TÉCNICO LISBOA

PhD in Computer Science and Engineering

Advanced Topics in Cybersecurity

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

Gessildo Bengui
gessildo.bengui@tecnico.ulisboa.pt

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

Agenda

1. Introduction

Overview | **IoT**

August
Smart Lock

Google Nest
Thermostat

Amazon
Echo Alexa

WeMo
Smart Plug

Foobot Air
Quality Monitor

Ring
Doorbell

Logitech Harmony
Universal Remote

Combination of

Embedded systems + real-time analytics + Machine Learning + commodity sensors + Wireless sensor networks

that enables

Devices embedded with sensors and software to connect each other and with other systems for exchanging data over Internet. **Margaret Rouse (2019)**

1. Introduction

Overview | IoT in Cloud-based Platforms


August Smart Lock

Google Nest Thermostat


Amazon Echo Alexa

WeMo Smart Plug


Foobot Air Quality Monitor


Ring Doorbell


Logitech Harmony Universal Remote

+

?

IoT Cloud-based 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

1. Introduction

Overview | IoT in Cloud-based Platforms | **Broker**

August Smart Lock

Google Nest Thermostat

Amazon Echo Alexa

WeMo Smart Plug

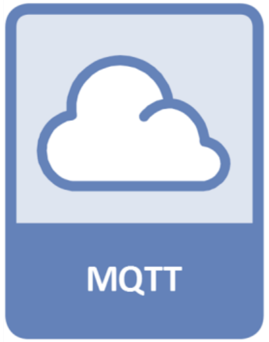
Foobot Air Quality Monitor

Ring Doorbell

Logitech Harmony Universal Remote

\+

? =



IoT Cloud-based Platforms

Watson IoT Platform | IBM

Alibaba Cloud

IoT Hub | Microsoft Azure

SUNING 苏宁易购

tuya.com

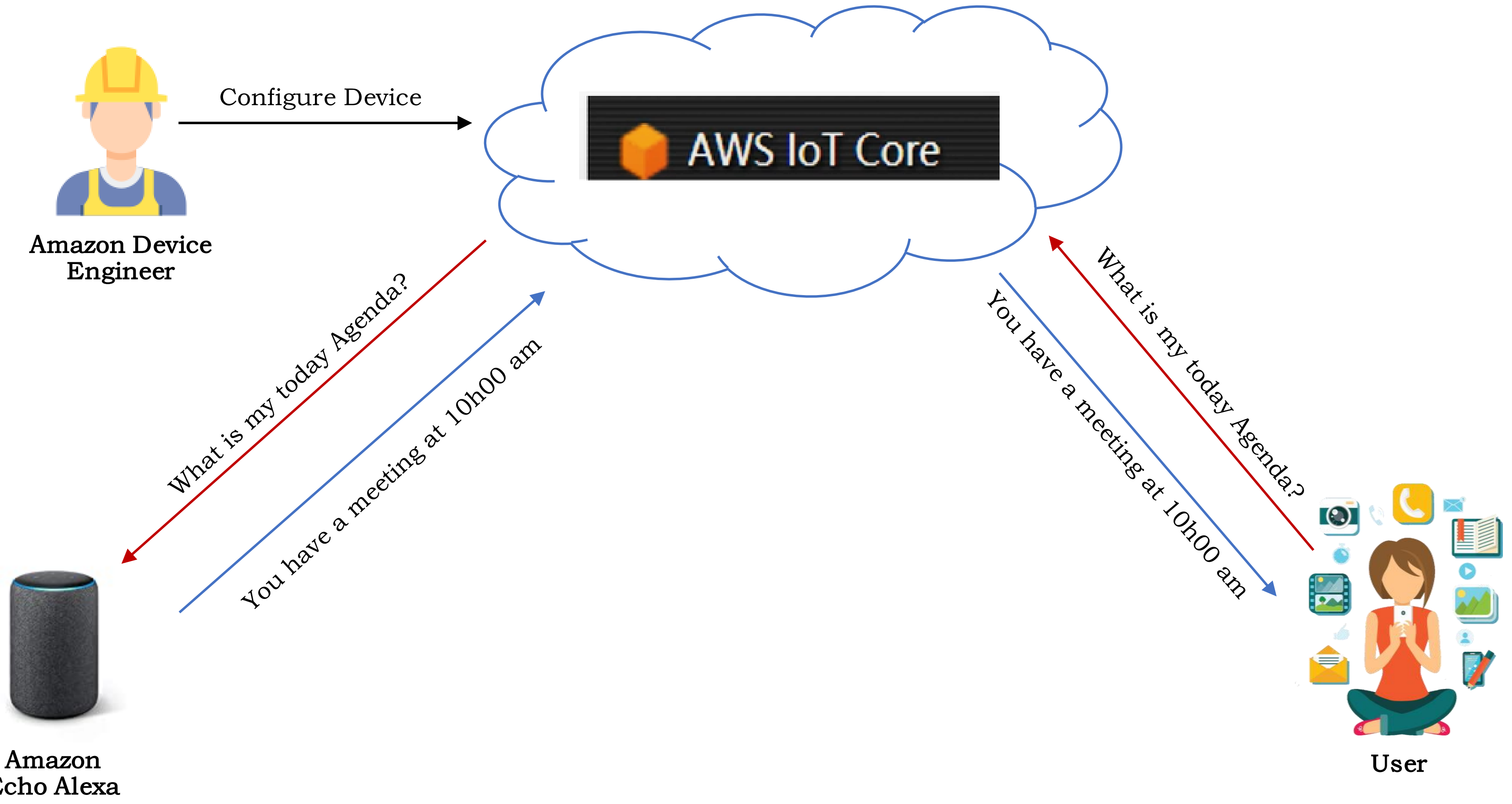
Cloud IoT Core | Google

AWS IoT Core

BAIDU AI CLOUD

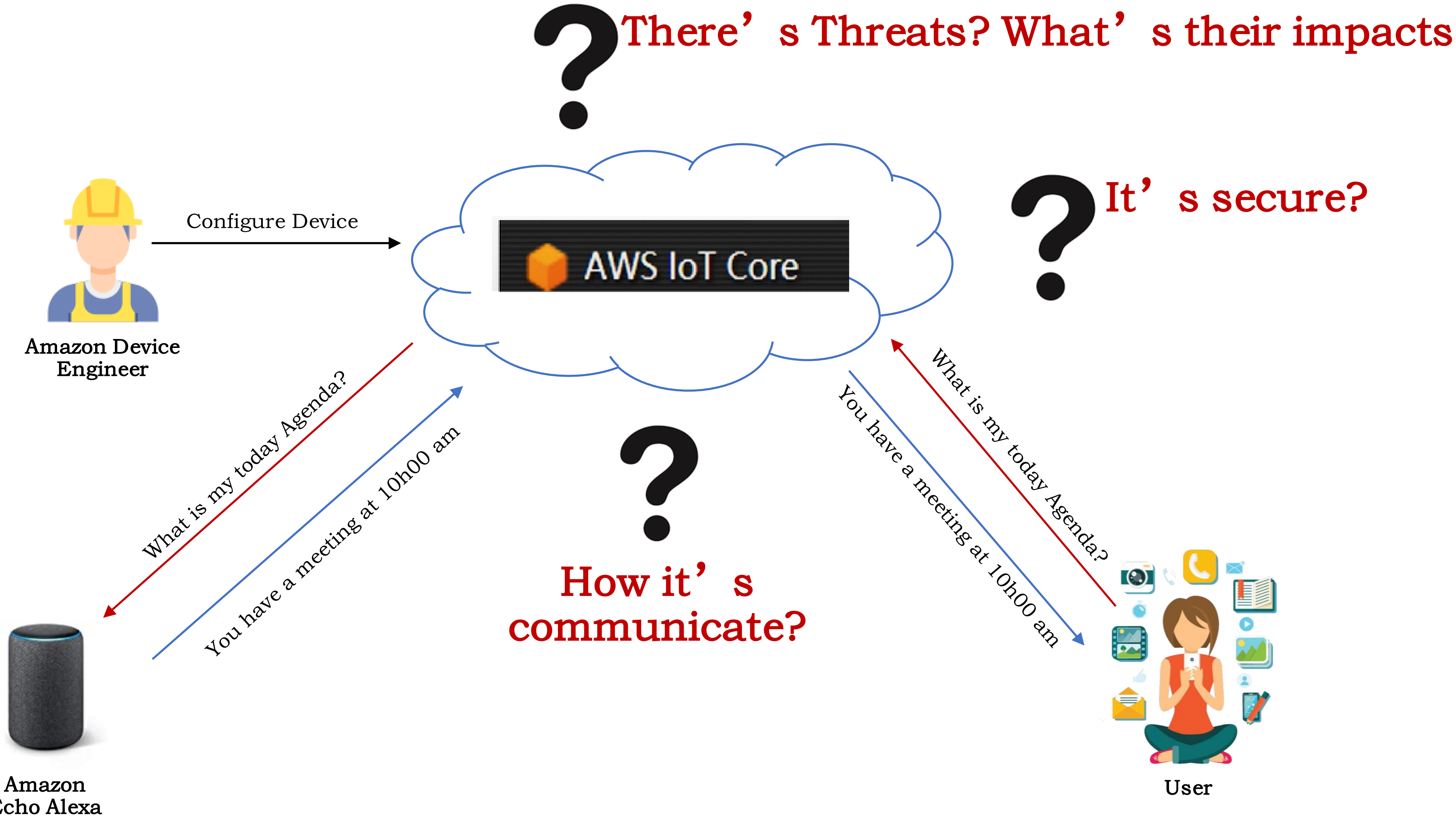
1. Introduction

Overview | IoT in Cloud-based Platforms | Broker | **Example**



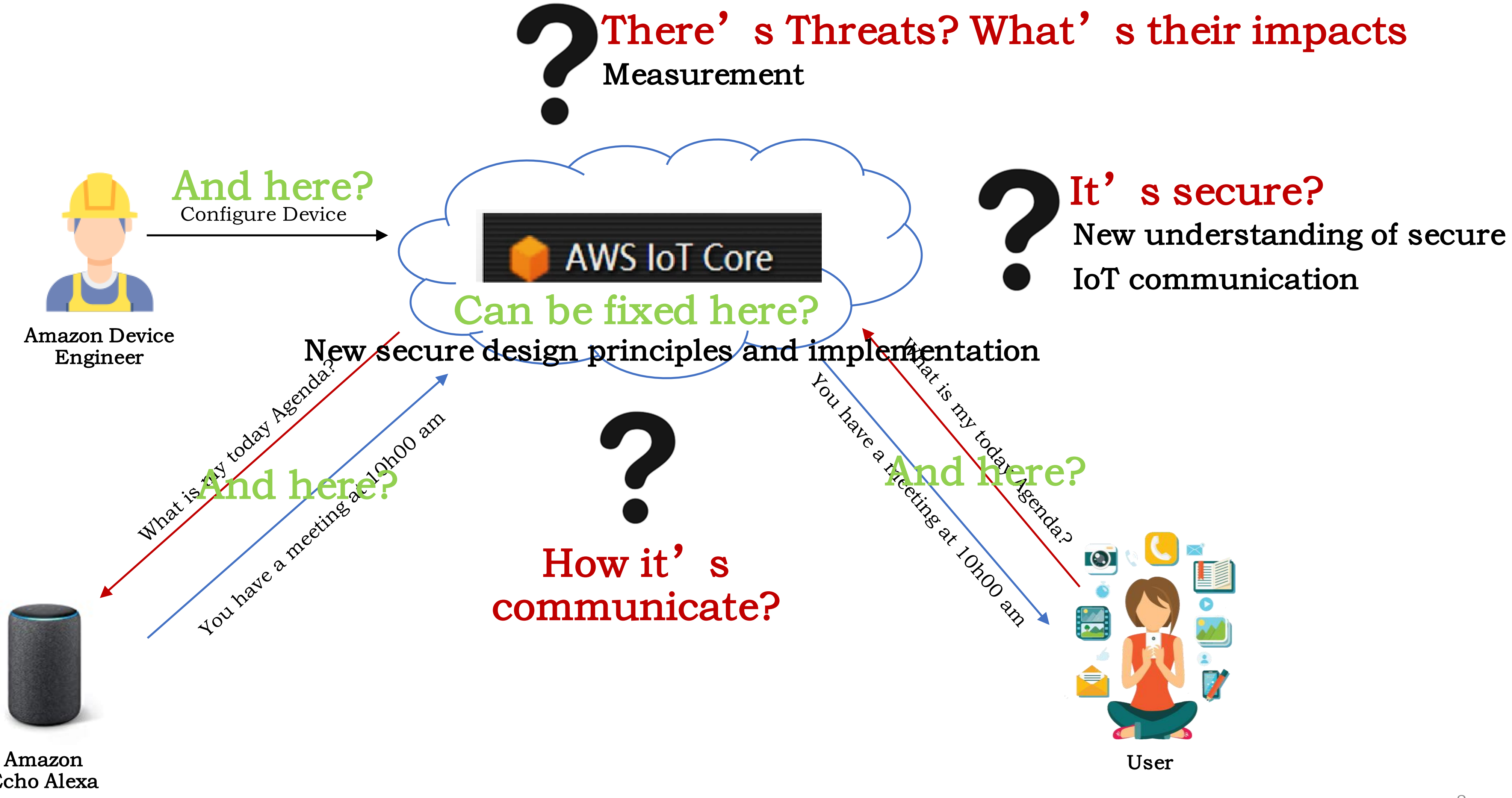
1. Introduction

Problems



1. Introduction

Contributions



2. IoT in Cloud-based Platforms

Communication | **Protocol**

- Cloud-based IoT is essentially support by a General Messaging.

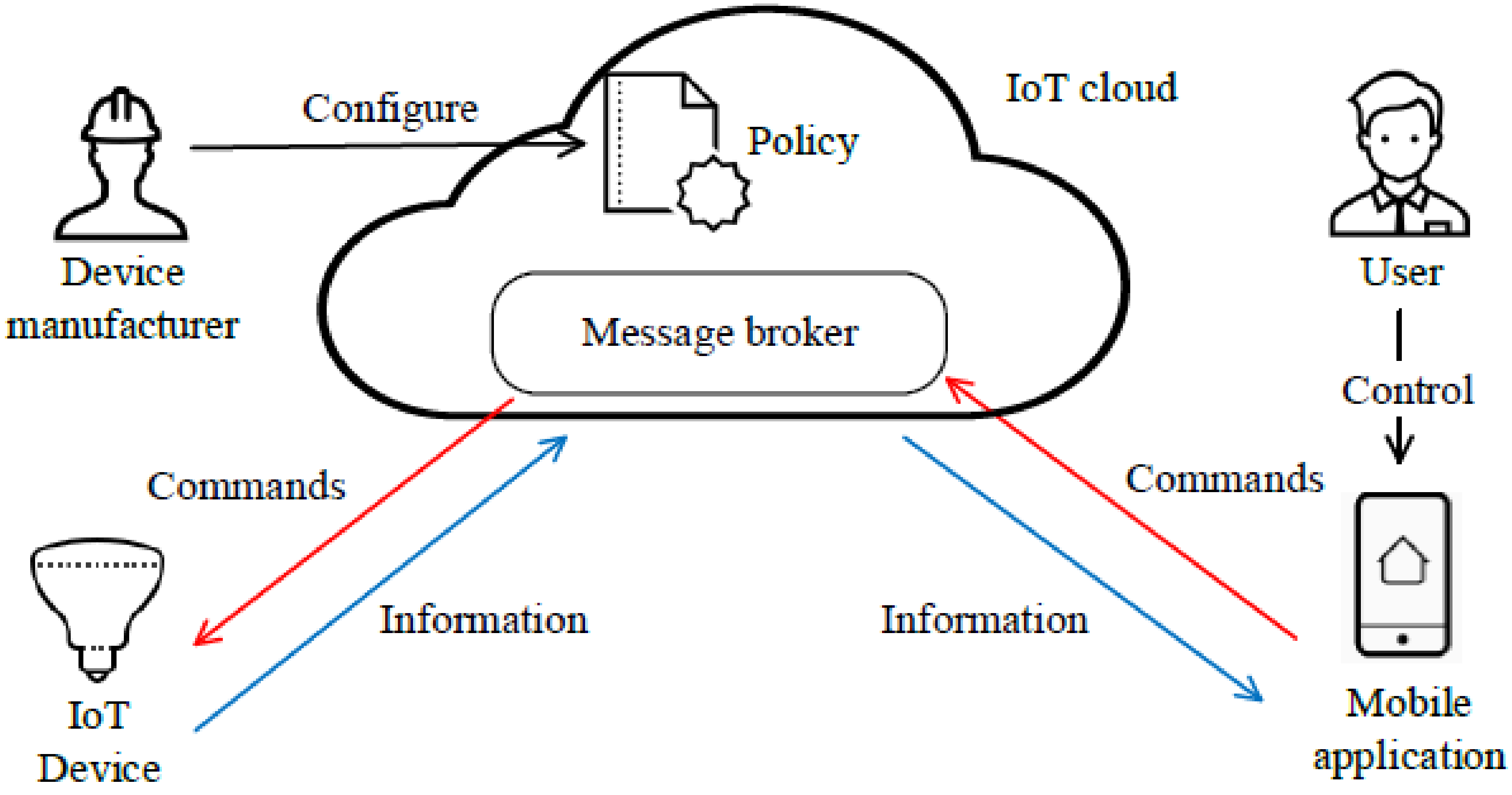
- The most used communication protocol for IoT implementations on most Cloud Platforms for IoT is **MQTT**.

- **MQTT** (**M**essage **Q**ueuing **T**elemetry **T**ransport) is a OASIS and ISO communication protocol used for remote locations where a "small code footprint" is required or network bandwidth is limited*****:
  - ✓ Lightweight.
  - ✓ publish-subscribe.
  - ✓ TCP/IP and WebSocket.

