Secure Communication over MQTT

Ahmet Onat
2018
onat@sabanciuniv.edu

Why Security?

- “Our data does not have commercial value”
- “There is no incentive for hackers to attack our systems”
- “I don’t bank online, I don’t store sensitive information on my machine! I only use it to check email → What could hackers possibly want from this machine?”
- A compromized system has value as:
  - Zombie (spam, DoS, CAPTCHA solver)
  - Server (phishing, malware, forbidden/explicit content)
  - Credentials (e-mail, accounts, banking, Twitter, Skype)
  - Many more...

Why Security?

- Modern attacks are auto coordinated.
- No distinction is made about the qualification of victim.

“Mirai” DDoS Attack from IoT Devices(2016)

- DDoS attack of 620Gbps (record volume at the time)
- Originating from:
  - IP security cameras
  - DVRs
  - Routers, set top box etc.
- All had inferior protection:
  - Open telnet ports,
  - Default / weak passwords
  - etc.
- Current record: 1.3Tbps (2x BluRay / sec)
- Search: Wikipedia “Mirai (malware)”, “Linux.Darlloz” etc.

Threats to MQTT Communication

- Compromised devices (key / password / certificate theft)
- Data in clients and brokers become accessible
- Comms: Intercepted, altered, re-routed or disclosed
- Injection of spoofed control packets, false packets
- Denial of Service (DoS) attack bot

How to Prevent

- Authentication of devices (and users)
- Authorization of access to server resources
- Control and payload packets:
  - Intergity
  - Privacy
- Transport encryption
- Payload encryption
- Physical protection (read protect, mechanical, etc.)

Security Features of MQTT

- MQTT is only a message transport protocol.
- Only basic security:
  Username/Password
  (sent without encryption...)

- User must provide message authentication/encryption.

- Transport Level Security (TLS)
  is the most common security method.

