Supporting the Security Modelling in Operational Technology by identifying capacities of Hidden Channels in ICS protocols

Robert Altschaffel, Sönke Otten, Stefan Kiltz, Jana Dittmann
Otto-von-Guericke University of Magdeburg
Magdeburg, Germany
e-mail: firstname.lastname@ovgu.de

***Abstract*—A domain affected by a rising threat landscape is Operational Technology (OT) and adjacent cyber-physical domains like Internet of Things (IoT). With the increased threat to OT, the need for security modelling in this domain grows. Security modelling requires reliable data about attack vectors and threats for optimal results. In this paper, we want to contribute to this data by presenting our approach to model the capacity of hidden channels in the OT-domain. This modelling is based on the systematic exploration of hidden channels in the MQTT network protocol, the creation of dedicated tools and the resulting data sets in order to investigate the capacity of these hidden channels within a control network. The approach is used to identify the capacity for 5 hidden channels usable in the MQTT protocol.**

Keywords-security modeling; iot; hidden channels.

I. Introduction

Industrial Control Systems (ICS) are facing rising threats, as shown by the trends identified in [1]. ICS are also, commonly referred to as Operational Technology (OT), are characterized by directly affecting physical processes. Hence, when an ICS is attacked, the result might endanger the safety of all the entities in the sphere of influence of the respective ICS. Actors posing as Advanced and Persistent Threats (APTs) play a significant role in this threat scenario (as demonstrated in [1]).

Advanced threat actors also use advanced techniques like hidden channels for aspects like malware infiltration, exfiltration of confidential data or command and control. This is exemplified by the inclusion of the technique *Data Obfuscation: Steganography* in the MITRE ATT&CK Matrix [2] and continuing overhauls. Another example is the widely-spread malware campaign SteganoAmor [3].

To protect against this threat, awareness and correct identification of the extent of danger originating from a threat are required. This includes the modelling of security and threats. Such security modelling requires reliable data about attack vectors and threats for optimal results. In this paper, we want to contribute to this data by presenting our approach to model the capacity of hidden channels in the OT-domain. This modelling is based on the systematic exploration of hidden channels in the Message Queuing Telemetry Transport (MQTT) network protocol [4] widely used in OT, the creation of dedicated tools and the resulting data sets in order to investigate the capacity of these hidden channels within a control network. The approach identifies the capacity for 5 hidden channels usable in the MQTT protocol and discusses countermeasures against these channels.

This paper is structured as follows: Section II will present some fundamentals. Section III describes the approach to identify the capacity of hidden channels as a foundation for security modelling after providing an overview on hidden channels in MQTT. Section IV discusses the technical implementation of the capacity measurement. Section V presents the results of the theoretical and practical considerations. Section VI discusses countermeasures against the hidden channels presented in this work. Section VII provides a summary and an outlook on future work.

II. State of the art

This section provides a brief overview on the terminology of hidden channels, hidden channels in ICS protocols and a brief overview on MQTT.

A. Hidden channels

Hidden Channels describe the use of techniques for hiding the communication altogether. This is often referred to as steganography. In general terms, a hidden message is concealed within legitimate cover communication.

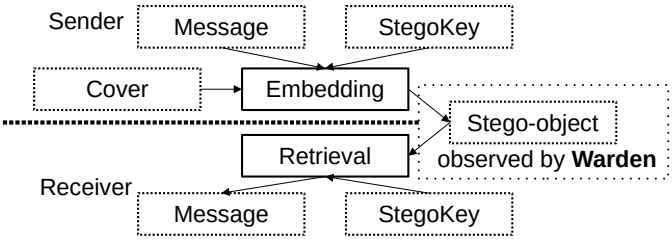

Figure 1. Entities and pipeline for concealed communication using a hidden channel

Figure 1 provides an overview on the general communication. On the sender side, a message is embedded into a Cover object. This process is parametrized by a StegoKey, which might include information about, e.g., embedding position or encoding. The resulting Stego-object is then transferred to the receiver of the message. The warden embodies the position of any security mechanism investigating the cover channel. The receiver then uses the knowledge of the StegoKey to obtain the message.

B. ICS and MQTT fundamentals

ICS control processes in the physical world by measuring them with sensors, affecting them with actuators and using

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

computing units to process sensor readings into control commands for the actuators. The same fundamentals hold true for the domain of IoT; the primary difference being the scale and objectives. In each case, such control networks rely on regular communication between these components using specialized protocols. For this paper, we use MQTT [4], which is a simple and compact protocol used in broad range of applications usually evolving around measuring (and potentially controlling) physical processes.

