A Generic Taxonomy for Steganography Methods

Steffen Wendzel [1], Luca Caviglione [2], Wojciech Mazurczyk [2], Aleksandra Mileva [2], Jana Dittmann [2], Christian Krätzer [2], Kevin Lamshöft [2], Claus Vielhauer [2], Laura Hartmann [2], Jörg Keller [2], Tom Neubert [2], and Sebastian Zillien [2]

[1]Hochschule Worms
[2]Affiliation not available

October 30, 2023

Posted on 5 Jul 2020 — CC-BY 4.0 — https://doi.org/10.36227/techrxiv.20215373.v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

Abstract

A unified understanding of terms and their applicability is essential for every scientific discipline: steganography is no exception. Being divided into several domains (for instance, text steganography, digital media steganography, and network steganography), it is crucial to provide a unified terminology as well as a taxonomy that is not limited to some specific applications or areas. A prime attempt towards a unified understanding of terms was conducted in 2015 with the introduction of a pattern-based taxonomy for network steganography. Six years later, in 2021, the first work towards a pattern-based taxonomy for steganography was proposed. However, this initial attempt still faced several shortcomings, e.g., the lack of patterns for several steganography domains (the work mainly focused on network steganography and covert channels), various terminology issues, and the need of providing a tutorial on how the taxonomy can be used during engineering and scientific tasks, including the paper-writing process.

As the consortium who published this initial 2021-study on steganography patterns, in this paper we present the first comprehensive pattern-based taxonomy tailored to fit all known domains of steganography, including smaller and emerging areas, such as filesystem steganography and cyber-physical systems steganography. Besides, to make our contribution more effective and promote the use of the taxonomy to advance research on steganography, we also provide a thorough tutorial on its utilization.

Our pattern collection is available at https://patterns.ztt.hs-worms.de.

A Generic Taxonomy for Steganography Methods

Steffen Wendzel *Member, IEEE*, Luca Caviglione, Wojciech Mazurczyk *Senior Member, IEEE*, Aleksandra Mileva *Member, IEEE*, Jana Dittmann, Christian Krätzer, Kevin Lamshöft, Claus Vielhauer, Laura Hartmann, Jörg Keller, Tom Neubert, Sebastian Zillien

***Abstract*—A unified understanding of terms and their applicability is essential for every scientific discipline: steganography is no exception. Being divided into several domains (for instance, text steganography, digital media steganography, and network steganography), it is crucial to provide a unified terminology as well as a taxonomy that is not limited to some specific applications or areas. A prime attempt towards a unified understanding of terms was conducted in 2015 with the introduction of a pattern-based taxonomy for network steganography. Six years later, in 2021, the first work towards a pattern-based taxonomy for steganography was proposed. However, this initial attempt still faced several shortcomings, e.g., the lack of patterns for several steganography domains (the work mainly focused on network steganography and covert channels), various terminology issues, and the need of providing a tutorial on how the taxonomy can be used during engineering and scientific tasks, including the paper-writing process.**

As the consortium who published this initial 2021-study on steganography patterns, in this paper we present the first comprehensive pattern-based taxonomy tailored to fit all known domains of steganography, including smaller and emerging areas, such as filesystem steganography and cyber-physical systems steganography. Besides, to make our contribution more effective and promote the use of the taxonomy to advance research on steganography, we also provide a thorough tutorial on its utilization.

Our pattern collection is available at https://patterns.ztt.hs-worms.de.

***Index Terms*—Steganography, Network Steganography, Covert Channels, Terminology, Taxonomy, Information Hiding, Science of Security, Information Security, Patterns, CPS, Filesystems, Digital Media, Linguistic Steganography, Cyber Security.**

I. Introduction

STEGANOGRAPHY is the art and science of concealing the existence of information transfer and storage. It has already been applied in Ancient Greece and the Roman Empire, with its methodology been developed further in the Medieval and Enlightenment Ages [1]–[3]. Especially during conflicts and wartimes, steganography developed further due to the need of sophisticated hiding methods to exchange secret messages. In the 20th century, steganography gained additional importance due to the digitization in organizational and private sectors as well as the introduction of the Internet [2]. Plenty of digital media formats, network communication protocols, and cyber-physical systems allowed to exploit hiding methods to conceal both, the storage and transfer of secret information.

S. Wendzel, L. Hartmann and S. Zillien are with the Hochschule Worms, Worms, Germany, S. Wendzel and L. Hartmann are also with the FernUniversität in Hagen, Hagen, Germany.

L. Caviglione is with the National Research Council of Italy (CNR), Genova, Italy.

W. Mazurczyk is with the Warsaw University of Technology (WUT), Warsaw, Poland, and the FernUniversität in Hagen, Hagen, Germany.

A. Mileva is with the University Goce Delcev, Stip, North Macedonia.

J. Dittmann, C. Krätzer and K. Lamshöft are with the University of Magdeburg, Magdeburg, Germany.

C. Vielhauer and T. Neubert are with the TH Brandenburg, Brandenburg a. d. Havel, Germany and the University of Magdeburg, Magdeburg, Germany.

J. Keller is with the FernUniversität in Hagen, Hagen, Germany.

Manuscript received ...; revised ....

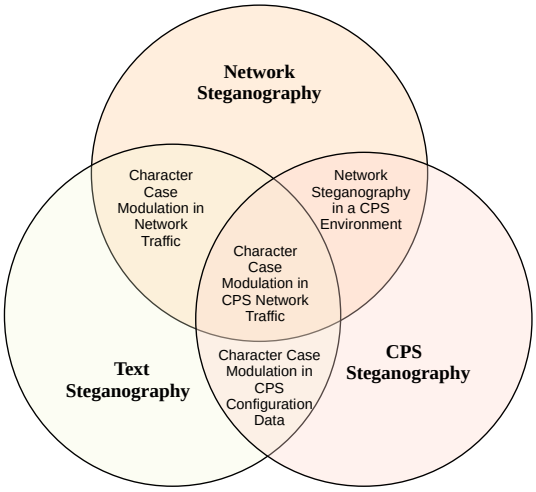

Fig. 1. Example for an overlapping focus of three steganography domains.

Today, steganography consists of several domains. As paradigmatic examples, we mention text steganography, network steganography, digital media steganography, filesystem steganography, and cyber-physical systems steganography. Some of these domains are influenced by multiple research directions and technological developments. For instance, network steganography is shaped by and overlaps with network covert channel research. Because of these domains, which develop further their own terms and methodology, it is crucial to synchronize and unify the understanding of central aspects, such as steganographic methods, as there are several overlaps, which have been exemplified in Fig. 1.

Therefore, in this article, we provide the following key contributions:

1) Presentation of the first comprehensive and fully-functional pattern-based taxonomy for steganography that includes all major steganography domains as well as recently evolving domains. Our taxonomy is partially applicable to non-digital domains, such as non-digital text steganography, but its focus is on digital forms of steganography.
2) Provision of a unified terminology to describe hiding methods of all steganography domains.
3) Supply of an accompanying tutorial to explain the application of the taxonomy in a way that enforces a unified

description of hiding methods. This aids replicability of future scientific studies.

The remainder of this article is structured as follows. Sect. II explains the current situation and covers related work on the subject while Sect. III discusses different domains where steganography is applied. Sect. IV introduces the methodology for creating the taxonomy. Sect. V presents the core taxonomy, followed by domain-specific sub-taxonomies in Sect. VI. Sect. VII presents countermeasures, while Sect. VIII discusses our approach including its limitations and Sect. IX gives a tutorial for using the taxonomy. Finally, Sect. X draws conclusions and provides an outlook on future research challenges.

List of Abbreviations

ARP	Address Resolution Protocol
BACnet	Building Automation Control Networks
BMP	Bitmap (file format)
CPS	Cyber-physical System
CR	Covert Receiver
CS	Covert Sender
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
exFAT	Extensible File Allocation Table
FAT	File Allocation Table
FIN	Finish
FTP	File Transfer Protocol
HAS	Human Audio System
HTTP	Hyper-text Transfer Protocol
HTML	Hyper-text Markup Language
HVS	Human Visual System
IEEE	Institute of Electrical and Electronics Engineers
ICMP	Internet Control Message Protocol
ICS	Industrial Control System
ID	Identifier
IoT	Internet of Things
IP(v4)	Internet Protocol (Version 4)
IP-IP	IPv4 in IPv4
IPSec	Internet Protocol Security
IPv6	Internet Protocol Version 6
LSB	Least Significant Bit(s)
MAC	Message Authentication Code
MBR	Manipulation by Reader
MIDI	Musical Instrument Digital Interface
MP3	MPEG-1/2 Audio Layer III
MQTT	Message Queuing Telemetry Transport
NNTP	Network News Transfer Protocol
NTFS	New Technology File System
OR	Overt Receiver
OS	Overt Sender
PCM	Pulse Code Modulation
PDF	Portable Document Format
PDU	Protocol Data Unit
PGM	Probabilistic Graphical Model
PLC	Programmable Logic Controller
PLML	Pattern Language Markup Language
PNG	Portable Network Graphics
POP3	Post Office Protocol (Version 3)
QoS	Quality of Service
RPM	Revolutions per Minute
RST	Reset
RTP	Real-time Transport Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TIA	Totally Integrated Automation (Portal)
TTL	Time to Live
UDP	User Datagram Protocol
VAD	Voice Activity Detection
VM	Virtual Machine
WAV	Waveform Audio File Format
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

II. Current Situation and Related Work

In Sect. II-A, we will first cover the current situation and related work on pattern-based descriptions for hiding methods. Afterwards, Sect. II-B discusses related surveys.

A. Pattern-based Taxonomies in Steganography

In research domains facing a vital development, it is not uncommon that terminology and definitions change over the years. This case was, for instance, observed for cyber-physical systems (CPS) [4]. Similarly, steganography faces an active and divergent development that lasts since several decades while new technology and domains, such as network steganography, emerged. This led to inconsistent descriptions of hiding methods — even *within* steganography domains [2]. For this reason, the concept of a pattern-based taxonomy was introduced by Wendzel et al. in 2015 for the domain of network steganography (mainly for the case of network covert channels) [5]. Patterns subsume hiding methods that share the same core idea. The authors proposed to describe patterns using the *pattern language markup language* (PLML), which provides a structured set of pattern attributes that are usually described in XML (see [6] for details). A key advantage of applying PLML is the fact that it allows to build links and derivations of patterns, which can then be used to form a taxonomy and to handle aliases since hiding methods appear under different names in the related literature [5].