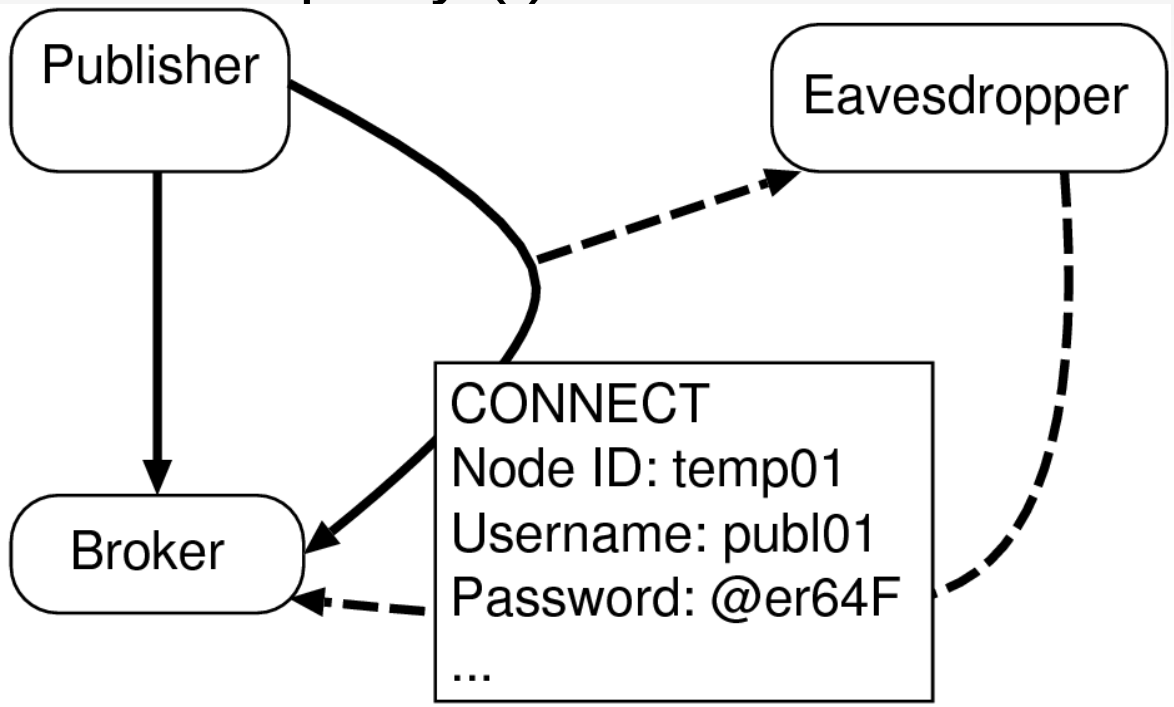
Authentication / Encryption

- Authentication: Proof that the content
  - Comes from the original source and
  - Was unaltered.

- Encryption: The content cannot be viwed by others during transport.

Simple Authentication

- Username and password sent by the client and authenticated by broker.
- Credentials sent openly (!)



- Transport protocol encryption is needed.

Public Key Cryptography

- Traditionally keys are “symmetric”
  - The key to lock, also unlocks.
- Client can encrypt a message and send, but a copy of the key must be delivered.
  → How can a message be encrypted without delivering the key?

- Public key cryptography.
  - The lock and unlock keys are different.

Distributed Decision of a Secret Key

- Use modulo functions. $f(b,x)=b^x \mid p=y$
- Where b, p are constants.

- Use the equality: $f(b,k)=l,\ \ f(b,m)=n$

$$f(n,k)=f(l,m)=z$$

Distributed Decision of a Secret Key

- Public information: $b=3,\ p=17$
- Nodes A, B each generate a random number, hash and exchange publicly.
- Each can generate the same crypto key.
- Eavesdroppers cannot compute 10 from $3,17,7,10$.