MQTT employs a client-server structure. Clients (**MQTT Clients**) connect to a server (**MQTT Broker**) and send or receive messages assigned to specific **Topics**. Clients regularly sending such messages to a given topic are sometimes referred to as **Publishers**. Messages are received by clients by subscribing to a specific topic. The Broker forwards any message by legitimate publisher to any client, which is referred to as a **Subscriber**.

C. Hidden Channels in ICS Protocols

Generalized hidden channels in network communication are described as patterns in [5]. These patterns are applied to the domain of ICS protocols in [6]. Closely following the concepts described in [5], [6] describes two primary groups - covert timing channels and covert storage channels.

The covert timing channels include 8 of the overall 18 patterns. All of these 8 patterns create a covert channel in common traffic by using different techniques for altering the timing of messages. Therefore, they transmit the hidden data trough e.g., manipulating the timing of massages or packets.

The covert storage channels manipulate the data transmitted in order to enable a hidden communication. This data could include header fields or certain aspects of the legitimate payload.

III. Identifying the capacity of hidden channels in MQTT as a foundation for security modelling

This section describes our approach to discern the capacity of hidden channels in MQTT.

A. Hidden channels in MQTT

The systematic approach presented in [6] can be an applied to the MQTT and leads to the identification of a set of potential patterns viable for hidden communication.

This feasibility is based on two essential aspects:

- The ability to implement a proof of concept of this pattern within the MQTT protocol. Specifically, the question is whether this pattern can be implemented at all in the MQTT protocol. An example of this would be the patterns **S4: Random Value**. This pattern uses a random value in the metadata of a message to fill it with pseudo-randomized values in order to encode the hidden message. However, this is not possible in MQTT, since MQTT does not contain metadata with random values (unlike, for example, TCP/Modbus).
- The feasibility also refers to the use of the implemented pattern under real-world conditions or in real deployment scenarios. A good example of this is **T6: Retransmission**. This pattern encodes the hidden message through artificial retransmissions. It is very easy to implement in MQTT, but it is difficult to use under real-world conditions, since natural retransmissions occur very frequently in those environments.

This investigation and the technical details of the specific patterns are beyond the scope of this paper - an overview on the results is presented in Figure 2.

Pattern from Mazurczyk et al. (2018)		Findings for MQTT	
ID	Pattern	Feasibility under optimal conditions	Realistic applicability
T1	Inter-packet Times	easy	hard
T2	Message Timing	easy	hard
T3	Rate/ Throughput	medium	hard
T4	Artificial Loss	medium	hard
T5	Message Ordering	easy	medium
T6	Retransmis-sion	easy	medium-hard
T7	Frame Collision	not possible	not possible
T8	Temperature	not possible	not possible
S1	Size Modulation	not possible	not possible
S2	Sequence Modulation	not possible	not possible
S3	Add Redundancy	not possible	not possible
S4	Random Value	not possible	not possible
S5	Value Modulation	easy	hard
S6	Reserved/ Unused	not possible	not possible
S7	Payload Field Size Modulation	easy	easy-medium
S8	User-data Corruption	easy	medium
S9	Modify Redundancy	hard	not possible
S 10	User-Data Value Modulation and Reserved/	easy	easy

Figure 2. Results of the systematic approach from [6] based on [5] applied to MQTT. 10 hidden channels are identified as potentially viable.

In this work, we focus on the steps necessary to discern the capacity of the following 5 patterns selected from those

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

deemed viable:

- **T2: Message Timing**
  The "Message Timing" pattern describes a technique that uses variations in the timing of messages to encode hidden information. In MQTT, the Publisher can modify this timing by altering the intervals between messages and the frequency with which they are sent.
- **T5: Message Ordering**
  The "Message Ordering" pattern transmits hidden information by altering the sequence of packets or messages within a communication flow. In MQTT, the Publisher can modify the arrangement of messages quite easily to encode such hidden data.
- **T6: Retransmission**
  The "Retransmission" pattern encodes hidden information through artificial retransmissions. In MQTT, this can be implemented quite easily while it introduces challenges. Retransmissions also occur naturally in real-world scenarios. As a result, distinguishing between intentional and naturally occurring retransmissions can make decoding the hidden message difficult.
- **S7: Payload Field Size Modulation**
  This pattern encodes hidden information by altering the size of the payload field. In MQTT, the payload field size is not explicitly defined in the protocol metadata. It depends on the chosen data type for transmission. However, this technique can still be implemented in a modified form by changing the actual size of the payload. For example, in an MQTT-based temperature measurement, additional decimal places could be added to subtly increase the payload size and encode hidden data.
- **S10: User-data Value Modulation and Reserved/Unused**
  This pattern encodes hidden information by modifying the payload in a way that is not significant or visible in the actual data. This is typically achieved by altering the least significant bits (LSBs) of the payload data. In real-world MQTT applications, the transmitted data usually consists of sensor measurements. Therefore, modifying the LSB of, for example, a temperature reading can be done quite easily and remains very unobtrusive.

