



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

## **Кіберзахист об'єктів критичної інфраструктури**

### **Лабораторний практикум №5**

#### **Впровадження MQTT з протоколом безпеки транспортного рівня використовуючи бібліотеку OpenSSL**

Перевірив:

Войцеховський А. В.

Виконав:

студент I курсу

групи ФБ-41мп

Сахній Н. Р.

Київ 2024

**Мета роботи:** Навчитися встановлювати безпечне/зашифроване з'єднання MQTT між клієнтами [MQTT](#) і [Mosquitto Broker](#), що працюють на комп'ютері з використанням бібліотеки OpenSSL, яка є однією з найбільш широко використовуваних криптографічних інструментів з відкритим кодом у системах на базі UNIX.

### Завдання до виконання:

1. Дерево сертифікатів та інсталяційних бібліотек для Mosquitto MQTT.

```
nazar@mqtt:~/Desktop$ tree .
.
├── certs
│   ├── broker
│   │   ├── broker.crt
│   │   └── broker.key
│   ├── ca
│   │   ├── ca.crt
│   │   ├── ca.key
│   │   └── ca.srl
│   ├── client
│   │   ├── client.crt
│   │   └── client.key
│   ├── publisher
│   │   ├── publisher.crt
│   │   └── publisher.key
│   └── subscriber
│       ├── subscriber.crt
│       └── subscriber.key
└── MQTT-TLS-Libraries
    ├── libdlt2_2.18.5-0.2_amd64.deb
    ├── libmosquitto1_1.6.12-1_amd64.deb
    ├── libwebsockets16_4.0.20-1_amd64.deb
    ├── mosquitto_1.6.12-1_amd64.deb
    └── mosquitto-clients_1.6.12-1_amd64.deb

7 directories, 16 files
```

Дерево сертифікатів використовується для встановлення MQTT(S)-з'єднань (M2M-комунікацій на основі SSL/TLS) із метою забезпечення автентифікації та шифрування між різними учасниками (брокерами, клієнтами, публікаторами (**publishers**) та підписниками (**subscribers**)).

## 2. Надсилання вимірювальних показників із датчиків температури OVEN.

```
nazar@mqt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert publisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 125°C
nazar@mqt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert publisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 140°C
nazar@mqt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert publisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 160°C
nazar@mqt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert publisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 200°C
nazar@mqt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert publisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m ***°C
```

Отже, для прикладу, в даному випадку датчик температури виступає в ролі “Publisher”, тому його завдання зводиться лише до публікації даних у бік брокера. Стандартний порт MQTT-брокера для вхідних TCP-з'єднань – 1883. При використанні захищеного підключення SSL/TLS – порт 8883.