	A	B
Random	43	51
Hashed	$f(3,43)=7$	$f(3,51)=10$
Secret	$f(10,43)=3$	$f(7,51)=3$

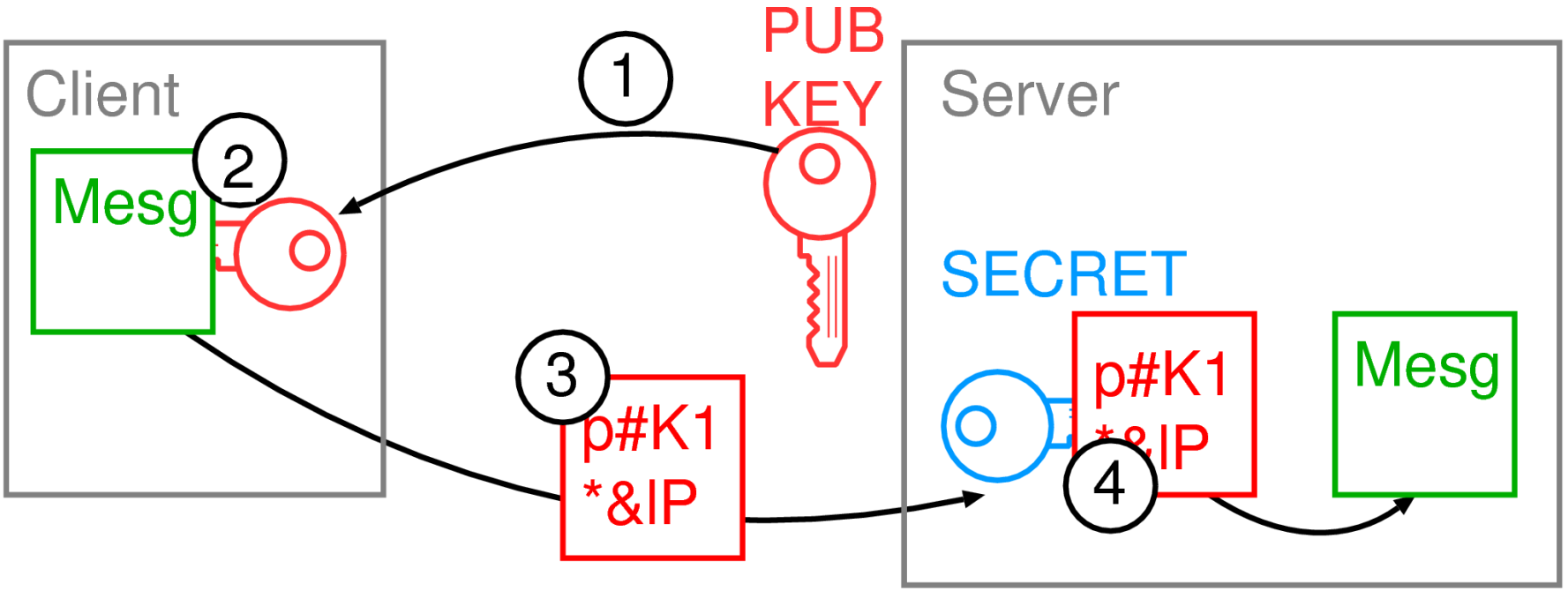
Public Key Cryptography

- Each lock has complementary keys.
- If one is used to lock, the other must be used to unlock.
- One key is guarded: Secret key
- The other key is publicly disclosed: Public key

- 3rd parties can lock with the public key→
  Only key owner can open with guarded key: Secrecy
- Owner can lock with secret key →
  3rd parties can unlock: Authentication

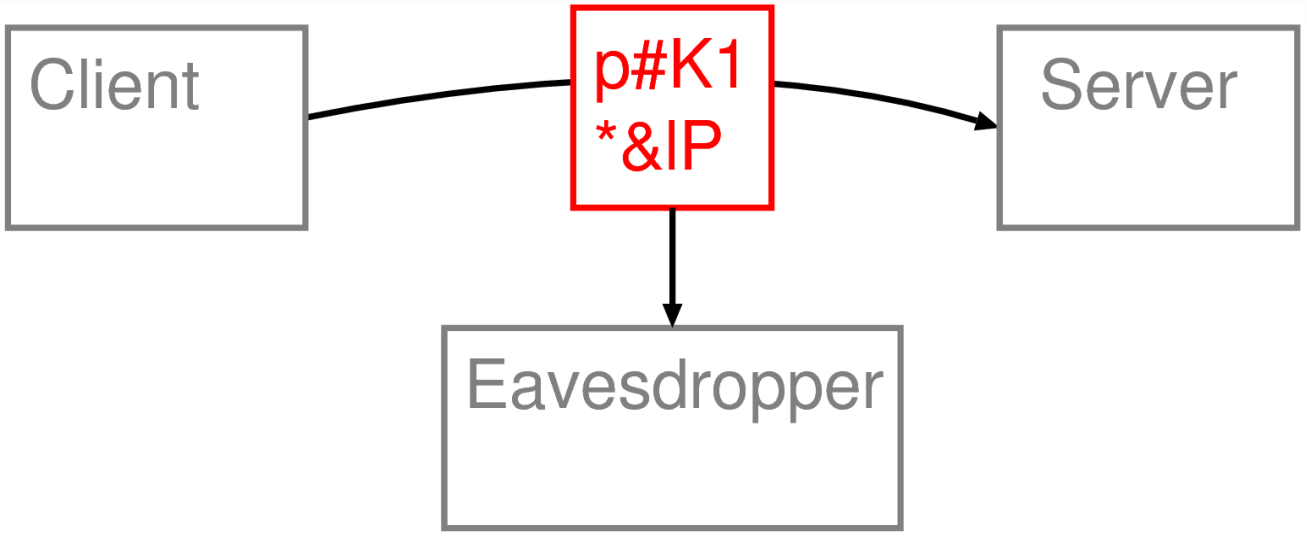
Public Key Cryptography

1. Client retrieves openly announced public key.
2. Client encrypts the message with the public key.
3. Message is transported in public network.
4. Server decrypts the message using secret key.



