Combining Different Existing Methods for Describing Steganography Hiding Methods

Steffen Wendzel[1][0000−0002−1913−5912], Christian Krätzer[2][0000−0002−0138−4638], Jana Dittmann[2][0009−0003−7985−8041], Luca Caviglione[3][0000−0001−6466−3354], Aleksandra Mileva[4][0000−0003−0706−6355], Tobias Schmidbauer[5][0000−0001−5912−0857], Claus Vielhauer[6][0009−0007−7125−2722], Sebastian Zander[7][0000−0002−2084−7204]

[1] Ulm University, Germany
steffen.wendzel@uni-ulm.com
[2] Otto-von-Guericke University Magdeburg, Germany
{dittmann,kraetzer}@ovgu.de
[3] Institute for Applied Mathematics and Information Technology, Italy
luca.caviglione@ge.imati.cnr.it
[4] Goce Delcev University, N. Macedonia
aleksandra.mileva@ugd.edu.mk
[5] Technische Hochschule Nürnberg Georg Simon Ohm
tobias.schmidbauer@th-nuernberg.de
[6] Brandenburg University of Applied Sciences, Brandenburg, Germany
claus.vielhauer@th-brandenburg.de
[7] Murdoch University, Perth, Australia
s.zander@murdoch.edu.au

Abstract. The proliferation of digital carriers that can be exploited to conceal arbitrary data has greatly increased the number of techniques for implementing network steganography. As a result, the literature overlaps greatly in terms of concepts and terminology. Moreover, from a cybersecurity viewpoint, the same hiding mechanism may be perceived differently, making harder the development of a unique defensive strategy or the definition of practices to mitigate risks arising from the use of steganography. To mitigate these drawbacks, several researchers introduced approaches that aid in the unified description of steganography methods and network covert channels.
Understanding and combining all descriptive methods for steganography techniques is a challenging but important task. For instance, researchers might want to explain how malware applies a certain steganography technique or categorize a novel hiding approach. Consequently, this paper aims to provide an introduction to the concept of *descriptive methods for steganography*. The paper is organized in the form of a tutorial, with the main goal of explaining how existing descriptions and taxonomy objects can be combined to achieve a detailed categorization and description of hiding methods. To show how this can effectively help the research community, the paper also contains various real-world examples.

Keywords: Steganography · Information Hiding · Covert Channels · Science of Security · Taxonomy · Terminology · Systematization.

This is a pre-print. The final version of this paper will be published in the proceedings of the *ARES 2025 Workshops* (CUING Workshop), Springer LNCS.

1 Introduction

The increasing diffusion of digital objects increases the opportunities for hiding data. For instance, the need for enforcing copyright, tracking the diffusion of information, or preventing that data is ingested without a suitable consent, culminated in a vast array of *watermarking* mechanisms [47]. At the same time, the massive softwarization of services in combination with the complexity of modern software supply chains required the design of new watermarks for concealing control information. As an example, software artifacts should be traceable to guarantee provenance and early detect tampering that may cause outages or data breaches [10]. In parallel, the ubiquitous diffusion of AI opened up new challenges. Information needs to be hidden in datasets and models to support licensing schemes, protect large monetary investments, and track quality in AI-as-a-Service frameworks [40].

However, the process of hiding information is not always performed for legitimate purposes. This is the case with malware endowed with steganographic capabilities. In essence, this class of malicious software tries to prevent detection or bypass security countermeasures by cloaking configuration data, offensive attack routines, or optional payloads within multimedia assets. As a result, malware is no longer monolithic, but is implemented through a multi-stage architecture, which reduces its footprint [6]. Another offensive approach concerns the creation of *covert channels*, i.e., parasitic communication paths that can be used to exchange data in a secret and unauthorized manner. Even with real-world attacks abounding in *carriers* that can be abused for the covert communication (e.g., patterns of syscalls or hardware behaviors), the most effective approaches take advantage of network traffic [45].

As a result, the works dealing with information hiding and steganography largely overlap both in terms of terminology and concepts. In some cases, the same idea is reinvented multiple times, wasting resources and complicating the retrieval of knowledge. Another major issue is rooted within the double-edged nature of mechanisms devoted to cloak data. On one hand, they have proven their effectiveness for tracking and copyright purposes (e.g., watermarks). On the other hand, they are definitely becoming an important resource in the toolbox of attackers (e.g., to implement stegomalware). This causes a "mismatch" in the perception of the various steganographic approaches, as some ideas may require to be designed contextually with a proper defensive strategy. Moreover, different backgrounds and use cases hinder the possibility of developing general countermeasures. For instance, the mitigation of network covert channels requires preventing ambiguities that could be exploited by an attacker early on,

i.e., during the design stage. Unfortunately, this conflicts with the need for offering techniques to mark packet flows and control their route through the Internet, such as for traffic engineering goals [7].

To cope with the aforementioned pitfalls, several authors introduced different approaches for a *unified description* of steganography methods and network covert channels. Specifically, this paper explains how existing steganography taxonomy and description objects can be used jointly to achieve a unified and clear explanation. The goal here is to provide a solid foundation for the scientific literature on steganography and covert channels. Our paper is accompanied by an inter-active online tool: `https://patterns.omi.uni-ulm.de/desrcovert/`.

In summary, the main contribution of this work is to provide an introduction to concepts related to *descriptive methods for steganography*. The paper provides a tutorial on how existing descriptions and taxonomy objects can be combined for designing precise descriptions of several hiding methods.

The remainder of this paper is structured as follows. We discuss the existing concepts in Section 2 and then describe how these can be combined in Section 3. We provide tutorial examples in Section 4 and a brief discussion in Section 5. Finally, Section 6 concludes.