B. Identifying the capacity of Hidden Channels

The capacity of hidden channels can be evaluated by using theoretical analysis and confirmed by practical evaluation results.

1) Theoretical Analysis of the capacity of Hidden Channels: The theoretical analysis involves the investigation of the specific pattern. The following questions serve as a baseline:

- **Identifying the type of message providing capacity** As a first step, the type of messages used for the encoding of the steganographic messages are identified. This could be messages only transmitted once per connection (e.g., messages during the establishment of a connection), multiple times (e.g., messages used to perform a more common occurrence, like subscribing to a specific topic in the case of MQTT) or messages that occur with every sensor reading transmitted.
- **Quantifying the occurrence of the type of messages providing capacity** The occurrence of the specific message for a given scenario is quantified, e.g., how often is a sensor reading being transmitted.
- **Identifying the capacity within a single message** After the amount of available messages is identified, the capacity within a single message is determined.

With these questions answered, the theoretical capacity of a hidden channel can be calculated by multiplying the occurrence of the type of messages providing capacity with the individual capacity of these messages.

2) Practical analysis of the capacity of hidden channels: The theoretical capacity requires confirmation and refinement using practical tests. These are also useful to identify special cases within a given communication scenario. For this, we chose a common application for the MQTT protocol in the context of IoT: climate control.

The transmission of steganographic messages within this use case is monitored. The basic structure of the test consists of 3 components:

- A temperature sensor acting as Publisher that is also acting as Sender and as such embeds the hidden message into the transmission of temperature readings which are used as Cover,
- A Broker,
- A thermostat acting as Subscriber and Receiver of the hidden message.

This setup is connected using a common network switch to create a closed network to prevent unwanted influences (e.g., packet loss, third-party network traffic). With this setup, a steganographic message of 1000 symbols is transmitted and the duration of the transmission monitored.

IV. MEASURING THE CAPACITY

Following this general outline, the test procedure is as follows: For every pattern we run two tests with different cover datasets (**CD**) as cover data. **CD1** and **CD2** each contain 1000 temperature measurements, which are collected from a real temperature sensor for a realistic variance in the measurements. This allows for an increased repeatability of the test.

During the communication between Publisher, Broker and Subscriber, the test data is collected on the device running the Broker software by using network capturing software (this setup is shown in Figure 3). In addition, the retrieved hidden message on the receiver end is obtained and compared against the embedded hidden message on the sender end to evaluate successful embedding and retrieval.

A. Technical Implementation

The following section outlines and explains the technical details, such as the selection of hardware and software. The aim of this section is to ensure both the traceability of the decisions made during the hardware and software selection process and the reproducibility of the experiment.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

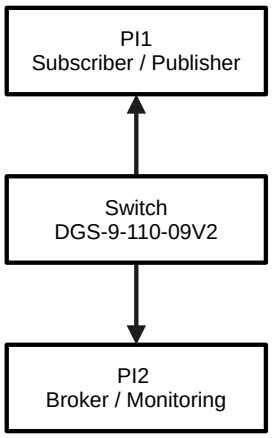

Figure 3. Setup of the test environment for the practical measurements of the capacity of the hidden channels in MQTT.

The following hardware components are chosen: A Raspberry Pi 3 is selected for both the Publisher and the Subscriber, as this device meets the technical requirements and offers a simple and accessible user interface. A Raspberry Pi 4 is used to monitor network traffic and to run the Broker. Additionally, to provide a closed network, a D-Link DGS-1100-09V2 switch is used, which features a monitoring port for observing network traffic.

The Publisher and Subscriber are located on the same device. This does not pose a problem, as the MQTT architecture allows publishers and subscribers to communicate independently with the Broker. There is no direct logical connection between the Publisher and the Subscriber; only between each client and the Broker. There for the Publisher and Subscriber communicate independently with the Broker. The Broker itself operates independently and serves as a central component for communication between clients. Furthermore, this setup simplifies the test environment without compromising the validity of the test data.

