE1 |
| **Термін дії** | Dec 13 2023 – Feb 20 2029 |
| **Публічний ключ** | EC, 256 біт (NIST P-256) |
| **Key Usage** | Digital Signature, **Certificate Sign**, **CRL Sign** (критичне) |
| **Basic Constraints** | **CA:TRUE**, pathlen:0 (може підписувати серти лише 1 рівня) |
| **Authority Info Access** | URI: http://i.pki.goog/r4.crt |
| **CRL Distribution Point** | URI: http://c.pki.goog/r/r4.crl |

Таблиця 3 – Дані третього сертифікату

|  |  |
| --- | --- |
| Поле | Значення |
| Версія | 3 (0x2) |
| Серійний номер | 7f:e5:30:bf:33:13:43:be:dd:82:16:10:49:3d:8a:1b |
| Алгоритм підпису | sha256WithRSAEncryption |
| Видавець (Issuer) | C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA |
| Суб’єкт | C=US, O=Google Trust Services LLC, CN=GTS Root R4 |
| Термін дії | Not Before: Nov 15 03:43:21 2023 GMTNot After : Jan 28 00:00:42 2028 GMT |
| Публічний ключ | Algorithm: id-ecPublicKey (secp384r1, NIST CURVE P-384)Public-Key: (384 bit) |
| Key Usage | Digital Signature, Certificate Sign, CRL Sign (critical) |
| Basic Constraints | CA:TRUE (critical) |
| Authority Info Access | CA Issuers - URI:<http://i.pki.goog/gsr1.crt> |
| CRL Distribution Point | URI:<http://c.pki.goog/r/gsr1.crl> |

**Завдання № 5 Перегляд логів Certificate Transparency**

Використовуючи сервіс [crt.sh](https://crt.sh/), дослідіть логи видачі сертифікатів на домен robotdreams.cc.

Коли (дата) був виданий найперший сертифікат для доменту robotdreams.cc (шукати по common name)?

Який термін дії мав цей перший сертифікат?

**Додаткова інформація:**

Інколи [crt.sh](https://crt.sh/) може повертати помилку 502. У цьому випадку спробуйте ще раз за кілька хвилин.