2 Related Work and Fundamentals

This section reviews previous approaches on how steganographic methods and techniques for the creation of covert channels can be organized or described in terms of hiding patterns. For the sake of brevity, this section does not embrace the literature on information hiding on a *tout court* manner. Rather, it should be considered as a complement of more comprehensive works, see, e.g., [5] for a general overview of cloaked/abusive communication paths.

Hiding Patterns. For the specific case of taxonomies, several authors already categorized information hiding and steganography methods. As a result, a relevant corpus of works has emerged [4,11,15,21,22,24,27,28,33,38,39,52,53,54]. However, the existing categorizations have either focused on high-level aspects or have been tailored to specific domains. For instance, many works only consider network steganography or image steganography.

In 2015, the concept of *hiding patterns* has been introduced to provide a generic taxonomy on the hiding process [50]. In essence, hiding patterns describe hiding methods in an abstract fashion. Until recently, the description of hiding patterns still has been domain-specific: the original taxonomy was tailored for network steganography [50] but an analysis of the taxonomy in the context of cyber-physical systems exists as well [20].

More recently, a *generic pattern-based taxonomy for all steganography domains* has been introduced [48]. Based on this taxonomy, hiding patterns either describe how a secret message is *embedded* in a carrier (so-called *embedding patterns*) or how the secret message is *represented* (so-called *representation patterns*). Representation patterns are derived from embedding patterns. The

enumeration and nomenclature is driven by clear rules. Two major types of embedding hiding patterns exist: (1) those that modulate a state or value, e.g., the pattern *E1.3. LSB State/Value Modulation*[8] subsumes LSB-based methods; (2) those that modify the occurrence of some element, e.g., the pattern *E2.1. Element Enumeration* encodes secret information through the number of some element (e.g., byte count of a file or size of a network packet) [48]. A brief overview on the major patterns is given by Tab. 1. Note that patterns can be media-specific, e.g., *E1.3n1. Network LSB State/Value Modulation* or *E1.3t1. Text LSB State/Value Modulation* for network and text steganography, respectively.

Pattern	**Brief Description**
E1. State/Value Modulation	Some state (e.g., of an actuator) or value (e.g., bit in a file) is modulated to hide a secret message.
E1.1. Reserved/Unused State/Value Mod.	Sub-variant where reserved or unused states/values (e.g., padding bits) are modulated to embed a secret message
E1.2. Random State/Value Mod.	Sub-variant that covers the modulation of random values (e.g., cryptographic hashes)
E1.3. LSB State/Value Mod.	Sub-variant covering all forms of LSB steganography
E1.4. Character State/Value Mod.	Sub-variant covering textual character modulations, e.g., changing the case of letters
E1.5. Redundancy State/Value Mod.	Sub-variant that covers hiding methods that change redundancy (to embed secret data), e.g., transcoding steganography
E2. Element Occurrence	Some element's occurrence is changed in some way (e.g., a network packet *appears*)
E2.1. Element Enumeration	Sub-variant covering methods where the number of elements is modulated to encode a secret message (e.g., number of bytes of a file)
E2.2. Element Positioning	Sub-variant covering methods where the location of an element in a cover object is used to encode a secret message (e.g., position of a specific pixel in an image or time of appearance of a signal)

Table 1: Overview of core hiding patterns of [48].

Local and Distributed Channels. Local steganography channels do not rely on distributing the secret message over multiple cover objects nor do they apply multiple hiding methods to the same cover object. The following terms were proposed in [50] to describe distributed hiding methods using patterns: *Pattern variation* refers to techniques that apply the same hiding pattern to different carrier objects, e.g., least significant bit modulation to a field in the IPv4 and the IPv6 header, without needed a whole new implementation. *Pattern combination* applies multiple hiding patterns to the same carrier object (e.g., embedding

[8] In this paper, we will use a highlighted font to indicate our proposed nomenclature, including the patterns introduced in [48].

a secret bit into the least significant bit of a timestamp in a file's metadata while embedding additional secret data into unused metadata bits). Finally, *pattern hopping* refers to the (pseudo-randomized) alternation of hiding methods [50]. Mazurczyk *et al.* slightly extended these terms in [31] by further splitting pattern variation into *host-*, *flow-* and *protocol-based scattering*, which represent approaches specific for network steganography. The core idea is to distribute secret message bits to different hosts, through multiple flows or through multiple protocols.

Direct and Indirect Channels. Some steganography methods, including network techniques for the creation of covert channels, are designed to establish indirect paths for secret messages, i.e., the covert sender does not directly send data to the covert receiver [53,52]. For instance, personal cloud storage services can be used to implement an encoding for transferring secret information. In this case, file operations (e.g., copying and renaming) can be grouped into patterns and are used to generate suitable signaling flows to convey secret data to the intended recipients [8].

When the covert communication path does *not* behave in a direct end-to-end flavor and is based on the (involuntary) integration of a third-party node, such as a network host or a process on a local system, the channel can be described by two indirect hiding patterns that have been introduced by Schmidbauer and Wendzel: *redirector* or *broker*, with the broker having the two sub-patterns *proxy* and *dead drop*) [41]. A brief description of these indirect hiding patterns is summarized in Tab. 2.

Active and Passive Channels. Covert senders do not necessarily need to create their own cover objects to embed data into, nor must covert receivers necessarily be the overt recipients of a cover object [56]. In this perspective, an *active* channel is one where the covert sender creates the cover object and the covert receiver is the overt recipient. In contrast, a *passive* channel would require a covert sender to modify a third-party cover object to embed the secret message into and the covert receiver to recognize the embedded message (e.g., as an on-path attacker). It is also possible to create channels of mixed form (*semi-active* [23] as introduced by Lamshöft and Dittmann; *semi-passive* as introduced by Zander [52]). In some cases *fully-passive* channels can be created (as introduced by Wendzel *et al.* [49]). In such cases, the cover object is untouched by sender and receiver. Instead, the sender solely *points* to the cover object and the receiver observes the traffic through eavesdropping or as a broadcast receiver.

