Burglars' IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds

Yan Jia, Luyi Xing,
Yuhang Mao, Dongfang Zhao,
XiaoFeng Wang, Shangru Zhao, and Yuqing Zhang

School of Cyber Engineering, Xidian University, China
National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences ,China
Indiana University Bloomington, USA

西安电子科技大学
XIDIAN UNIVERSITY

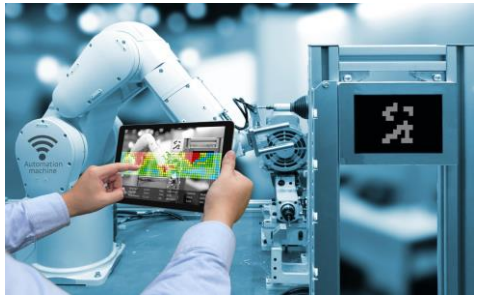
中国科学院大学
University of Chinese Academy of Sciences

INDIANA UNIVERSITY
BLOOMINGTON

Content

- Background of IoT
  - How IoT devices communicate with the cloud and mobile phones
  - Messaging protocol
- New vulnerabilities/attacks
- Measurement study of the attack impacts
- Mitigation
- Lessons

Internet of Things (IoT)



IEEE S&P

IoT Cloud

Lock the door

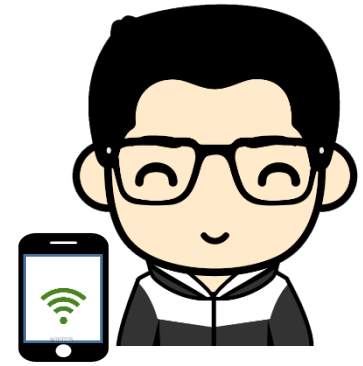
The door is locked

Lock the door

The door is locked

IoT Device

1. Register
2. Bind
3. Control
4. unbind



User

IoT Cloud

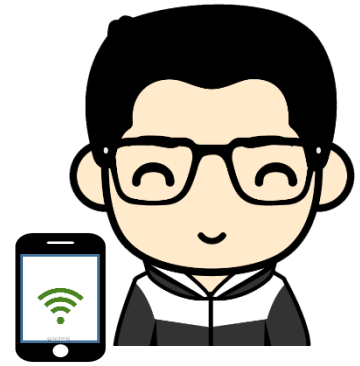
Lock the door

The door is locked

Lock the door

The door is locked

1. Register
2. Bind
3. Control
4. unbind



IoT Device

User

Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms[C]//USENIX Security 19

IEEE S&P

IoT Cloud Platforms

Watson IoT Platform | IBM

Alibaba Cloud

IoT Hub | Microsoft Azure

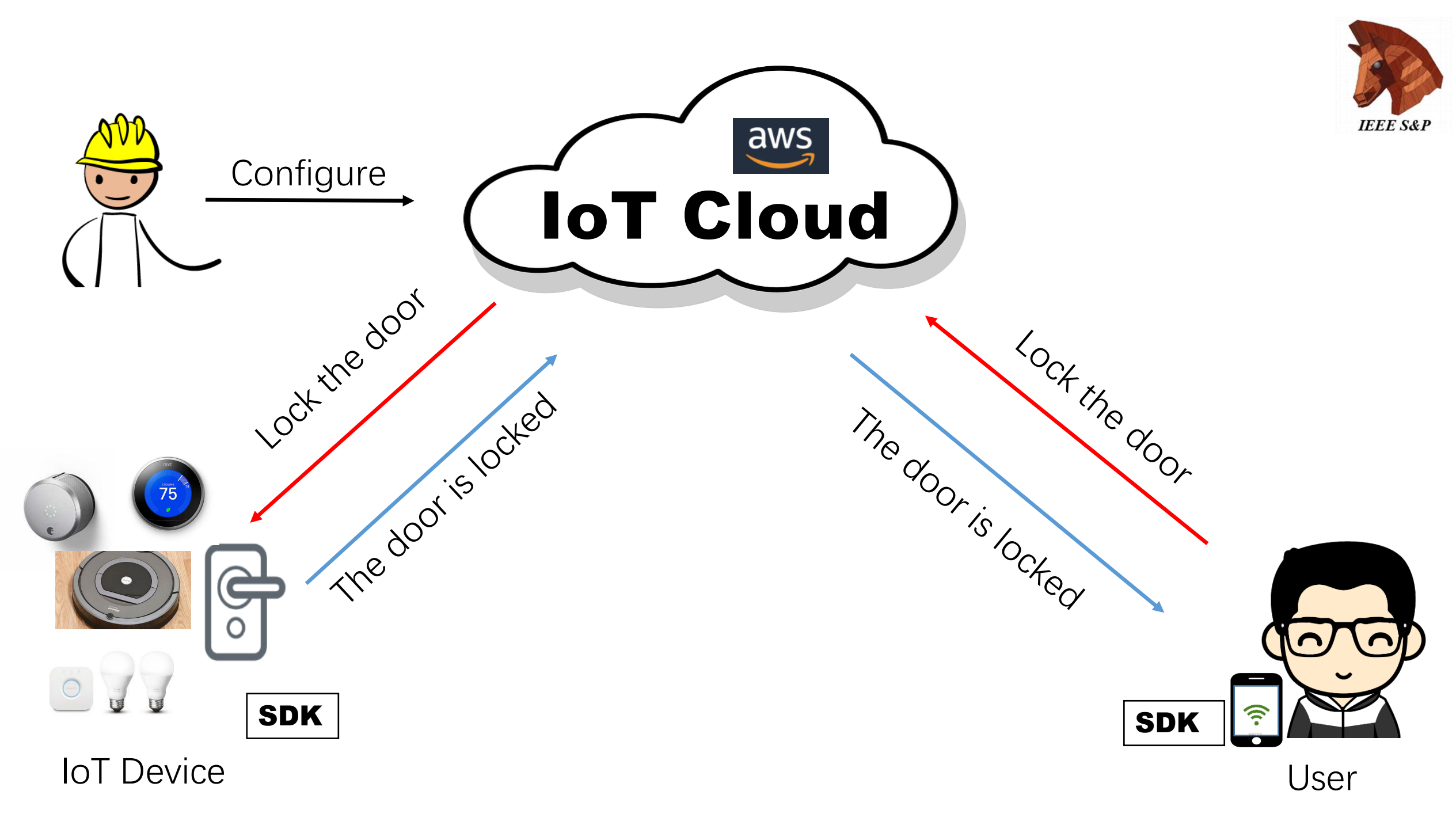
SUNING 苏宁易购

tuya.com

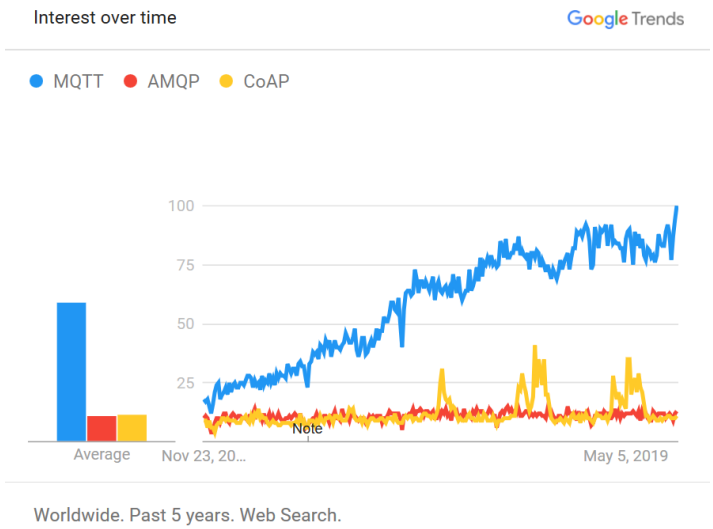
Cloud IoT Core | Google

AWS IoT Core

BAIDU AI CLOUD


IEEE S&P
Configure
aws
IoT Cloud
Lock the door
The door is locked
Lock the door
The door is locked
SDK
SDK
IoT Device
User