cryptography-1/lesson-9/homework at main · robot-dreams-code/cryptography-1

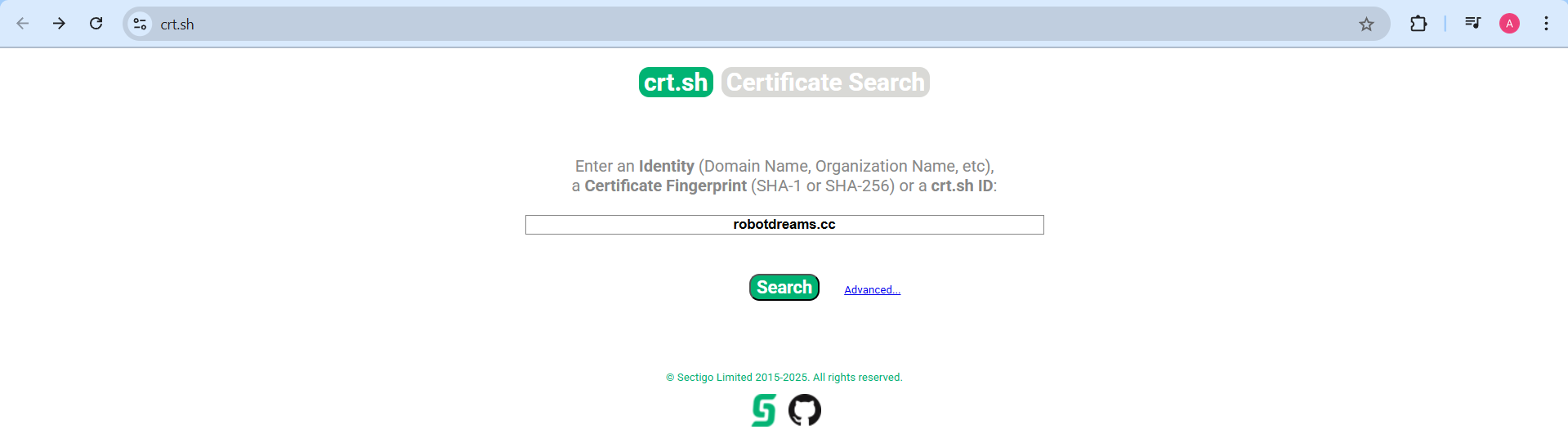


Рисунок 17 – Сайт

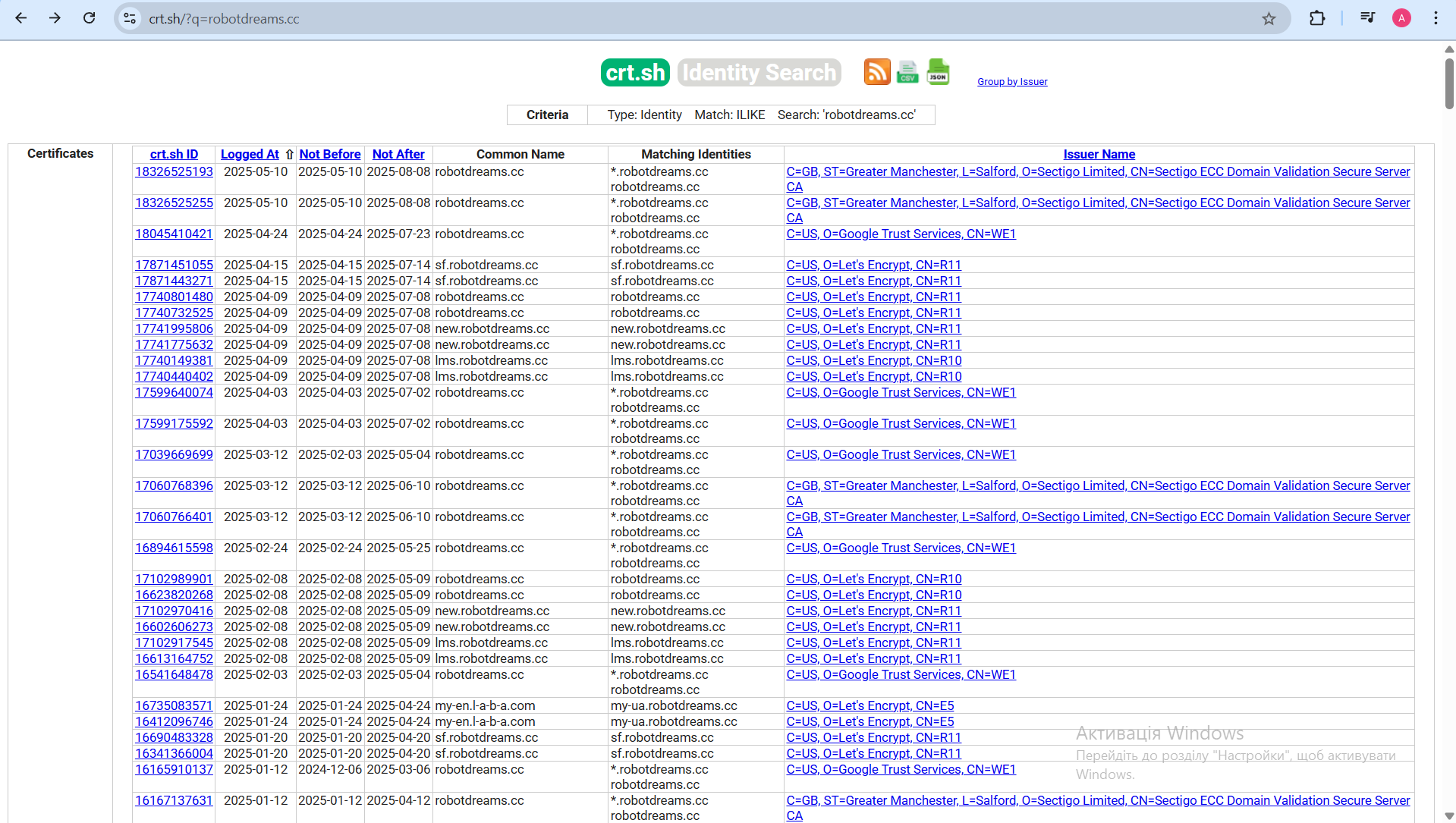


Рисунок 18 – Список сертифікатів

Для дослідження було використано сервіс [crt.sh](https://crt.sh), за допомогою якого було отримано інформацію про сертифікати, видані для домену robotdreams.cc:

Найперший сертифікат, який містить домен robotdreams.cc у Subject або Common Name (CN), було видано 18 травня 2020 року.

Common Name (CN) цього сертифіката — sni.cloudflaressl.com, а у Subject присутній wildcard сертифікат для \*.robotdreams.cc.

Термін дії цього сертифіката тривав з 18 травня 2020 року до 9 жовтня 2020 року (приблизно 5 місяців).

**Контрольні запитання**

1. Які криптографічні алгоритми використовуються в TLS 1.3?

1. Симетричні шифри – AES-GCM, ChaCha20-Poly1305;

2. Алгоритми хешування – SHA-256, SHA-384;

3. Асиметричні алгоритми – ECDSA (Elliptic Curve Digital Signature Algorithm), RSA (іноді);

4. Алгоритми обміну ключами – ECDHE (Elliptic Curve Diffie-Hellman Ephemeral).

2. Що таке Forward Secrecy і як TLS реалізує цей принцип?

Forward Secrecy — це властивість протоколу, яка забезпечує, що компрометація приватного ключа сервера не дасть змоги розшифрувати минулі сесії. TLS 1.3 реалізує це за допомогою тимчасових ключів обміну (Ephemeral Diffie-Hellman), які генеруються для кожної сесії окремо.

3. Що таке сертифікат X.509 і які основні його складові?

Сертифікат X.509 — це стандартизований цифровий документ, який підтверджує зв’язок між публічним ключем і ідентичністю власника.  
