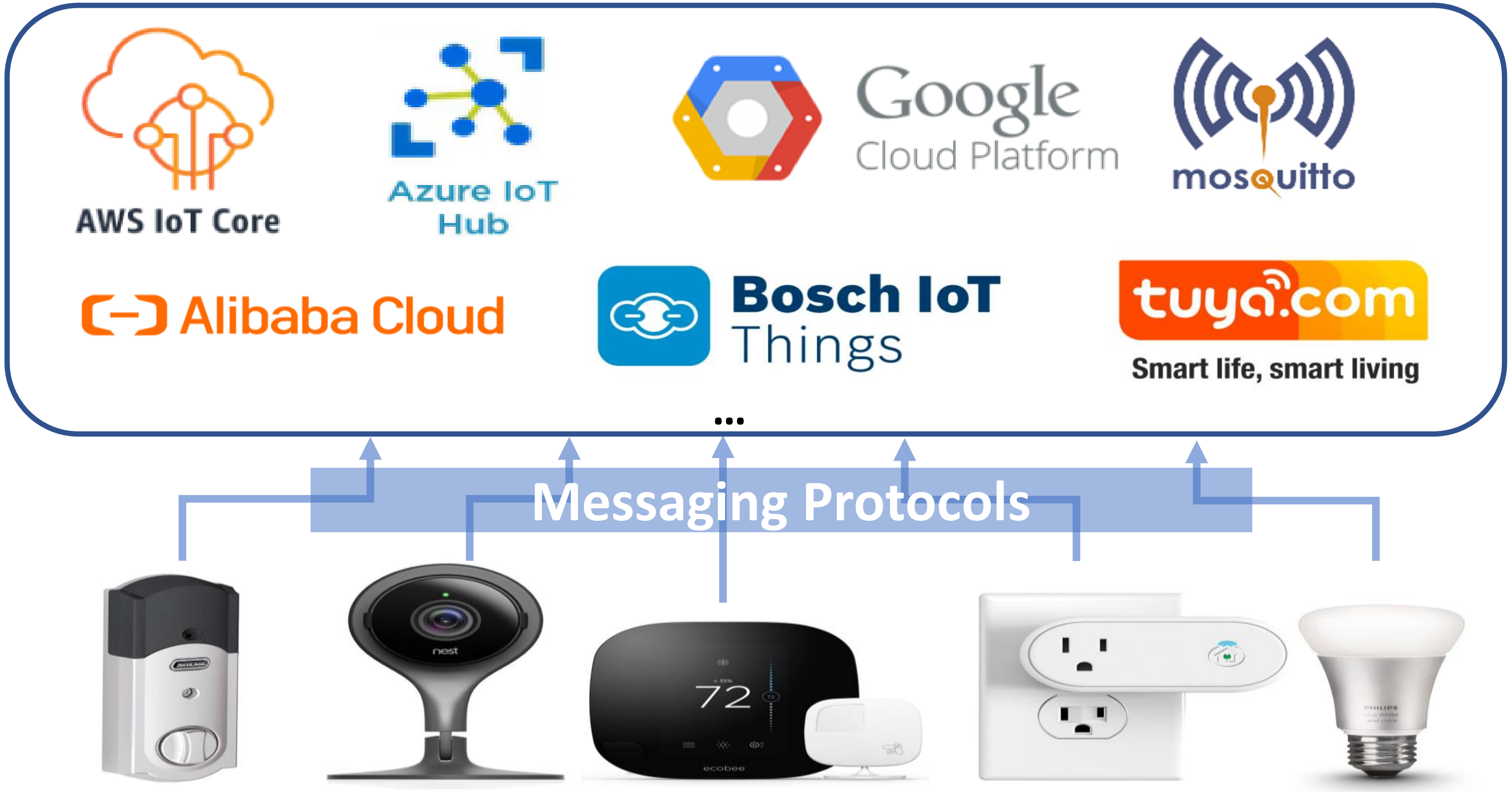
MPInspector: A Systematic and Automatic Approach for Evaluating the Security of IoT Messaging Protocols

Qinying Wang   Shouling Ji   Yuan Tian   Xuhong Zhang   Binbin Zhao
Yuhong Kan   Zhaowei Lin   Changting Lin   Shuiguang Deng   Alex X. Liu   Raheem Beyah

2021

Cloud based IoT Platforms

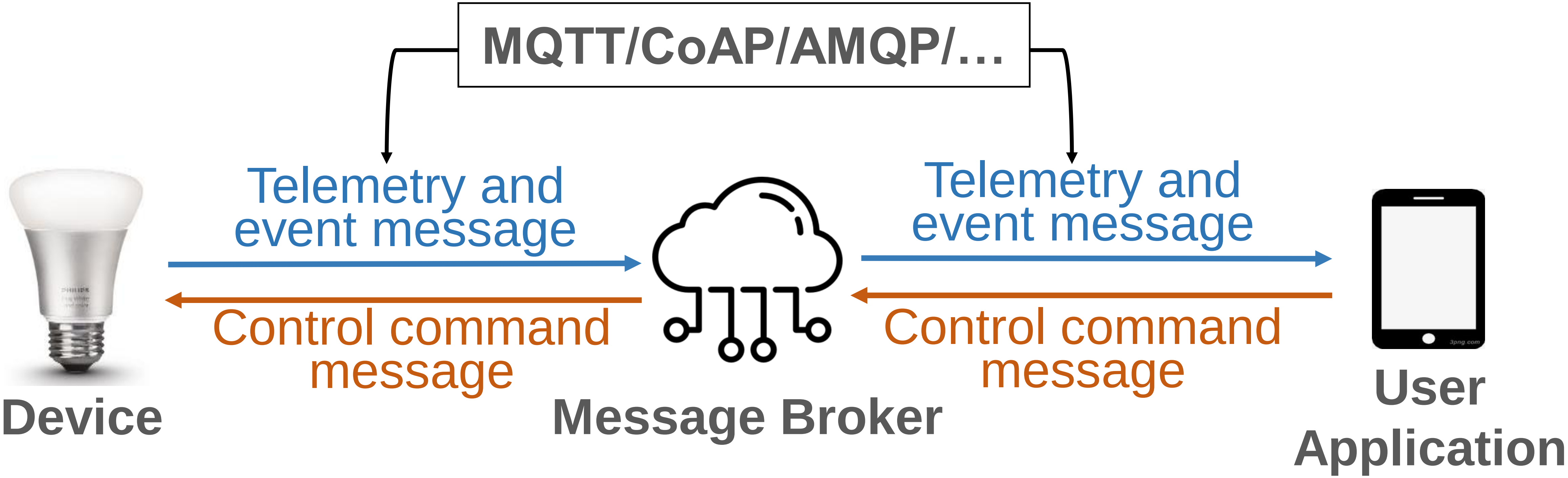
Most IoT platforms offer MP (Messaging Protocol) implementations.



…

Messaging Protocols

Typical Architecture of IoT platform



MQTT/CoAP/AMQP/…

Telemetry and event message

Control command message

Telemetry and event message

Control command message

Device

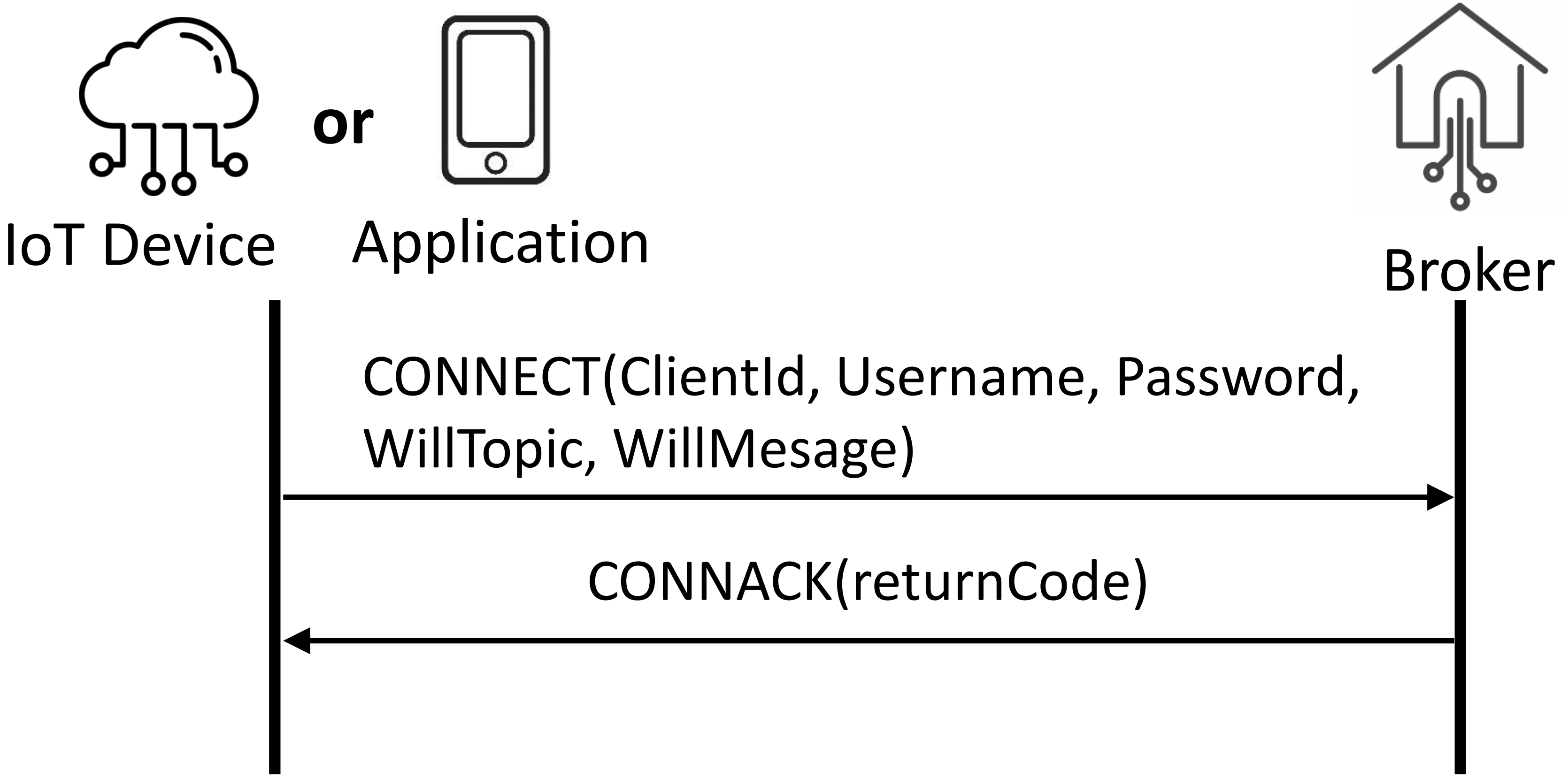
Message Broker

User Application

An MP Workflow

An example workflow of MQTT standards:

CONNECT | SUBSCRIBE | PUBLISH | UNSUBSCRIBE | DISCONNECT



IoT Device or Application

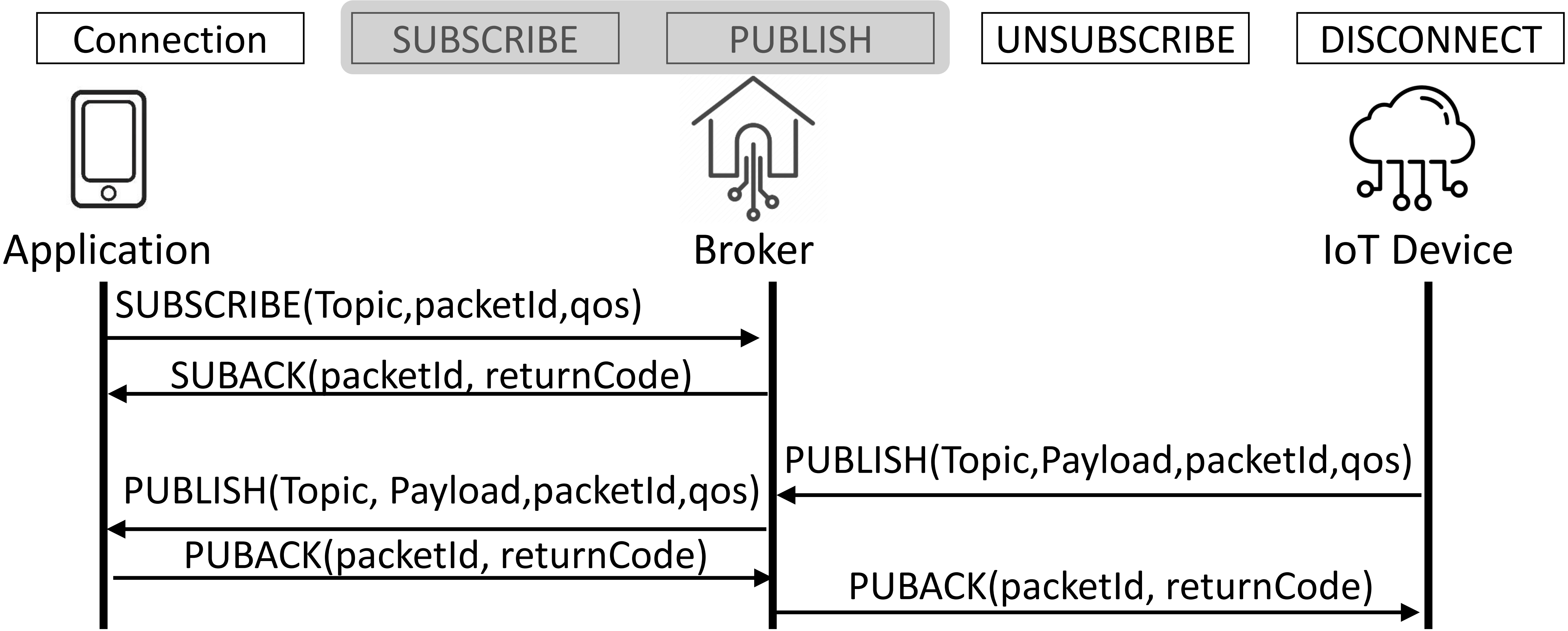
Broker

CONNECT(ClientId, Username, Password, WillTopic, WillMesage)