HTTP, CoAP



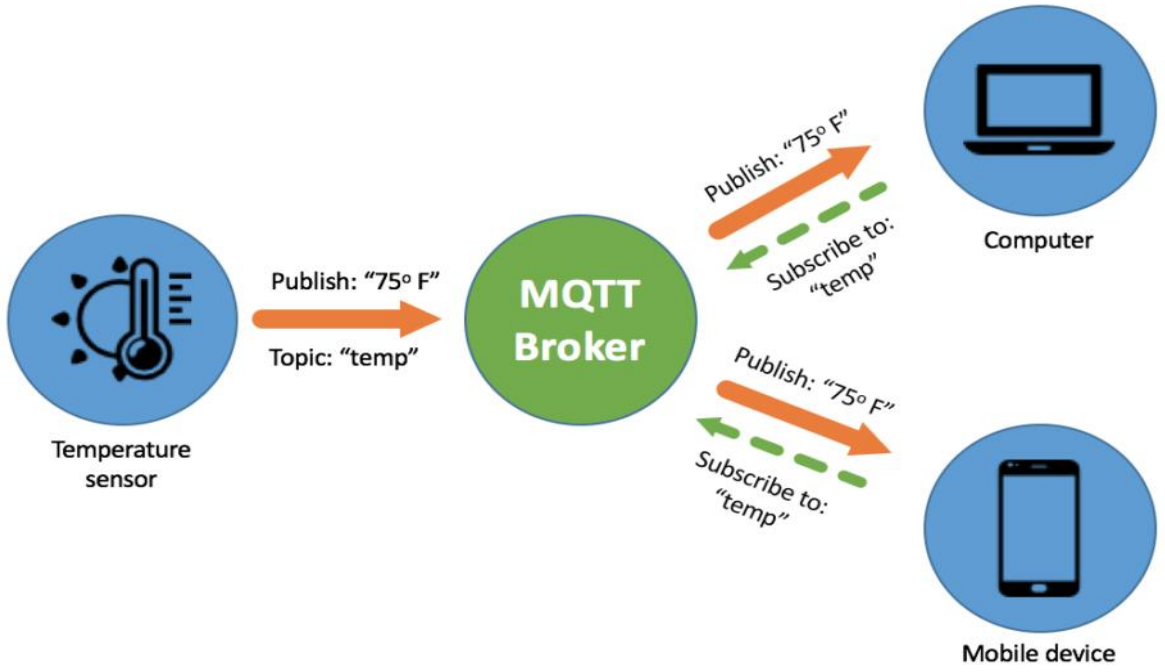
MQTT

AMQP

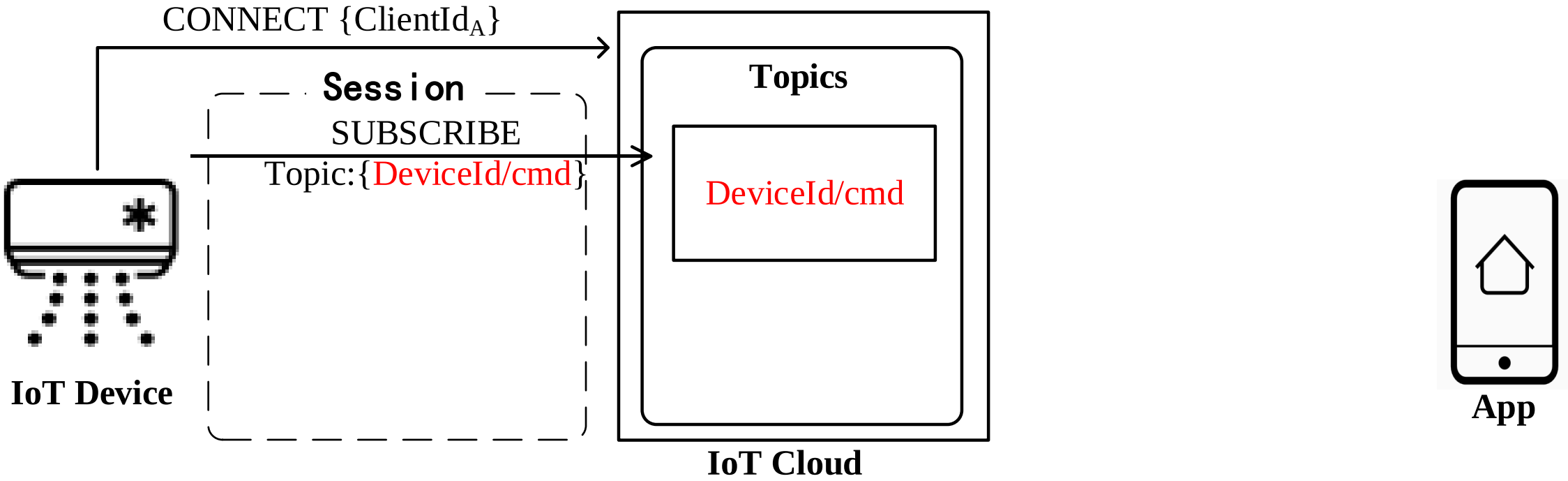
let's use MQTT as the messaging protocol

Message Queuing Telemetry Transport (MQTT)

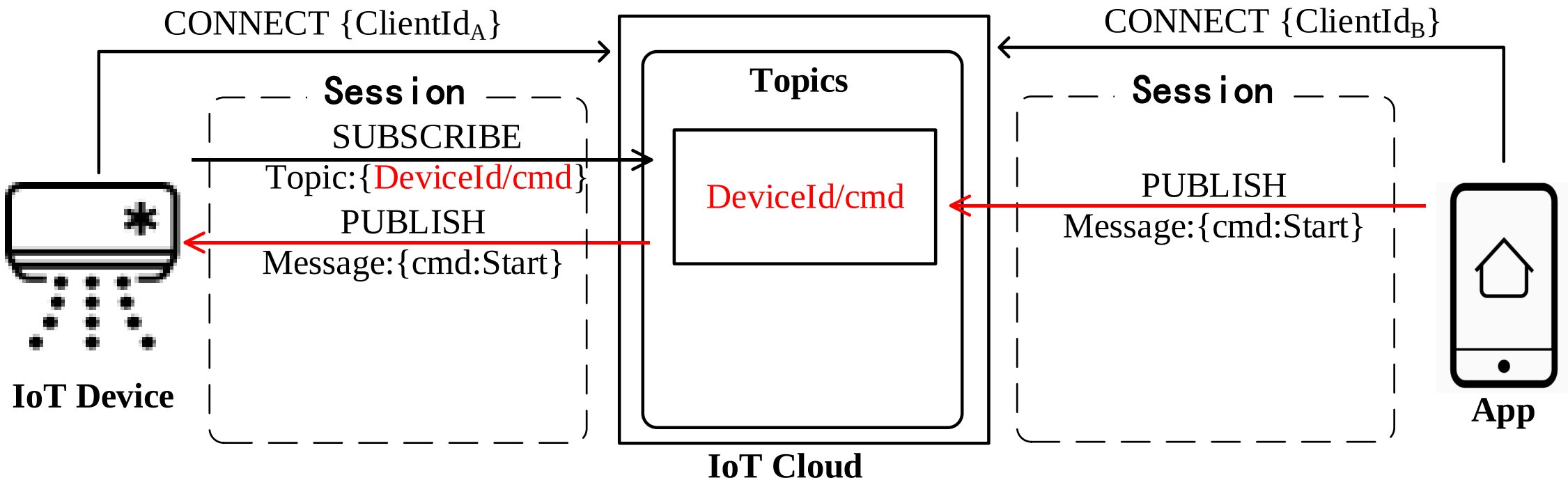
- Lightweight
- Publish-subscribe
- Over TCP/IP, Websocket



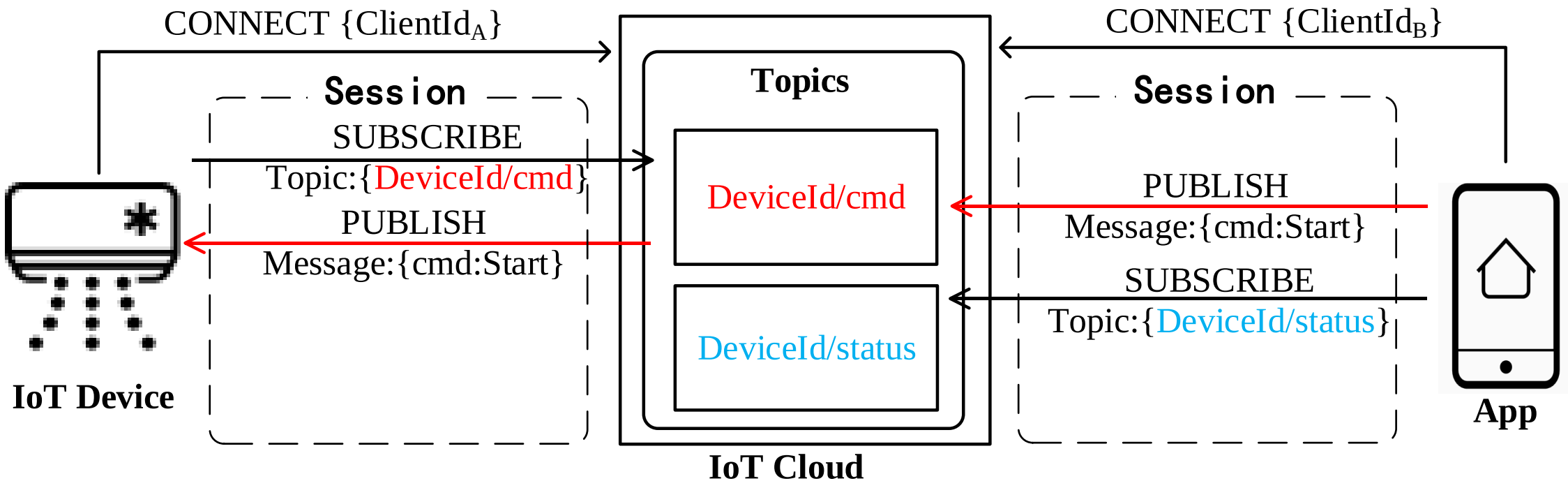
How MQTT works



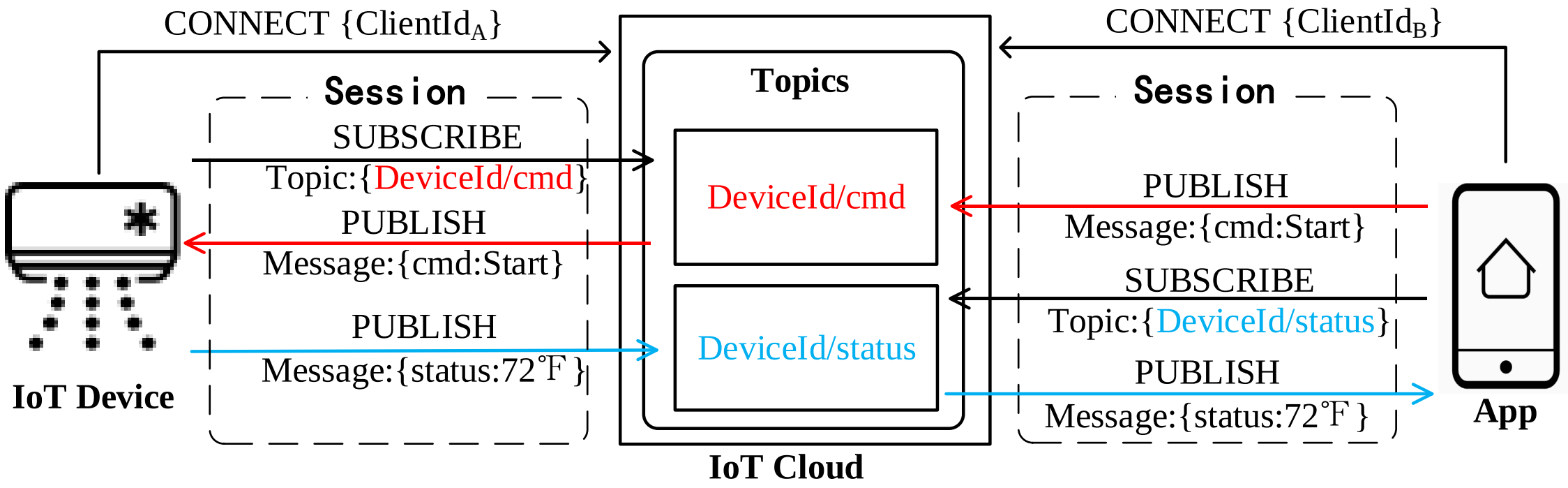
How MQTT works



How MQTT works



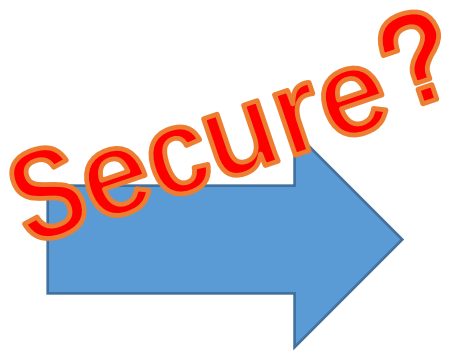
How MQTT works



Message Queuing Telemetry Transport (MQTT)