Multi-level Steganography. Multiple authors proposed nesting steganographic objects inside other steganography objects, leading to multiple levels (or layers) of steganography. Multi-level steganography can be found in different domains, including filesystem steganography [29] and network steganography [14]. Multi-level steganography can be used to reach a plausible deniability, where the outer steganography level is presented to an observer, but inner levels are not revealed and kept secret.

Indirect Hiding Pattern	Brief Description
Redirector	A sender forces a third-party node to unintentionally redirect steganography objects to a covert receiver. Example: a covert sender transmits steganography objects as payload within a spoofed network packet. The packet contains a request (e.g., ICMP or IGMP) and is sent to a third-party node that responds to the spoofed address (which is the one of the covert receiver) [41].
Broker	In comparison to the *Redirector*, a broker does not redirect steganography objects but manipulates the third-party node so that these steganography objects can be extracted by a covert receiver [41]. A *Broker* is either a *Proxy* or a *DeadDrop*.
Proxy	Sub-variant of the *Broker* where a covert sender influences a third-party node in such a way that the influence can be recognized by a covert receiver. Example: A local covert sender process might cause heavy load on a third-party process handling his requests. This load influences the third-party process' performance and can be measured by a covert receiver's process. The influence on the performance represents the secret message.
Dead Drop	Sub-variant of the *Broker* where the steganography object is stored on a third-party node. Example: a sender might influence the network protocol's cache of a third-party node to embed a secret message. The cache's content is then read by the covert receiver [41].

Table 2: Summary of indirect hiding patterns as introduced in [41]; descriptions have been generalized to remove the network-specific context.

A similar concept introduced by Ogiela and Koptyra is called *multi-secret steganography* [35]. Instead of nesting one layer inside another, multi-secret steganography embeds a set of secret messages into the same carrier. Several of these messages are *false stego-objects* that are comparably easy to detect while the actual secret message is more challenging to detect.[9]

Pointers to "Historic" and "Future" Data. A steganography transmission can embed the actual secret message or a *pointer* that refers to the desired secret message, so that only a small fraction of the information is used instead of transferring the entire message [49]. In the case of a pointer, one can refer to either already existing data (called historic data, even if it was *just* created, e.g., a few CPU cycles ago) or to anticipated future data (e.g., expected regular ARP requests).

[9] Multi-secret steganography could alternatively be considered a special variant of the previously-mentioned *pattern combination* [50] as multiple hiding methods are combined to add secret message (fragments) to the same cover object.

Unified Description Method. In 2016, a *unified description method* (UDM) was introduced for network information hiding methods, which was slightly modified in 2025 to fit all domains of steganography and better integrate the updated patterns taxonomy [48, supplemental material Sect. A.6]. The UDM covers several attributes that are described in a comparable manner, including the application scenario, the hiding patterns of a hiding method, the required properties of a cover object that are necessary to realize the steganography channel, and the channel properties (e.g., robustness, countermeasures, and capacity). Finally, optional information can be provided on a channel-internal protocol. The UDM has been designed to improve the replicability, comparability, and identification of research gaps in the steganography literature.

3 Proposal for a Combination of Description Methods

Fig. 1 shows the general structure of our naming convention. Following the recent steganography taxonomy [48], here we apply the proposed pattern naming convention for "hiding patterns" (see the most-right component in the figure). At the same time, we also adjust it to incorporate the surrounding terms for categorization (see the remaining boxes in the figure). A dash (-) indicates a default category, i.e., it can be omitted if a channel does not contain a specific feature. In particular, it must not be mentioned explicitly that a steganography channel is non-distributed, direct, active, uses solely a single level of embedding, or when the steganography data is embedded into the object itself (present-focused) rather than referring to history or anticipated (future) data.

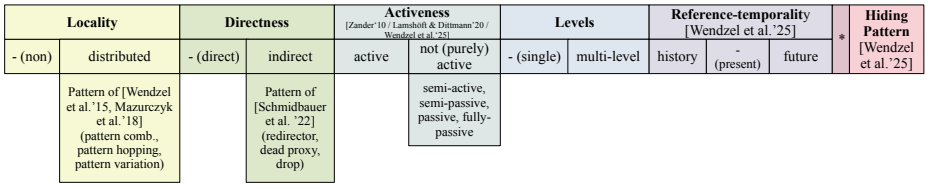

Fig. 1: Proposed Naming Convention

All components of our naming convention are additionally explained by our inter-active online tool: `https://patterns.omi.uni-ulm.de/desrcovert/`

3.1 Naming Components in Detail

Referring to Fig. 1, we now discuss the components that can be used to develop the proposed naming convention. The naming components are described below.

Locality. This clarifies whether a steganography hiding method is local or distributed. To this end, the first (but optional) component of the naming can be "distributed", followed by the distribution pattern (e.g., "pattern combination") enclosed in brackets [50,31]. Such methods might employ different hiding patterns simultaneously. As will be detailed later in Sect. 4.2, multiple methods should be mentioned jointly.

Directness. This clarifies if the hiding method represents a direct or indirect channel. The optional attribute "indirect" would be followed by the name of the particular *pattern* of Schmidbauer and Wendzel [41] in brackets, e.g., "redirector". Multiple examples featuring such indirect patterns will be provided in Sect. 4.

Activeness. This defines whether the channel is *active* (must not be mentioned explicitly) or *passive* [56]. Here, the above-mentioned passivity terms (*semi-active*, *semi-passive*, *fully-passive*) can be be placed in brackets behind the "passive" attribute.

Level Characteristic. This denotes whether the steganography method establishes a multi-level steganography system, or not. The attribute should be omitted if it is a single-level system.