CONNACK(returnCode)

An MP Workflow

An example workflow of MQTT standards:



Connection

SUBSCRIBE

PUBLISH

UNSUBSCRIBE

DISCONNECT

Application

Broker

IoT Device

SUBSCRIBE(Topic,packetId,qos)

SUBACK(packetId, returnCode)

PUBLISH(Topic,Payload,packetId,qos)

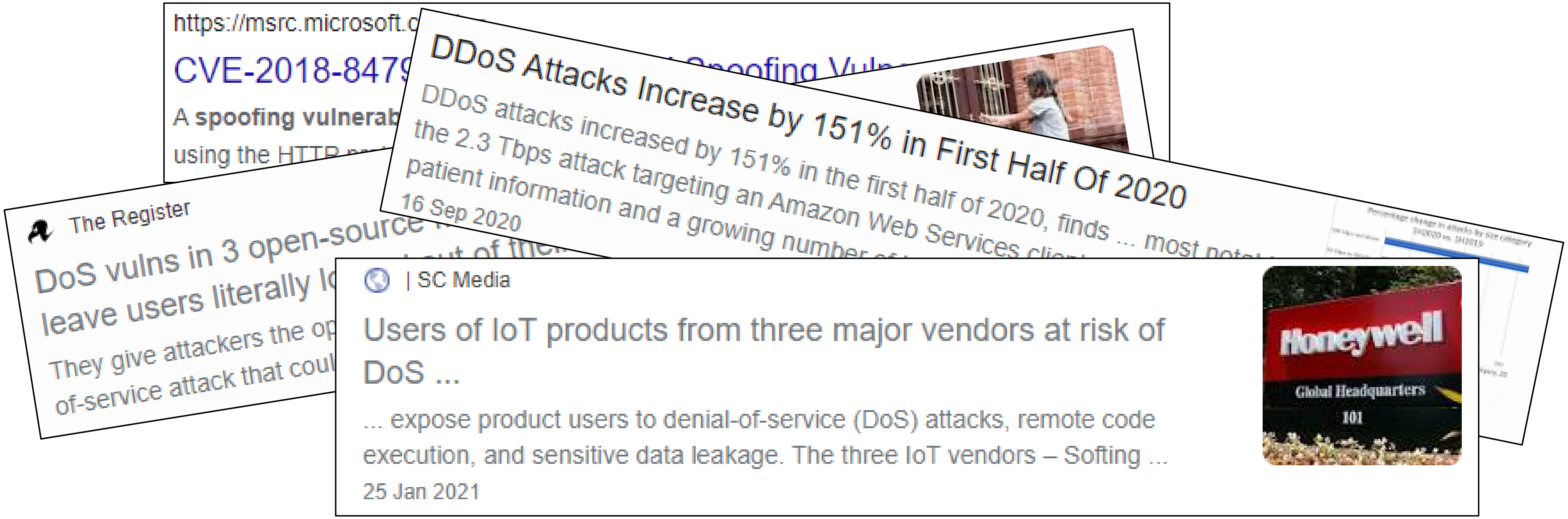
PUBLISH(Topic, Payload,packetId,qos)

PUBACK(packetId, returnCode)

PUBACK(packetId, returnCode)

Security and Privacy Threats on MP

Several MP flaws have been spotted including **denial of service**, **sensitive data theft**, **malicious command injection**, etc.



https://msrc.microsoft.c

CVE-2018-8479 ... Spoofing Vuln

A **spoofing vulnerab**

using the HTTP pr

The Register

DoS vulns in 3 open-source ... out of their

leave users literally lo

They give attackers the op

of-service attack that coul

DDoS Attacks Increase by 151% in First Half Of 2020

DDoS attacks increased by 151% in the first half of 2020, finds ... most nota

the 2.3 Tbps attack targeting an Amazon Web Services clien

patient information and a growing number of

16 Sep 2020

| SC Media

Users of IoT products from three major vendors at risk of DoS ...

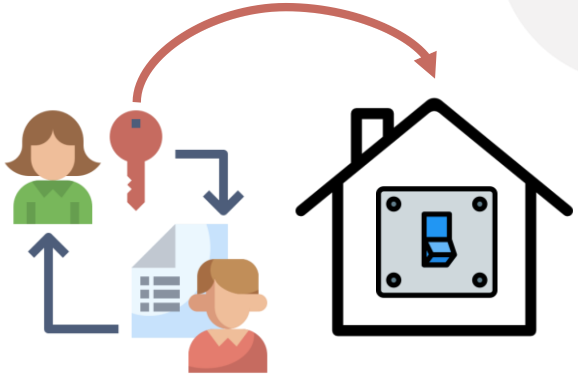
... expose product users to denial-of-service (DoS) attacks, remote code execution, and sensitive data leakage. The three IoT vendors – Softing ...

25 Jan 2021

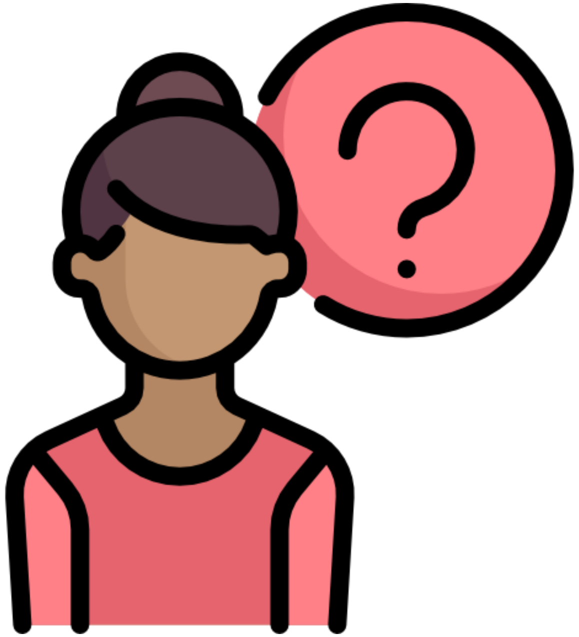
Threat Model

- **Neighbor scenario**
  - The victim and the attacker are in the same network.
  - The attacker can **eavesdrop, drop, modify, inject messages.**

- **Tenant scenario**
  - The victim uses some devices previously being used by the attacker.
  - The attacker can **collect the device identity** (e.g., password).
  - The attacker can **leave a backdoor** on the device.



How to build a **systematic** and **automatic** tool to evaluate the security of IoT MPs?



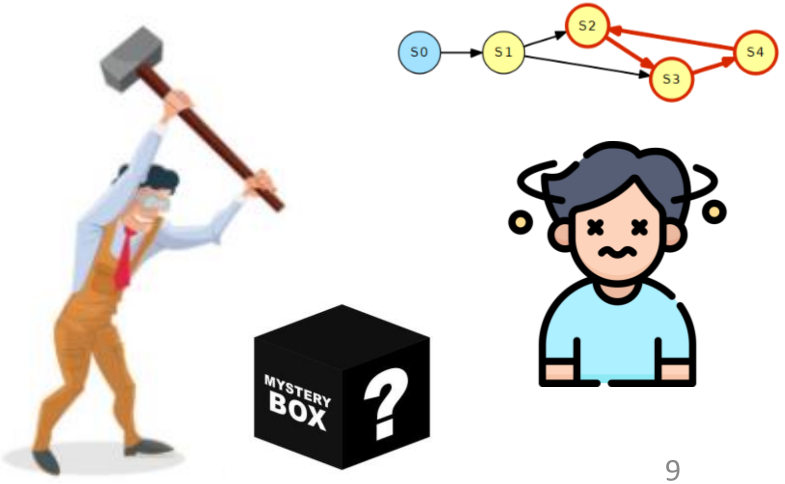
Challenges

¤ **Diverse** and **customized** MP implementations

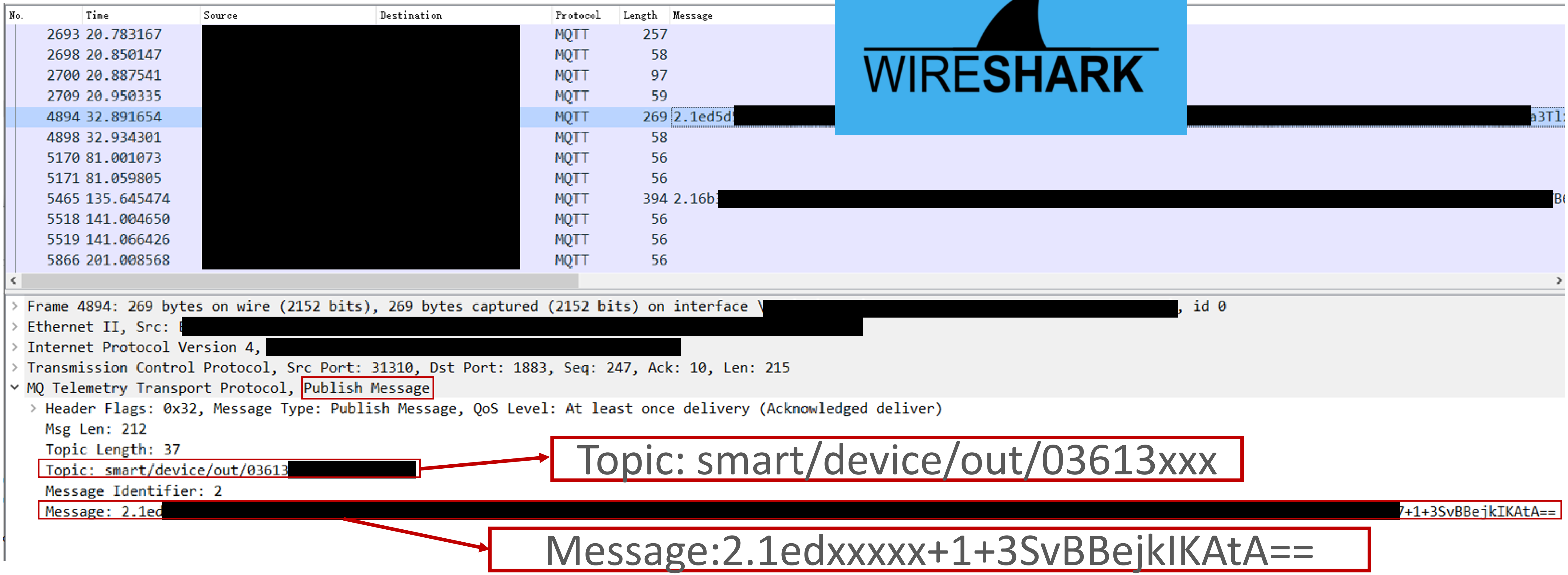
- ✓ Multiple types of MPs
- ✓ Customized implementations on different platforms

¤ Complex and closed-source MP workflow

- ✓ Stateful procedure with multiple messages
- ✓ The implementation are closed-source



Customized MP Implementations



```
No.   Time        Source      Destination   Protocol  Length  Message
2693  20.783167   [redacted]                MQTT      257
2698  20.850147   [redacted]                MQTT       58
2700  20.887541   [redacted]                MQTT       97
2709  20.950335   [redacted]                MQTT       59
4894  32.891654   [redacted]                MQTT      269     2.1ed5d[redacted]a3Tl
4898  32.934301   [redacted]                MQTT       58
5170  81.001073   [redacted]                MQTT       56
5171  81.059805   [redacted]                MQTT       56
5465  135.645474  [redacted]                MQTT      394     2.16b[redacted]B
5518  141.004650  [redacted]                MQTT       56
5519  141.066426  [redacted]                MQTT       56
5866  201.008568  [redacted]                MQTT       56
```

```
> Frame 4894: 269 bytes on wire (2152 bits), 269 bytes captured (2152 bits) on interface [redacted], id 0
> Ethernet II, Src: [redacted]
> Internet Protocol Version 4, [redacted]
> Transmission Control Protocol, Src Port: 31310, Dst Port: 1883, Seq: 247, Ack: 10, Len: 215
v MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x32, Message Type: Publish Message, QoS Level: At least once delivery (Acknowledged deliver)
    Msg Len: 212
    Topic Length: 37
    Topic: smart/device/out/03613[redacted]
    Message Identifier: 2
    Message: 2.1ed[redacted]7+1+3SvBBejkIKAtA==
```