The software consists of the following components:

- The operating system used on the Raspberry Pis [7] is Raspberry Pi OS 5.2 (64 bit version) [8]. Raspberry Pi OS was chosen because it is compatible with all common MQTT clients, brokers, and tools, while also offering an accessible and flexible platform for software development.
- Wireshark version 4.4.2 [9] is used to record network traffic, as it captures and logs network data in realtime. Additionally, Wireshark supports MQTT, making it particularly suitable for analysis. Each implementation on the receiver side includes functions that record all transmitted messages and the decoded bits at runtime and save them in a text file for later evaluation.
- Apache ActiveMQ Artemis [10] is used as the Broker, as it enables complex applications and is Open Source software.
- The programming language Python with version 3.13.1[11] is chosen for the implementation of the MQTT clients and the patterns, as it is well-suited due to its extensive libraries and easy integration of MQTT frameworks.
- The framework used is “paho-mqtt” version 2.1.0 [12], which provides an interface for implementing MQTT clients. Additional libraries used include “datetime” or “date” for capturing and recording metadata, “json” for processing data sets, and “struct” for data conversion.

These libraries and tools are chosen for the development of the implementations because they enable efficient and flexible programming.

V. Evaluation

The theoretical capacity of 5 selected hidden patterns is identified before performing the practical measurements using the setup described before. The results of both the theoretical considerations and the practical measurements are shown in Figure 4.

Pattern from Mazurczyk et al. (2018)		Findings for MQTT			
ID	**Pattern**	**Feasibility under optimal conditions**	**Realistic applicability**	**theoretical capacity**	**measured capacity**
T2	Message Timing	easy	hard	1 bit / packet	0,999 bit / packet
T5	Message Ordering	easy	medium	1 bit / flow	0,999 bit / packet
T6	Retransmission	easy	medium-hard	1 bit / time unit	0,66 bit / packet
S7	Payload Field Size Modulation	easy	easy-medium	3 bit / packet	1 bit / packet
S10	User-Data Value Modulation and Reserved/ Unused	easy	easy	1 bit / packet	1 bit / packet

Figure 4. Findings from the theoretical analysis and practical measurements of the capacity of 5 selected hidden channels in MQTT.

The primary observation is, that the practical capacity is lower than the theoretical capacity. This is caused by the difference between the theoretical applicability of a hidden channel - under optimal conditions - and the applicability in a realistic communication setting. In general, the application in a realistic setting is more difficult. Effects like network latency or retransmissions of failed packets interfere with some types of hidden channels. An example is the notably lower capacity of the hidden channel **T6** during the measurement.

A notable effect is the difference in measure and theoretical capacity in channel **S7: Payload Field Size Modulation**. This channel cannot be realized in MQTT as described, because MQTT does not define a fixed payload-field-size in its metadata. The payload size depends on the chosen data format of the transmitted payload and is adjusted flexibly to the payload’s actual length. In typical MQTT scenarios, payloads are sensor readings that usually share the same data type and have similar sizes. By artificially manipulating the payload content, a variation of the “payload field size” in a broader sense could therefore be achieved. The theoretical capacity must

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

consequently be determined for each use case. In the context of our experimental series, the payload consists of a float representing temperature sensor data recorded in the twenties (20–30 °C) with up to three decimal places. Thus, up to three additional decimal places could be used covertly to encode a hidden message unobtrusively, which corresponds to a theoretical capacity of up to 3 bits per packet. The measured capacity, however, is 1 bit per packet. This discrepancy is due to the simple implementation used, in which a 0 encodes a common payload size and a 1 encodes a significantly larger payload; using this scheme, 1,000 messages transmitted 1,000 bits. While this implementation has the disadvantage of lower capacity, it is substantially simpler and less conspicuous.

This shows that theoretical consideration alone is not able to provide a sufficient baseline for security modelling in the domain of hidden channels in ICS networks.

VI. Security Measures

The threat discussed in this work requires means to protect against it. Such measures are highly pattern-specific. Some patterns, e.g., timing patterns like **T6: Retransmission** or **T5: Message Ordering**, can be detected effectively through network monitoring using tools such as Wireshark [9] or specialized heuristics. With careful analysis, certain regularities can be identified by examining the order and frequency of messages. In particular, **T5** is readily detectable in this way, since changes in message ordering are not typical and therefore more conspicuous.

Detection is more challenging for payload-storage patterns (**S7–S10**). These can only be identified through deeper network analysis, which requires decrypting individual packets and inspecting their contents. An example is **S10: User-data Value Modulation and Reserved/Unused** by examining individual packets in Wireshark and performing detailed analysis, one may discover a characteristic pattern and thereby not only identify the covert channel and its associated pattern but potentially also decode the hidden message.