The authors of the article on hand suggested the first step towards a pattern-based taxonomy for the *whole* domain of steganography in 2021 [7], which can be considered a pre-study for this article used to gain feedback from the scientific community. To this end, a consortium of authors from different steganography domains was formed. While this new pattern-based taxonomy provided several advancements and solutions for problems of the 2015-taxonomy, it also faces several limitations. Summarized, the key advancements of the 2021-taxonomy over state-of-the-art are the following [7]:

1) **Separating Embedding and Extraction Patterns.** The previous taxonomic attempts contained no differentiation between the embedding and extraction process of secret

data. This lack led to inconsistent descriptions of hiding methods as embedding and extraction patterns can differ. As the extraction also determines how the covert receiver sees the data represented, extraction patterns will also synonymously be called representation patterns in this article.

2) **Domain-overlapping Patterns.** The previous 2015-taxonomy (and its several updates) were solely focusing on network steganography. The new 2021-taxonomy therefore proposed embedding patterns that can be conveniently applied to all domains of steganography.

However, there are several key limitations of that 2021-taxonomy which we address in this article:

1) **Domain-overlapping Patterns Limited to Embedding Patterns.** As mentioned above, the 2021-taxonomy introduced a generic pattern-based taxonomy. However, this generic approach was limited to embedding patterns. Extraction patterns were only exemplified for network steganography, not for other steganography domains. For this reason, we provide a taxonomy for embedding *and* representation patterns. This includes the provision of representation patterns for steganography sub-domains beyond network steganography.
2) **Strong Domain-focus.** The previous incarnations of the various taxonomies had a strong dedication to specific steganography domains, such as network steganography, even in the 2021-taxonomy. This resulted in redundancies, e.g., when character case modification of plaintext HTTP headers is performed in *network* steganography while actually *text* steganography methods are applied to network data.
   A similar case occurs in CPS steganography, where sometimes network traffic of a CPS is modified, but in fact the methods come from network steganography. We thus propose to make distinctions between the different steganography domains only when necessary but to unify the taxonomy among these domains. In other words, domain-specific sub-taxonomies are required to only contain domain-specific aspects.
3) **Lack of a Tutorial and Replication Support.** There exists no guide on how to apply the pattern-based taxonomy during research. This is why this paper provides an accompanying tutorial featuring a **unified description method for scientific papers**: while there exists a unified description method for *network* steganography, it is not applicable to other domains of steganography and does not consider the revised patterns taxonomy. The structured and unified description of hiding methods allows their easier assessment, comparability and replicability.
4) **Artificial Distinction of Temporal/Non-temporal Methods.** The distinction between temporal and non-temporal hiding patterns leads to inconsistent categorizations, e.g., when a hiding method orders the sequence of TCP packets it would be both, temporal and non-temporal, at the same time (the timing of the packet matters but also their stored sequence number). This imprecise distinction between storage and timing covert channels evolved many years ago, and we aim to solve the issue by avoiding this type of categorization completely, as it would remain problematic. Instead, we differentiate in our taxonomy by the type of the modified object (an element or a state/value).
5) **Inconsistent Pattern Naming.** The 2021-taxonomy introduced a nomenclature for patterns that is not fully consistent, as sometimes multiple objects are combined in a pattern's name (e.g., "Event/Element Interval Modulation"). This is solved in this article by borrowing the concept of *binomial* nomenclature from biology for most of our patterns.[1]
6) **Inconsistent Terminology.** In the 2021-taxonomy several objects were introduced: intervals, events, elements, features and states/values. Elements contain features. While elements can, for instance, be network packets, text characters or audio files, features would be, e.g., header fields and optional headers in network packets or fractions in audio files. However, these features could consist of multiple features themselves, which is not consistent. Thus, these should be considered as elements, too (as elements can contain other elements). Further, the 2021-taxonomy proposes that so-called events are sequences of elements when they appear, such as a text string that contains multiple characters. However, combined sequential characters can also be considered as separate elements with the characteristic that they all appear at specific locations (behind each other), which is the view of the paper at hand. Similarly, so-called intervals are not necessary as they can be considered as the distance between two elements (either in time or in space). Finally, some events, such as frame collisions in network packets can be considered as features (the feature of a frame collision), which means that events overlap with features in such cases.
   For these reasons, we propose a clarified terminology as well as a reduced number of relations between the terms.

The article at hand is a strongly revised and extended version of the 2021-taxonomy [7]. The particular enhancements were made during several meetings of the steganography patterns working group that all authors belong to.

B. Related Surveys

In addition to the discussed pattern-based surveys, multiple attempts have been made to survey steganography methods and to work on their terminology.

Several generic works provide fundamental terms and taxonomies for information security, such as Avizienis et al. [8] for dependable and secure computing. Similar approaches were conducted for unifying the terminology in information hiding, starting with results from an informal plenary meeting at the Information Hiding Workshop in 1996 [9] as well as work by Pfitzmann and Köhntopp who addressed related terms, such as anonymity and unobservability [10].

[1] In biology, the name of a species consists of the generic name (genus) and the specific name, e.g., *Canis lupus* (gray wolf).

An early survey was published by Petitcolas et al. [1]. In their article, the authors cover the whole information hiding domain, of which steganography, anonymity research, covert channels and copyright marking were considered separate sub-disciplines. A more recent coverage of steganography and its history was published by Zielińska et al. [11].

Network covert channel terminology, hiding methods and countermeasures have been surveyed by Zander et al. [12], Zhiyong et al. (based on entropy) [13], Mileva et al. [14] and Wendzel et al. [5]. The overlapping domain of network steganography was surveyed by Lubacz et al. [15], where some fundamental methods were categorized. Terminology and taxonomy (as well as patterns) between network covert channels and network steganography were later unified by Mazurczyk et al. [2]. Recently, Schmidbauer and Wendzel surveyed *indirect* network covert channels [16]. The authors assigned patterns of the 2021-taxonomy to the particular hiding methods.

Linguistic (text) steganography was also covered by some surveys, such as by Bennett [17] and Ahvanooey et al. [18]. A bibliographic collection is available through the work of Bergmair [19].

Digital media steganography was surveyed in several textbooks and articles. Fridrich covers a wide range of terminology, hiding methods and countermeasures for digital media steganography [3]. Li et al. [20] summarize methods used in image steganography and steganalysis. Johnson and Katzenbeisser [21] surveyed steganography methods in general.

Countermeasures for steganography methods have been surveyed by Gianvecchio et al. [22], Li et al. [20], Goher [23] et al., Mazurczyk et al. [2] and Caviglione [24] — just to mention a few. Moreover, steganography methods are also in use to cover censorship circumvention systems. Such methods were analyzed by Houmansadr et al. [25]. Similarly, systematic enhancements of steganographic channels through internal control protocols have been summarized by Wendzel and Keller [26].

None of the above-mentioned surveys provides a unified taxonomy for hiding methods that considers all (or a majority of) the currently separated steganography domains.

III. Analysis of Existing Steganography Domains

In general, steganographic methods can be utilized either to enable a *covert transfer* or *covert storage*, as well as in a combined manner as depicted in Fig. 2. In both cases, the secret information is *embedded* into a *cover object*, which should be selected to not represent an anomaly and have a suitable embedding capacity. Typically, a steganographic application or method is closely related to the features characterizing the chosen hidden data carrier. In more detail, for the case of covert transfer, a *covert sender* (CS) *embeds* secret information in a cover object for a *covert receiver* (CR), which needs to extract the secret information from the cover object. Thus, the CS and CR must agree on the *representation* of the secret information in the cover object. Even if many mechanisms for covert transfer exist, the most popular group of information hiding solutions exploit network traffic and protocols [27]. Instead, for the case of covert storage, the steganographer is

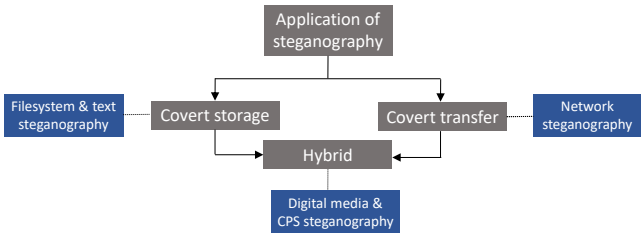

Fig. 2. Various applications of steganography.

interested only in storing sensitive data on a local information carrier (e.g., on a hard drive or in a document containing text information), in such a way that the data cannot be spotted by a third party observer unaware of the information concealment. An example of such a technique is filesystem steganography where some additional overlay filesystem for data hiding purposes is created by using features of an underlay filesystem like the unused space in partially-allocated blocks [28] or by using image collections as a carrier [29], [30]. A particularity of filesystem steganography is that besides creating and reading, also modifying the steganographic data occurs regularly. Some authors also consider storing any kind of sensitive data in a filesystem, e.g., in artificially created file metadata attributes [31], as filesystem steganography.

Finally, for some cover objects, it is possible to perform covert transfer or covert storage, depending on the required application (hybrid approach). This is the case, for instance, in digital media steganography where one can perform a hidden data exchange by embedding secret data into the content transferred by services like video or audio streaming, or even if one sends an e-mail with an image containing secret information. Alternatively, the steganographer can utilize digital images on the local disk as a vault to locally store his/her secrets [32]. CPS steganography can be treated as another example of a hybrid solution [33]. Recently, another new set of methods emerged that combines the covert transfer (over network covert channels) and the covert storage (within the caches of network protocols), which is called a *Dead Drop* [16], [34].

The remainder of this section highlights how major steganography domains differ in terms of their cover objects and embedding strategies.

a) Network Steganography: As hinted, the principal characteristics of network steganography are already covered by the existing terminology (see, e.g., [35] and the references therein). In essence, the main cover objects used in network steganography are provided by manipulating or injecting information in some digital artifacts belonging to the network traffic, e.g., the header or the payload of a Protocol Data Unit (PDU) as well as in the behavior of flows/conversations consisting of a coherent sequence of packets. In general, two main flavors of network steganography exist: *i*) embedding/representing of secret data within the PDU, and *ii*) influencing the timing or the sequence of adjacent/succeeding packets. Compared to other steganography domains, and with the rare exception of above-mentioned Dead Drops, the goal of network steganography is not to *store* but to *transfer* the data [35]. The capacity of a steganographic method targeting

networks is limited by the traffic type and the length of a transmission. Typically, this leads to a slower embedding process compared to digital media steganography [27], [32]. The data is hidden in an ephemeral manner, and the application of network steganography can increase delays and packet loss. This can impact on the stealthiness of the resulting covert transmission due to the reduction of some functionality provided by the protocol or a degradation of the transmission quality [35].

b) Digital Media Steganography: The term digital media steganography (or short: media steganography) addresses the wide field of digital steganography research and development focusing on digital media (i.e., media encoded in machine-readable formats) as cover data for a plausible, secured and hidden communication. Digital media were initially designed to address the human audio-visual system (by delivering information to a screen and/or loudspeaker) and include many heterogeneous forms such as images, audio data, videos, 3D models, etc. As with the media themselves, digital media steganography comes in a wide variety of different types that can be classified by various categories. In particular, digital media steganography can focus on the media type(s) (e.g., audio steganography), the transmission method (e.g., data as spatial image or as audio stream vs. audio files) and the basic strategy concerning the existence and plausibility of a cover data (such as a data stream to embed into) [21].