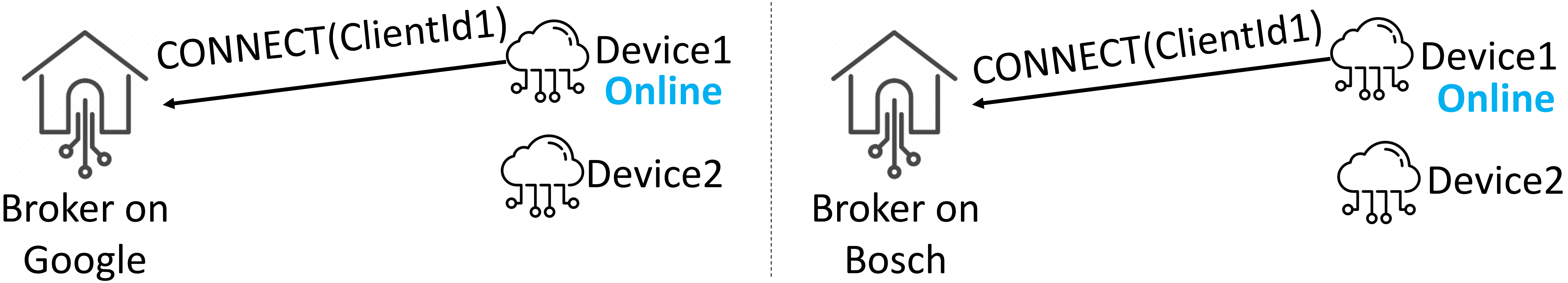
Topic: smart/device/out/03613xxx

Message:2.1edxxxxx+1+3SvBBejkIKAtA==

Customized parameters *Topic* and *Message*

Customized MP Implementations

- MP interaction logic



CONNECT(ClientId1)

Device1 Online

Device2

Broker on Google

CONNECT(ClientId1)

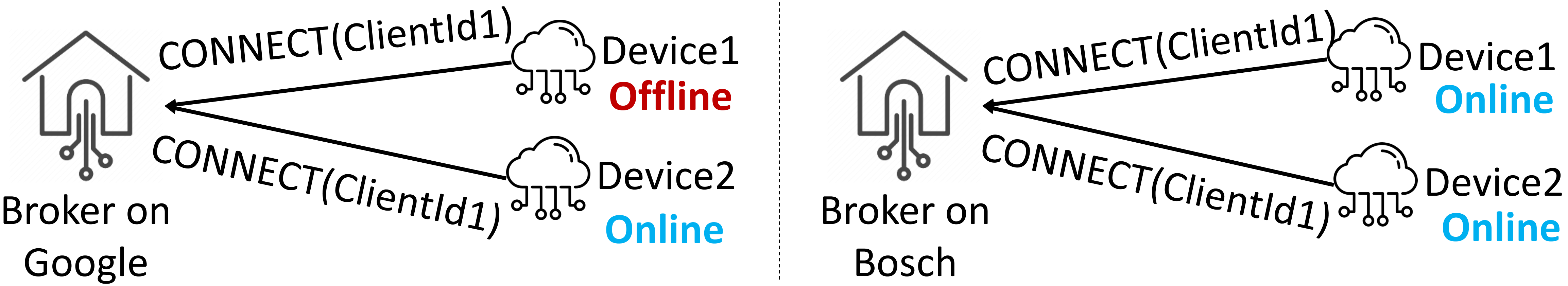
Device1 Online

Device2

Broker on Bosch

Customized MP Implementations

- **MP interaction logic**

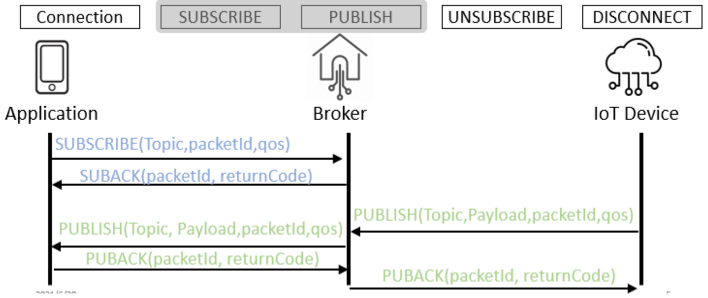


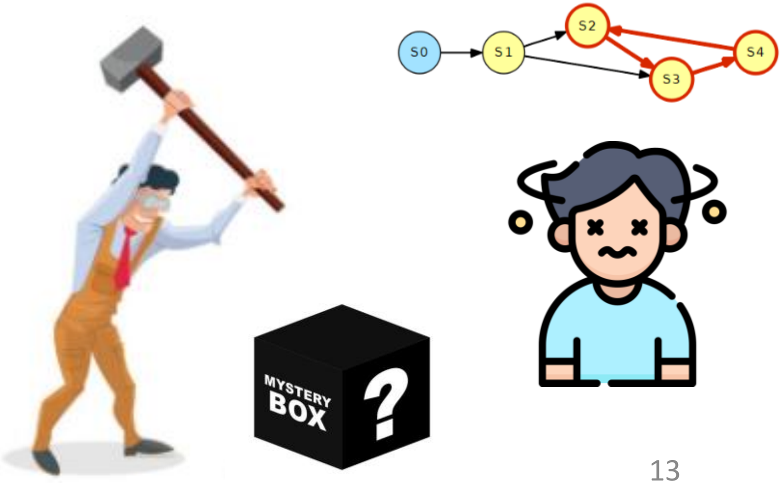
Customized interaction logic on duplicate connections with the same *ClientId*

Challenges

- ¤ Diverse and customized MP implementations
  - ✓ Multiple types of MPs
  - ✓ Customized implementations on different platforms

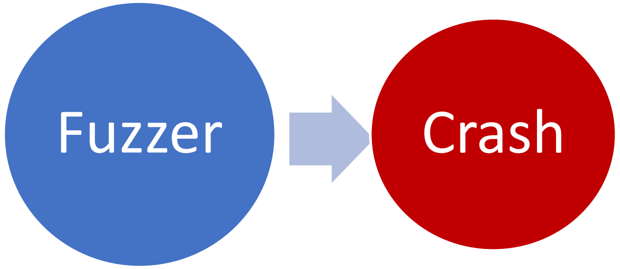
- ¤ **Complex** and **closed-source** MP workflow
  - ✓ Stateful procedure with multiple messages
  - ✓ The implementations are closed-source


Connection
SUBSCRIBE
PUBLISH
UNSUBSCRIBE
DISCONNECT
Application
Broker
IoT Device
SUBSCRIBE(Topic,packetId,qos)
SUBACK(packetId, returnCode)
PUBLISH(Topic, Payload,packetId,qos)
PUBACK(packetId, returnCode)
PUBLISH(Topic,Payload,packetId,qos)
PUBACK(packetId, returnCode)

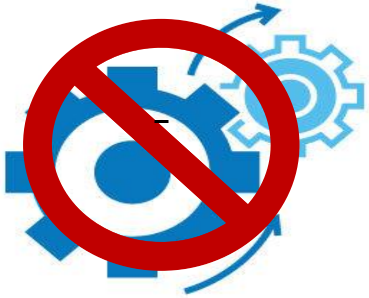


Limitations of Existing Attack Finding Strategies for IoT Protocols

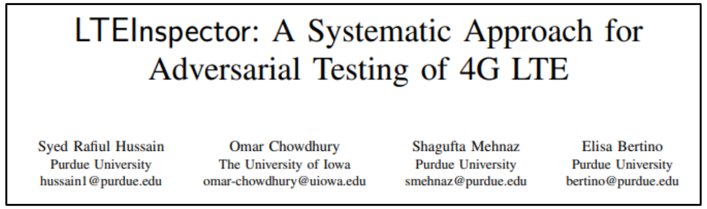
Few analysis on implementation, mostly analyze the specification



Few logic vulnerabilities which do not cause crashes

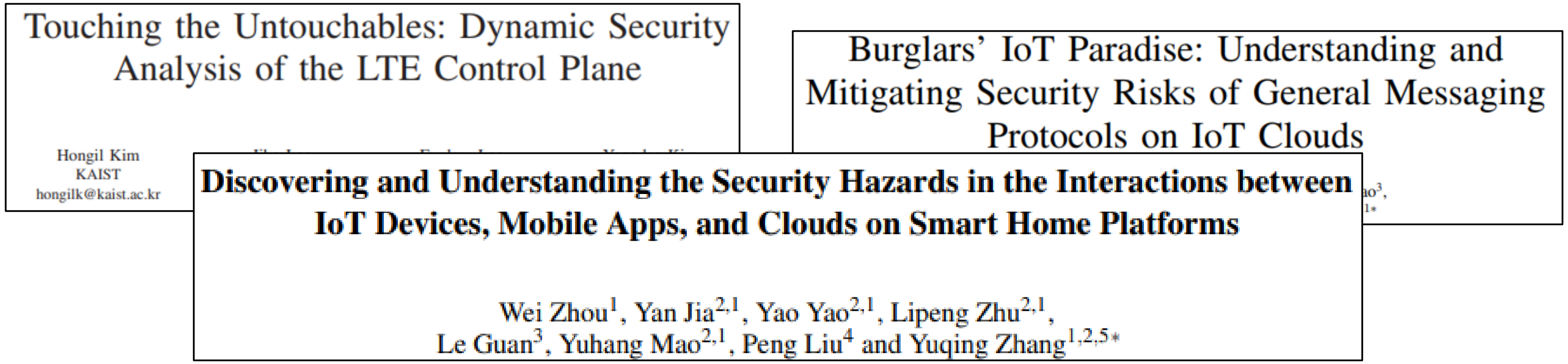


No systematic and automatic approach



LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE

Syed Rafiul Hussain, Purdue University, hussain1@purdue.edu
Omar Chowdhury, The University of Iowa, omar-chowdhury@uiowa.edu
Shagufta Mehnaz, Purdue University, smehnaz@purdue.edu
Elisa Bertino, Purdue University, bertino@purdue.edu



Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane

Hongil Kim
KAIST
hongilk@kaist.ac.kr

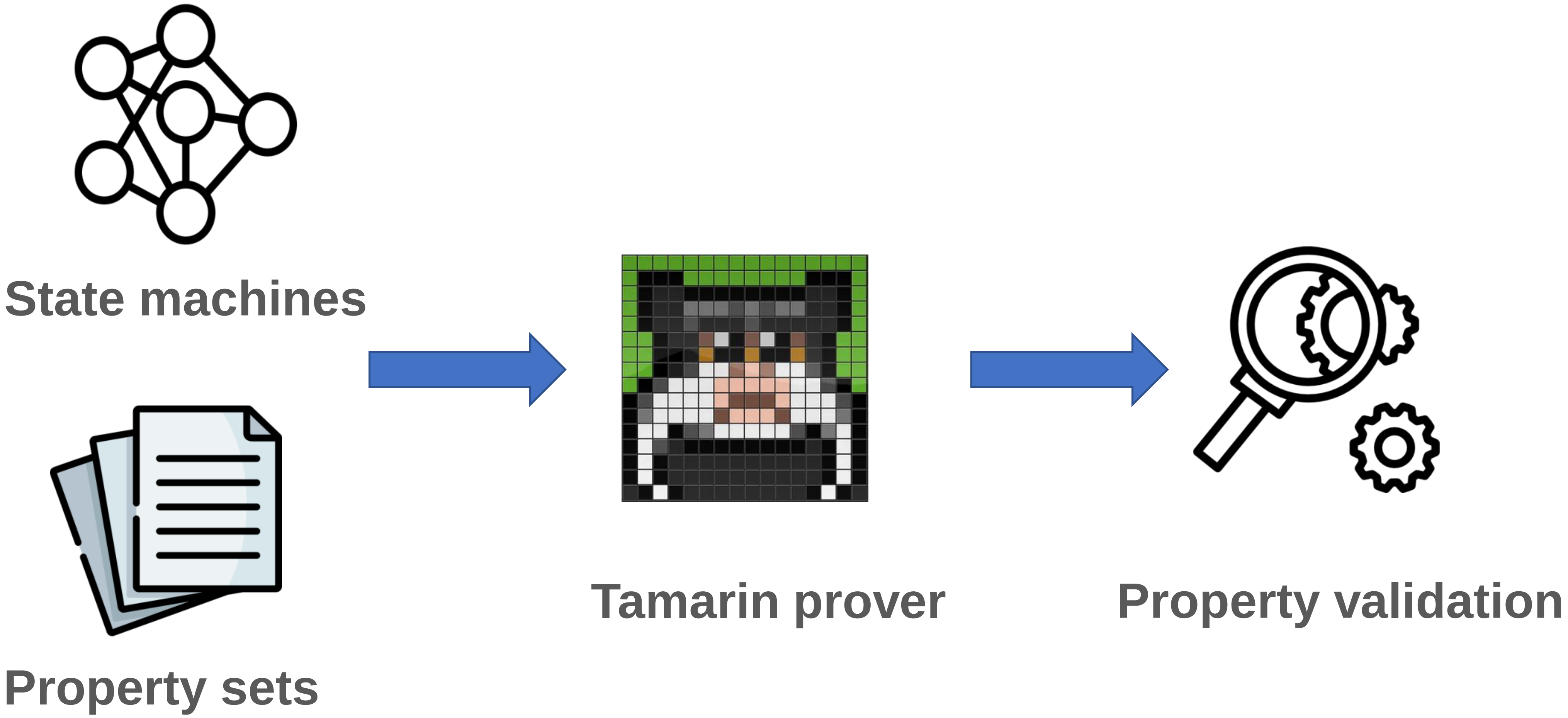
Burglars' IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds

Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms

Wei Zhou[1], Yan Jia[2,1], Yao Yao[2,1], Lipeng Zhu[2,1],
Le Guan[3], Yuhang Mao[2,1], Peng Liu[4] and Yuqing Zhang[1,2,5*]

Insight

A **property-driven** and **model-based** testing philosophy.