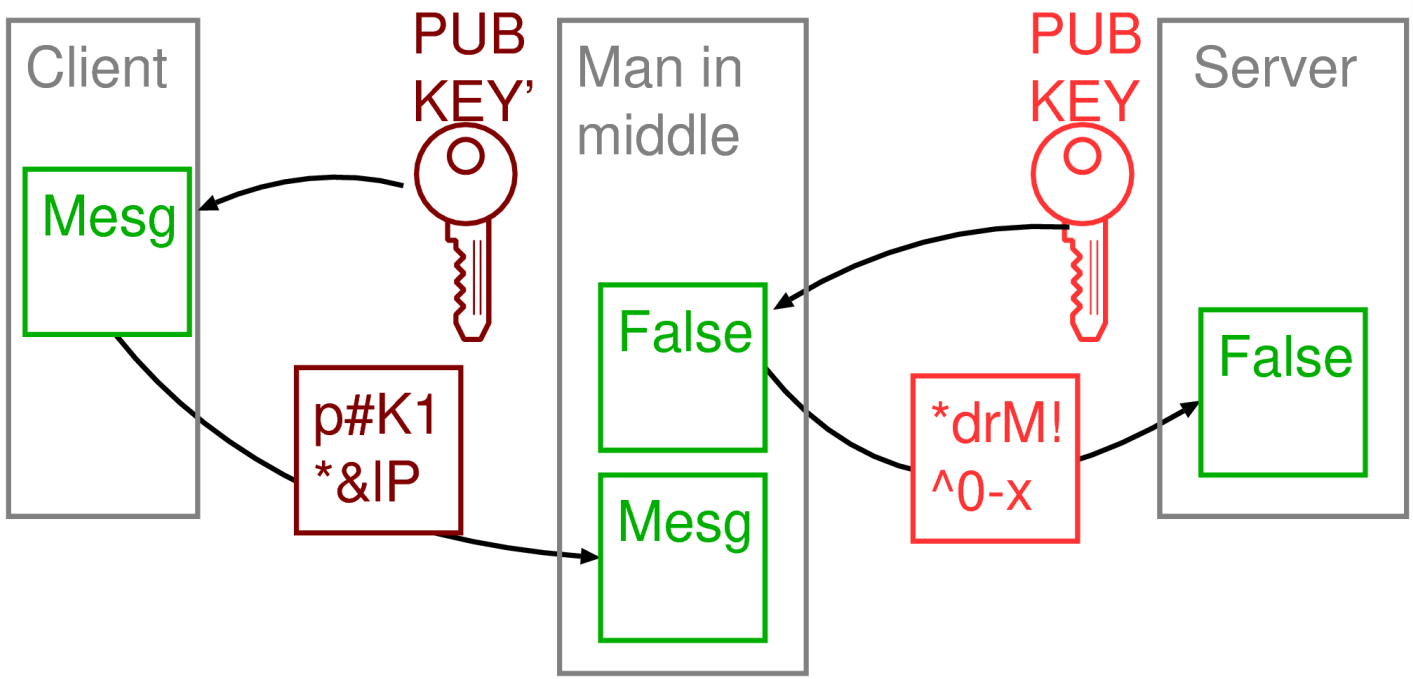
Encryption

- It can protect from eavesdroppers →
- Encryption by public key can only be decrypted using the secret key.