Established text books in this field, such as [3], argue that three basic paradigms for the message embedding can be applied in media steganography: steganography by modification (i.e., changing the content or representation of a media object to embed the message), steganography by synthesis (i.e., synthetically creating a media object containing the message based on pre-determined source characteristics) and steganography by selection (i.e., using a pre-exchanged codebook to signal the message by selecting unmodified media objects from a pool). In terms of occurrence, Fridrich points out that steganography by selection is by far the most published upon basic paradigm [3]. It is also the one encountered in most implemented media steganography tools. In case of steganography by modification, the basic coding strategies of message insertion (i.e., where to embed in the cover data), the structure of how to embed the message in the cover data (usually represented as a signal or coded signal data), as well as the usage of the steganographic key are important parameters for a scheme. All of them depend on the choice of media type to be used as cover objects. Since becoming an active research field in the 1990s, a great number and variety of scientific works have been published on media steganography and steganalysis. The vast majority of these publications (as well as most of the tools available) have been focusing on image steganography as the most prominent domain in this field [20].

It can be stated that any continuous digital media (in the sense of temporally-changing media content) can be designed both for covert storage and covert transfer. This obviously applies mainly to audio and video, which can be streamed or stored as files. Recently, streaming services received an increasing degree of interest, as they appear to become the new main form of media delivery and consumption in entertainment (see, e.g., [36]).

The capacity of digital media steganography is limited by the type and size of the digital media. For digital media steganography, capacity always depends on two other characteristics to be achieved: robustness and imperceptibility for the detection of the hidden message (also related to undetectability). Some methods of media steganography can survive conversion to another format, but a plausible cover object is always required. The application of digital media methods might decrease the quality of the cover object (e.g., image quality).

c) Text Steganography: This distinct branch of steganography is sometimes also referred to as *linguistic* steganography. It relies on hiding information in textual messages and textual documents as cover data, including those in magazines, newspapers, word processing documents, personal notes, source code of programs, and music notes — just to mention a few. In contrast to digital media steganography, it uses manipulation of some lexical, syntactic or semantic features of the text content, modification of different features of the text's elements (e.g., characters, paragraphs, sentences, words, lines) or generation of a new text that simulates some features of the normal text. Several examples of such methods are presented in [1] and more recently in [37]. The latter has identified the following concepts as embedding principles in the literature: *i*) word spelling, *ii*) semantic method, *iii*) line shifting, *iv*) abbreviation, *v*) word shifting, *vi*) syntactic method, and *vii*) new synonym text. Since at least three of these (i.e., *ii*, *iv*, and *vii*) can be considered of purely semantic nature, and since in comparison to digital media steganography, text steganography also involves printed (non-digital) text, the distinction between them and the field of digital media steganography seems reasonable.

Similar to digital media steganography, text steganography allows the permanent hiding of information, as the texts are not of ephemeral nature like network traffic. To this end, the vast majority of proposed concepts can be categorized as covert storage methods. However, concepts of embedding hidden information in text streams (e.g., keystrokes or scrolling text) appear feasible.

The capacity of text-based steganographic methods is mainly limited by the size and structure (including grammar, sections and use of white-spacing) of a text. A suitable cover text is required to make it plausible, as auto-generated texts might appear synthetic to an observer. Similar to digital media steganography, text steganography may decrease the quality of the cover object, even if imperceptible.

d) Other Steganography Domains: Additional domains of steganography bring different characteristics with them. For instance, in filesystem steganography, the cover object might be a file, unused space in a partially allocated block [38], cluster distribution of an existing file [28], or an inode [39]. In the case of cyber-physical systems (CPS), the focus of steganography is linked to the definition of the term CPS. According to a recent analysis by DeFranco and Serpanos, the definition of a CPS is not unified and overlaps with the definition of the IoT [4]. We thus consider CPS steganography

in a broad manner that involves the IoT domain to reflect the various definitions surveyed by DeFranco and Serpanos. In CPS steganography, secret information might be embedded into a sensor value [40], an actuator state [33], unused registers [33], or into the control logic of a PLC [41]. Hidden data might even be embedded into the number of cyber-physical events of some machine. Hildebrandt et al. published the only available pattern-based classification for CPS steganography [42], built on top of the existing one for network steganography. Their taxonomy adds additional categories, namely for firmware accessible and program accessible patterns.

e) Summary: When we look at the aforementioned steganography domains, it becomes clear that cover objects appear to be highly different, involving several different elements and values/states, not just digital files or network packets. For this reason, the novel taxonomy must allow for the inclusion of highly heterogeneous types of cover objects.

In general, a unified theory/taxonomy can be more suitable for a research domain than multiple domain-specific classifications/theories, in a similar manner that universal programming languages can be advantageous over domain-specific programming languages.

IV. Methodology

In this section, we first describe the general methodology applied for the development of the taxonomy in Sect. IV-A, followed by an explanation of the literature selection methodology in Sect. IV-B.

A. General Methodology

In 2020, we set up a consortium consisting of experts from seven institutions located in four countries. During regular consortium meetings, the following methodology emerged. Given the success and the functionality of hiding patterns in the community, we decided to keep the concept of hiding patterns for the new taxonomy. It was further agreed that the consortium will stick to the PLML-based pattern specification that was already applied by [5]. PLML provides a comparable and unified systematic for the description and management of patterns [6] that is also applied in other domains, such as software engineering. A PLML-based description contains certain attributes, such as a name for the pattern, aliases, an illustration, code snippets, evidence in the form of references, example cases, and links to related patterns [6] — just to mention a few. A PLML-based specification also allows exploiting existing methodology, such as the unified description method for hiding methods [43] and the existing framework for determining whether some hiding method represents a new pattern, or not [44]. Furthermore, PLML enables easy indexing, extensibility and linkage of patterns to keep the provided taxonomy up-to-date on the long run. By allowing the inclusion of *aliases* in PLML-based specifications, different terminology can be unified in a common term as well, limiting the chance for so-called scientific re-inventions [44].

Purpose and Requirements of the Taxonomy: The main purpose of this work is to provide a taxonomy that unifies the terminology for hiding methods in all domains of steganography and allows the unified description of embedding and extraction processes.

Based on this purpose and the limitations of the existing taxonomy (Sect. II), we identified the following requirements for the new taxonomy:

1) The new taxonomy should be applicable to all domains of steganography and thus needs to be generic.
2) While being generic, it should still allow the domain-specific description of hiding methods, e.g., methods that can only be applied to cyber-physical systems should be categorizable with the taxonomy.
3) Embedding and extraction (representation) process should be described separately to cover hybrid and indirect hiding methods.
4) Patterns should be applicable when new hiding methods are described or presented; the description should be unified and comparable.
5) The taxonomy should be easy to apply in scientific publications that describe new hiding methods in detail so that it is also attractive for scientists. The description should aid reproducibility.
6) The taxonomy should be extensible.

How these Requirements are Addressed: Requirements (1) and (2) are addressed by defining generic patterns in an abstract manner, while it is feasible to derive domain-specific patterns from these generic patterns. The domain specific patterns contain additional domain know-how and details. Further, by providing a unified description method for new hiding methods on the basis of our taxonomy, detailed characteristics of a hiding method (bandwidth, robustness, application scenario etc.) can be communicated, see Fig. 3.

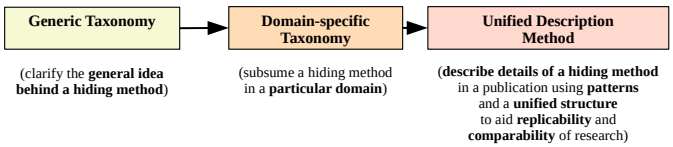

Fig. 3. Description detail using generic and domain-specific patterns as well as the unified description method

Requirement (3) is addressed by defining embedding patterns and deriving representation (extraction) patterns from the existing embedding patterns. In case a representation pattern cannot be derived from an embedding pattern, it can also be described from scratch. Deriving representation from embedding patterns however minimizes the chances for asynchronous descriptions of embedding and representation patterns. It further minimizes redundancies in descriptions. The unified description of patterns (Requirement (4)) is achieved by applying the PLML structure, as already done in earlier works [5]. PLML also enables the inclusion of future patterns. Further, the description of additional hiding method characteristics in a structured manner is achieved by our unified description method (Fig. 3) that is accompanied by a tutorial in Sect. V and aids replicability (Requirements (5) and (6)).

B. Methodology for Literature Selection

The literature selection followed general principles for finding related work. First, common literature databases for computer science (DBLP, ACM Digital Library, IEEE Xplore, Springerlink etc.) as well as Google Scholar were searched with the following keywords: *steganography*, *steganographic*, and *covert channel*, without limiting our search for a specific range of years. Whenever novel ideas not covered by earlier work were discovered, we added them to our references. However, when newer publications did not present hiding methods whose concepts differ from previous ones (e.g., when the encoding of a channel was improved or a combination of multiple already known ideas were combined), we excluded these from this article as they would cause redundancy in our generic taxonomy.

Furthermore, the consortium members added peer-reviewed published research papers from their local repositories, that emerged over the last decade(s) of active research in this domain. The literature found was screened and, if relevant, references in and citations of those articles were checked again for further relevant literature.

Each hiding method of every published paper found was classified with the taxonomy of Sect. V. For research results that did not fit into the taxonomy, the consortium discussed necessary adaptions and/or extensions of the taxonomy, until a stable situation was reached. The taxonomy was thus optimized in an iterative process.