State machines

Property sets

Tamarin prover

Property validation

Insight

A **property-driven** and model-based testing philosophy.

- Secrecy properties extracted from the specification
  - ✓ A set of **parameters** from messages that should be confidential

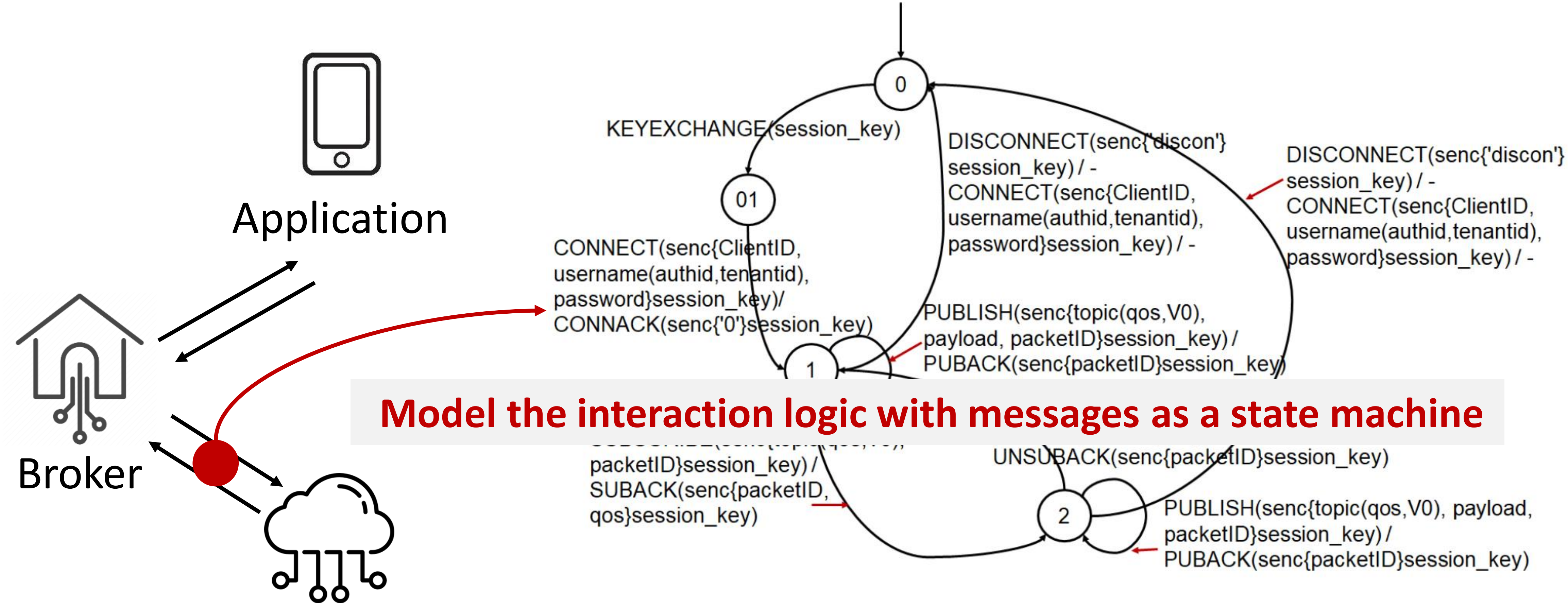
SecProp_Set={ClientId, Password, PUBLISH payload, …}

- Authentication properties extracted from the specification
  - ✓ A set of **messages** that should be authenticated

AuthProp_Ser={CONNECT, PUBLISH, SUBSCRIBE, …}

Insight

A property-driven and **model-based** testing philosophy.



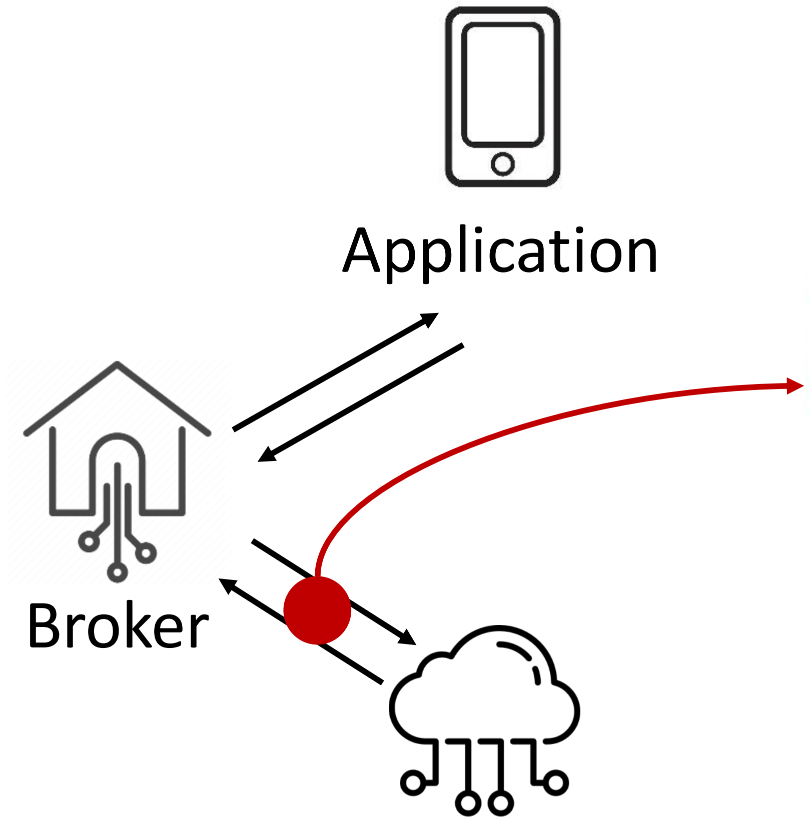
Application
Broker
IoT Device
0
KEYEXCHANGE(session_key)
01
CONNECT(senc{ClientID, username(authid,tenantid), password}session_key)/ CONNACK(senc{'0'}session_key)
DISCONNECT(senc{'discon'} session_key) / - CONNECT(senc{ClientID, username(authid,tenantid), password}session_key) / -
DISCONNECT(senc{'discon'} session_key) / - CONNECT(senc{ClientID, username(authid,tenantid), password}session_key) / -
1
PUBLISH(senc{topic(qos,V0), payload, packetID}session_key) / PUBACK(senc{packetID}session_key)
packetID}session_key) / SUBACK(senc{packetID, qos}session_key)
UNSUBACK(senc{packetID}session_key)
2
PUBLISH(senc{topic(qos,V0), payload, packetID}session_key) / PUBACK(senc{packetID}session_key)

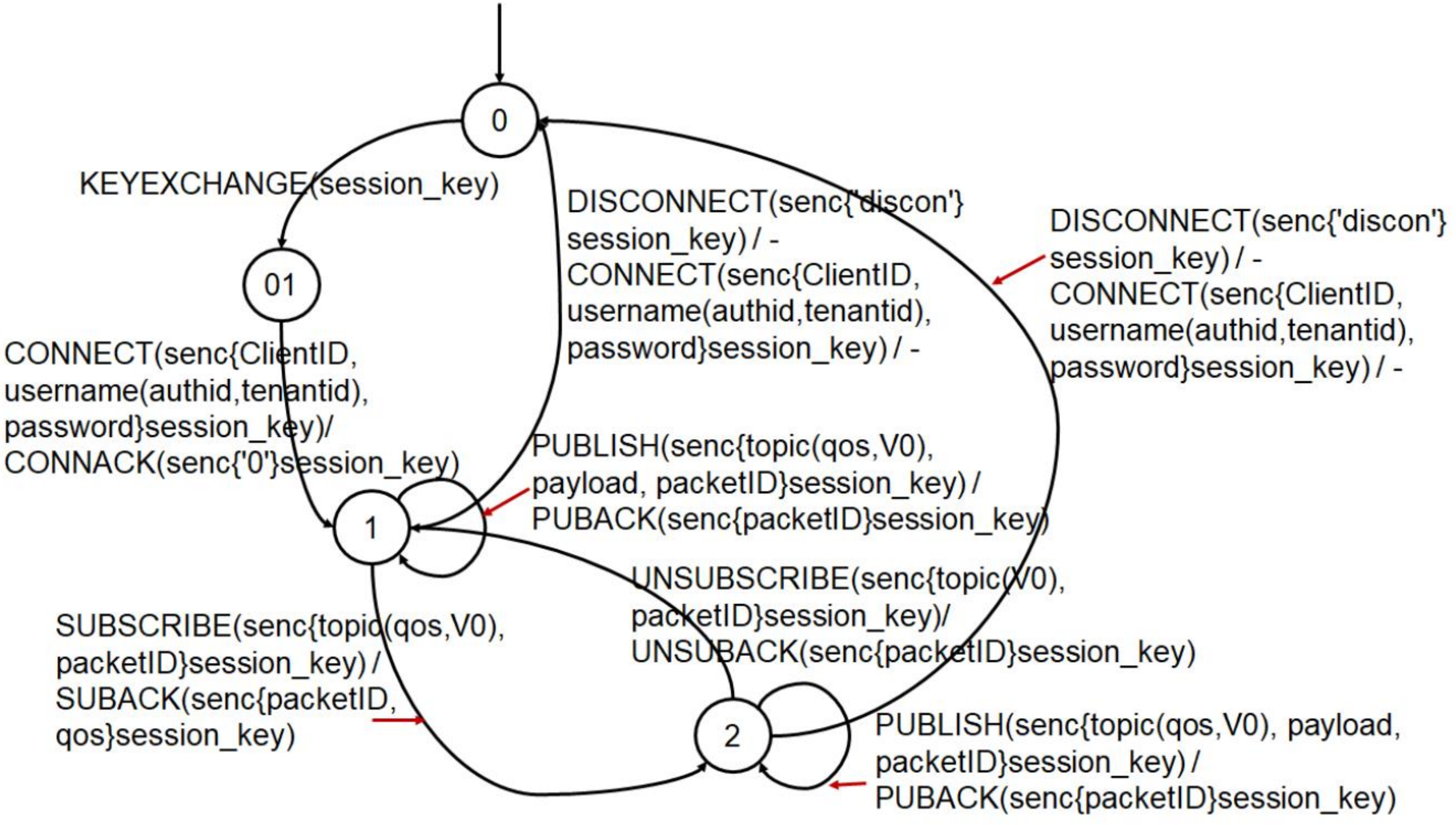
Model the interaction logic with messages as a state machine

State machine of the broker and device

Insight

A property-driven and **model-based** testing philosophy.




0

KEYEXCHANGE(session_key)

01

CONNECT(senc{ClientID,
username(authid,tenantid),
password}session_key)/
CONNACK(senc{'0'}session_key)

DISCONNECT(senc{'discon'}
session_key) / -
CONNECT(senc{ClientID,
username(authid,tenantid),
password}session_key) / -

DISCONNECT(senc{'discon'}
session_key) / -
CONNECT(senc{ClientID,
username(authid,tenantid),
password}session_key) / -

1

PUBLISH(senc{topic(qos,V0),
payload, packetID}session_key) /
PUBACK(senc{packetID}session_key)

UNSUBSCRIBE(senc{topic(V0),
packetID}session_key)/
UNSUBACK(senc{packetID}session_key)

SUBSCRIBE(senc{topic(qos,V0),
packetID}session_key) /
SUBACK(senc{packetID,
qos}session_key)

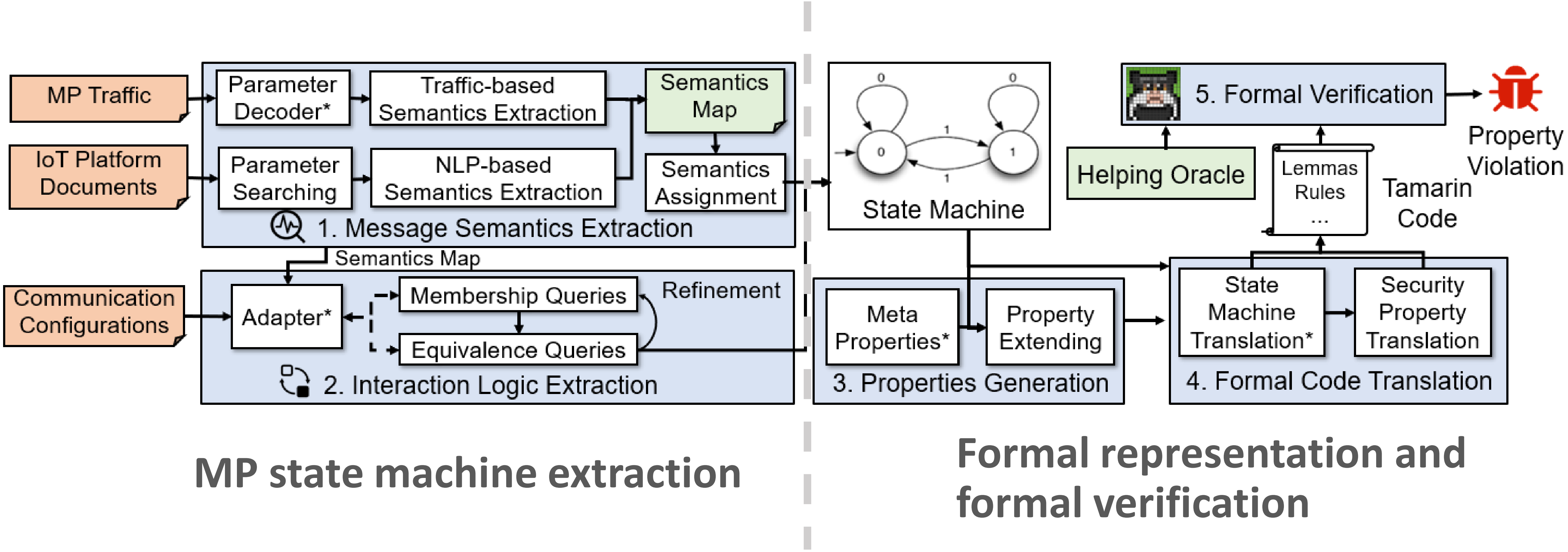
2

PUBLISH(senc{topic(qos,V0), payload,
packetID}session_key) /
PUBACK(senc{packetID}session_key)

State machine of the broker and device

Overview of MPInspector

MPInspector has 5 modules and 3 inputs.