Reference-temporality. This attribute can be used to specify if secret data is "moved" through the channel by means of pointers, that is if the channel points to previously found/written data (history covert channel) or to anticipated (future) data as introduced in [49].

Star-property (*). This attribute is a star property (*****) that allows arbitrary details to be added. For instance, one might employ terms like *cover selection* [15] or *coverless steganography* [55,25]. Another option is to state whether a channel is a unidirectional, bidirectional, or broadcast channel. If desired, one might explicitly mention rough robustness criteria, e.g., that a channel is *noisy* or *noise-free* or if the cover is *predictable*, *variable* or *randomized* as done in the work of Zander [52]. Finally, one might state that a hiding method is *reversible* [9,19,30,42], i.e., if an (intermediate) covert receiver can restore the cover object to its status before a secret message was embedded.

Hiding Pattern. This attribute must mention the hiding pattern of the 2025 taxonomy of steganography hiding methods [48]. This is a very important aspect, as this attribute contributes to aligning the various concepts within the literature towards a common knowledge.

3.2 Describing Sophisticated Methods

In general, the aforementioned *seven* attributes already provide some flexibility to describe sophisticated methods. For example, one might apply the distributed

host-based scattering method introduced in [31] by using an indirect communication through *dead drops* [41] that serve as nodes for storing the secret data. The actual hiding method might be a network-based LSB state-value modulation. To this end, we would call such a method a *distributed (host-based scattered) indirect (dead drop) E.1n1. network LSB state/value modulation*.

However, if the steganography method *utilizes different hiding patterns*, each must be named separately. For instance, one might be LSB and the other reserved/unused state/value modulation, so we would gain two descriptions, (a) and (b): *distributed (host-based scattered) indirect (dead drop) (a) E1.3n1. network LSB state/value modulation and (b) E1.1n1. network reserved/unused state/value modulation)*.

Unfortunately, methods might also apply *multi-level steganography*, i.e., nesting stego objects inside other stego objects. This requires more elaborate naming, but the availability of a solid convention/taxonomy makes the process simple and reduces ambiguities. In this case, each stego-layer could utilize a *different* hiding pattern. For instance, a multi-layer filesystem steganography method might be classified as *multi-level (a) E1.3f1. filesystem LSB state/value modulation, (b) E1.1f1. filesystem reserved/unused state/value modulation and (c) E1.2f1. filesystem random state/value modulation)*.

This means that the outermost layer performs LSB state/value modulation, while the middle layer performs reserved/unused state/value modulation, and the innermost layer is LSB state/value modulation on filesystem data.

It would also be reasonable to have a multi-*level*-multi-*media* approach. For instance, network steganography might hide data inside network packets, and embedded data could contain a second layer featuring digital image steganography data. An example of such a case would be *multi-level (a) E1.1n1. network reserved/unused state value modulation, (b) E1.3d1. digital media LSB state/value modulation*.

Allowing such multi-media descriptions also contributes to fill a gap identified by the latest taxonomy [48, cf. Fig. 1]. Especially, hiding methods have been described until now as belonging to only *one* domain, i.e., neglecting another domain if they utilize objects or hiding methods from multiple domains.

3.3 Utilization of the Unified Description Method

In case our nomenclature cannot capture all the nuances of the targeted steganography approach, the missing details can be covered by borrowing ideas from the UDM [48, cf. electronic supplement]. Fig. 2 shows the structure of the UDM. As shown, such a unified framework foresees that a steganography method is described by using hiding patterns as its core component as well as by several additional attributes. These additional attributes (application scenario, required properties of the cover object, etc.) leave room for a structured description of typical attributes.

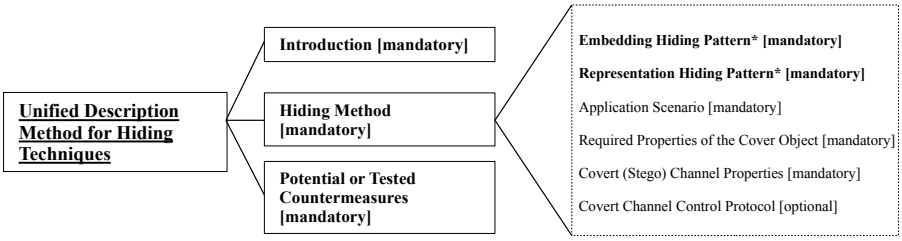

Fig. 2: The UDM of [48]. Our nomenclature is used within the attributes specifying the patterns (indicated by *).

Handling Representation Patterns. Some steganography methods utilize a different embedding and representation pattern. In this case, the attribute "representation hiding pattern" of the UDM can be used to mention the representation pattern, also following our nomenclature (see example in Sect. 4.2). In other cases where the representation pattern is simply the representing variant of the embedding pattern, the UDM attribute "representation hiding pattern" can just state "representation variant of embedding pattern".

4 Examples

This section describes existing steganography methods by using our combined approach. Tab. 3 provides an overview on our examples. We start with a simple network-specific technique, followed by two indirect/hybrid network steganography techniques. Afterwards, we cover an example from digital media steganography, one from cyber-physical systems steganography, and one from text steganography.

4.1 Simple Network Steganography Method

In network steganography, secret data is hidden inside the content of network packets (e.g., overwriting unused bits or replacing values in header fields) or by modulating temporal characteristics of the traffic (e.g., influencing packet occurrence or flow duration) [33,50,53]. The first methods for network steganography (or: network covert channels) have been described in the late 1980s, e.g., [16,51]. Recently, network steganography has become a major branch of steganography-capable malware [45,6].

In this first example, we assume that a covert sender hides data within the unused "reserved" bit of the IPv4 header. Without any additional sophisticated characteristics, such as multi-level steganography or an indirect manipulation of values, the whole description consists of the pattern *E1.1n1. Network Reserved/Unused State/Value Modulation*. As the covert receiver interprets the same reserved bit of the IPv4 header, the representation pattern is *R1.1n*.