V. A Generic Taxonomy for Steganography

This section presents our taxonomy for hiding patterns in a way that incorporates the characteristics of the discussed steganography domains. We first introduce the key concept of embedding and representation patterns in Sect. V-A, followed by a coverage of applied naming conventions in Sect. V-B and a glossary for the used terms in Sect. V-C. The generic taxonomy for steganography embedding patterns is presented in Sect. V-D. To finalize the generic taxonomy, we discuss design rules, including hybrid patterns, in Sect. V-E, and representation patterns in Sect. V-F.

A. Introduction

The central aspect of our taxonomy is to split all patterns into two categories:

1) *Embedding Patterns* describe how secret information is embedded into a cover object, such as an image file or a network packet.
2) *Representation Patterns* describe how the secret information is represented in a cover object, which is the essential knowledge for the receiver to extract the secret information from the cover object.

It must be noted that when secret data is embedded via the pattern A, it is not necessarily represented by the same pattern, but it can be. Two examples illustrate this statement:

1) *Embedding Pattern = Representation Pattern:* CS sends an IP packet to CR in which it manipulates the least significant bit(s) (LSB) of the Time to Live (TTL) field. CR reads the very same value. Thus, the embedding uses the so-called **State/Value Modulation** pattern while the hidden information is also represented by this pattern.
2) *Embedding Pattern ≠ Representation Pattern:* An example for a covert channel where the embedding and representation patterns differ is the *virtual machine* (VM) migration channel proposed by Spiekermann et al. [45], where the CS relocates a VM (e.g., from Europe to Australia, using commands sent through the **State/Value Modulation** pattern) and the CR observes the round trip time to the particular VM, i.e., the CR measures the temporal location (i.e., *position*) of network packets (**Element Positioning** pattern).

B. Naming Conventions

Hiding patterns are identified by a number (really an identifier as alpha-numeric symbols are used, cf. Sect. V-B1) and a name (Sect. V-B2).

1) Enumeration of Patterns: The enumeration of patterns follows the convention **{E,R}i[D]**, where the regular expression {A,B,C} means that one of the values A, B, or C must be selected and the regular expression [D] indicates that D is an optional parameter.

The first parameter **{E,R}** must be an upper-case **E** if the pattern is an embedding and an upper-case **R** if the pattern is a representation pattern.

The parameter **i** is a natural number to enumerate a pattern (so-called tier-1 pattern). Embedding patterns follow the enumeration convention **Ei** (*embedding; number i*). A single natural number *i* here refers to the highest level (tier 1) of a pattern hierarchy, whereas sub-patterns add an additional number preceded by a dot, e.g., **Ei.j** (the *j*-th sub-pattern of the embedding pattern E*i*). Additional hierarchy layers can be represented accordingly, such as **Ei.j.k** or even **Ei.j.k.l**, if necessary. Pattern numbers are assigned in an incremental manner so that future patterns can be assigned a new number easily.

Patterns can be domain-specific, e.g., a pattern could only be defined for the domain of text steganography. Therefore, the optional parameter **D** is used, where **[D]=[{n,d,t,c,f,o}x]**, with **n** (network steganography), **d** (digital media steganography), **t** (text steganography), **c** (cyber-physical steganography), **f** (filesystem steganography), and **o** (other domain of steganography). **x** is a domain-specific pattern number that starts with 1 for the first pattern and is incremented for succeeding sub-patterns at the same hierarchy level. For instance, the representation pattern **R1t1** tells us that it is a representation pattern with the number 1 and in particular it is the first text-steganography sub-pattern of **R1**.

In special cases, **x** can contain additional hierarchy levels. Thus, the pattern **E2t1.1** tells us that the pattern is a sub-pattern of the text steganography domain pattern **E2t1**, which is the text steganography version of the pattern **E2**. While not being applied in the current version, one could also derive domain-specific sub-patterns of generic sub-patterns, e.g., **E2.2t1.1**.

2) Naming of Patterns: The naming of patterns follows a clear structure and contains the following three components:

- its *number* as already described in Sect. V-B1, followed by a dot;
- the *modifiable object* that is altered for the steganographic data (an *Element* or *State/Value*);
- the *action* to be performed on the object (e.g., a *Modulation* or a *Positioning*).

The full pattern name separates all three components by spaces, e.g., **E1. State/Value Modulation**, i.e., the pattern is an embedding pattern with the number 1, and it encodes hidden information by influencing the state or value of some element. Sect. V-C provides a list of objects and actions. When necessary, additional objects and actions might be defined in the future after careful consideration of side-effects.

For all tier-1 patterns, the combination of modifiable object and action must be binomial (i.e., containing only two words). For sub-patterns, binomial names can be extended by additional terms (this is also the case in biology, e.g., *Canis lupus familiaris*, i.e., the domestic dog). While in biology the additional term would be put at the end of the name, for the sake of clarity we decided to put it in front of the modifiable object, e.g., "Random" for "Random State/Value Modulation".

The first letter of each word of a pattern name shall be written with an upper-case letter in texts (e.g., "E2. State/Value Modulation" instead of "e20. state/value modulation") to indicate that the term refers to a pattern. In this paper, all patterns of the novel taxonomy are written in **bold** font, while historic patterns are not.

C. Glossary

Even if the creation of an exhaustive, non-ambiguous vocabulary for steganographic applications is outside the scope of this work, reducing possible confusions or overloading of terms is fundamental to not void the efficiency and expressiveness of the taxonomy. Specifically, the term *modifiable object* we define as the *general* object type that will be used to contain or represent the secret information. The process of hiding data within the cover depends on the mechanism or pattern used. In the following, we refer to such a process as *embedding*, *injecting* or *hiding*. Because of the focus on objects, our taxonomy can be considered *object*-oriented; it describes the steganographic operation performed on objects using *actions*.

In general, patterns can be used to describe the process of hiding information for storage purposes as well as for secretly moving data among two endpoints. To avoid burdening the text, when the "transmissional" nature of the embedding process is not obvious, we will explicitly identify the covert sender and receiving side as to emphasize the origin and the destination of the steganographic communication.

For the specific case of defining the taxonomy as well as to describe patterns, the following formal definitions have been introduced. Please note that in comparison to previous attempts (Fig. 4, left side), we heavily simplified our terms and their relations due to the inconsistent pattern namings and terminology aspects already mentioned in Sect. II.

1) Modifiable Objects (see Fig. 4, right side, and Tab. I for examples):
   - An *Element* represents the object to be created, modified or deployed for steganographic data hiding. Elements can occur alone or in a sequence, e.g., *i*) a network packet; *ii*) a word/character of a text; *iii*) a pixel of an image; *iv*) a CPS command, e.g., a BACnet Read Property command or Who-Is message, a CPS sensor, a CPS sensor's polling rate, packet sending rate of a flow; *v*) a single file, folder or inode.
     Sub-elements: Elements can contain further elements, which can be considered their sub-elements (see Fig. 5 for an illustration). Sub-elements are equal to elements — the term *sub-element* just states that some element belongs to a higher-level parent element. Sub-elements can be stored in their parent element or can be a property of that parent element. For instance, an IPv4 network packet can contain option headers which can all be considered elements. Each IPv4 packet and each of its option headers has header fields, which are also elements. Further, each IPv4 packet has a header size, a time at which it occurs, etc., and so does each header field. These properties are also considered elements, although they are *virtual*, i.e., not explicitly stored in some memory location. Another example could be a CPS sensor as an element that contains several sub-elements, such as a temperature attribute.
   - The *State* or *Value* of an element is the actual state or value that an element currently has, i.e., not the property itself, e.g., not the size attribute of a network packet but the actual size value. Examples: *i*) the value of a TCP header field (the TCP header *element*'s actual bit-value) or the IPv4 IHL value; *ii*) the name of a character's font *iii*) the x-, y- or z-coordinate of a player in a 3D game; *iv*) an actual setpoint of a CPS device; *v*) the size value of a single file.
2) Actions (see Tab. I for examples):
   - The *Modulation* of an element's *state/value* is the selection of one particular state/value out of multiple possible states/values. For instance, if the element is the IPv4 TTL field, its *value* would have a range between 0 and 255, and modulation would select one of the possible values to embed a secret inside the TTL. Similarly, the *state* of an element that is a CPS window actuator could switch between *open* and *closed*.
     General modulation can be applied to various types of states/values, including such that represent *reserved/unused* or *random* values. These special cases are represented as sub-patterns in Fig. 6.
   - *Occurrence* is a predominant special case of modulation that appears several times in steganography research and was thus separated from the *general* modulation above. Occurrence refers to the modulation of the temporal or spatial location: an element, such as an IP packet, can have a sub-element (here in the sense of a property) that describes the time or location of appearance. This appearance element's value is then modified. For instance, an IP packet occurs at some point in time and a character of text occurs at some

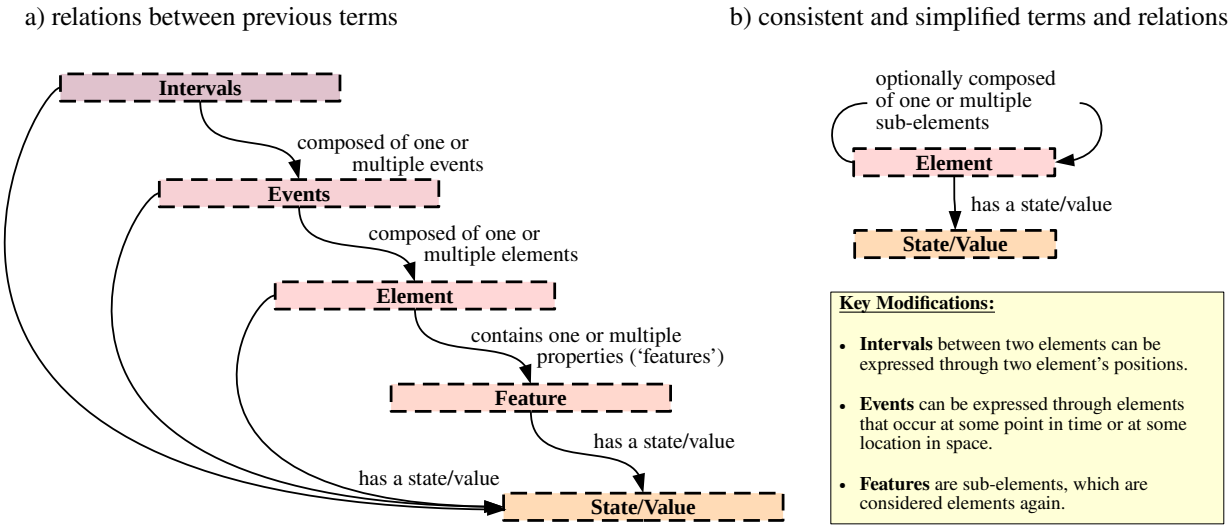

Fig. 4. Relations and terms have been simplified between the 2021-taxonomy (a) and the novel taxonomy (b).