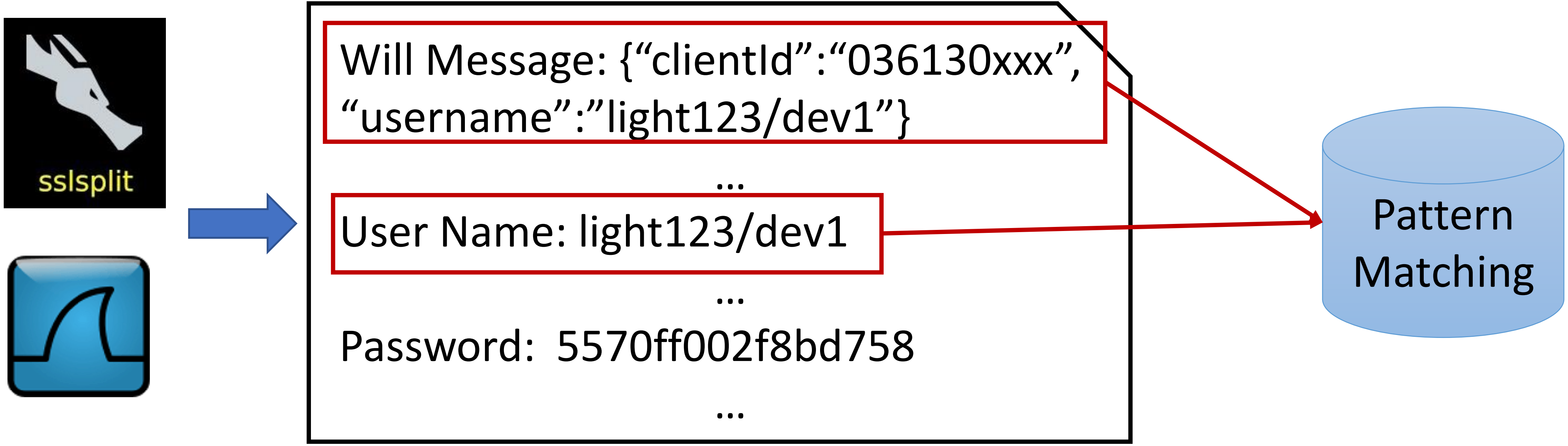


MP state machine extraction

Formal representation and formal verification

Message Semantics Extraction Workflow

- ¤ **Extracting the customized message semantics is not trivial.**
  - ✓ Traffic and document based analysis
  - ✓ Patter matching & NLP



Will Message: {“clientId”:“036130xxx”, “username”:”light123/dev1”}

…

User Name: light123/dev1

…

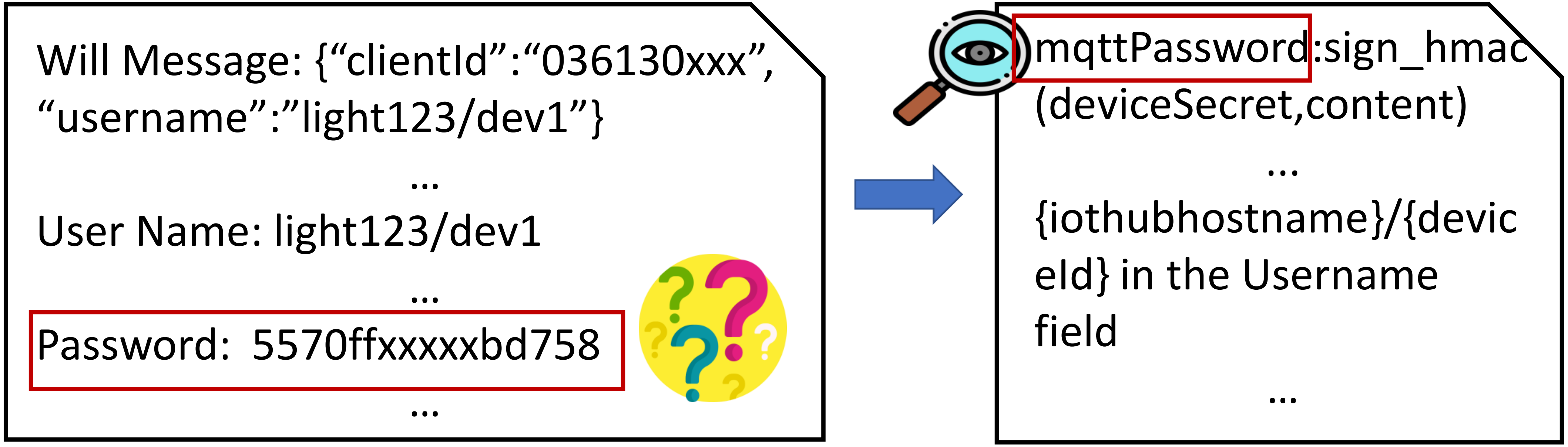
Password: 5570ff002f8bd758

…

Traffic file

Message Semantics Extraction

NLP assisted IoT platform documents analysis.



Will Message: {“clientId”:“036130xxx”, “username”:”light123/dev1”}

…

User Name: light123/dev1

…

Password: 5570ffxxxxxbd758

…

mqttPassword:sign_hmac (deviceSecret,content)

…

{iothubhostname}/{deviceId} in the Username field

…

Traffic file

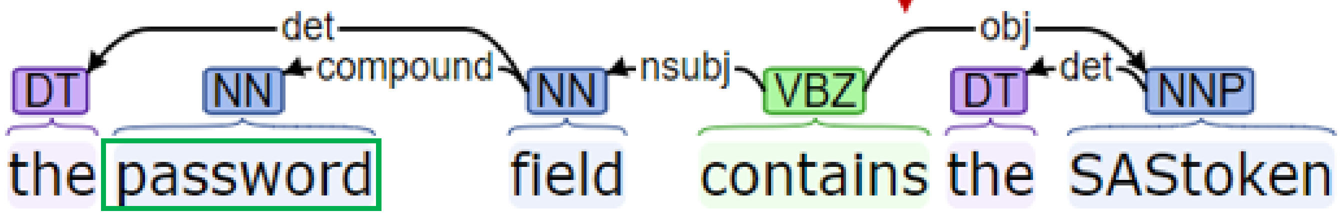
IoT platform documents

Message Semantics Extraction

NLP assisted IoT platform documents analysis.

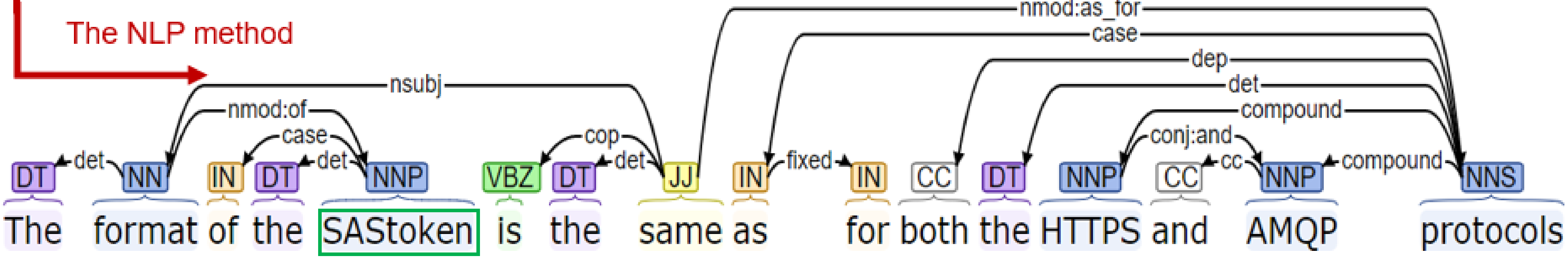
(1) mqttPassword sign_hmac(deviceSecret,timestamp,rawpassword) → Pattern matching

(2) the password field contains the SAStoken. → The NLP method



(3) The format of the SAStoken is the same as for both the HTTPS and AMQP protocols: SharedAccessSignatures={signature-string}&se={expiry}&sr={URL-encoded-resourceURI} → Pattern matching

The NLP method

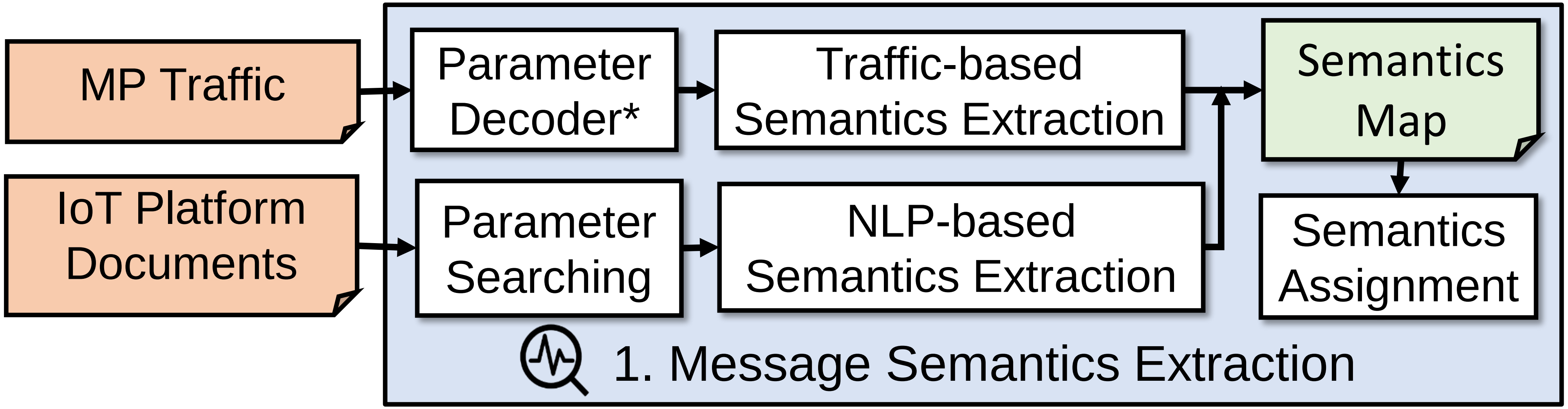


Message Semantics Extraction Workflow

¤ **Extracting the customized message semantics is not trivia.**

✓ Traffic and document based analysis

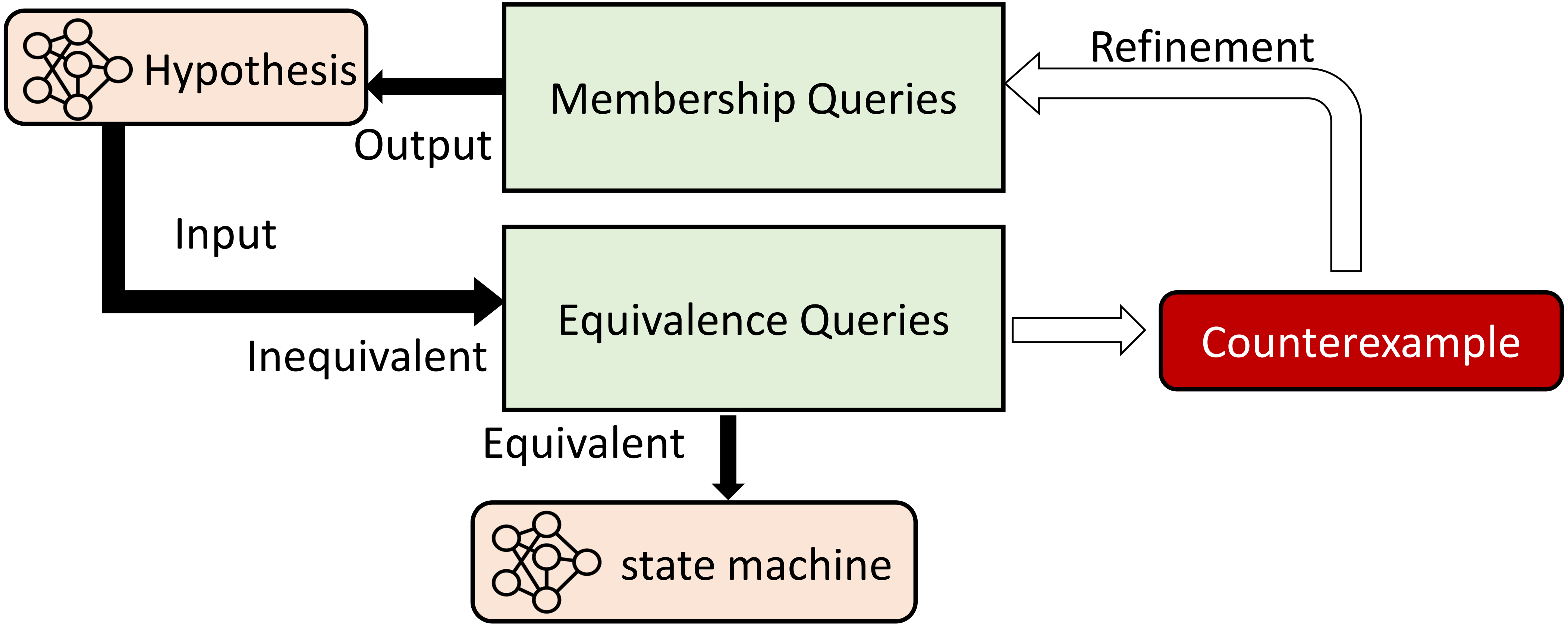
✓ Patter matching & NLP



EXP: {“CONNECT”:{ “ClientID”:“”,“username”:{“composition”:[“authid”, “tenantid”]},“password”:{“encryption”:”HMAC”, “encryptedTerms”:[“authid”,”timestamp”]}}}