*OASIS MQTT version 5.1 documentation

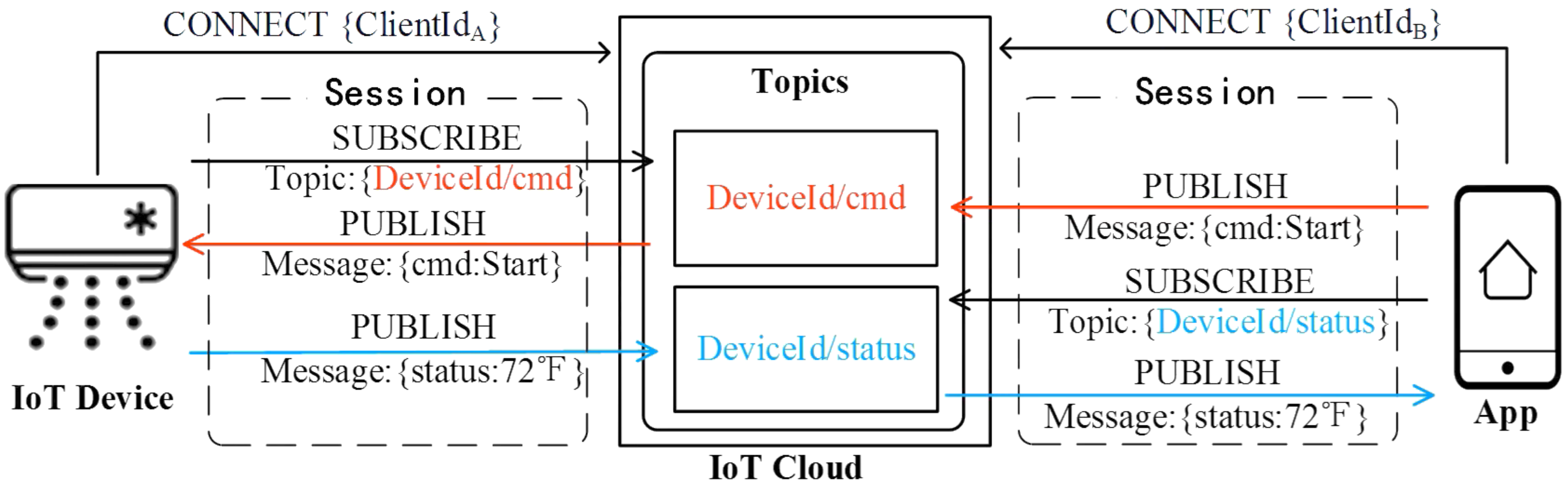
2. IoT in Cloud-based Platforms

Communication | **Architecture**



2. IoT in Cloud-based Platforms

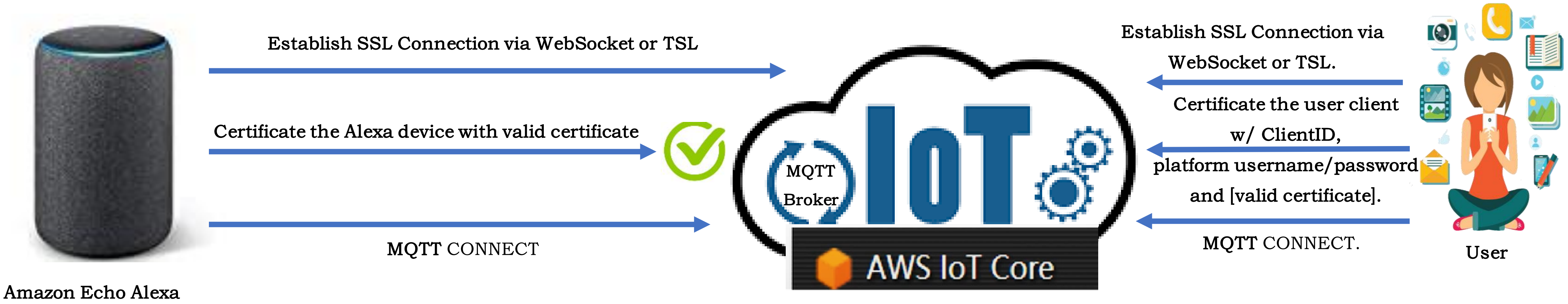
Communication | **Use of MQTT**



2. IoT Cloud-based Platforms

Protection | **Authentication**

- MQTT connections go through WebSocket and TLS.
  - ✓ Both uses cryptographic certificate;
  - ✓ Some cloud-based Platforms uses:
    - ➢ Its own username/password authentication mechanism (e. g. Amazon, Azure).
    - ➢ Single sign-on through Facebook/Google.

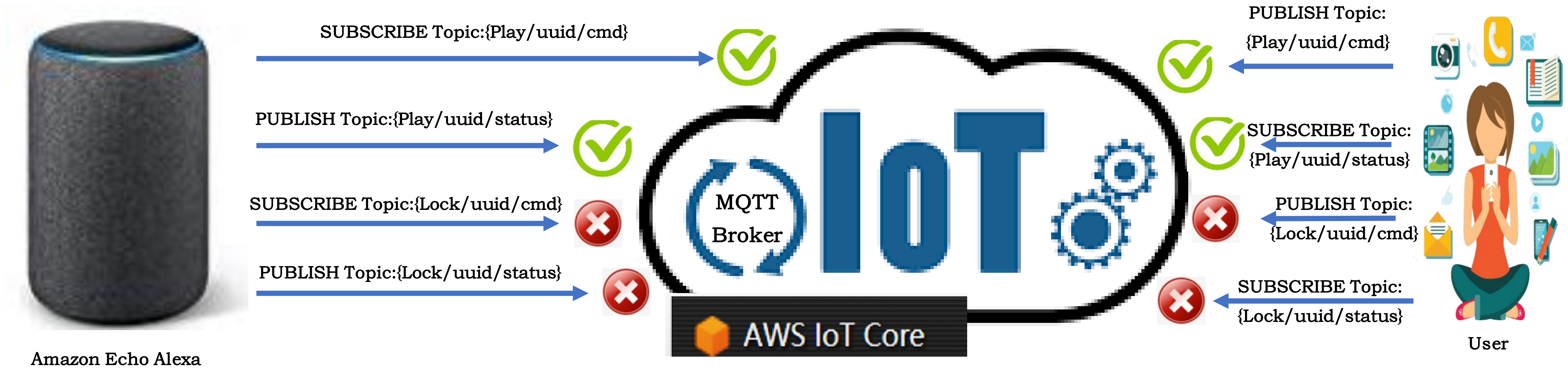


2. IoT Cloud-based Platforms

Protection | **Authorization**

- MQTT IoT cloud-based platforms aim to ensure that each user/client (Device or App user) can only send commands to and receive messages from the devices it is allowed to used.

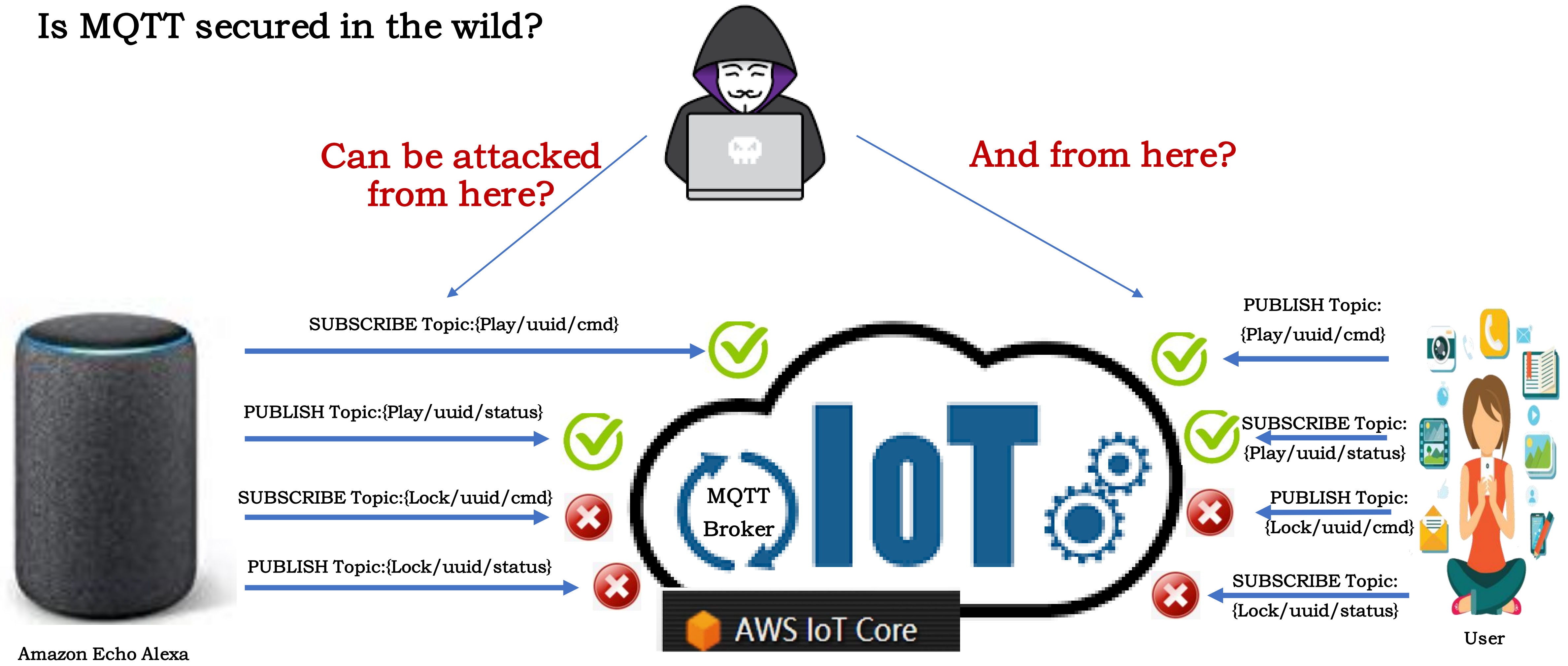
- These authorization mechanisms are ensured through **publish/subscribe** model.



2. IoT Cloud-based Platforms

Protection | Threat Model

Is MQTT secured in the wild?