TABLE I
DIFFERENTIATION BETWEEN THE TYPES OF *objects* USED IN THIS PAPER.

Domain	Element Examples	State/Value Examples
network steganography	network packet (e.g., IP packet); header field (e.g., TCP seq. no.); packet size property; time of occurrence property of a packet	actual packet size in bytes; actual TCP sequence number; time of sending/arrival
text steganography	a text; a paragraph; a character; line spacing; font of a character; size of a character; text length	actual color value; actual font name; actual length of text
digital media steganography	pixel of an image; PNG file header attributes; color attribute of a pixel; image size property	actual color value; actual image size value
cyber-physical systems steganography	a sensor; an actuator; control command (e.g., BACnet *ReadProperty*); temperature value of a sensor; status of an actuator	actual state of an actuator (open/closed); actual temperature value of a sensor
filesystem steganography	file; inode; file creation/deletion timestamp attributes; file size attribute; file header attribute; inode attribute (e.g., inode number field)	file's actual status (e.g., existent/deleted); actual inode number's value

location in a text document.

There are two specific variants of Occurrence:

a) *Enumeration* means that the overall number of occurrences of elements is altered. Enumeration is usually used for a higher-level element (e.g., a network flow) containing lower-level elements (e.g., network packets), e.g., the *number of lower-level elements* is considered an element (again in the sense of a property) of the higher-level element. For instance, one could have a network flow that contains either 10 or 20 IPv4 packets or a text document that contains 100 or 101 letters to represent a secret information.

b) *Positioning* selects the temporal/spatial position of one element (optionally in a sequence of elements). Again, the position of an element is an element (property) by itself. For instance, a particular TCP segment could be positioned at some location in a flow to embed a secret.

D. Generic Taxonomy Patterns

Our novel taxonomy of hiding patterns contains two major branches (see Fig. 6): patterns that describe how information is *embedded* in a cover object and patterns that describe how embedded secret data is *represented* through it. Whenever possible, i.e., as long as there is a matching embedding pattern, the representation patterns are derived from the embedding patterns and are not re-defined explicitly.

We define the following embedding patterns:

- **E1. State/Value Modulation**: The covert message is embedded by modulating the state or value of an element. Examples: *i*) modulating physical states, such as proximity, visibility, force, height, acceleration, speed, etc. of certain devices; *ii*) changing the values of the network packet header fields; *iii*) modulating the view angles of a player in a 3D multiplayer online game [46]. Pattern **E1** contains five sub-patterns that represent special variants of it:
  - **E1.1. Reserved/Unused State/Value Modulation**: The covert message is embedded by modulating reserved/unused states/values of elements.

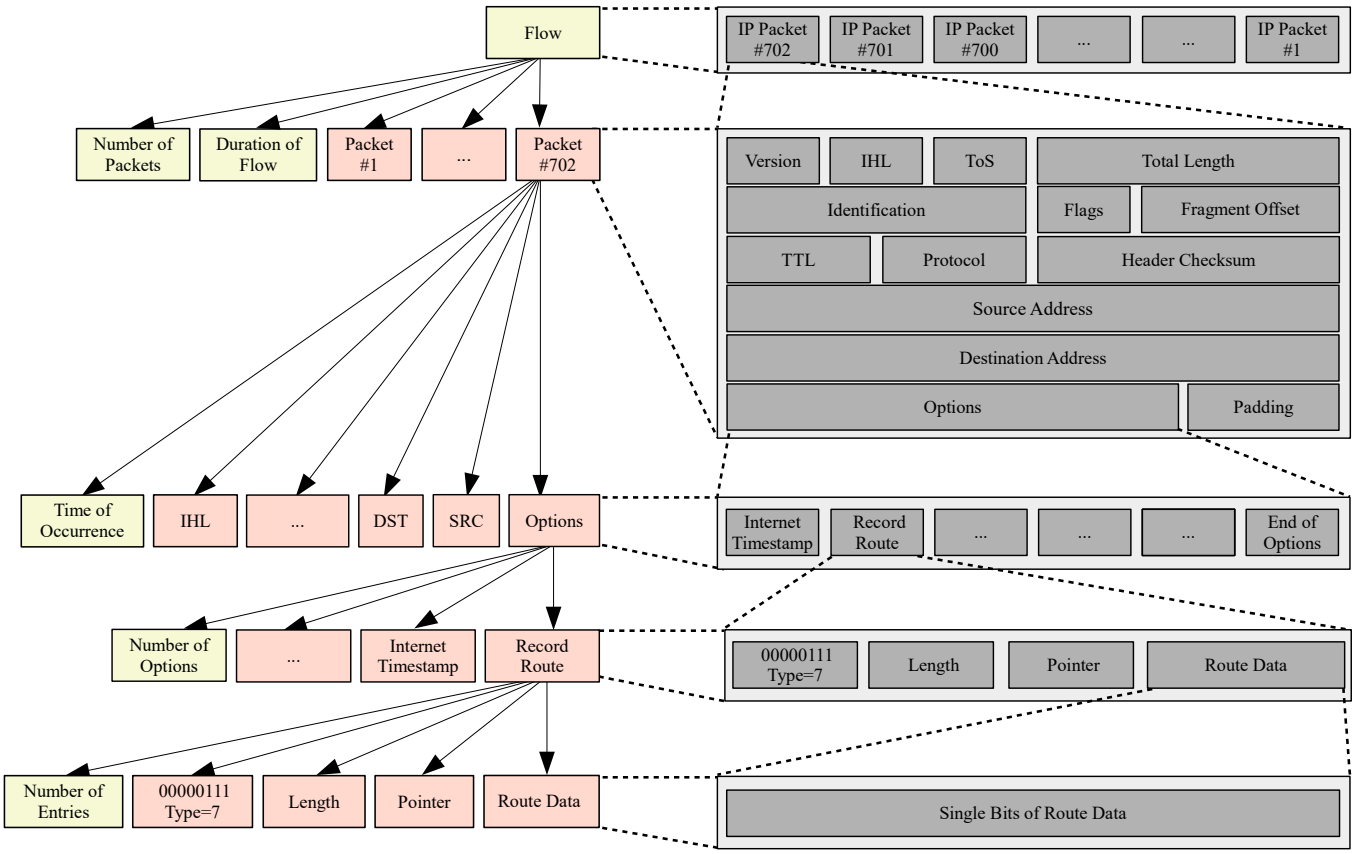

Fig. 5. Exemplified illustration of elements and sub-elements for a network flow. The element 'flow' contains several sub-elements, including some packets. Each of these packet elements contains further sub-elements, of which the illustrated 'options' can be one, containing further sub-elements, and so forth. **Note:** Yellow elements are virtual elements, while light-red elements reflect those actually stored inside a parent element.

Examples: *i*) overwriting a network packet's reserved bits; *ii*) modulation of unused registers in embedded CPS equipment [33].

– **E1.2. Random State/Value Modulation**: A (pseudo-) random value or state is replaced with a secret message (that is also following a pseudo-random appearance).
Examples: *i*) replacing the pseudo-random content of a network header field with encrypted covert content; *ii*) encoding a secret message in the randomized selection of a starting player's character type in an online game.

– **E1.3. LSB State/Value Modulation**: The least significant bits of elements are modulated.
Examples: *i*) modulation of the LSBs in TTL fields of network packets; *ii*) LSB modulation in pixels within digital images.

– **E1.4. Character State/Value Modulation**: The features of characters in elements are modulated.
Examples: *i*) modification of a characters' case in a network packet; *ii*) perturbation of characters' glyph in FontCode [47]; *iii*) modulation of a characters' color [48] or scale [49].

– **E1.5. Redundancy State/Value Modulation**: The redundancy of an element's content is modulated (this is usually applied by a succeeding pattern that fills the gained space with covert data), e.g., by means of compression.
Examples: *i*) transcoding a payload field; *ii*) compressing the content of a digital file.

• **E2. Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number of elements appearing in a flow (succeeding messages *occur* shortly or long after previous ones).
Examples: *i*) sending a specific network packet at 6pm; *ii*) influencing the time at which a drone starts its journey to some destination (or its arrival time); *iii*) letting a sentence appear at some location on a white page; *iv*) creating a video file with either n or m ($n \neq m$) frames.
Pattern **E2** has two sub-patterns, which are special forms of element occurrences:

– **E2.1. Element Enumeration**: An attribute describing the quantity of sub-elements is modulated. This pattern is also applied when the size of an element is increased by adding more sub-elements to an element (e.g., adding more payload characters or additional header components to a network packet or by adding additional characters to a text).
Examples: *i*) fragmenting a network packet into either n or m ($n \neq m$) fragments; *ii*) repeating selected characters or paragraphs in texts; *iii*) letting network packets occur multiple times; *iv*) Encoding secret information

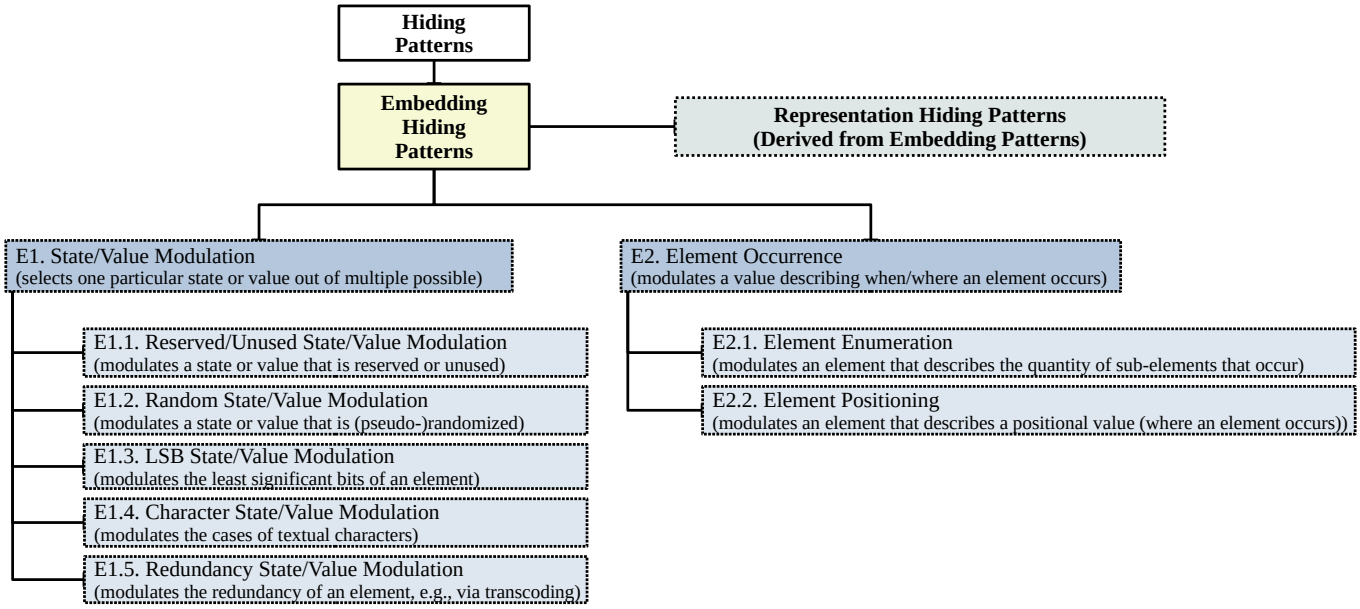

Fig. 6. The generic taxonomy's core component: embedding hiding patterns.

through the number of IPv6 extension headers or IPv4 option headers.