## 3. Отримання даних брокером від **publisher** та надсилання їх до **subscriber**.

```
nazar@mqt:~/Desktop/certs$ sudo mosquitto -v -c /etc/mosquitto/mosquitto.conf
1732008102: mosquitto version 1.6.12 starting
1732008102: Config loaded from /etc/mosquitto/mosquitto.conf.
1732008102: Opening ipv4 listen socket on port 8883.
1732008102: Opening ipv6 listen socket on port 8883.
1732008102: mosquitto version 1.6.12 running
1732008106: New connection from 127.0.0.1 on port 8883.
1732008106: New client connected from 127.0.0.1 as mosq-BDtpH0chR1gs8Yf7fR (p2, c1, k60).
1732008106: No will message specified.
1732008106: Sending CONNACK to mosq-BDtpH0chR1gs8Yf7fR (0, 0)
1732008106: Received SUBSCRIBE from mosq-BDtpH0chR1gs8Yf7fR
1732008106: /home/kitchen/oven_temperature (QoS 0)
1732008106: mosq-BDtpH0chR1gs8Yf7fR 0 /home/kitchen/oven_temperature
1732008106: Sending SUBACK to mosq-BDtpH0chR1gs8Yf7fR
1732008110: New connection from 127.0.0.1 on port 8883.
1732008110: New client connected from 127.0.0.1 as mosq-KLxuZD2w9H45W4o1gl (p2, c1, k60).
1732008110: No will message specified.
1732008110: Sending CONNACK to mosq-KLxuZD2w9H45W4o1gl (0, 0)
1732008110: Received PUBLISH from mosq-KLxuZD2w9H45W4o1gl (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008110: Sending PUBLISH to mosq-BDtpH0chR1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008110: Received DISCONNECT from mosq-KLxuZD2w9H45W4o1gl
1732008110: Client mosq-KLxuZD2w9H45W4o1gl disconnected.
1732008112: New connection from 127.0.0.1 on port 8883.
1732008112: New client connected from 127.0.0.1 as mosq-jtR08fGaabTJzlnIYb (p2, c1, k60).
1732008112: No will message specified.
1732008112: Sending CONNACK to mosq-jtR08fGaabTJzlnIYb (0, 0)
1732008112: Received PUBLISH from mosq-jtR08fGaabTJzlnIYb (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
```

```

1732008112: Received DISCONNECT from mosq-jtR08fGaabTJzlnIYb
1732008112: Client mosq-jtR08fGaabTJzlnIYb disconnected.
1732008114: New connection from 127.0.0.1 on port 8883.
1732008114: New client connected from 127.0.0.1 as mosq-PKBUSAZc6nr2eK5PbM (p2, c1, k60).
1732008114: No will message specified.
1732008114: Sending CONNACK to mosq-PKBUSAZc6nr2eK5PbM (0, 0)
1732008114: Received PUBLISH from mosq-PKBUSAZc6nr2eK5PbM (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008114: Sending PUBLISH to mosq-BDtPH0chr1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008114: Received DISCONNECT from mosq-PKBUSAZc6nr2eK5PbM
1732008114: Client mosq-PKBUSAZc6nr2eK5PbM disconnected.
1732008116: New connection from 127.0.0.1 on port 8883.
1732008116: New client connected from 127.0.0.1 as mosq-RkNEpd1FW13L3DAZPa (p2, c1, k60).
1732008116: No will message specified.
1732008116: Sending CONNACK to mosq-RkNEpd1FW13L3DAZPa (0, 0)
1732008116: Received PUBLISH from mosq-RkNEpd1FW13L3DAZPa (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008116: Sending PUBLISH to mosq-BDtPH0chr1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008116: Received DISCONNECT from mosq-RkNEpd1FW13L3DAZPa
1732008116: Client mosq-RkNEpd1FW13L3DAZPa disconnected.
1732008118: New connection from 127.0.0.1 on port 8883.
1732008118: New client connected from 127.0.0.1 as mosq-pNGrydNdMz2p8PhI57 (p2, c1, k60).
1732008118: No will message specified.
1732008118: Sending CONNACK to mosq-pNGrydNdMz2p8PhI57 (0, 0)
1732008118: Received PUBLISH from mosq-pNGrydNdMz2p8PhI57 (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008118: Sending PUBLISH to mosq-BDtPH0chr1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_temperature', ... (6 bytes))
1732008118: Received DISCONNECT from mosq-pNGrydNdMz2p8PhI57
1732008118: Client mosq-pNGrydNdMz2p8PhI57 disconnected.

```

Брокер – це центральний вузол MQTT, що забезпечує взаємодію клієнтів. Обмін даними між клієнтами відбувається лише через брокера. У ролі брокера може виступати як серверне ПЗ, так і контролер. До його завдань входить отримання даних від клієнтів, обробка та збереження даних, доставка даних клієнтам та контроль за доставкою їх повідомлень.

#### 4. Отримання даних **subscriber**, надісланих від імені **publisher** ч/з брокера:

```

nazar@mqtt:~/Desktop/certs/subscriber$ mosquitto_sub -p 8883 --cafile ../ca/ca.crt --cert subscriber.crt --key subscriber.key -h localhost -t /home/kitchen/oven_temperature
125°C
140°C
160°C
200°C
***°C

```

Роль “Subscriber” означає, що клієнт підписується на оновлення показників.