Interaction Logic Extraction

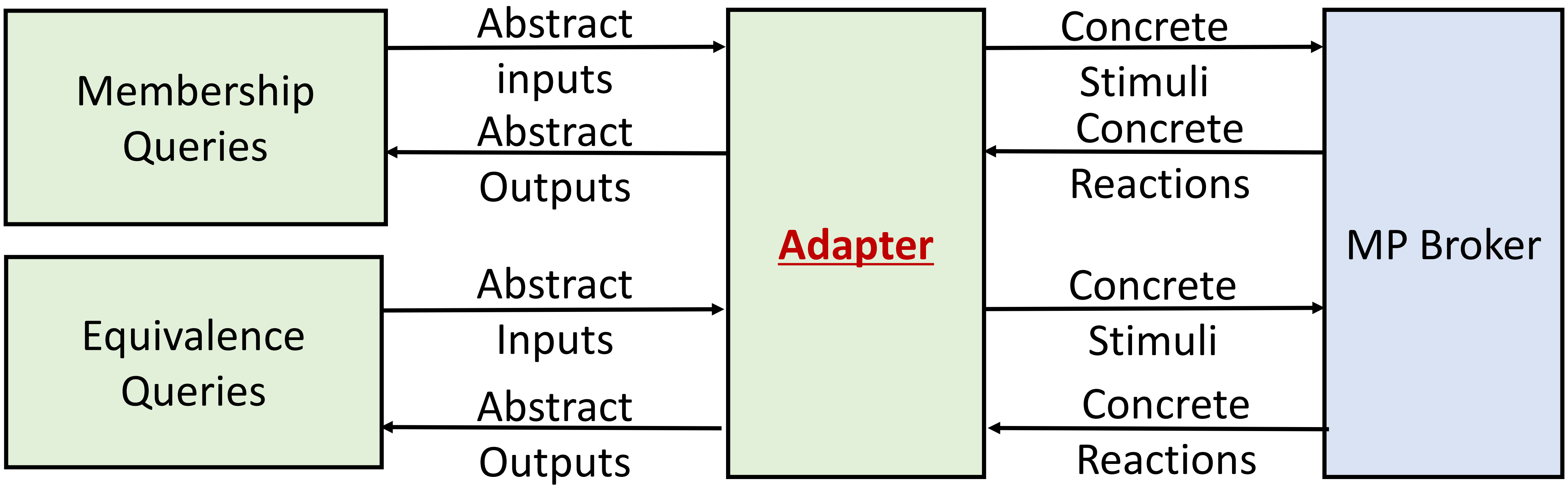
Apply active model learning to infer the interaction logic.



Interaction Logic Extraction

¤ **Only supports two parties and response messages.**

- ✓ Extend the adapter to support multi-parties and monitoring the connection state



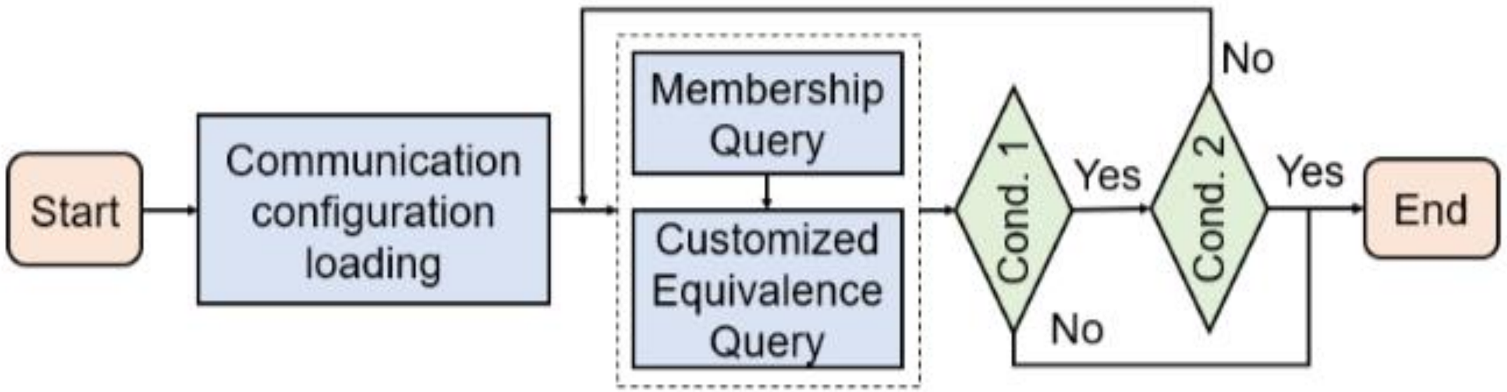
Sequences of message types

Sequences of real messages

Interaction Logic Extraction Workflow

- **The equivalence query is time-consuming while dealing multitype of messages.**
  - A customized equivalence algorithm to cut down unnecessary queries

- **Model learning may be trapped into an endless procedure;**
  - An enhance learning algorithm

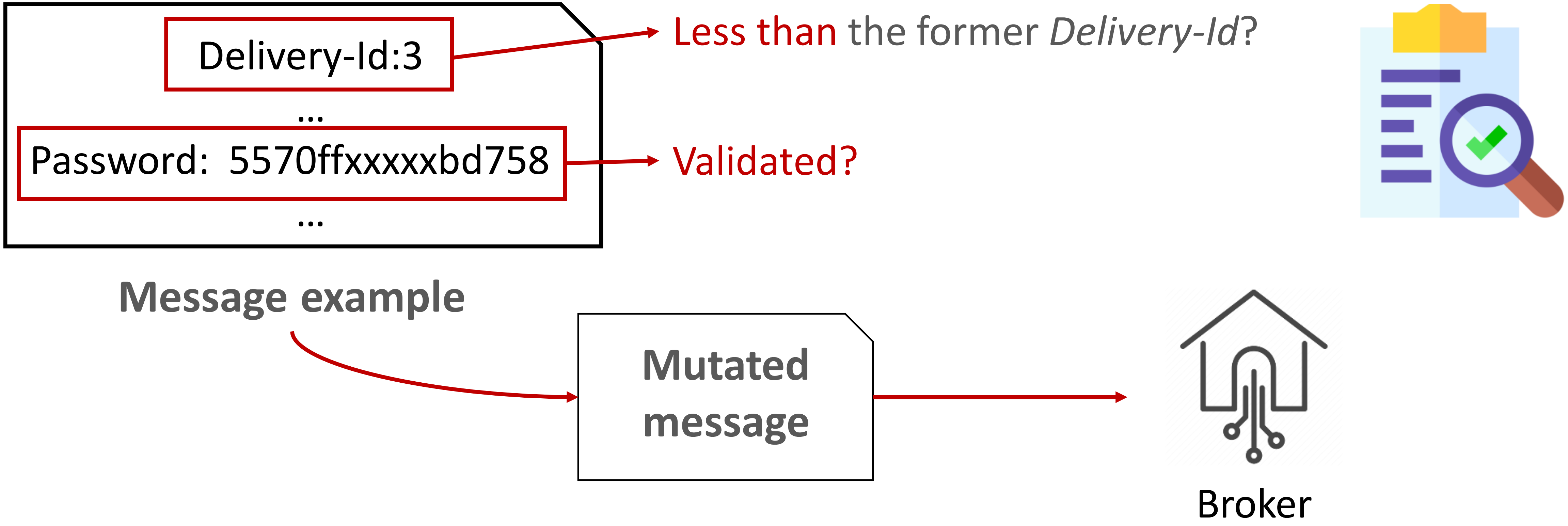


Start
Communication configuration loading
Membership Query
Customized Equivalence Query
Cond. 1
Yes
No
Cond. 2
No
Yes
End

◆ Cond. 1: If an counterexample is found?
◆ Cond. 2: Is the number of same hypotheses greater than the threshold?

Interaction Logic Extraction

Modeling validity predicates.



Delivery-Id:3

…

Password: 5570ffxxxxxbd758

…

Less than the former *Delivery-Id*?

Validated?

Message example

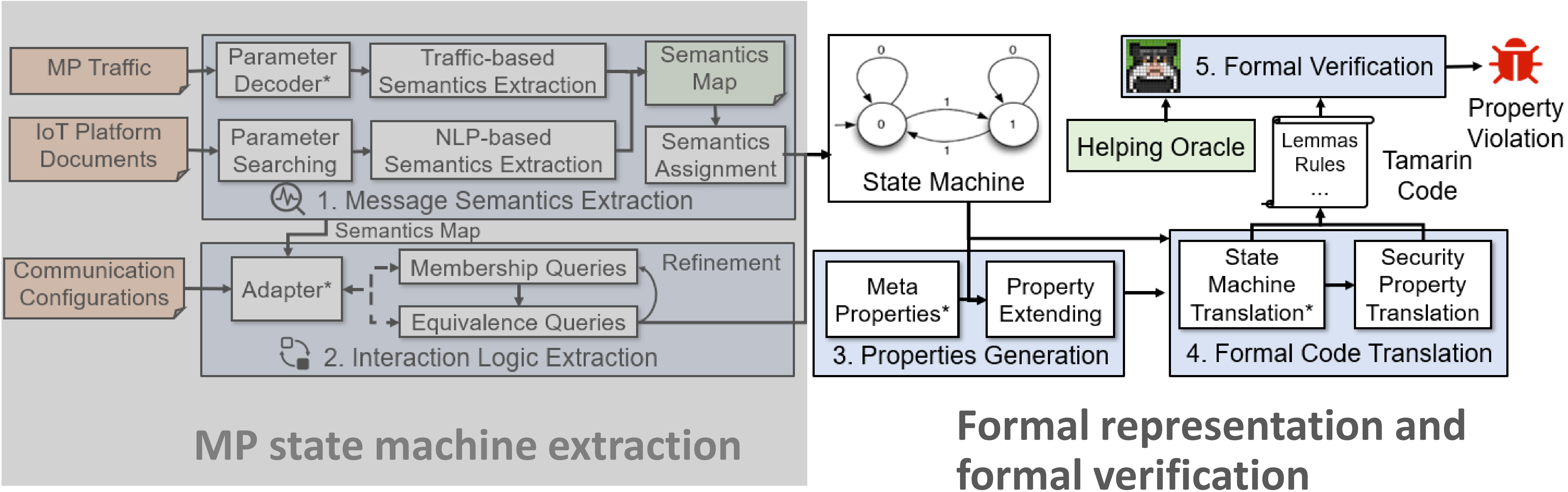
Mutated message

Broker

Validity predicate testing by sending mutated message to the broker

Overview of MPInspector

MPInspector performs formal representation and formal verification.



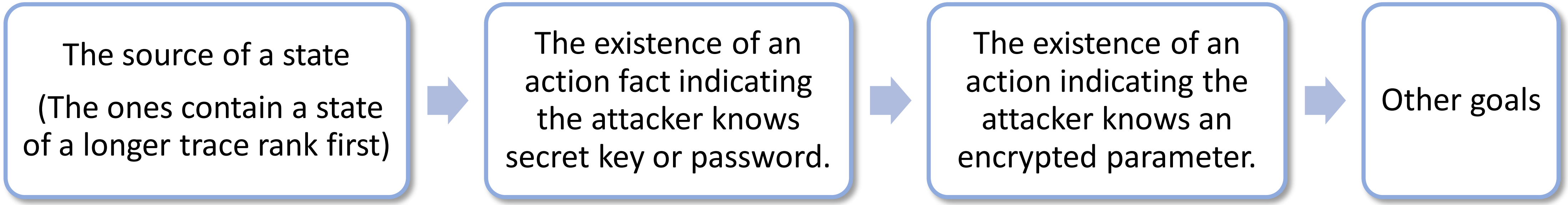
MP state machine extraction

Formal representation and formal verification

Formal Verification

- **The search space of possible states may potentially explode.**
  - ✓ An inherent limitation of Tamarin Prover

- ✓ Helping oracle ranks the open goals based on our strategies



The source of a state

(The ones contain a state of a longer trace rank first)

The existence of an action fact indicating the attacker knows secret key or password.

The existence of an action indicating the attacker knows an encrypted parameter.