2. IoT Cloud-based Platforms

Protection | Threat Model | **Context**

- Any user (Attackers also) can:
  - Open accounts:
    - ✓ With IoT Devices;
    - ✓ In IoT Cloud Platforms.
  - Collect and analyze network traffic between IoT cloud platform, IoT device and App under his legal control.
- IoT Cloud-based Platform is a pure device-sharing context:
  - Familiar apartments;
  - Hotels;
  - Airbnb (Temporary and vacation rental).
- In the above context, users are always been granted temporary to the devices.

3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | **Overview**

The main idea in in this analysis was to check how MQTT in Cloud-based Platforms consider security aspects and related threats in perspectives of:

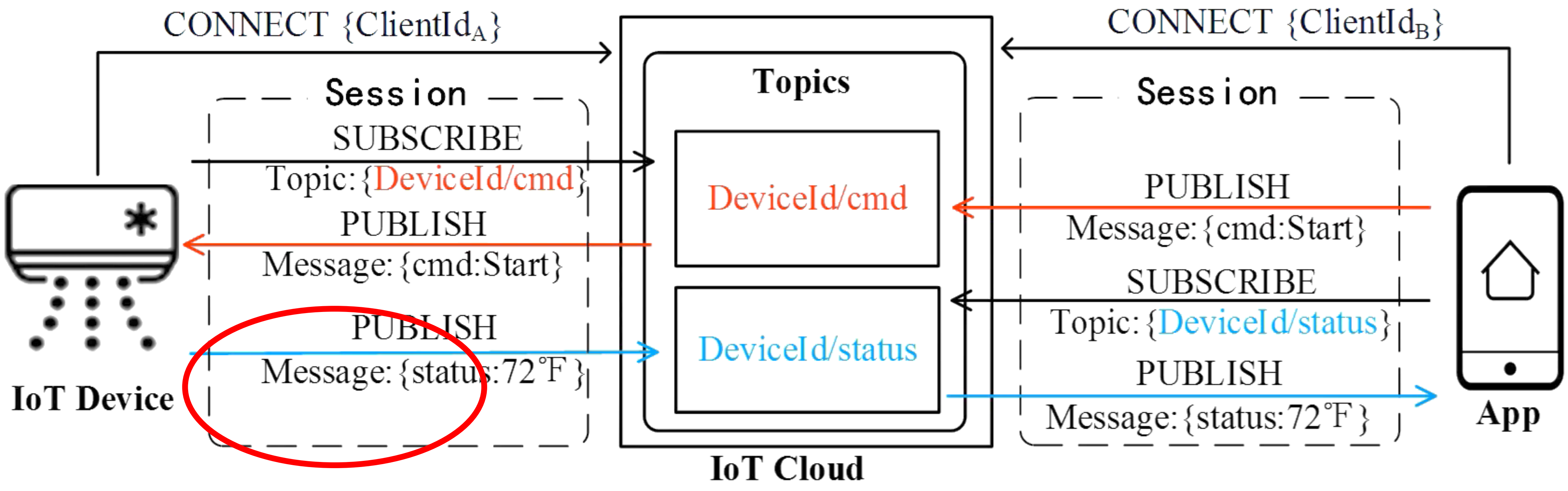
- Message;
- Session;
- Client Identity;
- Topics.

Analyze the the gaps in:

- MQTT original version;
- MQTT customized versions by Cloud-based platforms

3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | **Attack #1: Unauthorized MQTT Messages**

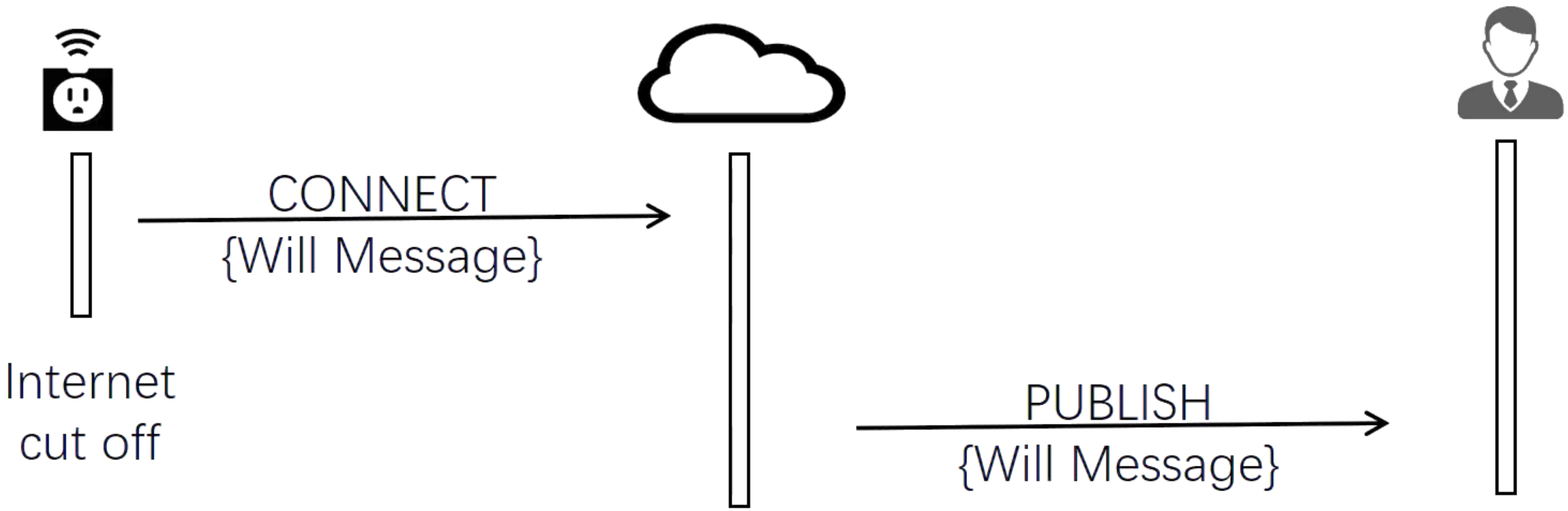


3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #1: Unauthorized MQTT Messages | **Context**

Will Message in MQTT

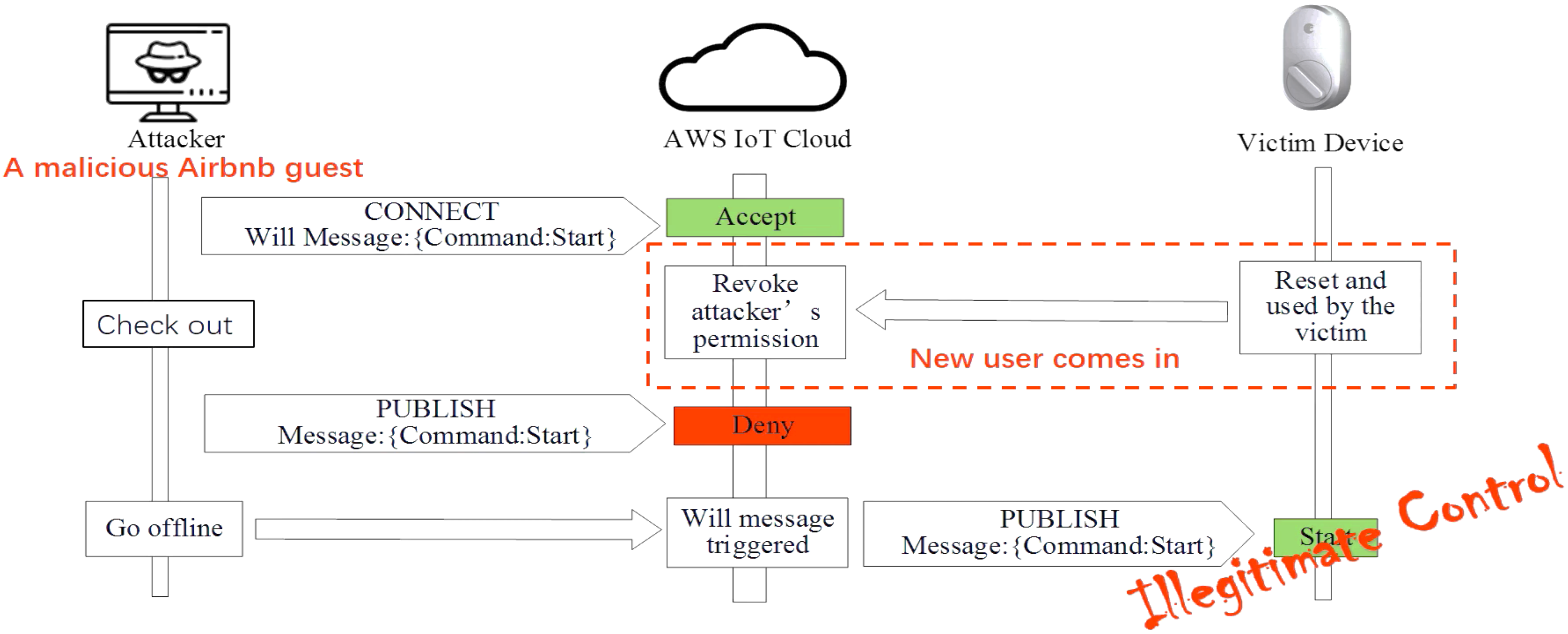
- A kind of MQTT message mostly used for exception handling scenario.
- Carries topics and payload (both commands and texts).
- Publish bu«y the server when client disconnects accidentally.