Ex.	Scenario	Description
1	Simple Network Method	*E1.1n1. Network Reserved/Unused State/-Value Modulation*
2	Hybrid Hiding Method	(1) *Indirect (Proxy) E1n1. Network State/-Value Modulation*, (2) *Indirect (Proxy) R2.2n1. Network Element Positioning*
3	Indirect Dead-drop	*Indirect (Dead Drop) E1.1n1. Network State/Value Modulation*
4	(History-focused) LSB Audio Steganography	*(History-focused) E1.3d1. Digital Media LSB State/Value Modulation*
5	(Distributed) OPC UA Steganography	*(Distributed) E1.3c1. CPS LSB State/Value Modulation*
6	Text Steganography	(1) *E2.1t1. Text Element Enumeration*, (2) *(Semi-active) E2.1t1. Text Element Enumeration*, (3) *Indirect (Dead-drop) E2.1t1. Text Element Enumeration*

Table 3: Overview on provided examples

4.2 Hybrid Hiding Method

Spiekermann *et al.* [44] present a migration covert channel. They propose that a covert sender migrates a virtual machine from one server to another, e.g., from Europe to Australia, so that a covert receiver can measure a different round trip time (RTT). While the embedding hiding pattern to migrate the virtual machine is a *E1n1. Network State/Value Modulation* (commands in a protocol are transferred so that the migration is triggered), the covert receiver must measure the RTT with probe traffic or by conducting some measurements. In particular, the time of occurrence of a response packet is measured to obtain the RTT, which is *R2.2n1. Network Element Positioning* (the network element (=packet) is "positioned" in time [48]).

To map the underlying patterns of this "migration" channel into the proposed nomenclature, we utilze Fig. 1 as a guide. The hiding method is *not* distributed, but it is an indirect method following the "proxy" pattern [41]. The method is an active method without multi-level component, and no history/future reference, i.e., these three aspects must not be mentioned explicitly. Thus, we call this hybrid method as follows: *Indirect (Proxy) E1n1. Network State/Value Modulation* (the embedding pattern) and *Indirect (Proxy) R2.2n1. Network Element Positioning* (the representation pattern). Note that the indirect attribute is included in both, embedding and representation patterns.

4.3 Network-based Indirect Dead-Drop Hiding Method

Velinov *et al.* introduced an indirect method that allows the establishment of a bidirectional network covert channel in the MQTT protocol, which has been

described as "ICC.1" in [46]. The channel is also referred to as "MQTT.1" in [41]. MQTT is specialized in conveying information between IoT devices, the clients of the MQTT server, via topics. Such topics may have numerous subtopics and can be subscribed to by clients. To achieve indirect covert communication, the covert sender and covert receiver (i.e., clients of the MQTT server) must agree on one first-level topic. The covert receiver has to subscribe to all subtopics of this first-level topic. The covert sender now embeds covert information in one of the subtopics that is stored by the MQTT-server. Due to subscription, the MQTT-server notifies the covert receiver and sends the embedded covert information, which can be extracted at the target of the covert channel. Due to the storing of information on the MQTT server, [41] considers this concept to be a so-called dead drop. For such an implementation, the covert channel is split into three phases: manipulation, storing, and extraction. Each phase may utilize different hiding methods, but in the case of this example, each phase uses the patterns E1n1. and R1n1. for embedding and representation of covert information, respectively.

In the proposed nomenclature, this covert channel can be defined as *non*-distributed but "indirect" (using the "dead drop" pattern). Furthermore, the method is active, does not apply multi-level steganography, and does not employ a history/future component. Thus, the complete categorization is *Indirect (Dead Drop) E1.1n1. Network State/Value Modulation*.

4.4 Audio Hiding Method

Despite dating back to the late 1990s, Least Significant Bit (LSB) modification is still one of the most relevant hiding methods used in audio steganography. From the approximately 800 audio steganography repositories currently available on Github, an estimated 70% features LSB modification based hiding methods, such as LSB Replacement (LSBR) and also one of the few audio stego-malware cases reported in the wild features LSBR as the embedding method [6,45].

Technically, in LSBR parts of the least significant bit plane of an uncompressed audio or image sample is adapted to match the bits in a secret message. In compact disc (CD) compliant pulse code modulated (PCM) WAV audio files, every sample value is encoded by a 16 bit value per channel. In that case, the lowest bit-plane value is modified, producing minimal changes in the audio content that are much below the hearing threshold of a human being, even if introduced into silent parts of the audio content. In general, LSBR for image and audio is considered to be easily detectable if high embedding rates (>20% of potential hiding places) are used since the message embedding in those cases overwrites/destroys the underlying characteristics of the cover signal. For very low embedding rates (i.e., very short messages hidden in long covers) the introduced embedding artifacts have been shown to be statistically indistinguishable from the cover statistics and therefore undetectable.

Due to the prominence of this method in digital media steganography the pattern *E1.3d1. Digital Media LSB State/Value Modulation* has been

proposed [48]. A hiding method would simply be described by its pattern if no sophisticated approach is given.

Instead, if a hiding method would only embed pointers to previously recorded data in a cover object, it would be categorized as *History-focused E1.3d1. Digital Media LSB State/Value Modulation*.