– **E2.2. Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of an element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5).
Examples: *i*) a packet sent at some certain point in time in a flow (temporal positioning); *ii*) a series of packets being sent to influence the inter-packet gap between these packets (every element is positioned separately); *iii*) a character of text placed at some particular location (spatial positioning).

E. Design Rules

Rule of Explicit Operations (Non-hybrid Patterns Rule): The operations performed by a hiding method must be stated explicitly, i.e., they must not be omitted to simplify the categorization through patterns.

Earlier research work considered several hiding methods as single patterns that *can* actually demand hybrid methods to be applied. One example is the *Size Modulation* pattern originally defined in [5]. To increase the size of a network packet, one needs to describe it with two patterns: First, the actual size of the packet must be increased, which is usually achieved by adding more header elements (e.g., to an IPv4 header) using pattern **E2.1. Element Enumeration**. Secondly, at least for the case of IPv4, the actual value of the IHL field must be also adjusted using the **State/Value Modulation** pattern. This pattern can thus be considered as a *hybrid* pattern when the size value must be adjusted. However, this usually depends on the implementation of a CS since sometimes, the CS process does not necessarily set the IHL field (as this is done by the operating system kernel instead and thus not explicitly part of the CS). In such cases, the pattern could be considered as a *non-hybrid* pattern. In other words, the differentiation between a hybrid and a non-hybrid pattern depends on the actual actions performed by the CS.

Similar cases are possible for other scenarios. For instance, when the order of TCP segments in a flow is jumbled to signal a secret message: each packet (element) is positioned (**E2.2. Element Positioning**) but it must also contain the related header information, especially the relevant sequence number (**E1. State/Value Modulation**).

Finally, the connection resets, reconnects, and disconnects are hybrid events. They require certain header bits or commands to be set (pattern **E1. State/Value Modulation**) while they also require a timing (element positioning in the temporal sense, i.e., pattern **E2.2 Element Positioning**).

Rule of Minimized Abstraction: In general, the lowest level of abstraction should be applied to keep patterns unified and the complexity low. This means that virtual properties, i.e., sub-elements that are not *contained* in an element but solely *describe* it (such as a number of packets of a flow or the length of a text) should not be used if they can be described with elements that are actually present (e.g., enumeration of packets or characters as described by pattern **E2.1. Element Enumeration**).

For instance, when we consider a network connection's disconnect, we can view it from the high-level perspective of a connection or flow, or from a lower-level perspective of separated packets. According to this rule, a flow's sub-element *connection-state*={*established,disconnected*} should be discarded in favor of a low-level view. Instead, the above-mentioned view (applying patterns **E1. State/Value Modulation**, e.g., to set a RST flag in a TCP segment, and **E2.2. Element Positioning**, to send such a TCP segment at the desired moment) should be applied.

Rule of Mandatory Occurrence of Elements: In most cases, elements, such as network packets or text characters,

need to occur to perform steganographic actions. If the occurrence is *not of key relevance* for the steganographic embedding method (e.g., a **E1.2. Random State/Value Modulation**), we follow the *modus operandi* of previous work which omits to mention an element occurrence pattern (i.e., pattern **E1.2. Random State/Value Modulation** is not a hybrid pattern as that would otherwise also require **E2. Element Occurrence** (or one of its sub-patterns)) for the sake of simplicity and since it would otherwise render *every* pattern a hybrid pattern. This rule represents the only exception from rule one (*Rule of Explicit Operations*).

Rule of Derivation: To ensure a consistency of the core taxonomy with all its sub-taxonomies, whenever possible, patterns should be derived from the core taxonomy instead of being defined from scratch. If a derivation from a core taxonomy's pattern is not feasible but another sub-taxonomy (or the currently used sub-taxonomy) provides a pattern from which can be derived from, it must be derived from the particular pattern. The derivation should be noted in a pattern's name (as shown by the sub-taxonomies of Sect. VI). Only if no such derivation is feasible, a new pattern shall be created.

Rule of Selecting a High-level Pattern when in Doubt: Sometimes, it might be unclear whether some hiding method should be categorized by some pattern X or its sub-pattern $X.y$. When unsure, the higher-level pattern X should be selected as it covers a broader view. This rule limits categorization errors.

F. Representation Patterns

As mentioned in Sect. V-B, representation patterns are derived from embedding patterns and start with the upper-case letter "R". To this end, the description of embedding patterns is slightly adjusted to reflect the representation of embedded data. We exemplify this approach using the embedding pattern **E1. State/Value Modulation**, which was defined as follows in the previous section:

> *The covert message is embedded by **modulating** the state or value of an element. [...]*

The representation pattern would be **R1. State/Value Modulation**. Pattern **E1**'s description is modified so that it reflects how the secret information is presented after the embedding pattern **E1. State/Value Modulation** has been applied:

> *The covert message is **recognized by observing modifications of** the state or value of an element. [...]*

It follows that the embedding pattern's examples can also be flipped based on the existing ones (modifications of the original examples are shown in *italic* font): *i*) *recognizing* physical states, such as proximity, visibility, force, height, acceleration, speed, etc. of certain devices; *ii*) *recognizing the* values of the network packet header fields (e.g., target IP address of ARP [50], Hop Count value in IPv6 [51] or the LSB in the IPv4 TTL); *iii*) *recognizing* the x-, y-, or z-coordinates of a player in a 3D multiplayer online game [46];

We omit further examples to prevent overloading the article, but as can be seen, the derivation of the representation from embedding patterns can be considered as simple.

VI. Sub-taxonomies for Domain-specific Patterns

In this section, we focus on sub-taxonomies. These are derived from the generic taxonomy. Sub-taxonomies describe patterns more precisely than patterns of the generic taxonomy as they can add details that are specific to a particular domain. We first present the sub-taxonomy for the domain of network steganography in Sect. VI-A, followed by text steganography in Sect. VI-B, digital media steganography in Sect. VI-C, CPS steganography in Sect. VI-D, and filesystem steganography in Sect. VI-E.

According to the naming convention in Sect. V-B, the pattern identifiers of the sub-taxonomies must end with a lower-case indicator for the particular domain, such as "n" for network steganography or "d" for digital media steganography.

The hierarchy level where sub-taxonomy patterns are added is reflected by their enumeration. For instance, a pattern with the identifier **E1n1** would be the first network-steganography sub-pattern of the core pattern **E1. State/Value Modulation** while **E1.2d3** would be the third digital-media steganography sub-pattern of the core pattern **E1.2. Random State/Value Modulation**.

A. Network Steganography Sub-Taxonomy

For network steganography, there already exists a taxonomy of hiding patterns [5] that was enhanced over the years by several papers [2], [52]–[55]. In the remainder, we categorize all existing methods described by these earlier patterns into our taxonomy and derive domain-specific patterns as needed.

First of all, we surveyed all existing hiding patterns from network steganography. Tab. II shows timing patterns, Tab. III storage patterns that focus on non-payload (i.e., headers and padding fields), and Tab. IV storage patterns that focus on payload.

As shown by the three tables, not all of the existing hiding patterns are actually representing hiding methods that can be considered whole patterns: some are highly specialist hiding methods that are no patterns due to the comments given in the tables. Moreover, some patterns must be considered as hybrid patterns as they incorporate more than one core pattern's functionality. However, several of the original patterns remain indeed as full patterns in our new taxonomy, including several overlaps, i.e., multiple patterns are special cases of the same core patterns. For instance, the core pattern **E2.2. Element Positioning** describes the original timing patterns PT1 and PT3 as well as the original storage patterns PS2 and PS2.a because all these patterns combine the positioning of elements, but from different angles (be it packets or elements within packets, such as header fields). In case of PT1. Inter-packet Times, the time at which packets are sent is adjusted (an element positioning) and the time between packets is used to represent a secret symbol. In the case of PT3. Rate/Throughput, the time of succeeding packets is adjusted to encode a hidden symbol through the number of packets that are sent per unit of time (each element is positioned independently, though). PS2 and PS2.a position elements *within* a packet. In all four cases, the covert sender essentially performs the same action: adjusting

TABLE II
INTEGRATION OF THE ORIGINAL NETWORK STEGANOGRAPHY **TIMING** PATTERNS INTO THE NEW TAXONOMY.