3. Sec. in MQTT in IoT Cloud-based Platforms

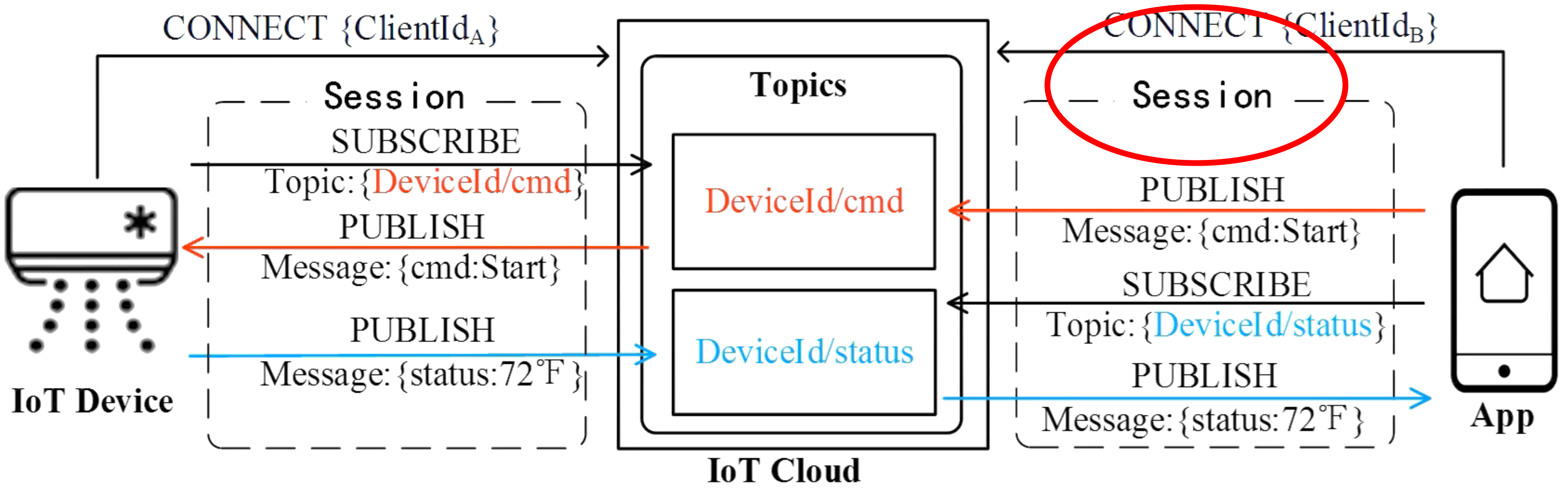
Analysis | Attack #1: Unauthorized MQTT Messages | **Context**

Unauthorized Will [and retained] Message



3. Sec. in MQTT in IoT Cloud-based Platforms

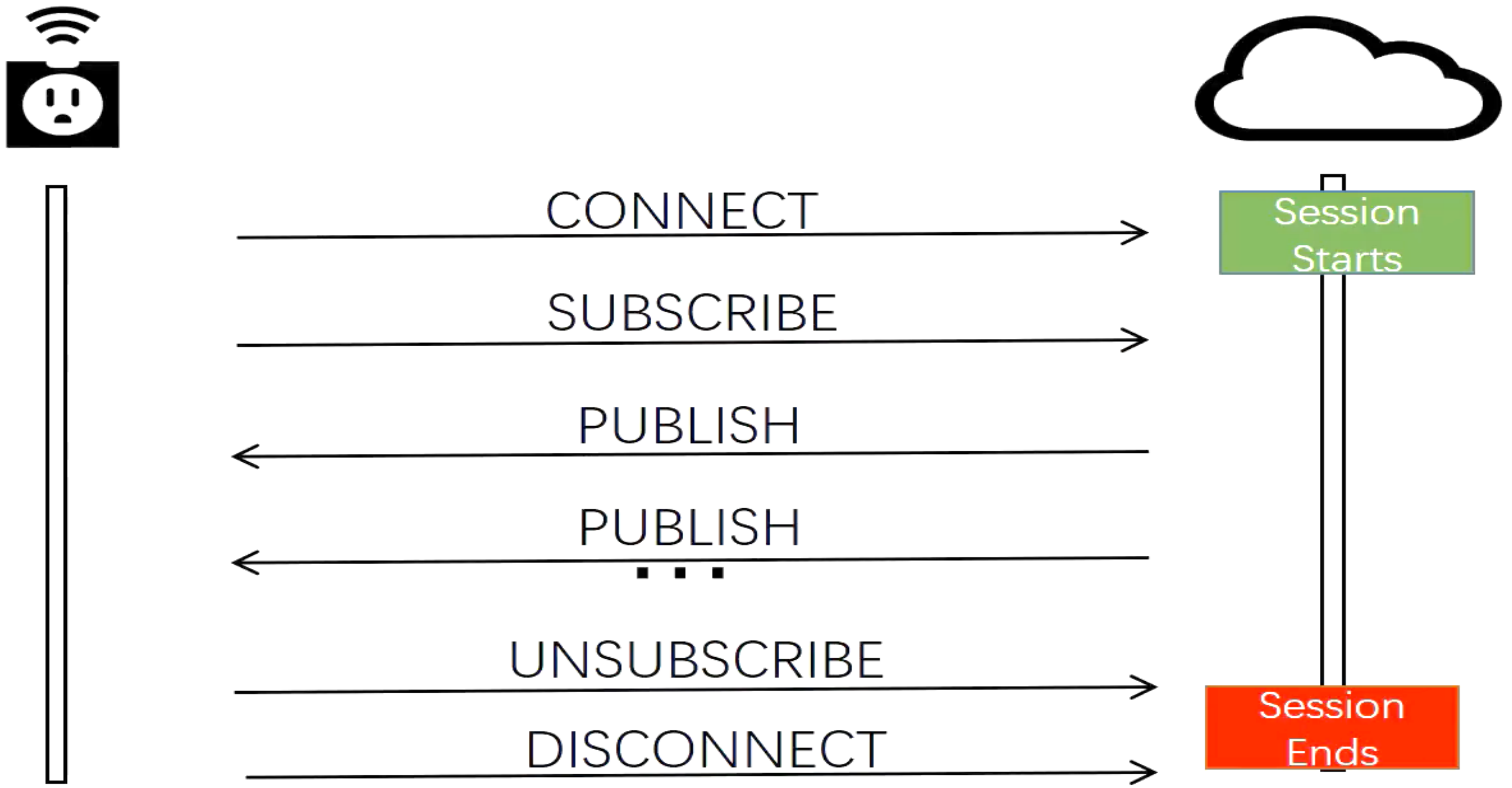
Analysis | Attack #2: Faults in Managing MQTT Sessions



3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #2: Faults in Managing MQTT Sessions| **Context**