4.5 OPC UA Hiding Method

Neubert *et al.* present and analyze three methods for steganographic embedding of hidden messages in OPC UA data packets [34]. OPC UA is a cross-platform, open-source protocol for Cyber-Physical Systems (CPS), in particular for Industrial Control System (ICS) network communication, and is developed by the OPC Foundation [36]. The goal of Neubert *et al.* is to generate and evaluate steganographic OPC UA network traffic including packets generated by simulation of a corrupted Programmable Logic Controller (PLC) within an ICS network. The PLC generates OPC UA packets with slight timestamp modifications in micro- and nanosecond range to embed hidden messages. OPC UA timestamps are composed in the format *"T_i = hh:mm:ss:mmm μμμ nnn"*, where h,m,s,m,μ and n stand for hour, minute, second, millisecond, microsecond and nanosecond respectively. All three methods make use of the least two digits of the microsecond timestamp and all three digits of the nanosecond values. This results in timestamps such as *"T_i = 10:00:00.123 4**56 789"* for each modified packet, where the five potentially modified digits are marked in bold, *"56"* representing the lower 2 digits of the microseconds and *"789"* the three digits for the nanoseconds. The three algorithms vary with respect to their strategy for constructing the actual embedding pattern and its position. The first and simplest method generates patterns by embedding two distinct digit values as embedding symbols for 1 and 0 for each hidden message bit in all three positions for the microsecond timestamp of three subsequent OPC UA packets (i.e. positions of *"456"* in the above example) in the communication flow. A second method involves basically the same scheme, but performs an embedding key-based permutation of the embedding symbols, as well as the embedding digit positions. The third method additionally involves the timestamp values at positions not considered for modification to generate patterns and uses XOR encryption.

All three examples fall in the category *E1.3c1. CPS LSB State/Value Modulation* because the least significant digits (although not exactly bits!) of timestamp elements are modulated. All other properties can be omitted, as they fall into the default categories, because the payload is bit-wise directly represented, all channels are active (due to the assumption of a compromised PLC component), do not consider multi-level steganography and carry the message directly. Thus, the directness, activeness, level characteristic and reference-temporality properties can be omitted. As a result, the simplest method can be described as *Non-Distributed E1.3c1. CPS LSB State/Value Modulation*, whereas the remaining two methods fall into the category of *Distributed E1.3c1. CPS LSB State/Value Modulation*. The two distributed methods utilize a key-based permutation of the embedding position of each single numeric

symbol within the 5 potential least significant numeric positions (i.e., *"56789"* in the above example) across subsequent data packets. The description for the pattern combination of the two distributed methods can be formulated as *"key-based symbol, embedding position and cover data permutation"* using the *-property of Fig. 1.

4.6 Text Hiding Method

Most of the methods applied in text steganography are simple. For example, repeating white space characters in a text (open space method) [3,2] is a form of the pattern *E2.1t1. Text Element Enumeration*. However, these methods can be applied in heterogeneous scenarios, with different naming components, such as directness and activeness. For example, Mileva *et al.* [32] suggest three different applications of the open space method and the pattern E2.1t1 using DICOM files as covert carriers. DICOM (*Digital Imaging and Communications in Medicine*, cf. [12]) is a standard for the digital handling of medical images, containing several attributes that can be filled with textual content.

In the first scenario, a covert sender and a covert receiver are the actual sender and receiver, which means that the created covert channel is active and direct, so we can describe it simply as *E2.1t1. Text Element Enumeration*. The second scenario covers a direct channel in which the covert sender is the actual sender, while covert receiver(s) monitor the network traffic intended for other receivers to extract the hidden message. This covert channel can be described as *Semi-active E2.1t1. Text Element Enumeration*. The last scenario is an example of an indirect active channel in which the covert sender (as the actual sender) stores the cover DICOM file in an archive (utilized as third-party intermediate component), while the covert receiver(s) can request some services regarding that DICOM file from the archive and extract the hidden message. Thus, this hidden communication mechanism can be described as *Indirect (Dead-drop) E2.1t1. Text Element Enumeration*.

5 Discussion

The examples provided here do not cover the full extent of existing hiding methods, such as steganography in filesystems [18,13], AI models [37,43,26], air-gapped covert channels [17,5] or covert channel-based traffic obfuscation for censorship circumvention [1]. However, we believe that our methodology can be easily extended to these domains, especially since the patterns taxonomy [48] already foresees many of them. At the same time, we are also aware that our work might be limited due to the lack of some categorizations and subtaxonomies of information hiding topics. However, the proposed approach and the related corpus of works at the basis of [48] are solid, thus making improvements and adjustments easier. As our work becomes accepted by the community, routine "maintenance" operations will be easier. In this perspective, information hiding topics that may emerge in the future due to the utilization of new technologies

not yet invented or not yet relevant could be added when needed. As an example, the vivid area of tracking AI-generated content was completely unforeseeable ten years ago but now drives vast research directions. This reinforces the need of having multiple and flexible attributes to describe how the steganography area evolves.

Another limitation of our work is that it does not cover countermeasures; they are solely an attribute of the UDM. However, since our work is focused on the categorization and description of hiding methods instead of the categorization and description of countermeasures to detect, limit, or prevent steganography, we consider the description of countermeasures as a separate project.

Future developments are notoriously difficult to predict. For this reason, our approach might be considered an intermediate step that allows extension in areas where it is needed (due to its *-property in Fig. 1).

6 Conclusion

We have introduced a meta-view on description and taxonomy approaches in steganography that several researchers focusing on information hiding have constructed over the years. Our combination of existing methodologies allows for a comprehensive description of steganography methods in a unified and comparable fashion. As a result, scientific re-inventions could be reduced and the knowledge between different fields (e.g., defensive watermarking and detection of stegomalware) could be exchanged much more effectively. Our provided interactive online tool aids the understanding of our methodology. Additionally, the tool can be used for didactic settings.

Future work will focus on the development of a similar methodology for countermeasures against threats endowed with some form of steganographic capabilities. Specifically, we are working towards the definition of a suitable abstraction/taxonomy to prevent ambiguities and imperfect isolation issues that may lead to exploitable hiding patterns.