**5. Перехоплення даних між брокером та іншими учасниками спілкування за допомогою “Wireshark” для підтвердження, що з’єднання шифроване.**

FileEditViewGoCaptureAnalyzeStatisticsTelephonyWirelessToolsHelp

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.53	DNS	75	Standard query 0x5673 A mqt OPT
2	0.000012826	127.0.0.1	127.0.0.53	DNS	75	Standard query 0x0d77 AAAA mqt OPT
3	0.000197242	127.0.0.53	127.0.0.1	DNS	91	Standard query response 0x5673 A mqt A 192.168.50.176
4	0.000262962	127.0.0.53	127.0.0.1	DNS	103	Standard query response 0x0d77 AAAA mqt AAAA fe80::55
5	5.586459966	127.0.0.1	127.0.0.1	TCP	74	53842 → 8883 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SAC
6	5.586502505	127.0.0.1	127.0.0.1	TCP	74	8883 → 53842 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MS
7	5.586511758	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1
8	5.635390545	127.0.0.1	127.0.0.1	TLSPv1.3	367	Client Hello
9	5.635403586	127.0.0.1	127.0.0.1	TCP	66	8883 → 53842 [ACK] Seq=1 Ack=302 Win=65280 Len=0 TSval
10	5.660204546	127.0.0.1	127.0.0.1	TLSPv1.3	2415	Server Hello, Change Cipher Spec, Application Data, Ap
11	5.660216348	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=302 Ack=2350 Win=63744 Len=0 TS
12	5.661872014	127.0.0.1	127.0.0.1	TLSPv1.3	2193	Change Cipher Spec, Application Data, Application Data
13	5.662170729	127.0.0.1	127.0.0.1	TLSPv1.3	1137	Application Data
14	5.662174806	127.0.0.1	127.0.0.1	TLSPv1.3	125	Application Data
15	5.662235280	127.0.0.1	127.0.0.1	TLSPv1.3	1137	Application Data
16	5.700935218	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=2488 Ack=4492 Win=65536 Len=0 T
17	5.700954219	127.0.0.1	127.0.0.1	TLSPv1.3	92	Application Data
18	5.700957406	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=2488 Ack=4518 Win=65536 Len=0 T
19	5.701017272	127.0.0.1	127.0.0.1	TLSPv1.3	125	Application Data
20	5.701092456	127.0.0.1	127.0.0.1	TLSPv1.3	93	Application Data
21	5.748346315	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=2547 Ack=4545 Win=65536 Len=0 T
22	10.506807954	127.0.0.1	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipp._tcp.local, "QM" questi
23	16.982033691	127.0.0.1	127.0.0.1	TCP	74	59810 → 8883 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SAC
24	16.982061918	127.0.0.1	127.0.0.1	TCP	74	8883 → 59810 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MS
25	16.982070268	127.0.0.1	127.0.0.1	TCP	66	59810 → 8883 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1
26	17.056570548	127.0.0.1	127.0.0.1	TLSPv1.3	367	Client Hello
27	17.056663366	127.0.0.1	127.0.0.1	TCP	66	8883 → 59810 [ACK] Seq=1 Ack=302 Win=65280 Len=0 TSval
28	17.057718334	127.0.0.1	127.0.0.1	TLSPv1.3	2415	Server Hello, Change Cipher Spec, Application Data, Ap
29	17.057725216	127.0.0.1	127.0.0.1	TCP	66	59810 → 8883 [ACK] Seq=302 Ack=2350 Win=63744 Len=0 TS
30	17.059389717	127.0.0.1	127.0.0.1	TLSPv1.3	2193	Change Cipher Spec, Application Data, Application Data
31	17.059668427	127.0.0.1	127.0.0.1	TLSPv1.3	1137	Application Data
32	17.059672553	127.0.0.1	127.0.0.1	TLSPv1.3	125	Application Data
33	17.059733903	127.0.0.1	127.0.0.1	TLSPv1.3	1137	Application Data
34	17.104400676	127.0.0.1	127.0.0.1	TCP	66	59810 → 8883 [ACK] Seq=2488 Ack=4492 Win=65536 Len=0 T
35	17.104470564	127.0.0.1	127.0.0.1	TLSPv1.3	92	Application Data
36	17.104474078	127.0.0.1	127.0.0.1	TCP	66	59810 → 8883 [ACK] Seq=2488 Ack=4518 Win=65536 Len=0 T
37	17.104539521	127.0.0.1	127.0.0.1	TLSPv1.3	128	Application Data
38	17.104592530	127.0.0.1	127.0.0.1	TLSPv1.3	114	Application Data, Application Data
39	17.105024508	127.0.0.1	127.0.0.1	TLSPv1.3	128	Application Data
40	17.105030897	127.0.0.1	127.0.0.1	TCP	66	53842 → 8883 [ACK] Seq=2547 Ack=4607 Win=65536 Len=0 T
41	17.105051699	127.0.0.1	127.0.0.1	TLSPv1.3	90	Application Data
42	17.105062233	127.0.0.1	127.0.0.1	TCP	54	59810 → 8883 [RST] Seq=2599 Win=0 Len=0