It has to be considered that, depending on the specific implementation, unambiguous identification of the pattern and the hidden channel and, above all, correct decoding of the hidden message may be difficult or impossible without background knowledge of the specific implementation details.

VII. Discussion

This work discusses how the capacity of hidden channels in MQTT networks employed in ICS can be identified by theoretical consideration and practical measurements. The tests show that the theoretical consideration has to be supplemented by practical measurements in order to accommodate the effects of real control networks. The considerations performed in this paper help support security modelling by providing a baseline of the extent of hidden communication an attacker could hide in MQTT communication.

The work is performed in a very limited network setting and would benefit from the use of a more complex network setup. Such a setup should follow are more realistic communication scenario including many more publishers and subscribers as well as the effect and the effect of latency and packet loss. Latency and packet loss could affect the capacity of hidden communication, although some MQTT brokers use various QOS mechanisms in some configurations to address latency and packet loss, which in turn affect specific hidden channels. E.g. Pattern **S10** is directly affected since some QoS features use the fields employed by this pattern. Also, packet loss might affect the transmission quality in the hidden channels itself, depending on the pattern. Pattern **T6** is affected to a great degree by natural retransmissions caused by the network quality. For other patterns, the flag *DUB* used in MQTT could be used to prevent impacts. In our setup, no transmission errors are detected.

Also only a subset of the potentially viable hidden channels in MQTT is evaluated, leaving the remaining channels open for future work.

Additionally, other protocols also have relevance in the field of Industrial Control Systems and might also be subject to hidden channels. We deem the approach presented in this paper to be usable for other protocols, even though the specific implementation of the specific patterns would vary and some patterns are not applicable to all common ICS protocols. Modbus/TCP is discussed in [6].

Acknowledgement

This work has been performed in the scope of the project "SYNTHESIS - Synthetically generated data segments with hidden malicious code functions for safety analysis in nuclear control technology" with the grant number FKZ: 1501666A which is funded by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

Author shares: All authors: fundamental discussion, exchange of ideas, refinement. RA: outline, background, theoretical concept. SO: theoretical and practical adaptation of the concepts from [6] to MQTT as student work, technical implementation (setup, evaluation and hidden channels), evaluation as student work. SK: initial IT-landscape design and test. JD: Input to the general design of hidden channels derived from [6].

References

[1] Dragos, Inc., "2025 OT/ICS Cybersecurity Report", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://hub.dragos.com/hubfs/312-Year-in-Review/2025/Dragos-2025-OT-Cybersecurity-Report-A-Year-in-Review.pdf?hsLang=en.

[2] MITRE, "MITRE ATT&CK: Techniques - Data Obfuscation: Steganography", Apr. 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://attack.mitre.org/techniques/T1001/002/.

[3] A. Badaev and K. Naumova, "SteganoAmor campaign: Ta558 mass-attacking companies and public institutions all around the world", 2024, Accessed: Sep. 15, 2025. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/steganoamor-campaign-ta558-mass-attacking-companies-and-public-institutions-all-around-the-world/.

[4] MQTT.org, "MQTT: The Standard for IoT Messaging", 2024, Accessed: Sep. 15, 2025. [Online]. Available: https://mqtt.org/.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

[5] W. Mazurczyk, S. Wendzel, and K. Cabaj, "Towards deriving insights into data hiding methods using pattern-based approach", *ARES 2018, 13th International Conference on Availability, Reliability and Security; Hamburg, Germany, August 27 - August 30, ISBN: 978-1-4503-6448-5*, pp. 1–10, 2018.

[6] K. Lamshöft and J. Dittmann, "Assessment of Hidden Channel Attacks: Targetting Modbus/TCP", *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 100–11 107, 2020, 21st IFAC World Congress, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.258.

[7] Raspberry Pi Foundation, "Raspberry Pi", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://www.raspberrypi.com/.

[8] Raspberry Pi Foundation, "Raspberry Pi software", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://www.raspberrypi.com/software/.

[9] Wireshark Foundation, "Wireshark", 2024, Accessed: Sep. 15, 2025. [Online]. Available: https://www.wireshark.org.

[10] Apache Software Foundation, "Activemq artemis", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://activemq.apache.org/components/artemis/.

[11] Python Software Foundation, "Python", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://www.python.org/.

[12] Eclipse Foundation, "Paho-mqtt", 2025, Accessed: Sep. 15, 2025. [Online]. Available: https://pypi.org/project/paho-mqtt/.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org