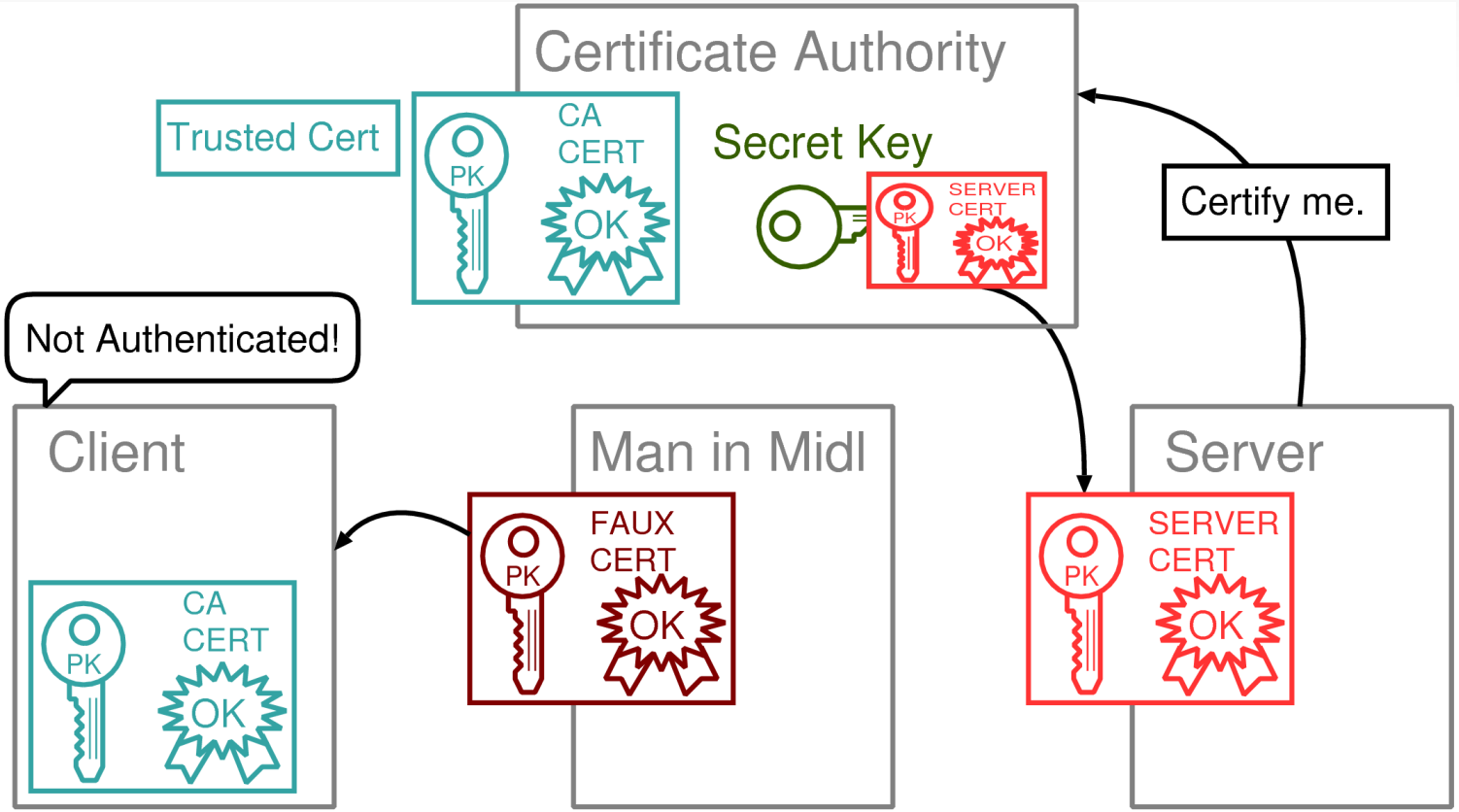
Encryption

- How about man-in-the-middle attacks?



Authorized Certificates

- Certificate Authority (CA) is a legal entity.
- Approves legitimacy of server.
- Encrypts server certificate with “holy key”



TLS Messaging Mechanism

- Transpor Layer Security:
  - Messages are encrypted
  - Authentication is performed.