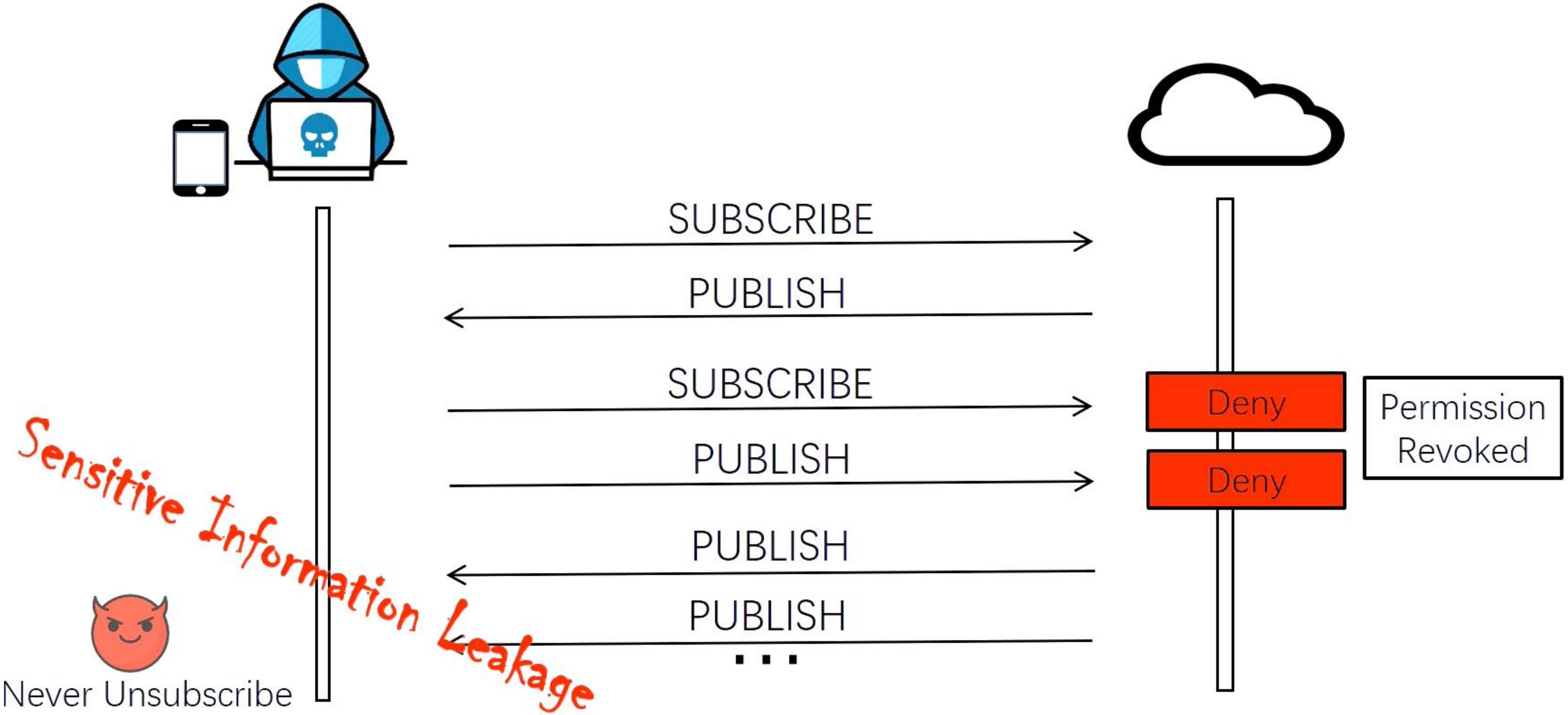
MQTT Session



3. Sec. in MQTT in IoT Cloud-based Platforms

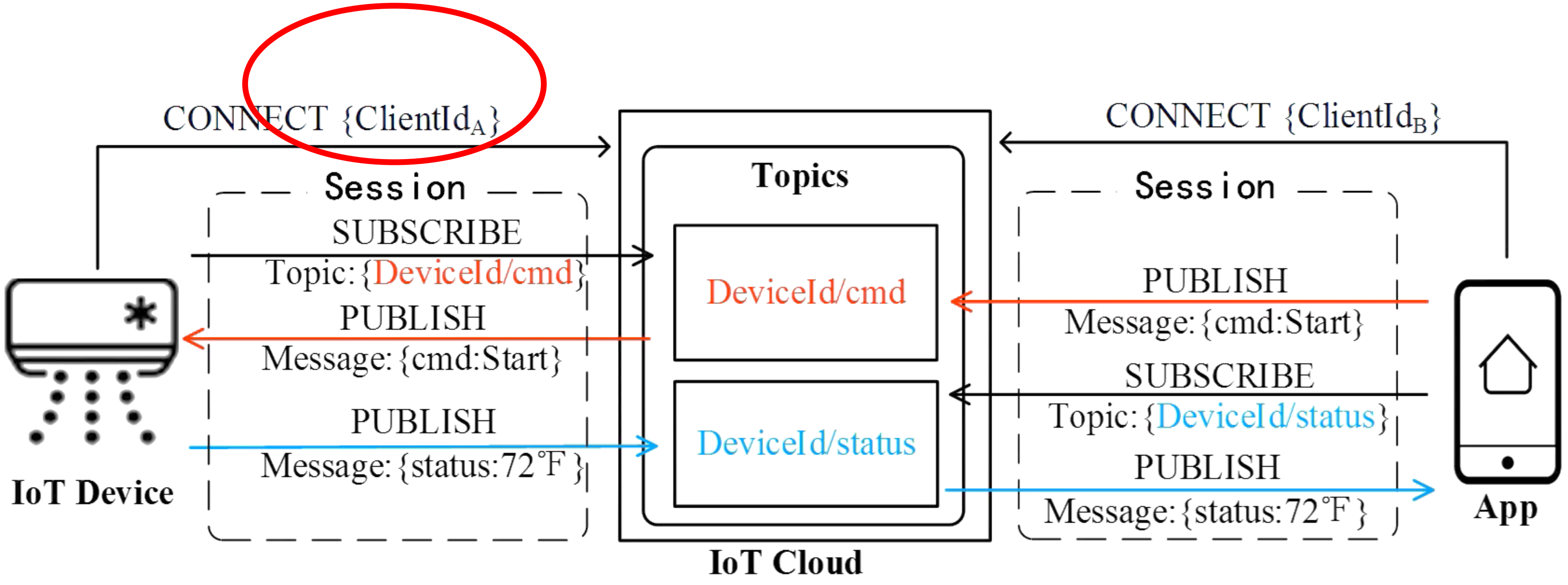
Analysis | Attack #2: Faults in Managing MQTT Sessions | **Context**

No-updated session subscription [and life cycle] state



3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | **Attack #3: Unauthenticated MQTT Identity**

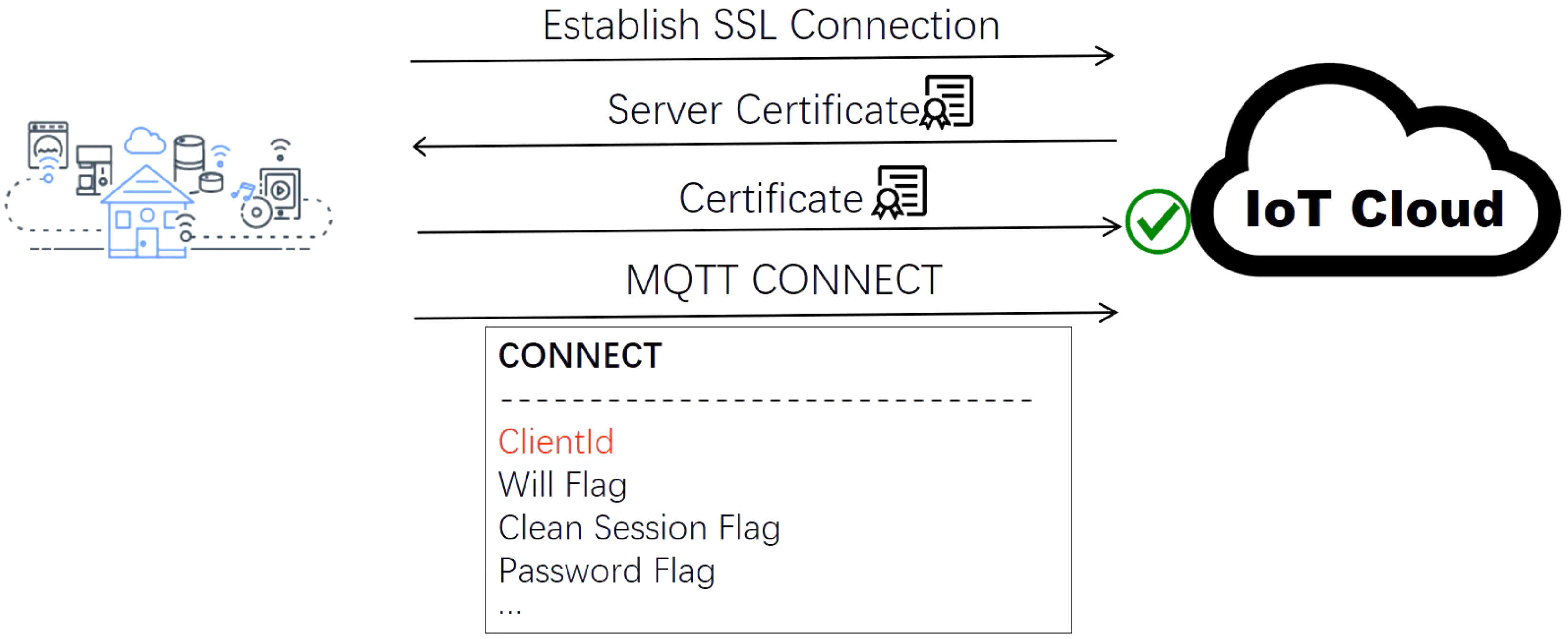


3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #3: Unauthenticated MQTT Identity | **Context**

Identity Management in MQTT | ClientId

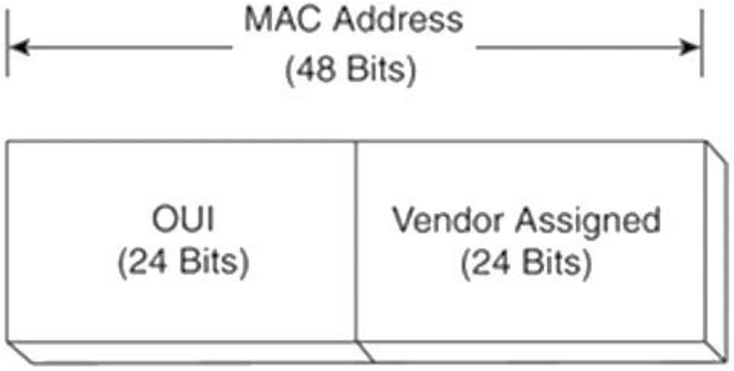
- The Client Identifier (ClientId) identifies the Client to the server. Each Client connecting to the Server has a **unique** ClientId.”
- If two clients claim the same ClientId, **the later one will kick the connected one off.**



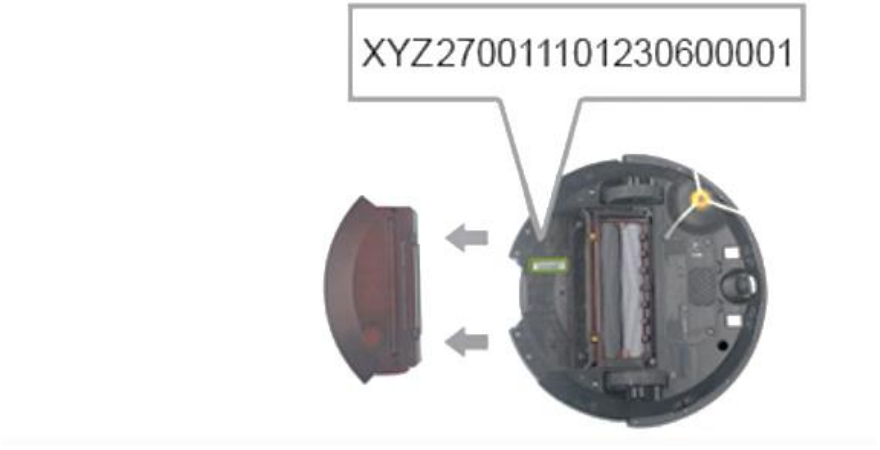
3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #3: Unauthenticated MQTT Identity | **Context**