Pattern of Existing Taxonomy	**Ref.**	**Short Description**	**Generic Emb. Pattern (New Taxonomy)**	**Type**	**Sub-taxonomy Pattern (New Taxonomy)**	**Comments**
PT1. Inter-packet Times (former: Inter-arrival Times)	[2], [5]	The CS alters the timing intervals between network PDUs (inter-packet times) to encode hidden data.	**E2.2. Element Positioning**	Pattern	**E2.2n1. Network Element Positioning**	An inter-packet time is represented by two packet (element) occurrences in time, one timestamp can then be subtracted from the other (or both could be added/multiplied to encode a secret symbol). Instead of adjusting an inter-packet time, each element is placed separately at the time of embedding.
PT2. Message Sequence Timing	[2]	The CS encodes secret symbols through the timing of message sequences.	**E2.1. Element Enumeration**	Pattern	**E2n1. Network Element Enumeration**	The number of occurrences of elements are altered to embed a secret symbol (usually followed by some reaction).
PT3. Rate/Throughput	[5]	The CS alters the data rate of traffic.	**E2.2. Element Positioning**	Pattern	**E2.2n1. Network Element Positioning**	Elements (packets) are positioned either quickly behind each other, or not. As in the case of PT1, each element (packet) is placed separately.
PT10. Artificial Loss	[2]	The CS signals secret symbols through artificial loss of transmitted PDUs.	**E2. Element Occurrence**	Pattern	**E2n1. Network Element Occurrence**	Which message is lost depends on which message is not lost, i.e., which elements occur.
PT11. Message Ordering (former: PDU Order/ Manipulated Message Ordering)	[2], [5], [52]	The CS encodes data using a synthetic PDU order.	**E2.2 Element Positioning** (& **E1. State/Value Modulation**)	Pattern / Hybrid Pattern	-	The PDUs are located at specific points in time, however, their sequence/identification numbers must also be modulated in some cases (when emitted by a CS, instead of being delayed by a CS-router).
PT12. Retransmission	[5]	The CS re-transmits previously sent or received PDUs.	**E2.1. Element Enumeration**	Pattern	**E2.1n1. Network Element Enumeration**	An element (packet) occurs multiple times.
PT13. Frame Collisions (former: PDU Corruption/Loss)	[2], [5]	The CS causes artificial frame collisions to embed secret symbols by letting two packets occur closely behind each other.	**E2.2. Element Positioning**	Hybrid Pattern	**E2.1n1. Network Element Enumeration**	Two elements (packets) are positioned within the same time slot, thus, causing a collision.
PT14. Temperature	[2]	The CS influences a third-party node's clock skew, e.g., using burst traffic.	-	Specific indirect and hybrid hiding method; not a pattern.	-	PT14 contains aspects of an embedding and of a representation pattern. It further mixes two domains: network steganography with CPS steganography (CPU temperature).
PT15. Artificial Reconnections	[54]	The CS employs artificial (forced) reconnections to transfer secret messages.	**E1. State/Value Modulation** & **E2.2 Element Positioning**	Specific indirect and hybrid hiding method, not a pattern	-	Reconnects (and disconnects) are hybrid events. They require certain header bits (e.g., FIN in TCP) or commands to be set (**E1. State/Value Modulation**) while they also require a timing (element positioning in the temporal sense, i.e., pattern **E2.2 Element Positioning**) as the time of reconnection is used to encode a secret message together with a sender's address. Moreover, this behavior represents an indirect covert channel as a sender influences the reconnections of third-party nodes observed by the CR.
PT16. Artificial Resets	[55]	The CS causes a connection reset of third-party nodes, whose connection states are observed by one or more CRs.	**E1. State/Value Modulation** & **E2.2 Element Positioning**	Same as PT15	-	See PT15, above. Resets are comparable to reconnections.

the temporal/spatial location of elements. The only difference is the element is either a whole packet or a header field.

However, our intention is not to generalize while losing descriptive detail. To cover all special variants of new patterns that combine the ideas of multiple original patterns, the following pattern descriptions cover all related aspects in their description:

- **E1n1. Network State/Value Modulation**: The covert message is embedded by modulating the state or value of a network element, such as a frame, packet, header element, payload field etc.
  Original network steganography patterns: PS11. Value Modulation (including all sub-patterns) and PS31. User-data Value Modulation, and *partially*: PT11. Message Ordering (only the aspect of modulating sequence numbers) and PT15. Artificial Reconnections/PT16. Artificial Resets (in both cases, some header fields, such as RST flags, must be modulated). Additionally: multiple patterns to cover **E1n1**'s sub-patterns (see particular descriptions below).
  Examples: *i*) changing values of the network packet header fields (e.g., target IP address of ARP [50], *ii*) Hop Count value in IPv6 [51] or *iii*) the LSB in the IPv4 TTL).
  This pattern has the following sub-patterns, which are special forms of state/value modulations:
  – E1.1n1. Network Reserved/Unused State/Value

TABLE III
INTEGRATION OF THE ORIGINAL NETWORK STEGANOGRAPHY **STORAGE** PATTERNS FOR **NON-PAYLOAD** (HEADERS AND PADDING) INTO THE NEW TAXONOMY.

Pattern of Existing Taxonomy	Ref.	Short Description	Generic Emb. Pattern (New Taxonomy)	Type	Sub-taxonomy Pattern (New Taxonomy)	Comments
PS1. Size Modulation	[5]	The CS uses the size of a header element or a PDU to encode a hidden message.	**E2.1. Element Enumeration**	Pattern	**E2n1. Network Element Enumeration**	A packet's size is increased by adding more sub-elements (e.g., header fields or payload components) to a packet.
PS2. Sequence	[5]	The CS alters the sequence of header/PDU elements to encode hidden information.	**E2.2. Element Positioning**	Pattern	**E2.2n1 Network Element Positioning**	The position of elements is modified as covered by **E2.2**.
PS2.a. Position	[5]	The CS alters the position of a given (single) header/PDU element to encode hidden information.	**E2.2. Element Positioning**	Pattern	**E2.2n1 Network Element Positioning**	Special case of PS2, but limited to one element.
PS2.b. Number of Elements	[5]	The CS encodes the hidden information by the number of header/PDU elements transferred.	**E2.1. Element Enumeration**	Pattern	**E2n1. Network Element Enumeration**	The number of an element's sub-elements is modified as covered by **E2.1**.
PS3. Add Redundancy	[5]	The CS creates a new space within a given header element or within a PDU to hide data into.	**E2.1. Element Enumeration** & **E1.1. Reserved/Unused State/Value Modulation**	Hybrid Pattern	-	First, additional sub-elements (packet elements) must be created (**E2.1**), afterwards, these new (but unused) elements are filled with secret data (**E1.1**).
PS10. Random Value	[5]	The CS embeds hidden data in a header element containing a (pseudo) random value.	**E1.2. Random State/Value Modulation**	Pattern	**E1.2n1. Network Random State/Value Modulation**	Pattern matches definition of **E1.2**, but is limited to the network scenario.
PS11. Value Modulation	[5]	The CS selects one of the n values that a header element can contain to encode a hidden message.	**E1. State/Value Modulation**	Pattern	**E1n1. Network State/Value Modulation**	General modulation of an element's value, thus **E1**.
PS11.a. Case Modulation	[5]	The CS uses case-modification of letters in header elements to encode hidden data.	**E1.4. Character State/Value Modulation**	Pattern	**E1.4n1. Character Network State/Value Modulation**	Letter case modulation matches **E1.4**.
PS11.b. LSB Modulation	[5]	The CS uses the LSB of header elements to encode the hidden data.	**E1.3. LSB State/Value Modulation**	Pattern	**E1.3n1. Network LSB State/Value Modulation**	An element value's LSB are modulated, thus special case of **E1.3**.
PS11.c. Value Influencing	[53]	The CS (directly or indirectly) influences some (out of n possible) values in a way that a covert channel receiver can determine the value. The value is not directly written, but influenced by altering another value or surrounding networking conditions.	depends on the implementation (temporal and spatial actions necessary)	Not a pattern	-	Indirect hiding method that combines embedding and representation methods in its description.
PS12. Reserved/ Unused	[5]	The CS encodes hidden data into a reserved or unused header/PDU element.	**E1.1 Reserved/Unused State/Value Modulation**	Pattern	**E1.1n1. Network Reserved/Unused State/Value Modulation**	Matches description of **E1.1**.

Modulation: The covert message is embedded by modulating reserved/unused states/values of network elements.
Original network steganography patterns: PS12. Reserved/Unused and, *parts* of PS3. Redundancy (see Tab. III), parts of PS30. Modify Redundancy (see Tab. IV), as well as parts of PS31. User-data Value Modulation and Reserved/Unused (also Tab. IV).
Examples: *i*) ten examples for the original Reserved/Unused pattern are surveyed in [5], showing that unused/reserved fields in IEEE 802.3 and 802.5 [56]–[58], IPv4 [56], IP-IP, IPv6 [51], TCP [56], [59], ICMP [60], [61], BACnet [62], DHCP [63], and IPSec [59] can be exploited by overwriting certain header fields, such as the IP Identifier. Moreover, additional and recent works have shown that more protocols are vulnerable due to their unused/reserved fields, such as *ii*) MQTT [53] and *iii*) SIP [64].

– **E1.2n1. Network Random State/Value Modulation**: A (pseudo-)random value or state of/in the network data is replaced with a secret message (that is also following a pseudo-random appearance).
Original network steganography patterns: PS10. Random Value and PS33. User-data Random Value Modulation.
Examples: *i*) [5] already mentions some examples, such as the utilization of the pseudo-random IP Identifier field [65], the TCP ISN [65], [66], the DHCP xid field [63] and the SSH MAC field [67]. Additional examples can be found in *ii*) cryptographic protocols that use nonces during the challenge-response process [68] as well as in *iii*) IoT protocols with random value fields, such as MQTT [53].
– **E1.3n1. Network LSB State/Value Modulation**: The LSB of network elements are modulated.
Original network steganography patterns: PS11.b.

TABLE IV
INTEGRATION OF THE ORIGINAL NETWORK STEGANOGRAPHY **STORAGE** PATTERNS FOR **PAYLOAD-BASED METHODS** INTO THE NEW TAXONOMY. [*] INDICATES PATTERNS WHICH WERE ADDED FOR COMPLETENESS BUT WERE NOT OFFICIALLY DEFINED.