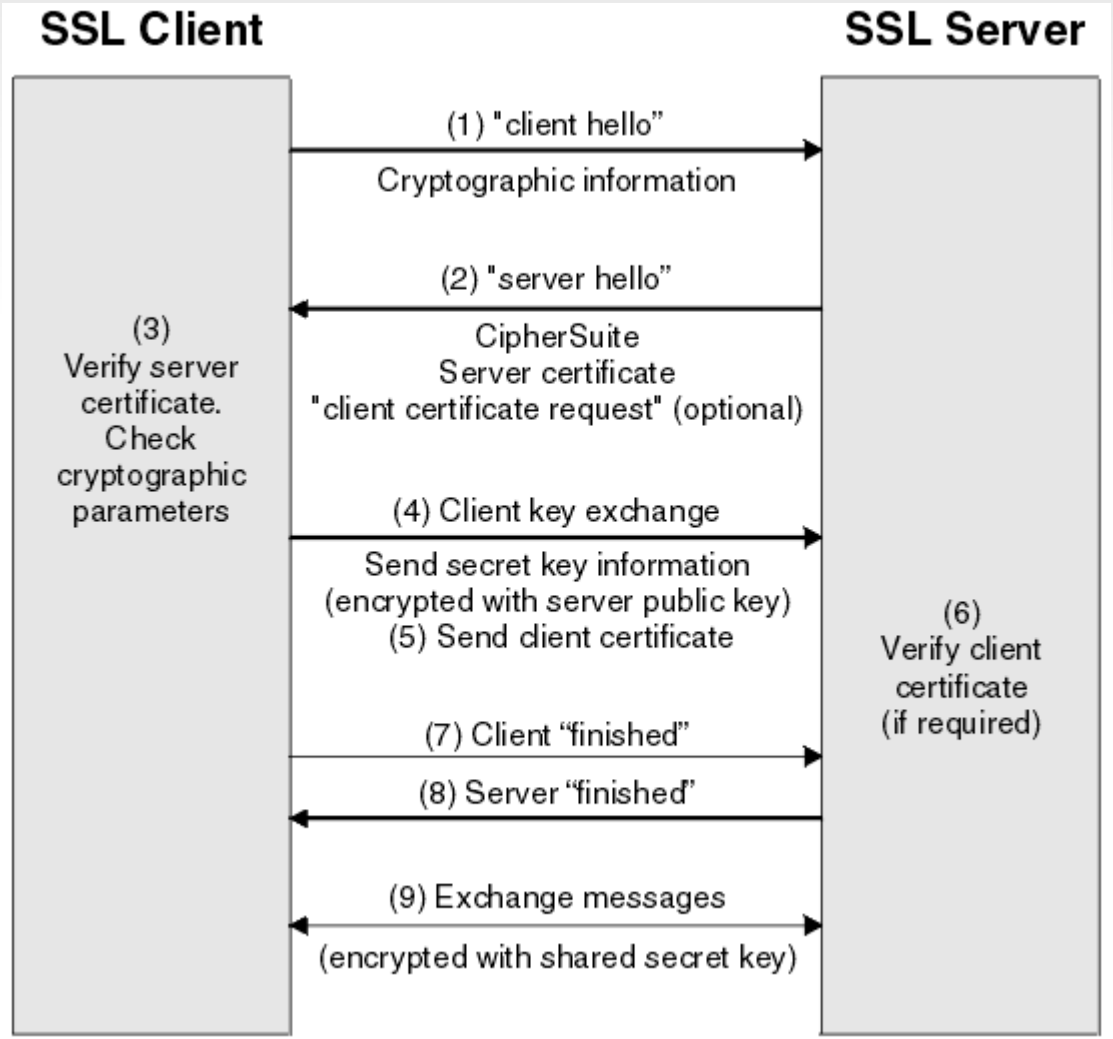
- It is possible both to:
  - Ensure authenticity
  - Prevent content theft

TLS Messaging Mechanism

- Handshake: Agree on crypto algorithms.
- Authenticate; each other by digital certificates.
- Generate; shared secret key using asymmetric encryption specifically for this session.
- Further communications are encrypted with shared secret key.