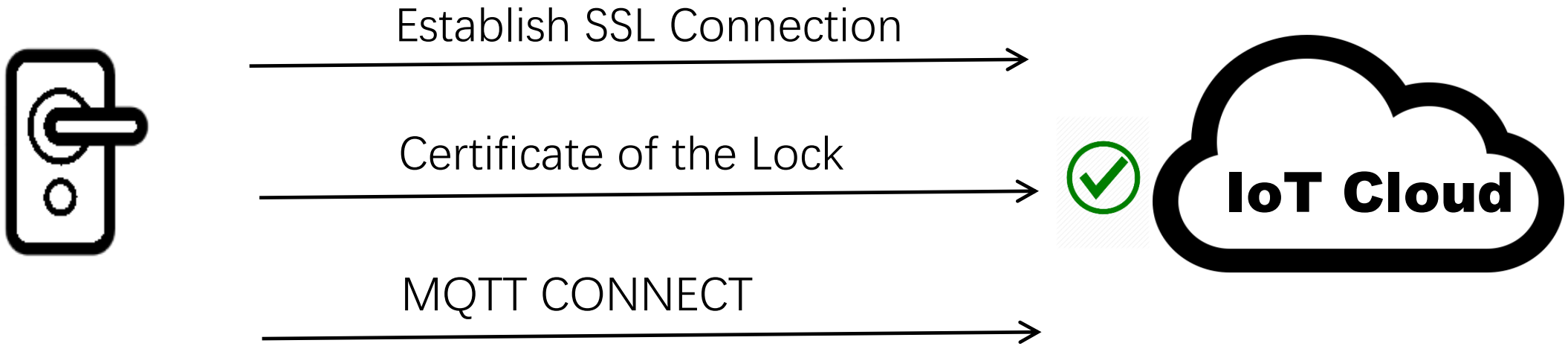
- It was created in 1990's and used to monitor proprietary oil pipelines through the desert, and communication with satellites.

WIKIPEDIA



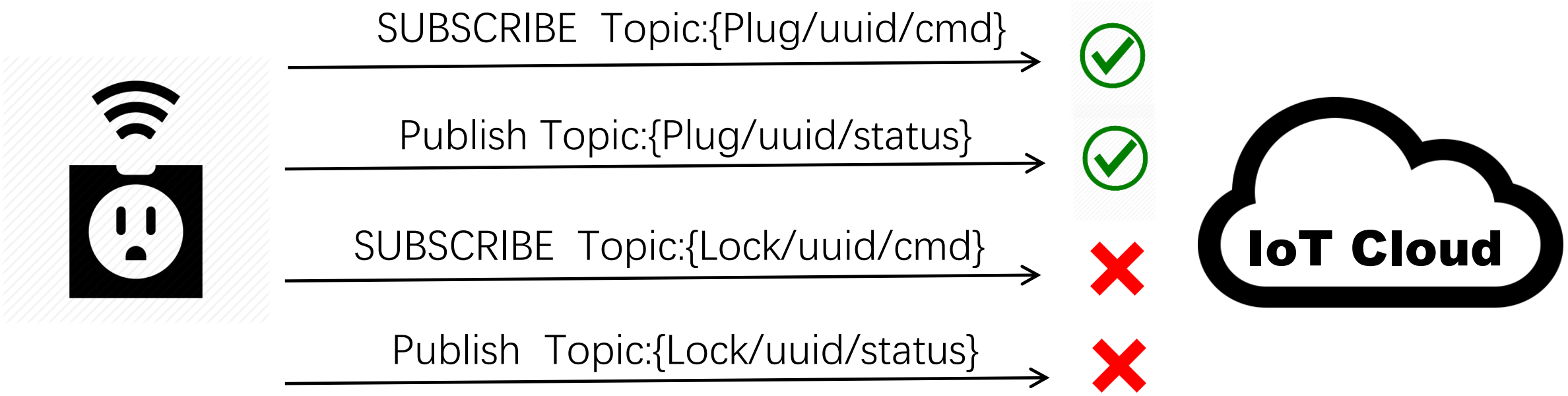
Protection of MQTT on IoT Clouds

- Authentication
  - X.509 Client Certificates
  - Username/Password in MQTT Connect
  - Other Identities (e.g., Amazon Cognito)



Protection of MQTT on IoT Clouds

- Authorization
  - SUBSCRIBE
  - PUBLISH



Is MQTT secured in the wild ?

Alibaba Cloud | Worldwide Cloud Services Partner

tuya.com

IoT Hub | Microsoft Azure

BAIDU AI CLOUD

AWS IoT Core

Cloud IoT Core | Google

SUNING 苏宁易购

Watson IoT Platform | IBM

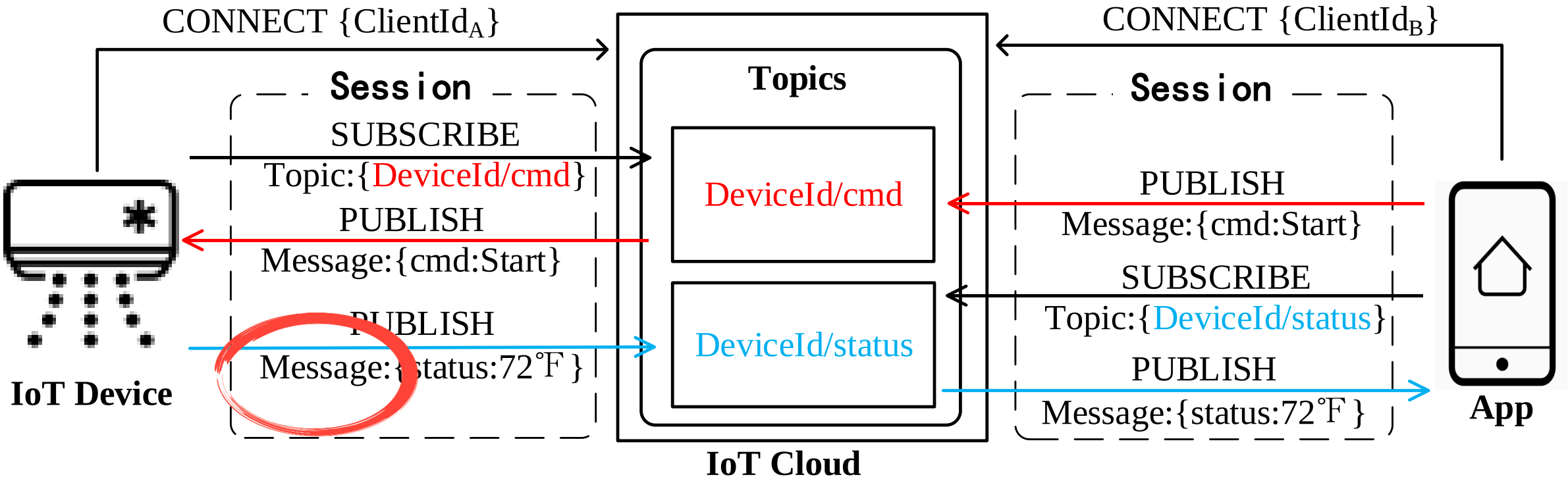
mosquitto

Threat Model

- The adversary can register user accounts with IoT device manufacturers and IoT clouds. He can analyze network traffic between the IoT cloud, the IoT device and the app under his control.
- He cannot eavesdrop on the communication of other users' devices and apps.
- We consider the device-sharing situation that becomes pervasive today. Hotels, Airbnb, apartments and other vacation rental homes are increasingly equipped with IoT devices and their guests are routinely granted temporary access to the devices.