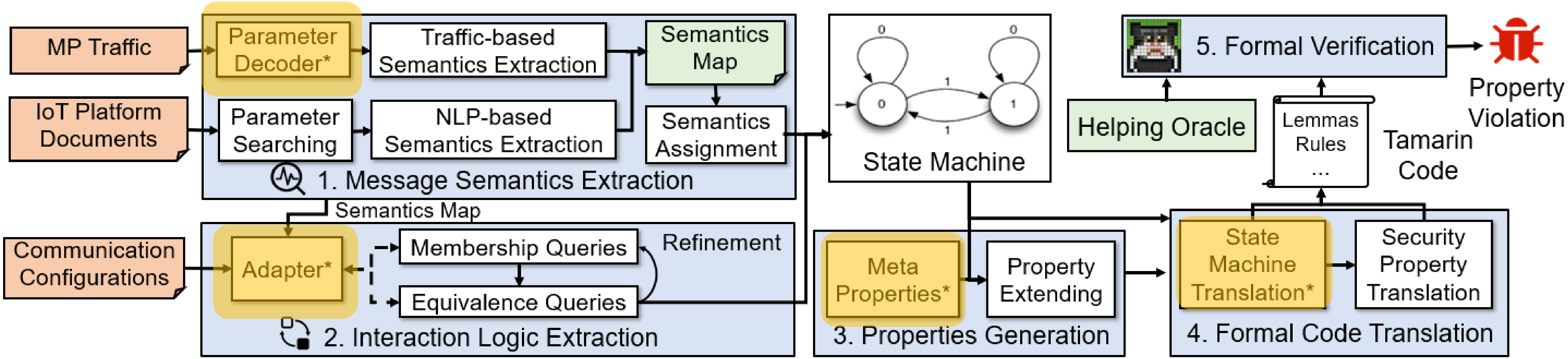
Other goals

New Extension for New Types of MP

¤ **A one-short effort for each new MP type**

✓ Message structure, meta properties and initial state of MP

✓ Concluded from the MP specification



Evaluations

Experiment settings

¤ Experiment settings

✓ Test **ten** MP implementations from **nine** leading IoT platforms

Google Cloud Platform

Microsoft Azure

aws

tuya.com Smart life, smart living

mosquitto

Bosch IoT Things

Alibaba Cloud

Apache ACTIVEMQ

MQTT V3.1.1

MQTT V5.0

AMQP V1.0

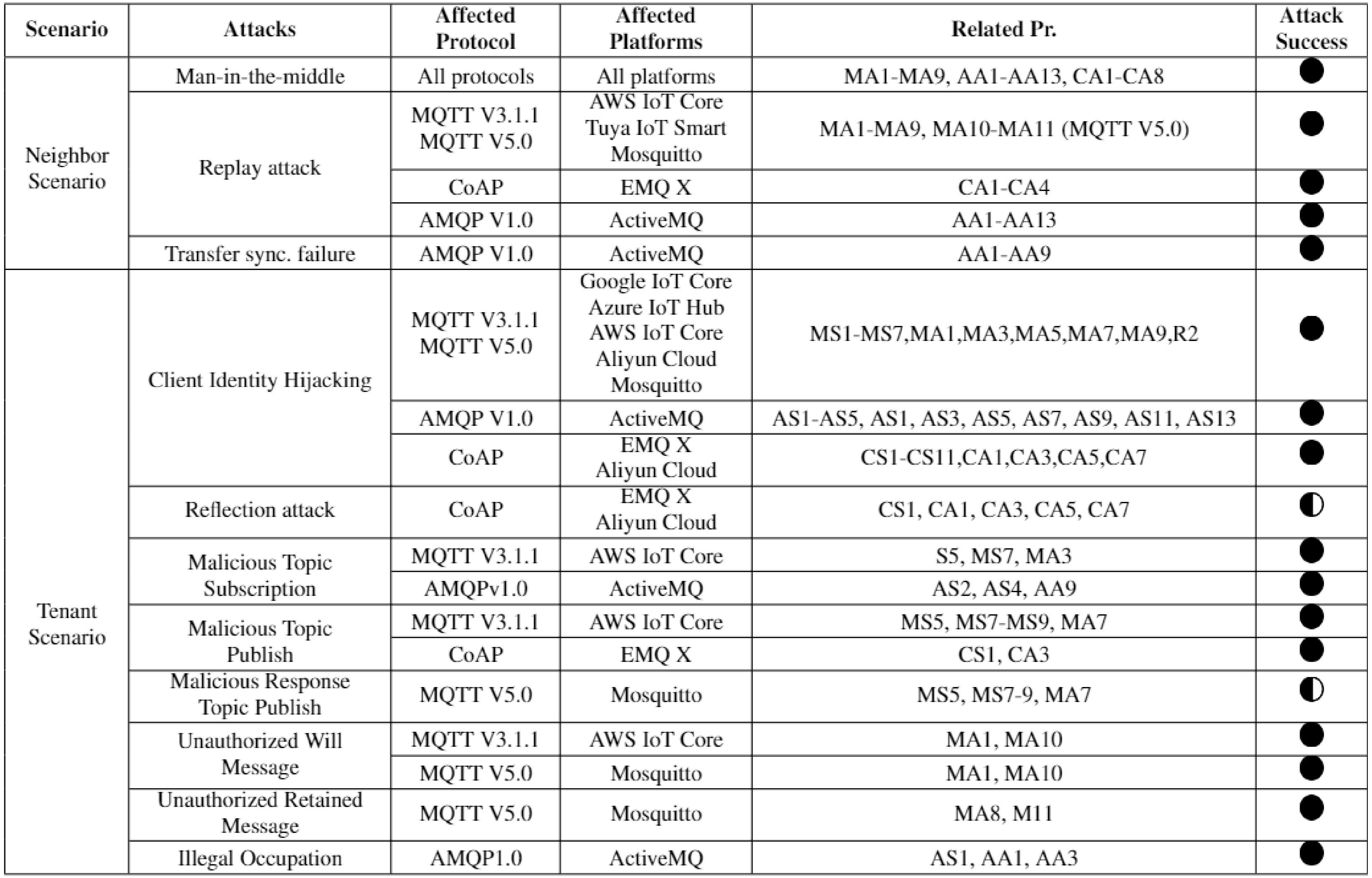
CoAP

✓ Test the SaaS applications on our own services

✓ Validate our attack on our own devices

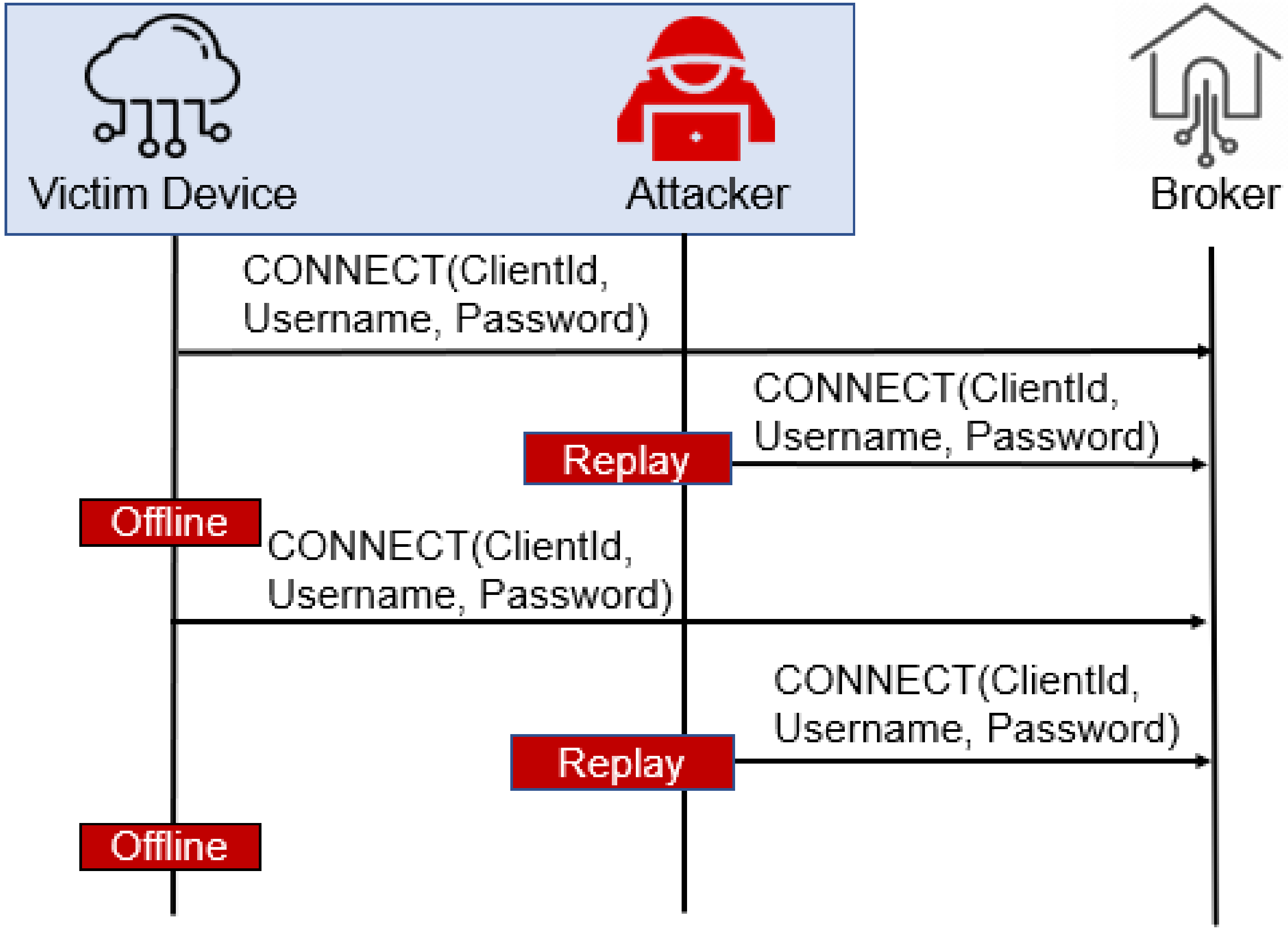
Findings

¤ **Uncovered 11 types of MP attacks**

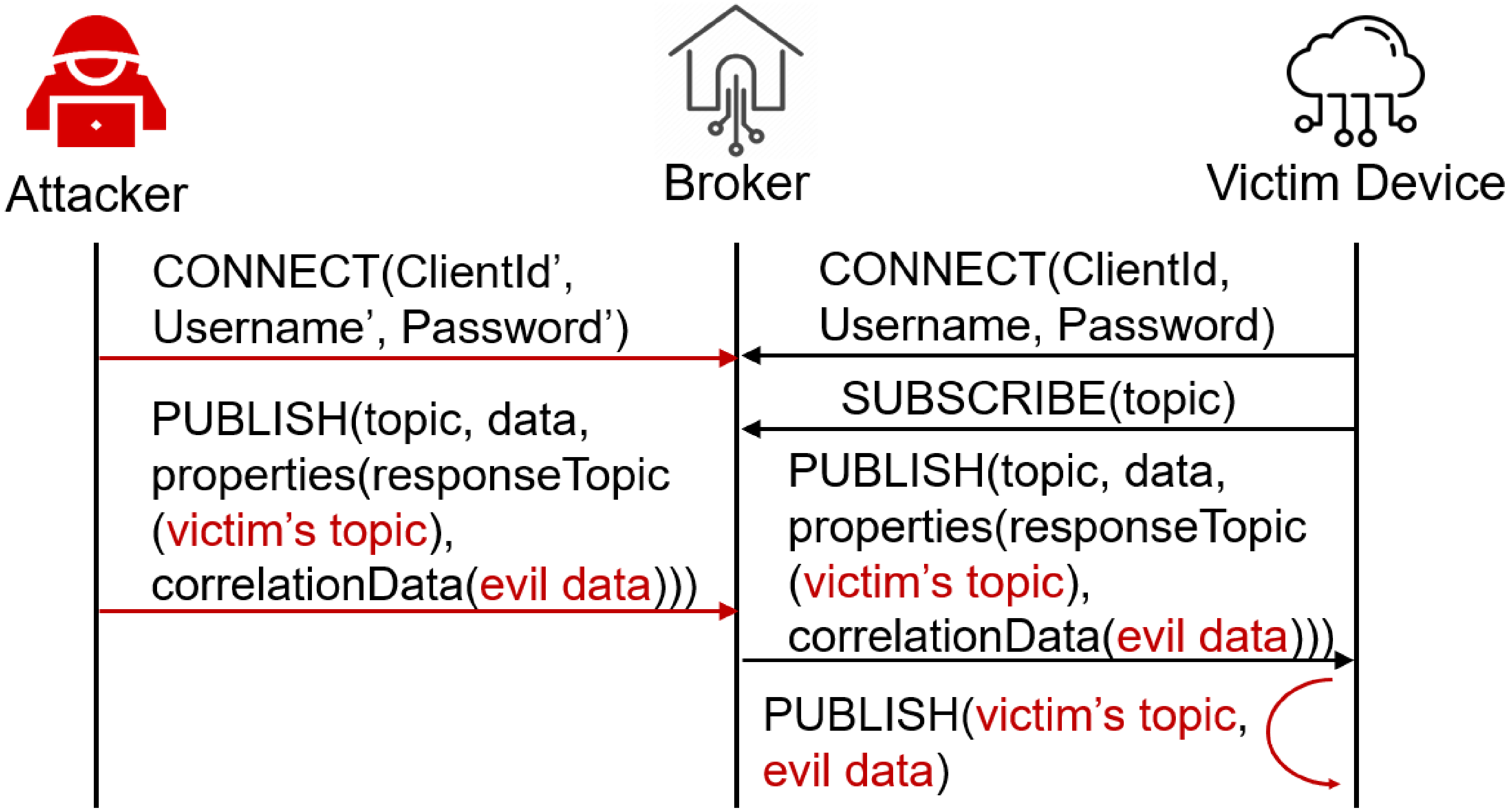


Scenario	Attacks	Affected Protocol	Affected Platforms	Related Pr.	Attack Success
Neighbor Scenario	Man-in-the-middle	All protocols	All platforms	MA1-MA9, AA1-AA13, CA1-CA8	●
	Replay attack	MQTT V3.1.1 MQTT V5.0	AWS IoT Core Tuya IoT Smart Mosquitto	MA1-MA9, MA10-MA11 (MQTT V5.0)	●
		CoAP	EMQ X	CA1-CA4	●
		AMQP V1.0	ActiveMQ	AA1-AA13	●
	Transfer sync. failure	AMQP V1.0	ActiveMQ	AA1-AA9	●
Tenant Scenario	Client Identity Hijacking	MQTT V3.1.1 MQTT V5.0	Google IoT Core Azure IoT Hub AWS IoT Core Aliyun Cloud Mosquitto	MS1-MS7,MA1,MA3,MA5,MA7,MA9,R2	●
		AMQP V1.0	ActiveMQ	AS1-AS5, AS1, AS3, AS5, AS7, AS9, AS11, AS13	●
		CoAP	EMQ X Aliyun Cloud	CS1-CS11,CA1,CA3,CA5,CA7	●
	Reflection attack	CoAP	EMQ X Aliyun Cloud	CS1, CA1, CA3, CA5, CA7	◐
	Malicious Topic Subscription	MQTT V3.1.1	AWS IoT Core	S5, MS7, MA3	●
		AMQPv1.0	ActiveMQ	AS2, AS4, AA9	●
	Malicious Topic Publish	MQTT V3.1.1	AWS IoT Core	MS5, MS7-MS9, MA7	●
		CoAP	EMQ X	CS1, CA3	●
	Malicious Response Topic Publish	MQTT V5.0	Mosquitto	MS5, MS7-9, MA7	◐
	Unauthorized Will Message	MQTT V3.1.1	AWS IoT Core	MA1, MA10	●
		MQTT V5.0	Mosquitto	MA1, MA10	●
	Unauthorized Retained Message	MQTT V5.0	Mosquitto	MA8, M11	●
	Illegal Occupation	AMQP1.0	ActiveMQ	AS1, AA1, AA3	●

Denial of Service (Neighbor Scenario)


Victim Device
Attacker
Broker
CONNECT(ClientId, Username, Password)
Replay
CONNECT(ClientId, Username, Password)
Offline
CONNECT(ClientId, Username, Password)
Replay
CONNECT(ClientId, Username, Password)
Offline

Unauthorized Response Topic publish (Tenant Scenario)



Attacker

Broker

Victim Device

CONNECT(ClientId’, Username’, Password’)

CONNECT(ClientId, Username, Password)

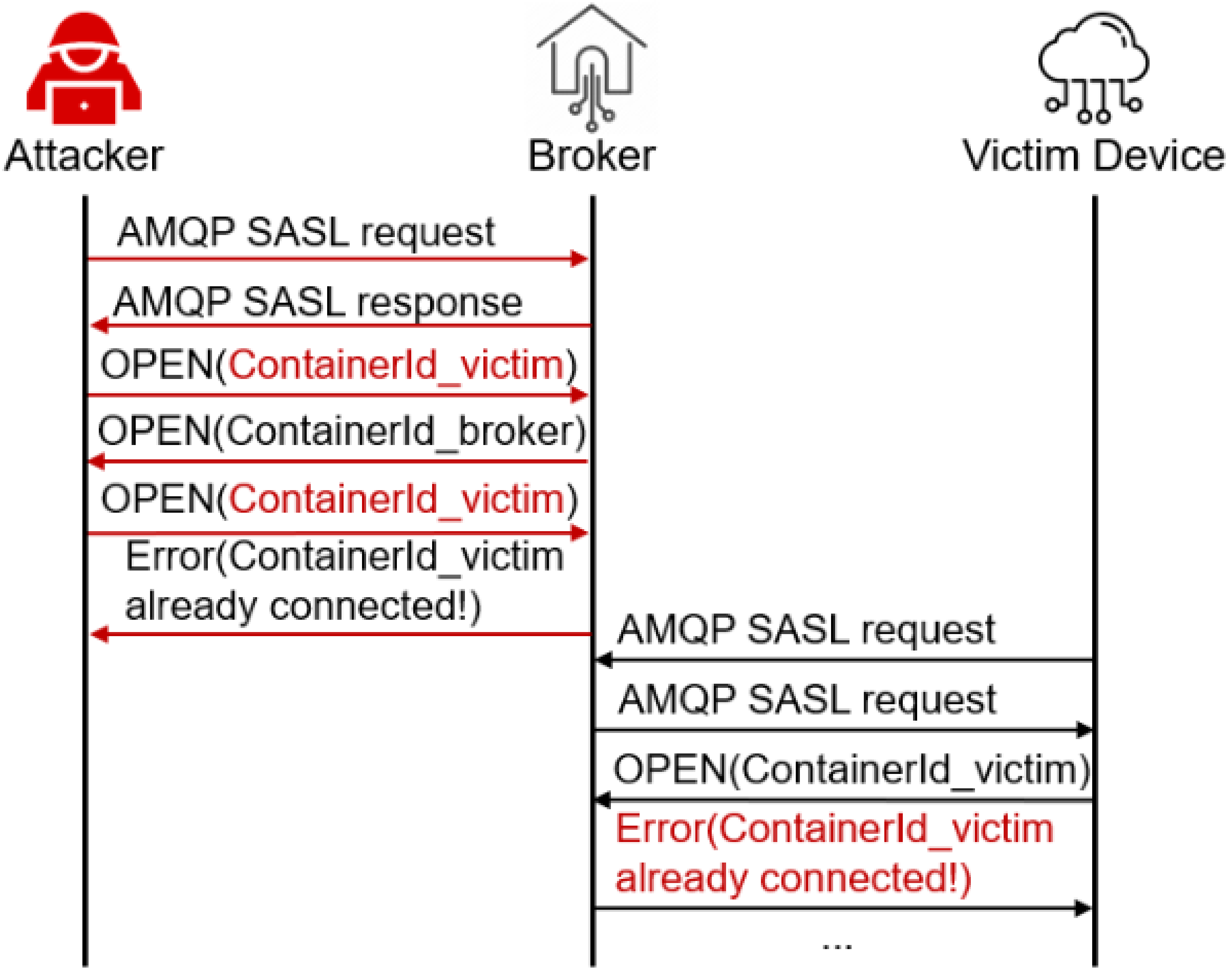
SUBSCRIBE(topic)