ClientId in Vendors View



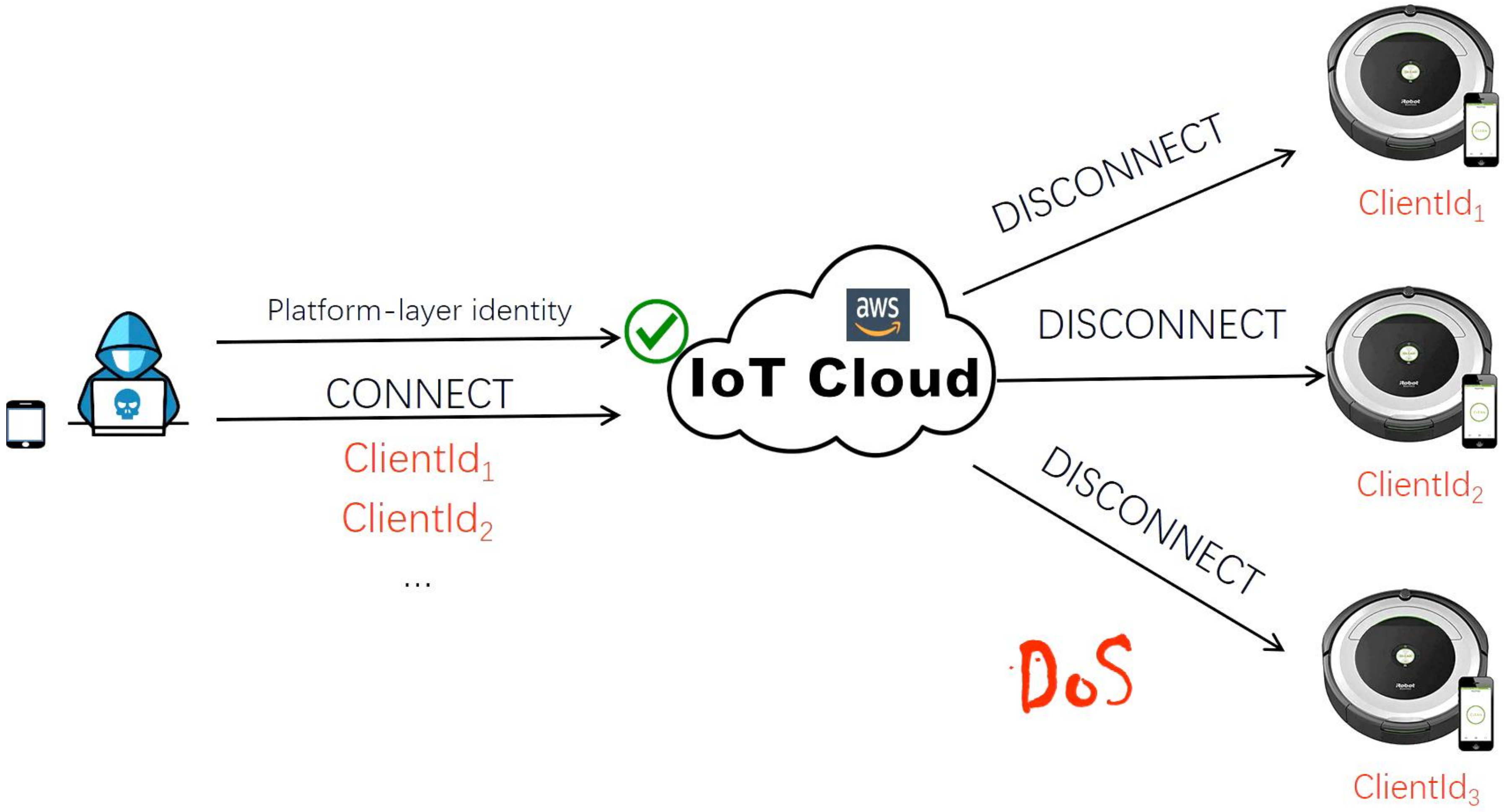
- Uniqueness
  - MAC Address.
  - Device Serial Number.
  - **Are Guessable.**
- One account can have multiple devices
  - Platform-layer identity.
  - Lack sufficient authentication.



3. Sec. in MQTT in IoT Cloud-based Platforms

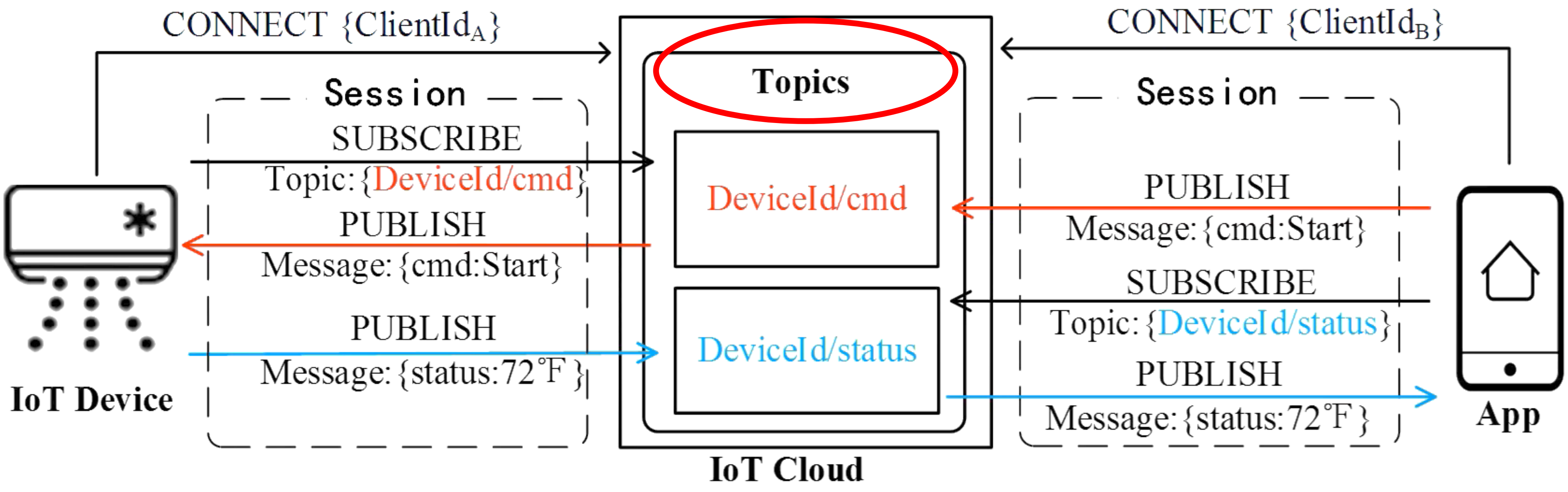
Analysis | Attack #3: Unauthenticated MQTT Identity | **Context**

Denial-of-Service



3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #4: Authorization Mystery of MQTT Topics



3. Sec. in MQTT in IoT Cloud-based Platforms

Analysis | Attack #4: Authorization Mystery of MQTT Topics | **Context**

Topics in MQTT

- Insecure shortcut in protecting MQTT topics
  - MQTT topics are confidential.
  - But not a secret for ex-users.
- Expressive syntax of MQTT
  - #.
- Privacy implications of leaked MQTT messages.
  - Personally Identifiable Information.
  - Information captured by the device (temperature, lock status, air quality, etc).
  - Living habit.

3. Sec. in MQTT in IoT Cloud-based Platforms

Measurement | **Scope and Magnitude**

- Focused on design defects.
- Applied on eight leading IoT cloud-based platforms.
- Cover the four dimension of threats related in security analysis:
  - ✓ Identify management;
  - ✓ Message authorization;
  - ✓ Session management;
  - ✓ Topic authorization.

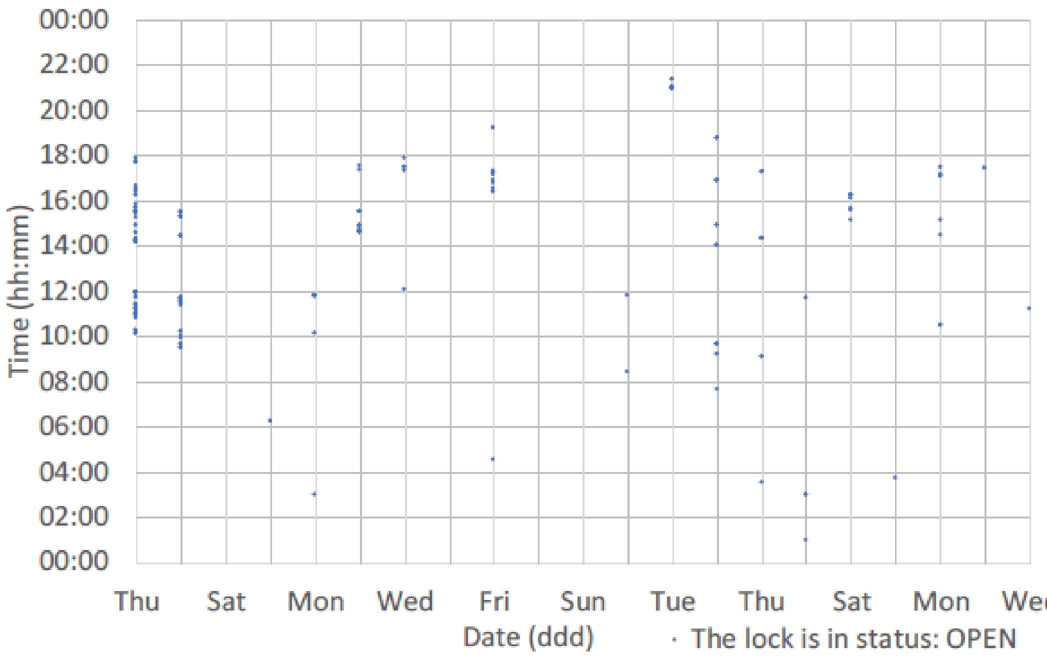
Security Weaknesses		Alibaba	AWS	Baidu	Google	IBM[1]		Microsoft	Suning	Tuya
ClientId Management		✓	✗	✗	✓	✓	✗	✗	✗	✗
Message Authorization	Will Message	N/A	✗	✗	N/A	N/A	✗	✗	N/A	✗
	Retained Message	N/A	N/A	✗	N/A	N/A	N/A	N/A	N/A	N/A
Topic Authorization		✓	✗	✓	✓	✓	✓	✓	✗	✓
Session Management	Subscription state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