Attack #1

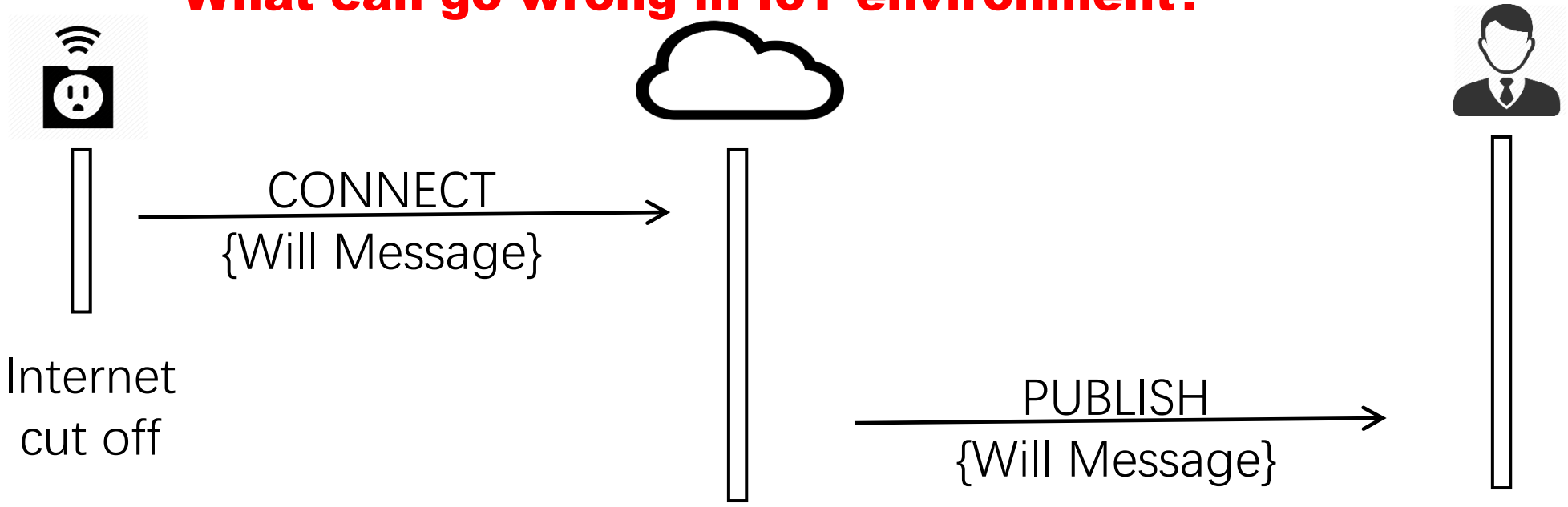
Unauthorized MQTT Messages



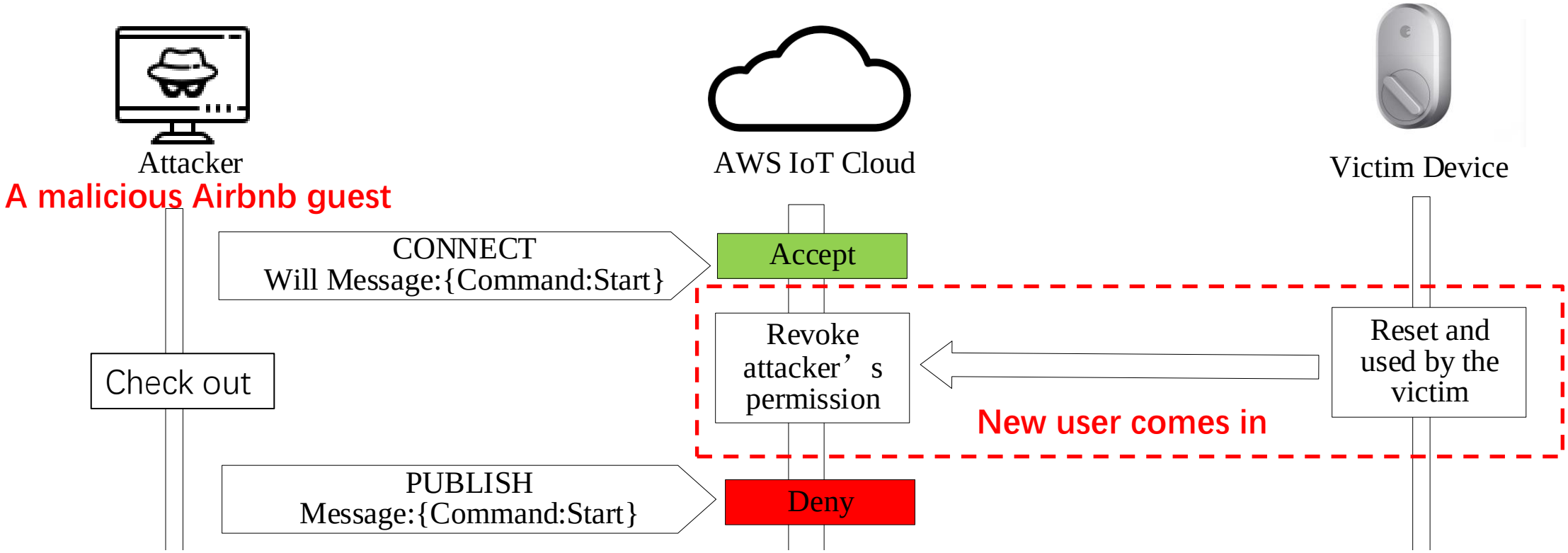
Will Message

- A kind of MQTT message, an exception handling feature
- Carries topics and payload (commands, texts)
- Published by the server when client disconnects accidentally

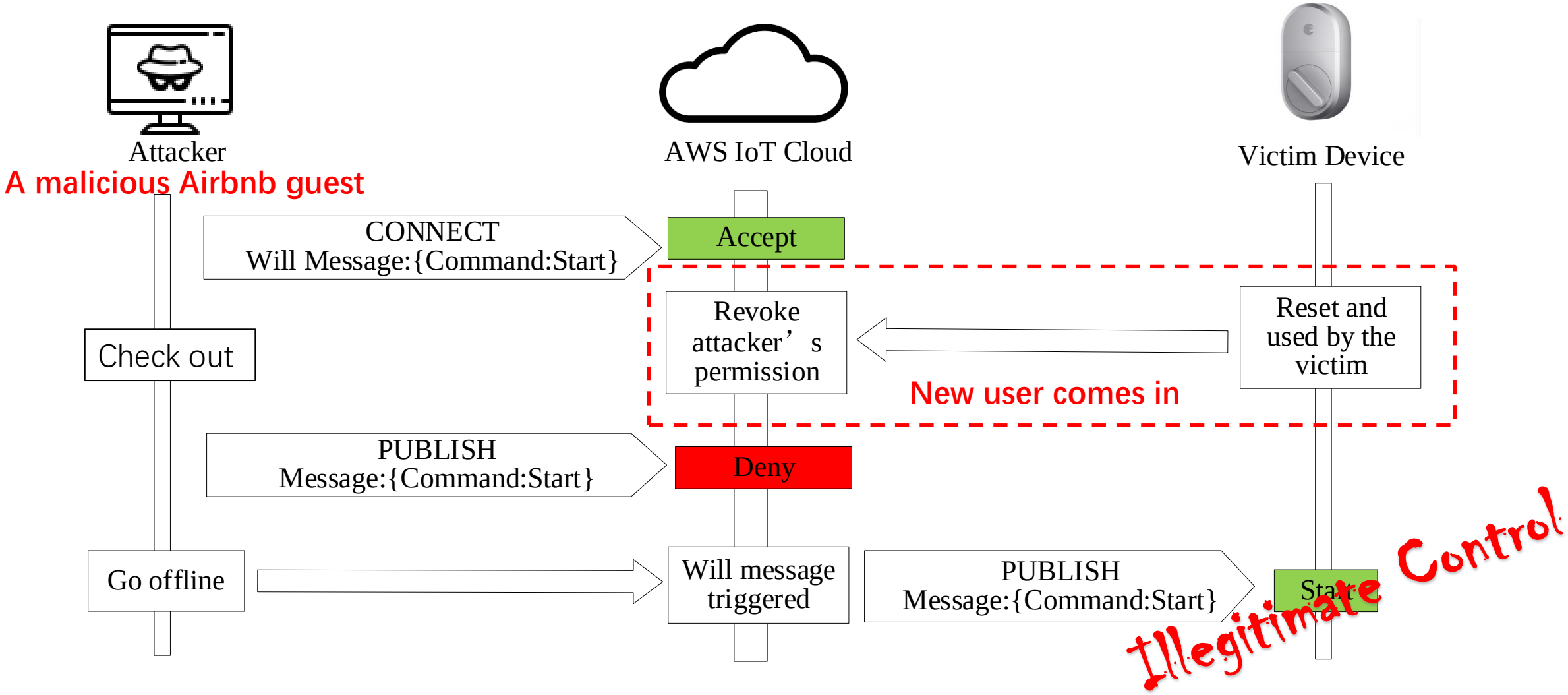
What can go wrong in IoT environment?