Acknowledgments. We like to thank the anonymous reviewers for their constructive feedback. We further like to express our gratitude to co-authors of previous works that helped paving the way to this paper: Wojciech Mazurczyk, Jörg Keller, Krzysztof Cabaj, Laura Hartmann, Sebastian Zillien, Tom Neubert, Bernhard Fechner, and Christian Herdin.

The work of Luca Caviglione has been been supported by PNRR project.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. DomEye: Detecting network covert channel of domain fronting with throughput fluctuation. Computers & Security **144** (2024). https://doi.org/10.1016/j.cose.2024.103976

2. Ahvanooey, M.T., Li, Q., Hou, J., Rajput, A.R., Chen, Y.: Modern text hiding, text steganalysis, and applications: A comparative analysis. Entropy **21**(4), 355 (2019)
3. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal **35 (Nos3&4)**, 313–336 (1996)
4. Bennett, K.: Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. Tech. Rep. 2004-13, CERIAS Tech Report, Purdue University (2004)
5. Carrara, B., Adams, C.: Out-of-band covert channels—a survey. ACM Computing Surveys (CSUR) **49**(2), 1–36 (2016)
6. Caviglione, L., Mazurczyk, W.: Never mind the malware, here's the stegomalware. IEEE Security & Privacy **20**(5), 101–106 (2022)
7. Caviglione, L., Mazurczyk, W.: You can't do that on protocols anymore: analysis of covert channels in ietf standards. IEEE Network **38**(5), 255–263 (2024)
8. Caviglione, L., Podolski, M., Mazurczyk, W., Ianigro, M.: Covert channels in personal cloud storage services: The case of Dropbox. IEEE Transactions on Industrial Informatics **13**(4), 1921–1931 (2017)
9. Chang, C.C., Lin, C.Y.: Reversible steganography for VQ-compressed images using side matching and relocation. IEEE Trans. Inform. Forens. and Sec. **1**(4), 493–501 (2006)
10. Dey, A., Bhattacharya, S., Chaki, N.: Software watermarking: Progress and challenges. INAE Letters **4**, 65–75 (2019)
11. Dhawan, S., Gupta, R.: Analysis of various data security techniques of steganography: A survey. Information Security Journal: A Global Perspective **30**(2), 63–87 (2021)
12. DICOM: The DICOM standard (2025), https://www.dicomstandard.org/
13. Eckstein, K., Jahnke, M.: Data hiding in journaling file systems. In: Digital Forensic Research Workshop (2005)
14. Fraczek, W., Mazurczyk, W., Szczypiorski, K.: Multi-level steganography: Improving hidden communication in networks. Journal of Universal Computer Science (J. UCS) **18**(14), 1967–1986 (2012). https://doi.org/10.3217/jucs-018-14-1967
15. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press (2009). https://doi.org/10.1017/CBO9781139192903
16. Girling, C.G.: Covert channels in LAN's. IEEE Transactions on Software Engineering **13**, 292–296 (February 1987)
17. Guri, M., Monitz, M., Mirski, Y., Elovici, Y.: BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In: 2015 IEEE 28th Computer Security Foundations Symposium. pp. 276–289 (2015). https://doi.org/10.1109/CSF.2015.26
18. Han, J., Pan, M., Gao, D., Pang, H.: A multi-user steganographic file system on untrusted shared storage. In: 26th ACSAC. pp. 317–326. ACM (2010). https://doi.org/10.1145/1920261.1920309, https://doi.org/10.1145/1920261.1920309
19. Heßeling, C., Keller, J., Litzinger, S.: Reversible network covert channel by payload modulation in streams of decimal sensor values. In: 2023 IEEE 19th International Conference on e-Science (e-Science). IEEE Computer Society, Los Alamitos, CA, USA (2023). https://doi.org/10.1109/e-Science58273.2023.10254946
20. Hildebrandt, M., Lamshöft, K., Dittmann, J., Neubert, T., Vielhauer, C.: Information hiding in industrial control systems: An opc ua based supply chain attack and its detection. In: Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security. pp. 115–120 (2020)