**Основні складові:**

1. Серійний номер;

2. Алгоритм підпису;

3. Видавець (Issuer);

4. Суб’єкт (Subject);

5. Публічний ключ;

6. Термін дії (Valid From, Valid To);

7. Розширення (Key Usage, Basic Constraints тощо);

8. Версія сертифіката;

9. Цифровий підпис.

4. Яка роль Центру Сертифікації (CA) у PKI?

Центр Сертифікації (CA) видає цифрові сертифікати, підтверджує ідентичність власників ключів і підписує сертифікати, встановлюючи довіру в PKI (Public Key Infrastructure).

5. Як працює механізм валідації сертифікатів? Що таке ланцюг довіри?

Валідація сертифіката полягає у перевірці його підпису, терміну дії, відкликання (CRL/OCSP) та довіреності до кореневого сертифіката.  
Ланцюг довіри — це послідовність сертифікатів від кінцевого користувача до кореневого CA, кожен із яких підписаний попереднім, що підтверджує автентичність.

6. Що таке кроспідпис сертифікатів і в яких випадках він використовується?

Кроспідпис — це підписування одного сертифіката іншим CA для встановлення довіри між різними PKI. Використовується при переході між CA або для розширення довіри.

7. Які основні етапи життєвого циклу сертифіката?

1. Генерація ключів та запит на сертифікат (CSR);

2. Видача сертифіката CA;

3. Використання сертифіката (автентифікація, шифрування);

4. Відкликання або закінчення терміну дії;

5.Оновлення / перевидача.

8. Як перевірити валідність сертифіката за допомогою браузера чи командного рядка?

У браузері: натиснути на замок біля адреси сайту → Переглянути сертифікат → Перевірити інформацію про термін, видавця, ланцюг довіри.

В командному рядку:

openssl x509 -in cert.pem -text -noout