Unauthorized Will Message



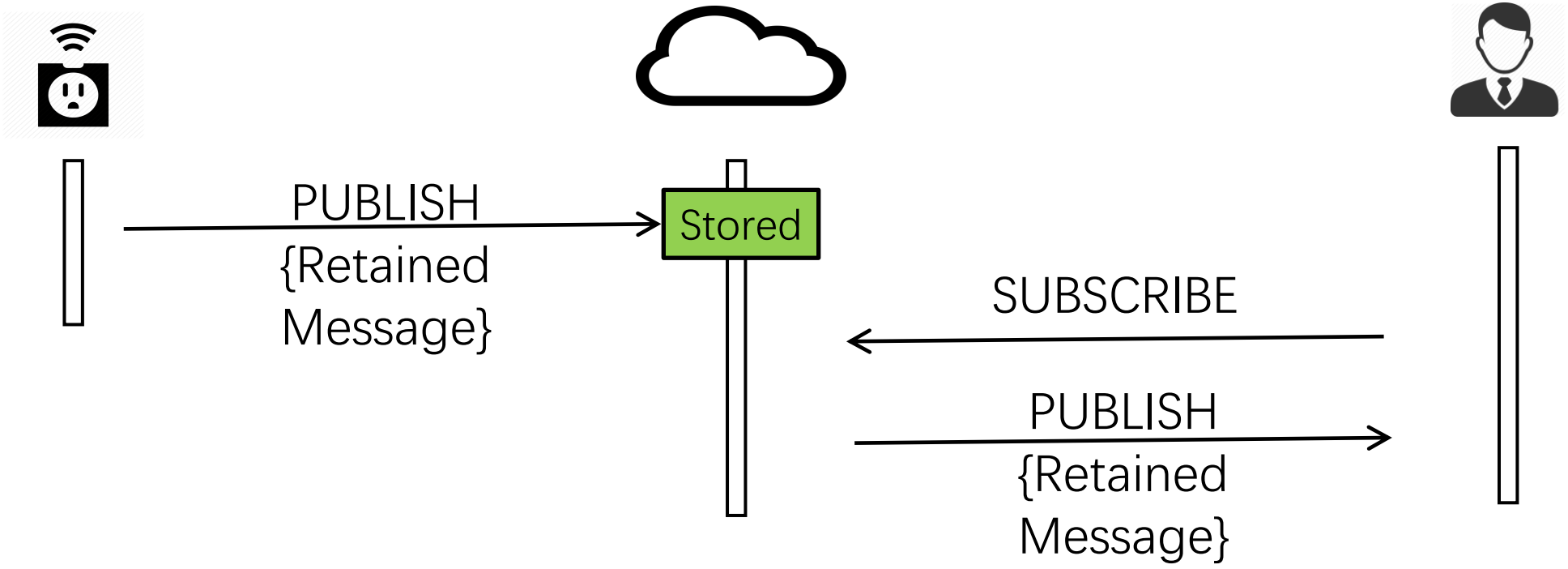
Unauthorized Will Message



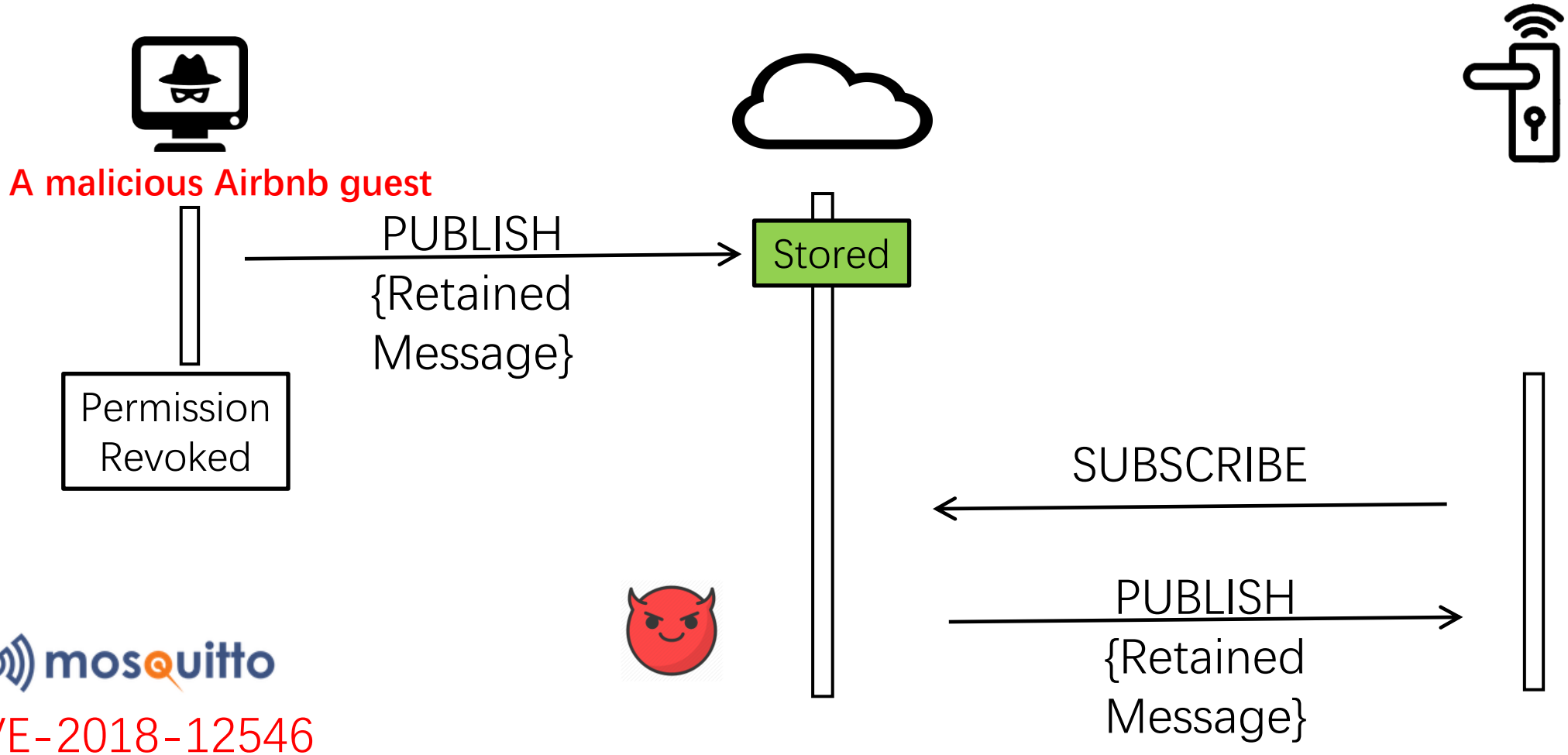
Will Message Attack Video Demo

Retained Message

- Designed to address a problem: when publishing a message, all subscribed clients are offline.
- What can go wrong in IoT?



Retained Message Attack



mosquitto

CVE-2018-12546

Why the problem happened?

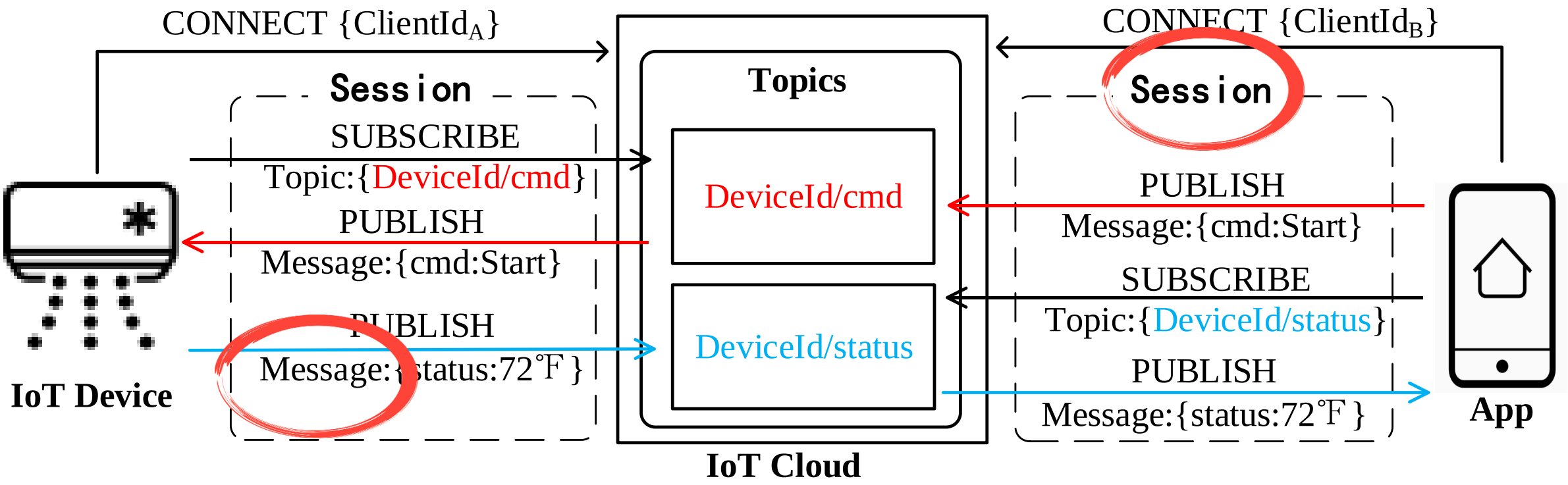
- Will and Retained messages are exception handling features not meant to work in the adversarial IoT environment, where the access right to a device can be transferred from one person to another
  - “The Will message is accepted at the time it is set. That act of acceptance grants the permission for it to be delivered at a later time. The client is out of the picture. “

Comments from MQTT TC

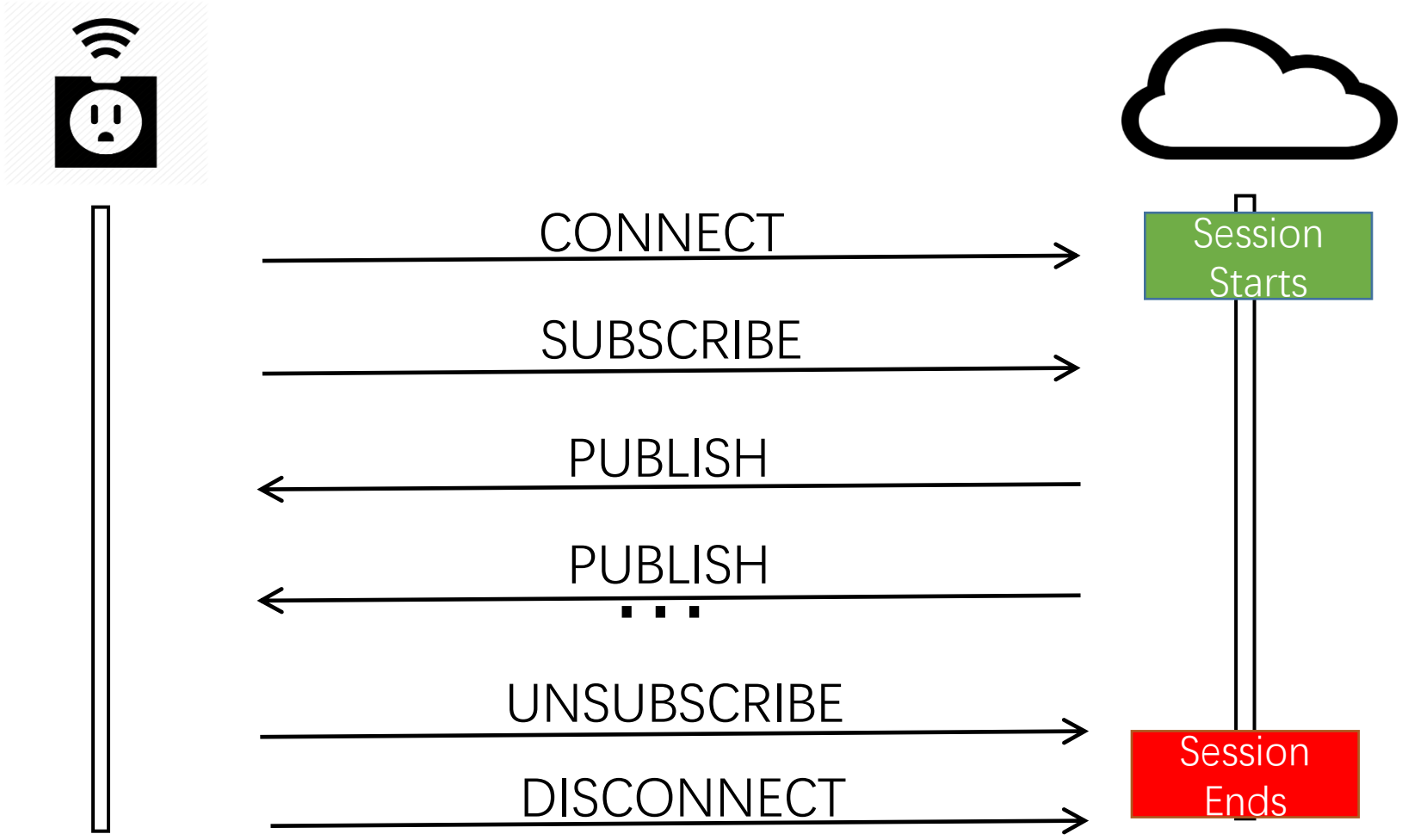
- Open discussion
  - [OASIS Open Issues MQTT-536]
  - mqtt-comment@lists.oasis-open.org