TLS Messaging Mechanism

- Transport Layer Security



TLS Messaging Essentials

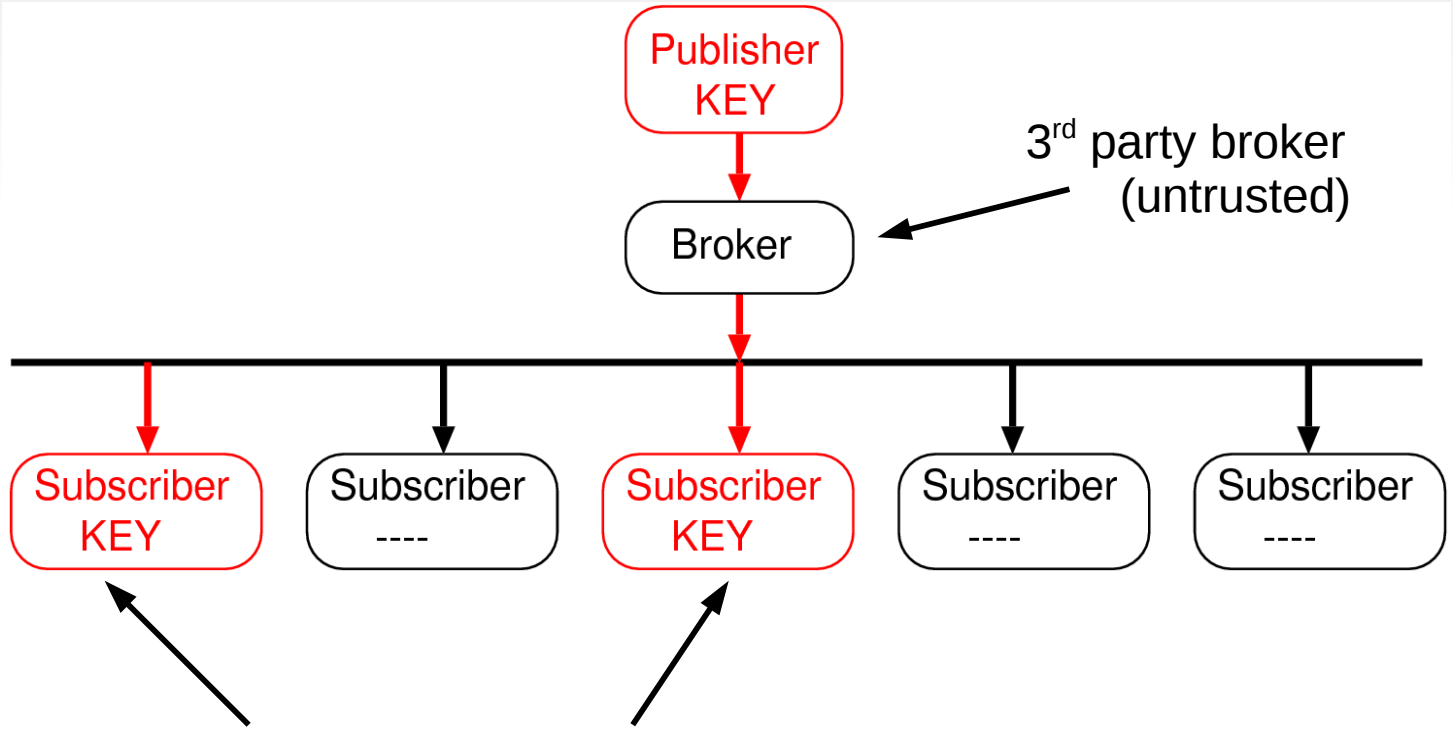
- The following information must be generated:
- Certificate Authority (CA) certificate (X509) (provides public key of authority.) ca.crt
- Server certificate (signed by the CA secret key) (Authentication of the server.) server.crt
- Server key server.key (For encrypting server messages)

Payload Encryption Only

- Payload data is encrypted at publisher.
- Decrypt at broker OR,
- Decrypt at subscriber.
- Meta-data is intact NOT encrypted:
  → topic, password, username (routing, QoS etc.)
- Payload authenticity can be verified.
- Broker cannot access payload.
- Only qualified subscribers may access payload.
- How to secure the encryption keys?