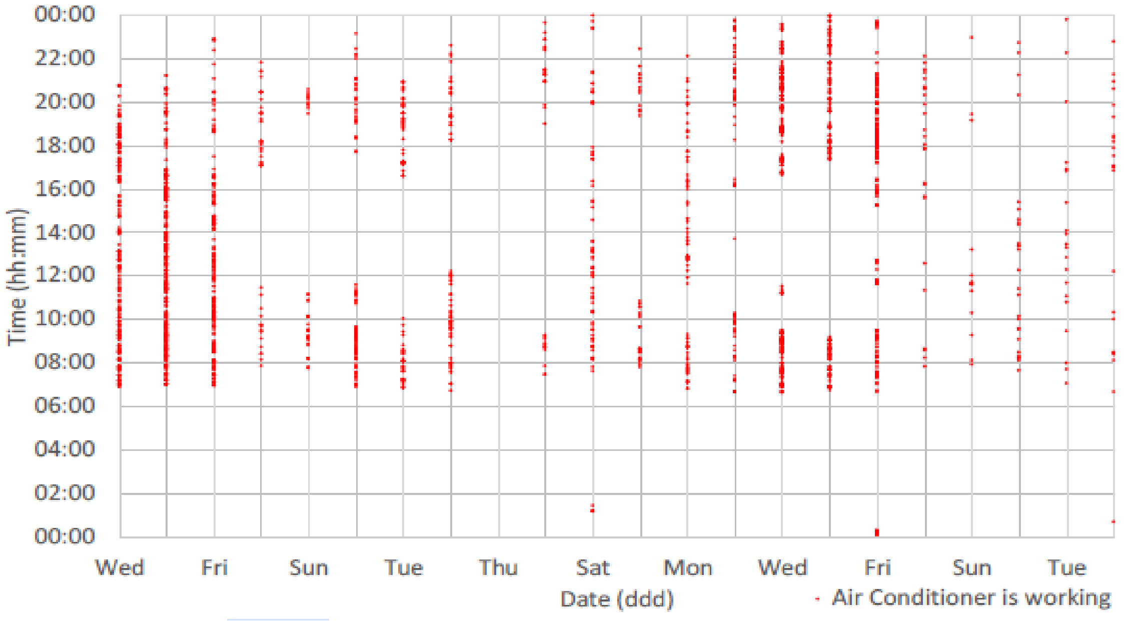
✗means the weakness was successfully exploited on the platform. ✓means we were not able to exploit the weakness on the platform.
N/A means the platform did not fully support the MQTT feature; or its security policy was too coarse-grained for us to test the fine-grained aspect, e.g., the platform did 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.

3. Sec. in MQTT in IoT Cloud-based Platforms

Measurement | **Privacy Implications of Leaked MQTT Messages**

- Attacks performed by exploring **Fault in Managing MQTT Sessions** and **Authorization Mystery of MQTT Topics**.
- Personal user data collected by authorized user (future attacker) by subscribing generic (all topics) with wildcard (#) and get these data after been revoked.
- When this information is combined for a longitudinal analysis, it’s possible to infer private habits, routine behavior, cohabitant relation, etc.





3. Sec. in MQTT in IoT Cloud-based Platforms

Mitigation | **Proposals**

- Managing Protocol Identities and Sessions.
- Message-Oriented Access Control Model.
- Implementation and Evaluation.

3. Sec. in MQTT in IoT Cloud-based Platforms

Mitigation | Proposals | **Managing Protocol Identities and Sessions**

A key design principle in the adoption of a messaging protocol to the complicated and adversarial IoT systems is, *protocol-layer identity (e.g., ClientId), if any, should be authenticated; additionally, if the identity is used as a security token (e.g., session token), its confidentiality should be guaranteed.*

Identification mechanism can be improved by combination of **platform username/password + OAuthA + user** to create **p_user_id**.

Additionally, sessions in a messaging protocol should be guarded following the principle: *in the presence of an adversarial environment where subjects (e.g., a user) are expected to have privilege changes, session states, including protocol-agnostic states (e.g., lifecycle states) and protocol-specific states (e.g., subscription states), should accordingly keep updated in response.*

3. Sec. in MQTT in IoT Cloud-based Platforms

Mitigation | Proposals | **Message-Oriented Access Control Model**

A Key to securing a messaging protocol on IoT systems is to protect its message communication: *the system should govern the subjects' rights to send/receive messages, and additionally manage security implications in receiving a message with respect to the recipients' security requirement.*

- Using Message-Oriented Usage Control Model **(MOUCON)** that is extension of **UCON** and builds familiar concepts, such as:
  - **Subject (S):** clients in the communication (users and devices).
  - **Subject Attributes (ATT(S)):** ATT(S) = {id, URIw, URIr}.
  - **Object(O):** The set of messages that subjects hold rights on.
  - **Object Attributes (ATT(O)):** ATT(O) = {content, URI, source}.
  - **Rights (R):** are privileges that a **subject(s)** can hold and exercise on an **object(o)**. Can be: **Write (W)** (e.g., publish a message) and **Read (R)** (e.g., subscribe/receive a message).
  - **Authorizations:** function that evaluate **Rights(R)** of **ATT(S)** against **ATT(O)**.
    - ✓ allowed(s, o, R)) => (o.URI € s.URIr) ^ (o.URI € o.source.URIw)

3. Sec. in MQTT in IoT Cloud-based Platforms

Mitigation | Proposals | **Implementation and Evaluation**

To implement the proposed solution (**Managing Protocol Identities and Sessions** and **Message-Oriented Access Control Model**), the authors used Mosquitto 1.5.4 (a open source IoT cloud-based platform customized MQTT implementation), by modifying:

- It’ s relevant data structures relating to its messages (struct *mosquitto_msg_store*) by adding Security-related attributes (e.g., message’ source);
- Adding authorization functions to its broker;
- Adding the proposed client identification mechanism (ClientId restriction in the broker’ s existing access control function used for establishing session).

Clients Num	1000			2000			4000			6000			8000		
	Without Protection	With Protection	Overhead (%)	Without Protection	With Protection	Overhead (%)	Without Protection	With Protection	Overhead (%)	Without Protection	With Protection	Overhead (%)	Without Protection	With Protection	Overhead (%)
Delay (s)	1.432	1.441	0.63	1.450	1.456	0.40	1.456	1.462	0.41	1.458	1.459	0.06	1.466	1.471	0.34
CPU (%)	19.1	22.2	5.52	23.2	25.6	10.34	24.4	26.9	10.25	27.6	29.6	7.34	29.5	32.2	9.15
MEM (KB)	6725	6734	0.13	6736	6740	0.05	6752	6756	0.05	6872	6880	0.12	6883	6963	0.16

4. Discussion and Future Work

Lessons learnt

- Most important:
  - Check and evaluate when applying a utility-oriented, common-purpose protocol to malicious parties.
  - Even after customized the protocol, please test and check and evaluate if all gaps are solved/closed.
  - The gap analysis must be based on **what the protocol can protect and what needs to be protected**.

4. Discussion and Future Work

Future Work

- Apply the same study in other similar protocols:
  - Firehose; CoAP;
  - AMQP;
  - JoyLink;
  - Alink
- Automated discovery of the flaws.
- MQTT 5.

5. Related Work

Main Work

- Security studies on MQTT
- Security studies on IoT cloud-based Platform

6. Conclusion

Key Outcomes

- First systematic study on security risks in use of general messaging protocol for IoT device-user communication.

- Identified, and shared with the eight most used/bigger IoT cloud-based platforms, the gaps between the protocol designed for simple and benign context versus in complicated and adversarial one, and the challenges in covering properly those gaps.

- Presented new design principles and proposed an enhanced access model.

- Proposal implemented and evaluated in real scenario and proved to be high effectiveness and efficiency.

- The new design principles and the enhanced protection model will lead to better protection of user-device interactions in the real world.

Appendix

Related Papers

- Study of Internet-of-Things Messaging Protocols Used for Exchanging Data with External Sources.
- SAFETHINGS: Data Security by Design in the IoT.
- Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol.
- Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol.
- On the Bulk Ingestion of IoT Devices from Heterogeneous IoT Brokers.
- CoAP and MQTT Based Models to Deliver Software and Security Updates to IoT Devices over the Air.
- Evaluation of Message Protocols for IoT.
- Implementation of MQTT Native Application for Tizen-Based Smartwatches.
- A Large-scale Empirical Study on the Vulnerability of Deployed IoT Devices.
- A Secure Corroboration Protocol for Internet of Things (IoT) Devices Using MQTT Version 5 and LDAP.

Thank You!