Attack #2

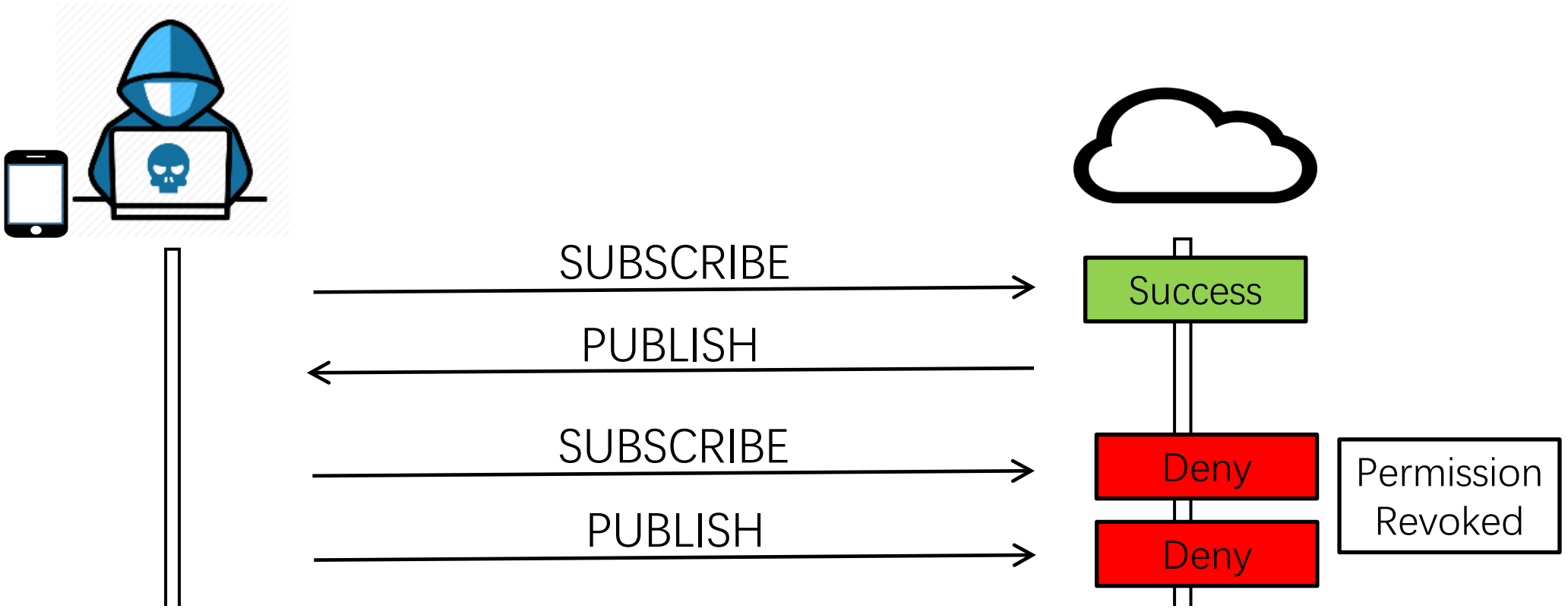
Faults in Managing MQTT Sessions



MQTT Session



Non-updated session subscription state



Non-updated session subscription state

SUBSCRIBE

PUBLISH

SUBSCRIBE

PUBLISH

PUBLISH

PUBLISH

. . .

Deny

Deny

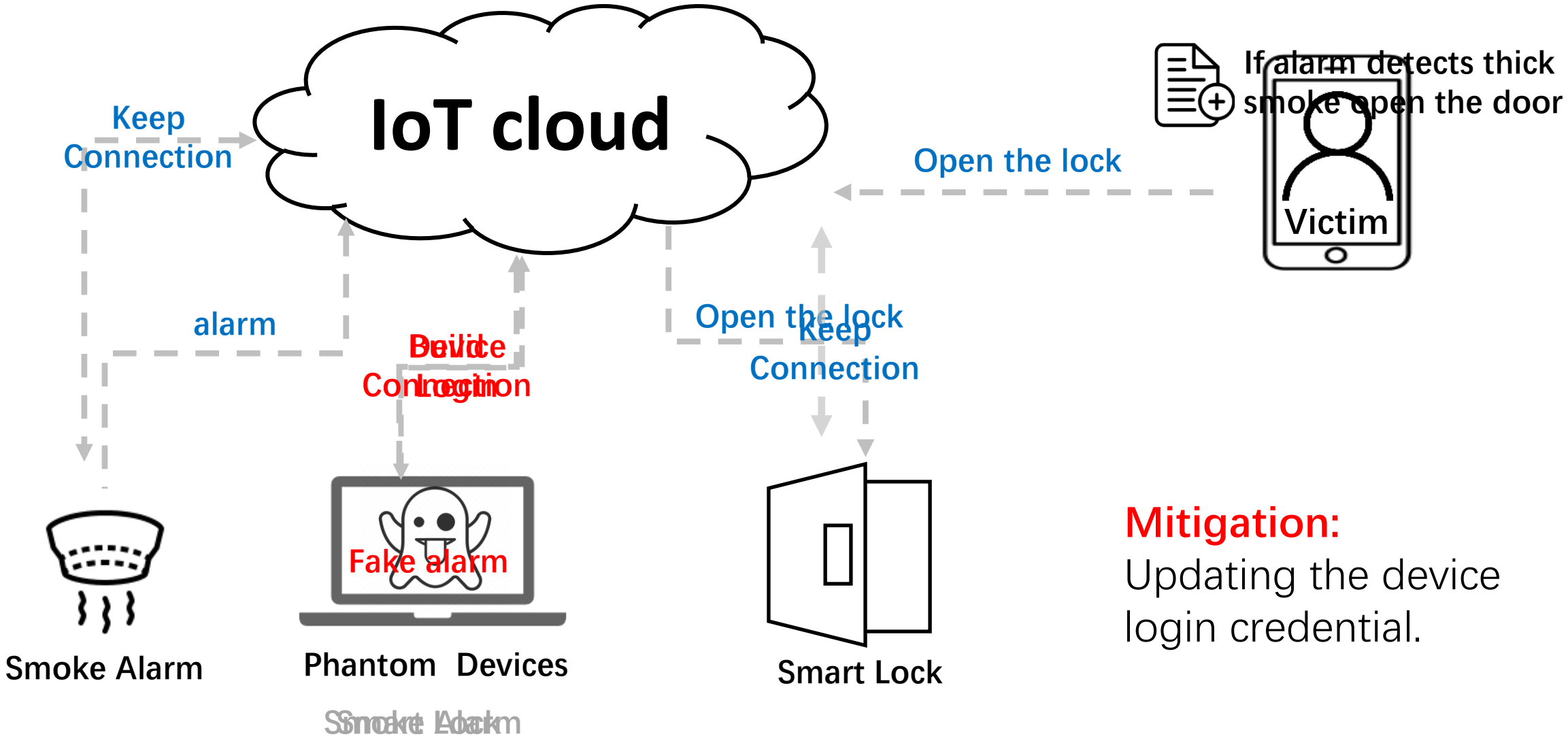
Permission Revoked

Sensitive Information Leakage

Never Unsubscribe

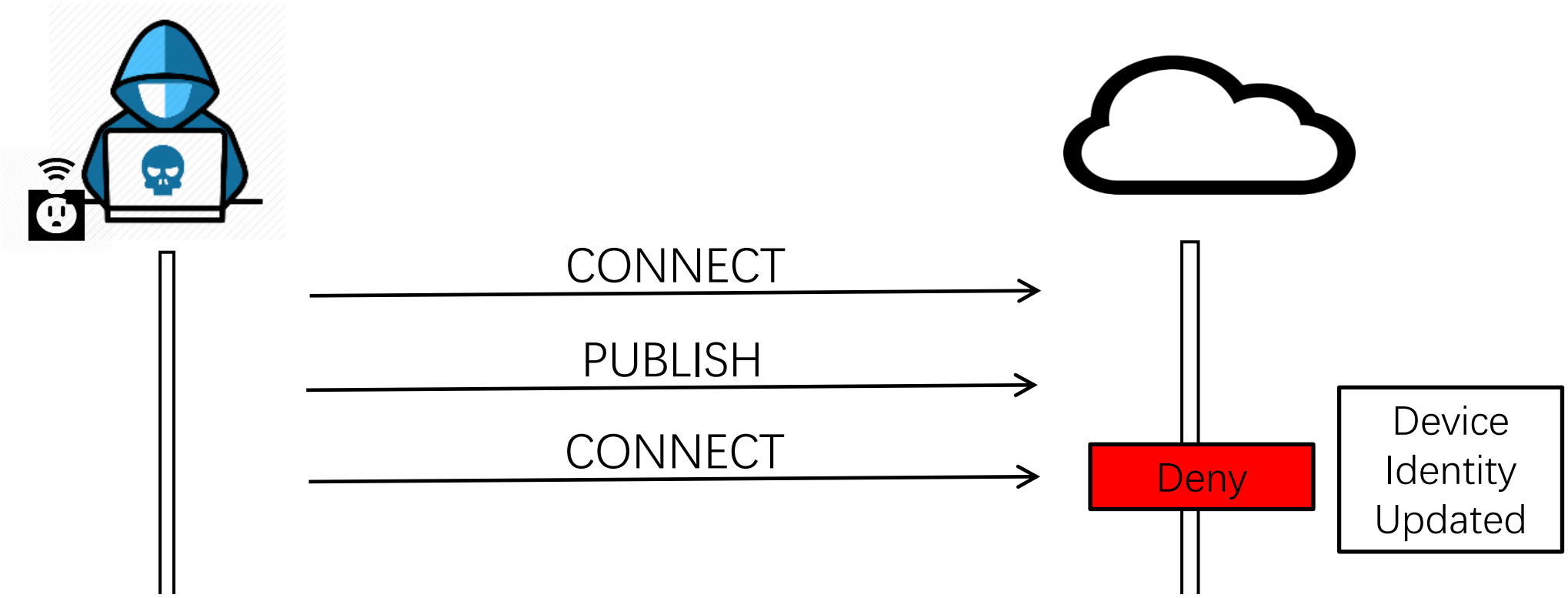
A malicious Airbnb guest

Background– Phantom Device Substitution Attack



Mitigation:
Updating the device login credential.

Non-updated session lifecycle state



Non-updated session lifecycle state

CONNECT

PUBLISH

CONNECT

Deny

Device Identity Updated

PUBLISH

PUBLISH

Never Disconnect

Why?