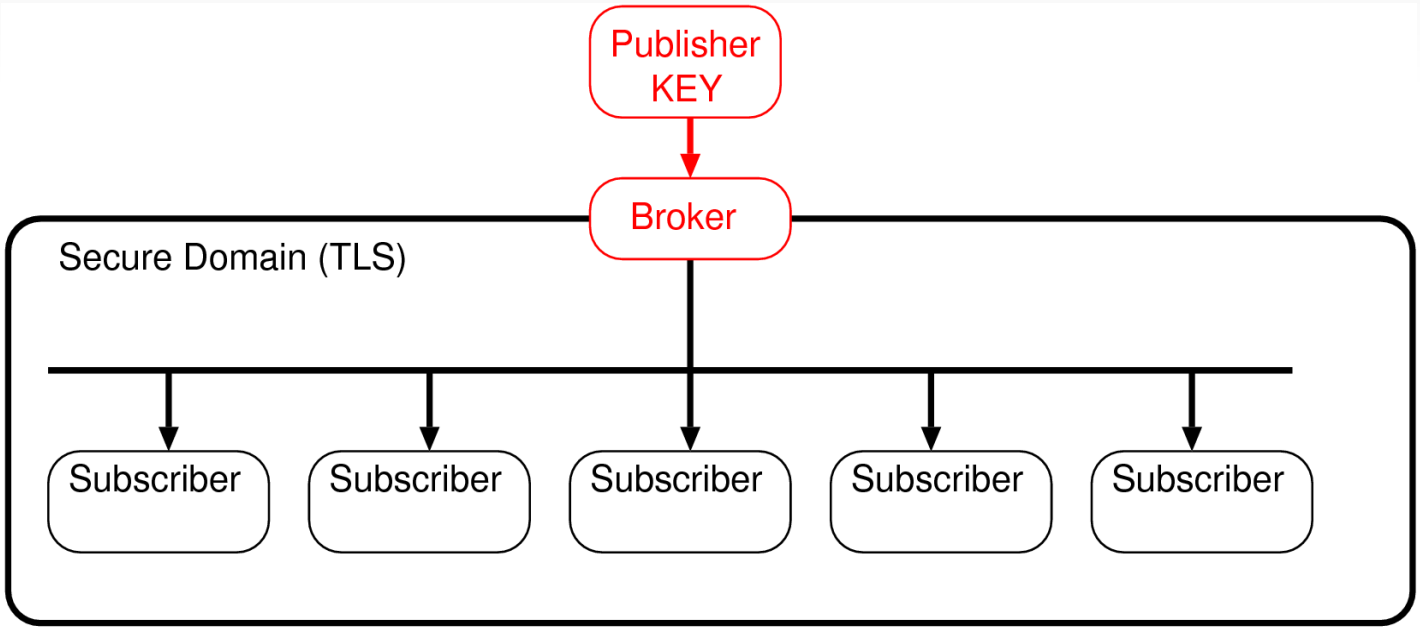
Payload Encryption: End to End

- Qualified subscribers can decrypt.
- How to keep keys safe?



Payload Encryption: Publisher to Broker

- Encryption publisher → broker
- Simpler key management: Only publishers and broker



MQTT TLS example - Certificates

- We will use “openssl” package.
- CA certificate. (We will sign our own certificates)
  openssl req -new -x509 -days 1000 -extensions v3_ca -keyout ca.key -out ca.crt
- Server key: openssl genrsa -out server.key 2048
- Server certificate request: openssl req -out server.csr -key server.key -new
- Sign server certificate: openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 1000
- The required files are highlighted.
- “ca.crt” must reside in the client to authenticate server.
- “server.crt”, “server.key” must reside in the server.
- Communication will be encrypted.

MQTT TLS example – Broker Setup

- Start broker:
  mosquitto -c mosquitto.conf :

mosquitto.conf:
port 8883
cafile /etc/mosquitto/ca_certificates/ca.crt
certfile /etc/mosquitto/certs/server.crt
keyfile /etc/mosquitto/certs/server.key

- Subscribe:
  mosquitto_sub -h 192.168.142.84 -t house --cafile ca.crt
  -p 8883 --tls-version tlsv1
- Publish:
  mosquitto_pub -h 192.168.142.84 -t house --cafile ca.crt -m "testing"
  -p 8883

- Unencrypted subscribe/publish requests are not honored.

Physical protection

- System is deployed at some remote location, open to the elements.
- By definition, all of the information required to achieve communication is within the device.
- The information must be physically protected using code locking, physical protection etc.

Contact Information

Ahmet Onat

- Sabanci University, Istanbul, Turkey
- Mail: onat@sabanciuniv.edu
- Web: http://people.sabanciuniv.edu/onat

Kerbs security DDoS attack URL

- https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/