PUBLISH(topic, data, properties(responseTopic (victim’s topic), correlationData(evil data)))

PUBLISH(topic, data, properties(responseTopic (victim’s topic), correlationData(evil data)))

PUBLISH(victim’s topic, evil data)

AMQP illegal occupation (Tenant Scenario)



Attacker

Broker

Victim Device

AMQP SASL request

AMQP SASL response

OPEN(ContainerId_victim)

OPEN(ContainerId_broker)

OPEN(ContainerId_victim)

Error(ContainerId_victim already connected!)

AMQP SASL request

AMQP SASL request

OPEN(ContainerId_victim)

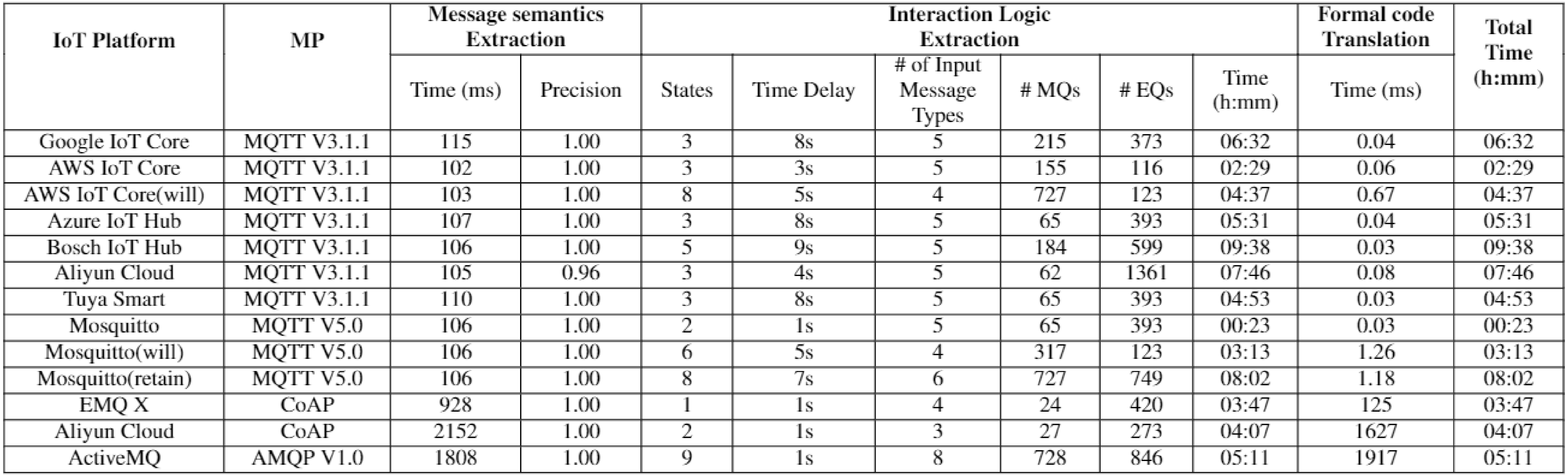
Error(ContainerId_victim already connected!)

...

Performance

¤ **The overhead of MPInspector**

- ✓ The average precision of property violations is 1.00
- ✓ The average overhead is ~4.5 hours

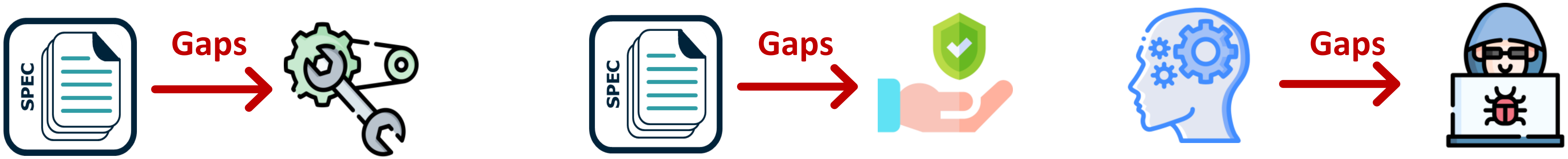


IoT Platform	MP	Message semantics Extraction		Interaction Logic Extraction						Formal code Translation	Total Time (h:mm)
		Time (ms)	Precision	States	Time Delay	# of Input Message Types	# MQs	# EQs	Time (h:mm)	Time (ms)	
Google IoT Core	MQTT V3.1.1	115	1.00	3	8s	5	215	373	06:32	0.04	06:32
AWS IoT Core	MQTT V3.1.1	102	1.00	3	3s	5	155	116	02:29	0.06	02:29
AWS IoT Core(will)	MQTT V3.1.1	103	1.00	8	5s	4	727	123	04:37	0.67	04:37
Azure IoT Hub	MQTT V3.1.1	107	1.00	3	8s	5	65	393	05:31	0.04	05:31
Bosch IoT Hub	MQTT V3.1.1	106	1.00	5	9s	5	184	599	09:38	0.03	09:38
Aliyun Cloud	MQTT V3.1.1	105	0.96	3	4s	5	62	1361	07:46	0.08	07:46
Tuya Smart	MQTT V3.1.1	110	1.00	3	8s	5	65	393	04:53	0.03	04:53
Mosquitto	MQTT V5.0	106	1.00	2	1s	5	65	393	00:23	0.03	00:23
Mosquitto(will)	MQTT V5.0	106	1.00	6	5s	4	317	123	03:13	1.26	03:13
Mosquitto(retain)	MQTT V5.0	106	1.00	8	7s	6	727	749	08:02	1.18	08:02
EMQ X	CoAP	928	1.00	1	1s	4	24	420	03:47	125	03:47
Aliyun Cloud	CoAP	2152	1.00	2	1s	3	27	273	04:07	1627	04:07
ActiveMQ	AMQP V1.0	1808	1.00	9	1s	8	728	846	05:11	1917	05:11

Discussion

Discussion

- **Mitigate security risks**



Gaps

Gaps

Gaps

Specification Implementation

Constraint Resource

Security Requirements

Security design

Adversary environment

- **Limitation and future work**
  - ✓ Fine-grained testing and more flexible model learning strategies
  - ✓ Automatic meta property extraction based on NLP
  - ✓ Applying MPInspector on more MPs and devices

Summary

Summary

¤ The **first systematic and automatic** framework for evaluating the security of MP implementations.

¤ A **large-scale experiment** on 3 popular MPs on 9 leading IoT platforms.

¤ Uncover **11 kinds of** attacks.

¤ https://github.com/wqqqy/MPInspector

Q&A

wangqinying@zju.edu.cn