- “The Server MAY use a security component to authorize particular actions on the topic resource for a given Client.” -- MQTT 5.0 specification
  - CONNECT
  - SUBSCRIBE
  - PUBLISH

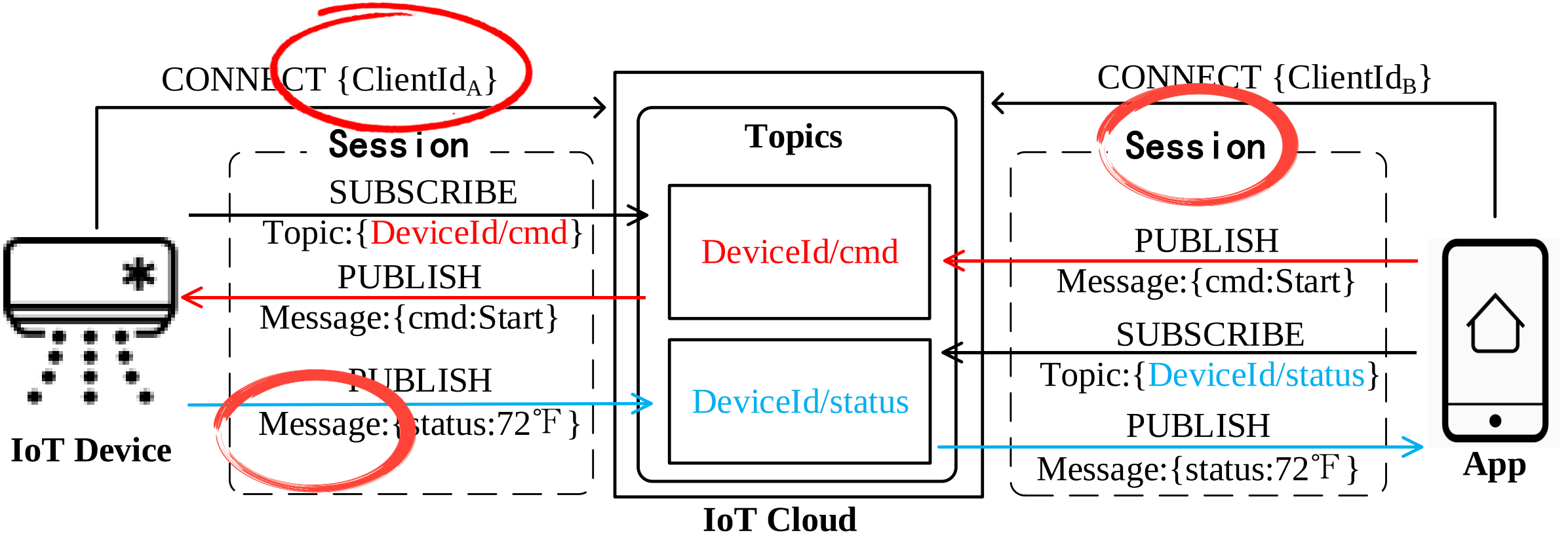
Session is not in picture

- Clients manage the session
  - SUBSCRIBE
  - UNSUBSCRIBE
  - DISCONNECT

Vendors need to extend the states of MQTT

Attack #3

Unauthenticated MQTT Identity



Identity Management in MQTT

Establish SSL Connection

Server Certificate

Certificate

IoT Cloud

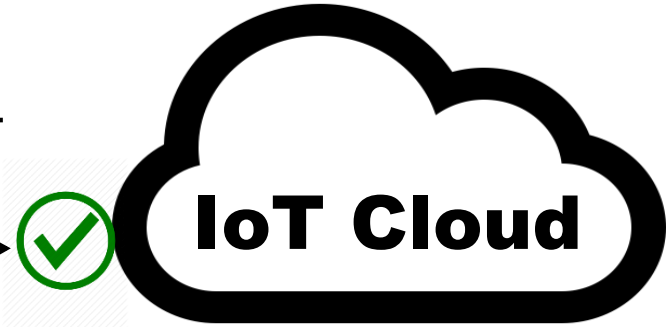
Identity Management in MQTT

Establish SSL Connection

Server Certificate

Certificate

MQTT CONNECT



CONNECT

ClientId
Will Flag
Clean Session Flag
Password Flag
...

Client Identifier (ClientId)

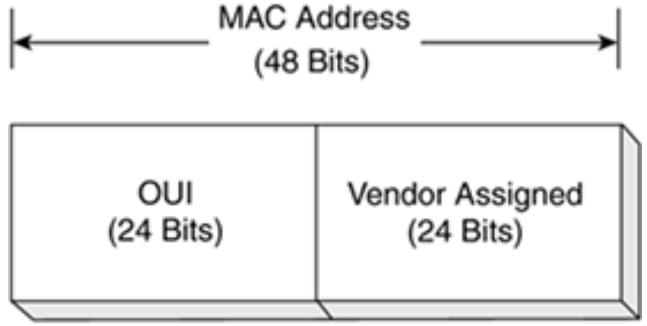
- “The Client Identifier (ClientId) identifies the Client to the Server. Each Client connecting to the Server has a unique ClientId.”

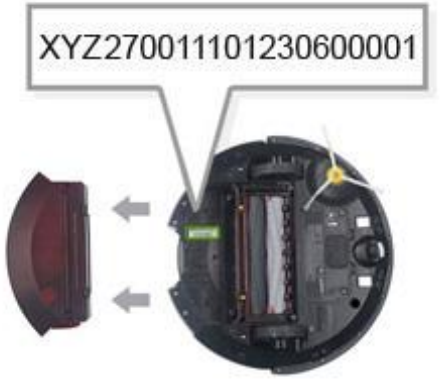
- If two clients claim the same ClientId, the later one will kick the connected one off.

MQTT Specification

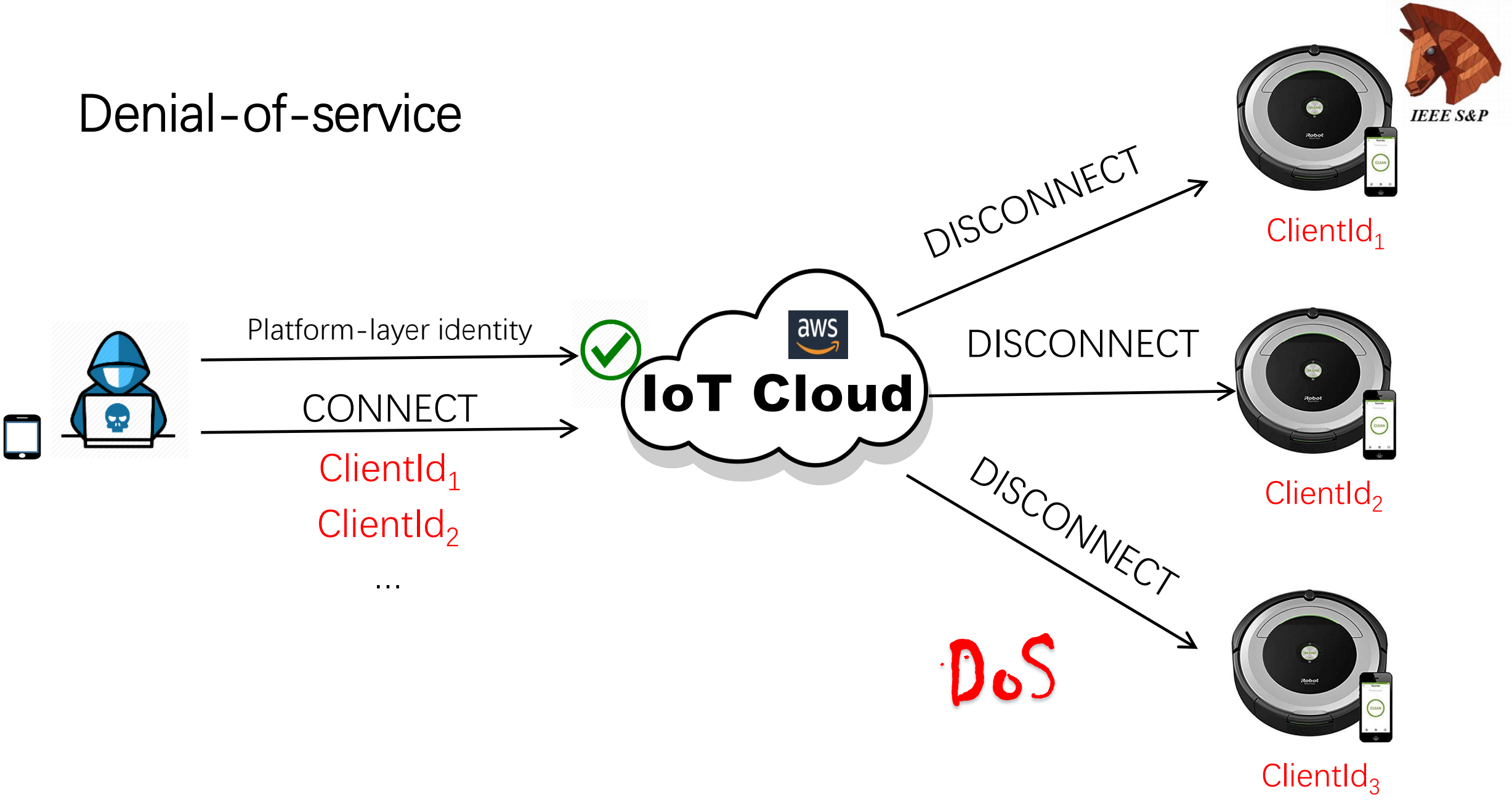
ClientId in Vendors View

- Uniqueness
  - MAC address
  - Serial number of device
  - Guessable
- One account can have multiple devices
  - Platform-layer identity
  - Lack sufficient authentication





Denial-of-service



Attack

- iRobot Roomba 690
  - Looks like a 16-digit serial number (e.g, 3147C60043211234)
  - Queried 200,000 numbers through a Web API
  - Found 10,000 ClientIds in wild after hours
  - The ClientId of mobile app can be changed

Attack

- iRobot Roomba 690
  - Looks like a 16-digit serial number (e.g, 3147C60043211234)
  - Queried 200,000 numbers through a Web API
  - Found 10,000 ClientIds in wild after hours
  - The ClientId of mobile app can be changed
  - Kick the 10,000 robots offline!

PoC Attack

- iRobot Roomba 690
  - Looks like a 16-digit serial number (e.g, 3147C60043211234)
  - Queried 200,000 numbers through a Web API
  - Found 10,000 ClientIds in wild after hours
  - The ClientId of mobile app can be changed
  - ~~Kick the 10,000 robots offline!~~
  - Only kick our own robot offline
  - One client identity, 2,000 concurrent connections
    (on our own AWS IoT endpoint)
- Session hijacking
  - Clean session flag

Why?