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 8883, Dst Port: 53842, Seq: 4545, Ack: 2547, Len: 62

Transport Layer Security

TLSPv1.3 Record Layer: Application Data Protocol: mqtt

Opaque Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 57

Encrypted Application Data: 577fe9e23fec66f8092918986618d95024c10f2cb87604b3...

0000 0

З'являються записи протоколу **TLSv1.3**, що вказує на те, що з'єднання зашифроване. Поле “Encrypted Application Data” підтверджує, що вміст переданих даних є зашифрованим. Неможливо побачити текст повідомлення без доступу до приватного ключа. Також у записах помітно пакет RST (Reset), який свідчить про завершення сеансу цього з'єднання.

Додатково перевірено, що без шифрування йде обмін пакетами MQTT.

## 6. Відповісти на контрольні запитання.

### ▪ Як працює взаємодія “Клієнт-Брокер-Публікатор-Підписник”?

#### У чому полягають переваги використання такої схеми?

Отже, в цій схемі MQTT-клієнти, тобто публікатор і підписник, не знають про існування один одного і не взаємодіють безпосередньо. Брокер може отримувати дані з різних джерел, проводити над ними маніпуляції, наприклад, розраховувати середнє значення від кількох публікаторів і вже оброблені дані повертати підписнику.

При цьому, асинхронність протоколу MQTT передбачає, що публікатор та підписник можуть бути онлайн у різний час, втрачати пакети, і бути недоступними. Брокер подбає про те, щоб зберегти в пам'яті останні дані, отримані від датчика, та забезпечити їх доставку.

### ▪ До якого рівня моделі OSI належить MQTT?

MQTT знаходиться на прикладному рівні моделі OSI, а водночас для структури пакету він також використовує Ethernet, TCP/IP та SSL/TLS.

### ▪ Які альтернативи MQTT ви можете вказати?

**Modbus**: Організація зв'язку між цифровими пристроями в області релейного захисту і автоматики. Може використовуватись для передачі даних через послідовні лінії зв'язку RS-485, RS-422, RS-232, а також мережі TCP/IP (Modbus TCP). Розроблений для використання в програмно-логічних контролерах.

**DNP3** (Distributed Network Protocol): Підтримка роботи з подіями типу: зміна стану і подія з міткою часу. Обмін даними по мережах Ethernet і через інтерфейси RS232/RS485. Використання при передачі повідомлення великого розміру. Широкомовна розсилка повідомлень. Віддалене конфігурування програмно-логічних контролерів.

- **Які основні кроки слід зробити, щоб налаштувати криптографічно захищену передачу інформації?**

- 1) *Вибір криптографічного алгоритму та методу шифрування* – визначення типу шифрування й алгоритму, які будуть використані.
- 2) *Генерація криптографічних ключів та сертифікатів* – після вибору алгоритмів потрібно згенерувати відповідні ключі та сертифікати для підтвердження автентичності кінцевого пристрою.
- 3) *Налаштування криптографічного захисту на пристроях* – після цього налаштовуються пристрої для використання обраних алгоритмів, що включає конфігурацію криптографічних модулів, налаштування для захищеного каналу для шифрованого зв'язку.

- **Для чого в роботі використовуються TCPdump та Wireshark?**

**TCPdump** і **Wireshark** є інструментами для перехоплення і аналізу мережевого трафіку. Вони дозволяють спостерігати, як дані передаються через мережу, що важливо для діагностики та моніторингу з'єднань.

Для з'єднань, що використовують шифрування (SSL/TLS), можна було побачити, що комунікації здійснюються через порт **8883**, який є стандартним для **MQTT(S)**. Проаналізувавши більш детально пакети, було підтверджено наявність TLSv1.3-пакетів, тому всі дані були зашифровані.