21. Johnson, N.F., Katzenbeisser, S.: A survey of steganographic techniques. In: Information hiding. pp. 43–78 (2000)
22. Knöchel, M., Karius, S.: Text steganography methods and their influence in malware: A comprehensive overview and evaluation. In: Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security. pp. 113–124 (2024)
23. Lamshöft, K., Dittmann, J.: Assessment of hidden channel attacks: Targetting Modbus/TCP. IFAC-PapersOnLine **53**(2) (2020). https://doi.org/10.1016/j.ifacol.2020.12.258
24. Li, B., He, J., Huang, J., Shi, Y.Q.: A survey on image steganography and steganalysis. J. Inf. Hiding Multim. Signal Process. **2**(2), 142–172 (2011)
25. Liu, H., Zhang, C., Wang, Z., Guo, C., Gou, P., Shan, L., Lu, Z.: To deliver more information in coverless information hiding. Multim. Tools Appl. **83**(3), 7215–7229 (2024)
26. Liu, T., Liu, Z., Liu, Q., Wen, W., Xu, W., Li, M.: Stegonet: Turn deep neural network into a stegomalware. In: Proc. ACSAC'20. p. 928–938. ACM (2020). https://doi.org/10.1145/3427228.3427268
27. Lubacz, J., Mazurczyk, W., Szczypiorski, K.: Principles and overview of network steganography. IEEE Communications Magazine **52**(5), 225–229 (2014)
28. Majeed, M.A., Sulaiman, R., Shukur, Z., Hasan, M.K.: A review on text steganography techniques. Mathematics **9**(21), 2829 (2021)
29. Masud, M.A., Akter, S., Sultana, N., Yousuf, M.A., Uddin, M.Z.: Multi-layered password-based steganography: A novel approach for tiered information hiding. techxiv (2024). https://doi.org/10.36227/techrxiv.173397886.68744435/v1
30. Mazurczyk, W., Szary, P., Wendzel, S., Caviglione, L.: Towards reversible storage network covert channels. In: Proc. ARES 2019. pp. 1–8
31. Mazurczyk, W., Wendzel, S., Cabaj, K.: Towards deriving insights into data hiding methods using pattern-based approach. In: Proc. ARES'18. pp. 10:1–10. ACM (2018). https://doi.org/10.1145/3230833.3233261
32. Mileva, A., Caviglione, L., Velinov, A., Wendzel, S., Dimitrova, V.: Risks and opportunities for information hiding in dicom standard. In: CUING, ARES '21: Proceedings of the 16th International Conference on Availability, Reliability and Security (2021). https://doi.org/10.1145/3465481.347007
33. Mileva, A., Panajotov, B.: Covert channels in TCP/IP protocol stack-extended version. Open Computer Science **4**(2), 45–66 (2014)
34. Neubert, T., Peuker, B., Schueler, E., Ullrich, H., Buxhoidt, L., Vielhauer, C.: An analysis framework for steganographic network data in industrial control systems. In: Proc. SECURWARE 2024. IARIA (2024)
35. Ogiela, M.R., Koptyra, K.: False and multi-secret steganography in digital images. Soft computing **19**(11), 3331–3339 (2015)
36. OPC Foundation: official website (2025), https://opcfoundation.org
37. Pan, X., Zhang, S., Zhang, M., Yang, M.: House of cans: Covert transmission of internal datasets via capacity-aware neuron steganography. In: Advances in Neural Information Processing Systems (2022)
38. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding-a survey. Proc. of the IEEE **87**(7), 1062–1078 (1999)
39. Provos, N., Honeyman, P.: Hide and seek: An introduction to steganography. IEEE security & privacy **1**(3), 32–44 (2003)
40. Regazzoni, F., Palmieri, P., Smailbegovic, F., Cammarota, R., Polian, I.: Protecting artificial intelligence ips: a survey of watermarking and fingerprinting for machine learning. CAAI Transactions on Intelligence Technology **6**(2), 180–191 (2021)

41. Schmidbauer, T., Wendzel, S.: SoK: A survey of indirect network-level covert channels. In: Proc. 17th Asia Conf. Computer and Communications Security (ASIACCS). pp. 546–560. ACM (2022). https://doi.org/10.1145/3488932.3517418
42. Song, C., Zhang, Y., Lu, G.: Reversible data hiding in encrypted images based on image partition and spatial correlation. In: Proc. International Workshop On Digital Watermarking (IWDW). pp. 180–194 (2018)
43. Song, C., Ristenpart, T., Shmatikov, V.: Machine learning models that remember too much. In: Proc. ACM CCS 2017. pp. 587–601. ACM (2017). https://doi.org/10.1145/3133956.3134077
44. Spiekermann, D., Keller, J., Eggendorfer, T.: Towards covert channels in cloud environments: a study of implementations in virtual networks. In: International Workshop on Digital Watermarking. pp. 248–262. Springer (2017). https://doi.org/10.1007/978-3-319-64185-0_19
45. Strachanski, F., Petrov, D., Schmidbauer, T., Wendzel, S.: A comprehensive pattern-based overview of stegomalware. In: Proc. ARES'24 (2024). https://doi.org/10.1145/3664476.3670886
46. Velinov, A., Mileva, A., Wendzel, S., Mazurczyk, W.: Covert channels in the MQTT-based Internet of Things. IEEE Access **7**, 161899–161915 (2019). https://doi.org/10.1109/ACCESS.2019.2951425
47. Wan, W., Wang, J., Zhang, Y., Li, J., Yu, H., Sun, J.: A comprehensive survey on robust image watermarking. Neurocomputing **488**, 226–247 (2022)
48. Wendzel, S., Caviglione, L., Mazurczyk, W., Mileva, A., Dittmann, J., Krätzer, C., Lamshöft, K., Vielhauer, C., Hartmann, L., Keller, J., Neubert, T., Zillien, S.: A generic taxonomy for steganography methods. ACM Comput. Surv. **57**(9), 1–37 (May 2025). https://doi.org/10.1145/3729165
49. Wendzel, S., Schmidbauer, T., Zillien, S., Keller, J.: DYST (did you see that?): An amplified covert channel that points to previously seen data. IEEE Transactions on Dependable and Secure Computing (TDSC) (2024). https://doi.org/10.1109/TDSC.2024.3410679
50. Wendzel, S., Zander, S., Fechner, B., Herdin, C.: Pattern-based survey and categorization of network covert channel techniques. Comp. Surveys **47**(3) (2015). https://doi.org/10.1145/2684195
51. Wolf, M.: Covert channels in LAN protocols. In: Proc. LAN Security, LNCS, vol. 396, pp. 89–101. Springer (1989)
52. Zander, S.: Performance of selected noisy covert channels and their countermeasures in IP networks. Ph.D. thesis, Swinburne Univ. (2010)
53. Zander, S., Armitage, G., Branch, P.: A survey of covert channels and countermeasures in computer network protocols. Communications Surveys & Tutorials **9**(3), 44–57 (2007)
54. Zhiyong, C., Yong, Z.: Entropy based taxonomy of network convert channels. In: Proc. 2nd Int. Conf. on Power Electronics and Intelligent Transportation System (PEITS). pp. 451–455 (2009)
55. Zhou, Z., Mu, Y., Wu, Q.M.J.: Coverless image steganography using partial-duplicate image retrieval. Soft Comput. **23**(13), 4927–4938 (2019)
56. Zi, X., Yao, L., Pan, L., Li, J.: Implementing a passive network covert timing channel. Computers & Security **29**(6), 686–696 (2010). https://doi.org/10.1016/j.cose.2009.12.010