- ClientId is not a secret
- No feature provided by (some) IoT clouds to restrict the ClientId
- Misleading development guide

```
   "Version": "2012-10-17",
   "Statement": [
{
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:000000000000:client/${iot:ClientId}"
      ]
    },
```

${iot:ClientId} or *

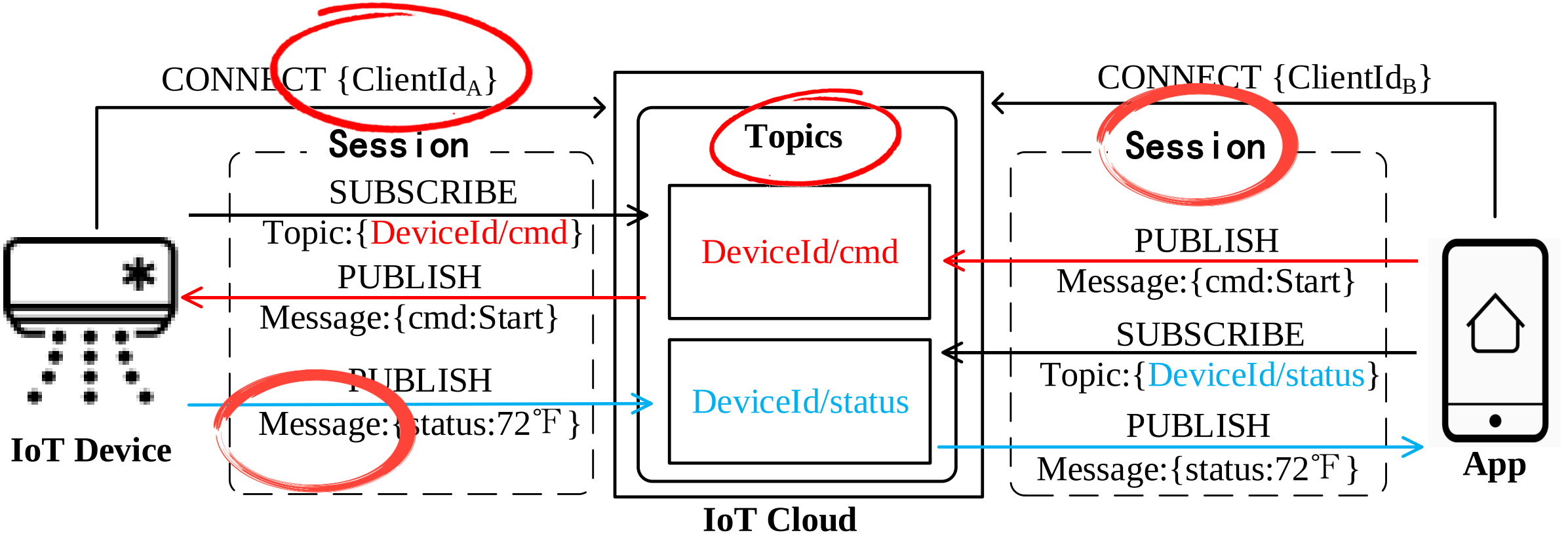
68.4% (26/38) recommended by AWS

85.4% (76/89) on Github

AWS Blog

Attack #4

Authorization Mystery of MQTT Topics



- Insecure shortcut in protecting MQTT topics
  - MQTT topics are confidential
  - Not a secret for ex-user

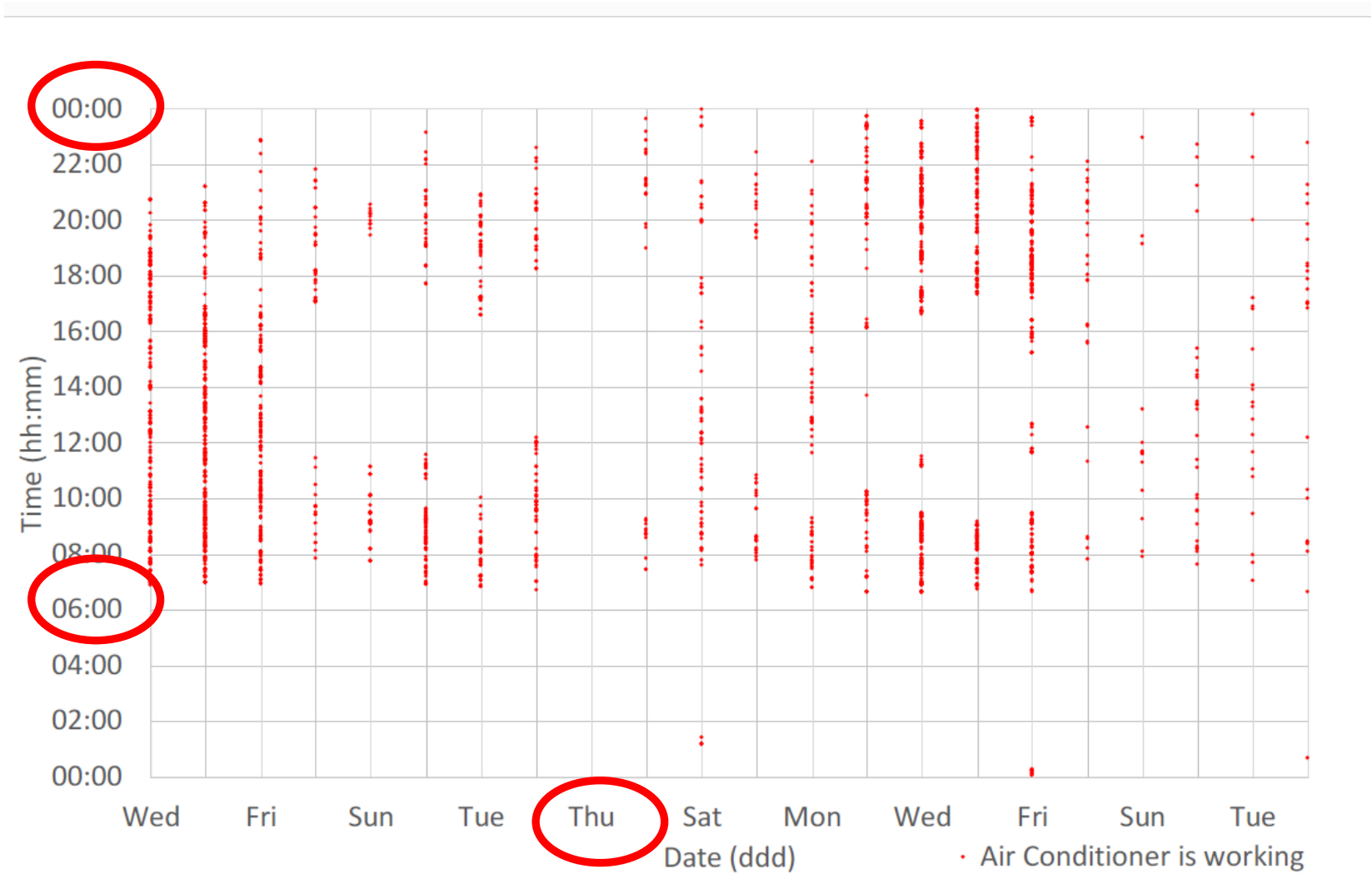
- Expressive syntax of MQTT
  - #

- Insecure shortcut in protecting MQTT topics
  - MQTT topics are confidential
  - Not a secret for ex-user

- Expressive syntax of MQTT
  - #
- Privacy implications of leaked MQTT messages
  - Personally Identifiable Information
  - Information captured by the device (temperature, air quality, etc.)
  - Cohabitants relation ("[Person Name set by user] opened the door")
  - Living habit
  - …

Sensitive Information Leakage

Measurement



Measurement

TABLE I
SUMMARY OF MEASUREMENT RESULTS

Security Weaknesses		Alibaba	AWS	Baidu	Google	IBM[1]		Microsoft	Suning	Tuya
ClientId Management		✓	✗	✗	✓	✓	✗	✗	✗	✗
Message Authorization	Will Message	N/A	✗	✗	N/A	N/A	✗	✗	N/A	✗
	Retained Message	N/A	N/A	✗	N/A	N/A	N/A	N/A	N/A	N/A
Topic Authorization		✓	✗	✓	✓	✓	✓	✓	✗	✓
Session Management	Subscription state	✗	✓	✗	N/A	N/A	✗	✗	✗	✗
	Lifecycle state	✓	✗	✗	✓	✓	✗	✗	✗	✗

✗means the weakness was successfully exploited on the platform. ✓means we were not able to exploit the weakness on the platform.
N/A means the platform did not fully support the MQTT feature; or its security policy was too coarse-grained for us to test the fine-grained aspect, e.g., the platform did not support to revoke a client's capability to subscribe, so we could not adequately test its management of "subscription state".

[1] The left and right columns under IBM show the results of testing using the *device* client and *user* client respectively.

Mitigation

- Managing protocol identities
- Update sessions
- Message-oriented access control

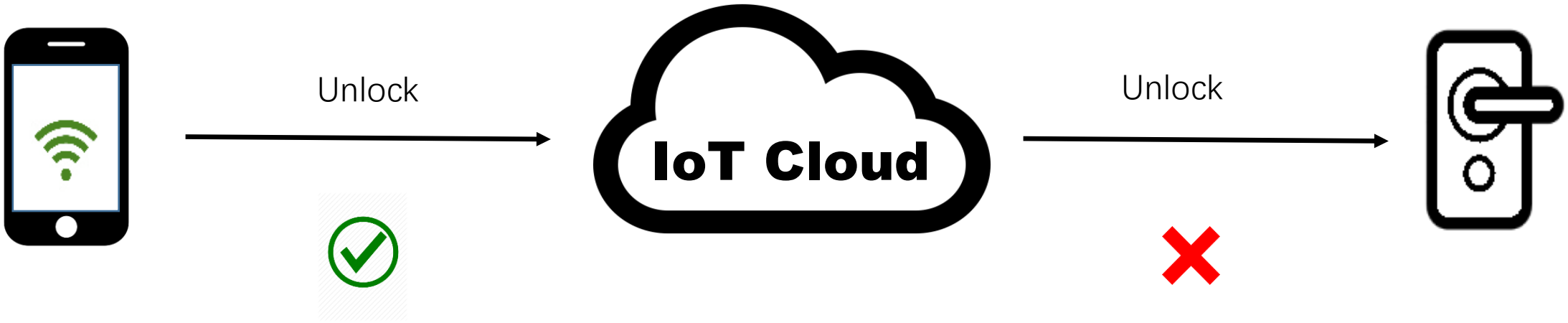
Object (O). The set of messages that subjects hold rights on.

Object Attributes (ATT(O)). An object's attributes are specified as $ATT(O) = \{content, URI, source\}$, and includes $content$ which is the application-layer information (e.g., message content), URI which represents the channel of the message (e.g. which MQTT topic the messages is published to or from), $source$ which represents the source of the object, i.e., the subject that created the message.

$$allowed(s, o, R) \Rightarrow (o.URI \in s.URI_r) \wedge (o.URI \in o.source.URI_w) \quad (1)$$

Mitigation

- Managing protocol identities
- Update sessions
- Message-oriented access control



Lessons Learnt

- Risks in applying a common-purpose protocol to IoT applications
  - Scenarios the protocol does not cover (permission revocation)
  - States of the protocol (ClientId, Session)

- Mitigating such flaws requires a joint effort from both the protocol designer and the IoT manufacturer

Thank You


西安电子科技大学
XIDIAN UNIVERSITY


中国科学院大学
University of Chinese Academy of Sciences


INDIANA UNIVERSITY
BLOOMINGTON