Pattern of Existing Taxonomy	**Ref.**	**Short Description**	**Generic Emb. Pattern (New Taxonomy)**	**Type**	**Sub-taxonomy Pattern (New Taxonomy)**	**Comments**
PS20. Payload Field Size Modulation (derived from PS1)	[52]	The CS uses the payload size to encode a hidden message.	**E2.1. Element Enumeration**	Pattern	**E2n1. Network Element Enumeration**	Equals original pattern PS1. Size Modulation, but with a focus on payload.
PS21. User-data Corruption	[52]	The CS blindly overwrites a packet's payload.	**E1. State/Value Modulation**	Not a pattern	-	Special case of **E1** being applied to network payload; the fact that the overwriting is *blind* does not make it an own pattern (in comparison to, e.g., **E1.1.** or **E1.2.** that focus on specific types of cover data). Moreover, example cases of [52] represent hybrid methods.
PS30. Modify Redundancy	[52]	The CS exploits the redundancy of user-data by transcoding them so that a free space for secret data is obtained (and then filled).	**E1.5. Redundancy State/Value Modulation** & **E1.1. Reserved/Unused State/Value Modulation**	Hybrid pattern	**E1.5n1 Network Redundancy State/Value Modulation** & **E1.1n1. Network Reserved/Unused State/Value Modulation**	First, an element's values are modified (e.g., by transcoding or compression) so that free space is created in a packet (**E1.5**); the space is then filled with secret data (**E1.1**).
PS31. User-data Value Modulation and Reserved/Unused	[52]	The CS performs a modulation of payload values.	**E1. State/Value Modulation / E1.1. Reserved/Unused State/Value Modulation**	Two *separate* patterns	**E1n1. Network State/Value Modulation** & **E1.1n1. Network Reserved/Unused State/Value Modulation**	Special case of **E1./E1.1.** being applied to payload elements.
PS32. User-data Sequence Modulation (plus sub-patterns)	[*]	The CS performs a PS2/PS2.a/PS2.b-like sequence modulation of payload fields.	**E2.1. Element Enumeration / E2.2. Element Positioning**	Two *separate* patterns	**E2.2n1. Network Element Enumeration** *-or-* **E2.1n1. Network Element Positioning**	Special case of the original patterns PS2/PS2.a/PS2.b being applied to payload elements.
PS33. User-data Random Value Modulation	[*]	The CS performs a PS10-like random value modulation of payload fields.	**E1.2. Random State/Value Modulation**	Pattern	**E1.2n1. Network Random State/Value Modulation**	Special case of the original pattern PS10 being applied to payload elements.

LSB Modulation.
Examples: *i*) [5] provides several examples, such as the modulation of the LSBs in the IPv6 Hop Limit field [51], IPv4 TTL field, modulation of the IP timestamp option's LSB [56], TCP timestamp option [69], DHCP's LSB of the secs field [63], the BACnet hop count field, or the XMPP id attributes LSB [70]. *ii*) Recent work has applied the LSB method to the Modbus protocol [71].

– **E1.4n1. Network Character State/Value Modulation**: The features of characters in network elements are modulated.
Original network steganography patterns: PS11.a. Case Modulation.
Examples: *i*) case modulation of characters in HTTP headers [72]. This method can also be applied to several other textual protocols, such as SMTP, IMAP, POP3, NNTP etc.

– **E1.5n1. Network Redundancy State/Value Modulation**: The redundancy of a network element's content is modulated (this is usually applied by a succeeding pattern that fills the gained space with covert data), e.g., by means of compression.
Original network steganography patterns: *part of* PS30. Modify Redundancy (cf. Tab. IV).
Examples: three examples of [52] can be adjusted in their formulation to reflect this pattern: *i*) compression of existing payload (gained space can be used by **E1.1n1. Reserved/Unused State/Value Modulation** afterwards) [73]; *ii*) transformation of the VAD-enabled IP telephony voice stream into a non-VAD one and fill the gaps using artificially generated RTP packets containing secret data by applying another pattern [74]; *iii*) approximation of the F0 parameter of the Speex codec which carries information about the pitch of the speech signal (again, the saved space can then be used by another pattern) [75].

- **E2n1. Network Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number of packets appearing in a flow (succeeding messages *occur* shortly or long after previous ones).
Original network steganography patterns: PT10. Artificial Loss and PT12. Retransmission as well as multiple patterns to cover **E2n1**'s sub-patterns (see particular descriptions below).
Examples: *i*) sending a specific frame or packet multiple times (retransmission) as done in case of IEEE 802.11 [76] or TCP [77]; *ii*) performing a high number of frame transmissions (e.g., so that their occurrences influence the rate/throughput of a network link [78]); *iii*) selecting one out of multiple possible IPv4 option headers to appear; *iv*) dropping TCP segments with an even sequence number (artificial loss, i.e., non-occurrence *or* occurrence of all

other elements of a flow, except the dropped ones), *v*) not acknowledging TCP packets [79] (again a form of non-occurrence).
This pattern has the following sub-patterns, which are special forms of element occurrences:

– **E2.1n1. Network Element Enumeration**: An attribute describing the quantity of network sub-elements is modulated. This pattern also applies if sub-elements are added to a network element to increase its overall size (e.g., by adding more sub-elements to the payload in network packets).
Original network steganography patterns: PT2. Message Sequence Timing, PT12. Retransmission, PS1. Size Modulation, PS2.b. Number of Elements, PS20. Payload Field Size Modulation, as well as *parts* of PS3. Add Redundancy (see Tab. III) and PS32. User-data Sequence Modulation (see Tab. IV).
Examples: *i*) fragmenting a network packet into either n or m ($n \neq m$) fragments [80]; *ii*) letting network packets or commands occur just once or multiple times (e.g., artificial TCP segment or FTP command retransmissions) [77], [81]; *iii*) Encoding secret information through the number of IPv6 extension headers or IPv4 option headers; *iv*) creating additional (unused) space in network packets, such as adding an "unused" IPv6 destination option [82] (a variant of the former PS1. Size Modulation) or integration of additional SMTP header lines [83]; *v*) modulating the number of DHCP options [63].

– **E2.2n1. Network Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of a network element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5).
Original network steganography patterns: PT1. Inter-packet Times, PT3. Rate/Throughput, PT13. Frame Collisions, PS2. Sequence, PS2.a. Position, as well as *parts* of PT11. Message Ordering (see Tab. II) and PT15./PT16. Artificial Reconnections/Resets (see Tab. II).
Examples: *i*) a specific packet sent at some certain point in time in a flow (temporal positioning); *ii*) modulating the position of an existing TCP segment in a TCP stream[2]; *iii*) position of a specific IPv4 option in the list of options [5] as well as the sequence of multiple IPv4 options in the list of options [5], the order of DHCP options [63] or the order of HTTP header lines [72] or FTP commands [81] (each element is positioned individually, but overall, they form a sequence); *iv*) influencing the inter-arrival time of packets by positioning individual packets [84]–[88] (this can also be done to influence the throughput of a connection, e.g., for a switch [78] or a serial communication port [56]).

[2]Non pre-existing TCP segments would require the adjustment of the sequence number, which is discussed in Sect. V-E and Tab. II for PT11.

B. Text Steganography Sub-Taxonomy

For text steganography (as for all the following steganography domains), no pre-existing pattern-based taxonomy was available. For this reason, a literature study was performed, which resulted in the following patterns that we derived from the core taxonomy (again, only *embedding* patterns are described).

Please note that this sub-taxonomy is not limited to digital text steganography as certain methods might also be applied in a non-digital form (e.g., with ink on paper).

- **E1t1. Text State/Value Modulation**: The covert message is embedded by modulating the state or value of a text element, such as a paragraph or a character.
Examples: *i*) overwriting of random characters; *ii*) modulation of a paragraphs' or a sentences' border [49]; *iii*) replacement of some character with a homoglyph (similarly to DNS-based homograph attacks [89], in which homoglyphs are used).
The following sub-patterns have been discovered for text steganography:
  – **E1.2t1. Text Random State/Value Modulation**: A (pseudo-)random value or state of/in the text data is replaced with a secret message (that is also following a pseudo-random appearance).
  Examples: several methods come to mind quickly here as any appearance of (pseudo-)random text strings, such as textual bitstrings, hash values or printed cryptographic keys as well as TAN lists for banking and textual nonces can be replaced with steganographic data.
  – **E1.3t1. Text LSB State/Value Modulation**: The LSB of text elements are modulated.
  Examples: modulate the LSBs of a text's font color (or its R, G and/or B component) [90].
  – **E1.4t1. Text Character State/Value Modulation**: The features of characters in text elements are modulated.
  Examples: *i*) all cases in which a textual character's case is adjusted, e.g., in text files, source code, HTML files etc. *ii*) modulation of a characters' glyph [47], underlining [49], font type [91], color or luminance intensity (color quantization) [92].
  – **E1.5t1. Text Redundancy State/Value Modulation**: The redundancy of a text element's content is modulated (this is usually applied by a succeeding pattern that fills the gained space with covert data), e.g., by means of semantic-preserving compression.
  Examples: All semantic methods that include synonym substitution [93], paraphrasing [94], changing word spelling [95], using of abbreviations and acronyms with typographical errors [96], etc. belong in this pattern. In general, the semantics of texts can be kept, even if shortened. For instance, the sentence "This very moment can be considered relevant." can also be expressed in a shorter form: "This moment is relevant.".
- **E2t1. Text Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number

of text elements appearing in a message flow (succeeding messages *occur* shortly or long after the previous ones). Examples: *i*) All syntactic methods that include changing the diction and structure of text without significantly altering meaning or tone, by means of punctuation [97], shifting the location of the noun and verb [98], pointed letters and extensions in Arabic language [99], etc., belong here; *ii*) selecting one out of multiple plausible HTML tags to set a character in bold font (e.g., '`<b>`' vs. '`<strong>`') [72]; *iii*) removing commas in sentences (i.e., the elements do not appear) [97]; *iv*) use of emoticons and/or lingoes for hiding data in SMS and chats [100], [101].

This pattern has the following sub-patterns, which are special forms of element occurrences:

- **E2.1t1. Text Element Enumeration**: An attribute describing the quantity of text sub-elements is modulated. This pattern also applies if sub-elements are added to a textual element to increase its overall size (e.g., by adding more characters to a text so that the overall length of the text is influenced).
  Examples: *i*) repeating a white space element (or any Unicode space character) by duplicating it (or not) in a text (open space method) [97], [102]; *ii*) adding/removing tags in an HTML file (e.g., to influence the overall number of tags).
- **E2.2t1. Text Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of a text element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5). Examples: *i*) a character of text placed at some particular location (spatial positioning); *ii*) placing a specific HTML tag's attribute in the list of attributes, e.g., `<img text=''...'' src=''...'' alt=''...'' />` vs. `<img alt=''...'' text=''...'' src=''...'' />`. *iii*) line and word shift coding [103], [104]; *iv*) combining specific invisible Unicode characters to represent different groups of n secret bits [105].
- Special case **E2t1.1. Generative Text Element Occurrence**. There is one group of text steganographic methods that are identified with the umbrella term "generative linguistic steganography" (or random and statistical methods [18]). These can be considered a special variant of **E2t1. Text Element Occurrence** and are thus categorized as **E2t1.1.**. In essence, such methods directly transform the secret data into steganographic text, without any cover text, by using natural language models. To this aim, they first use a known model to learn the statistical language model from a large set of natural language sentences, and then they embed the secret data by encoding the conditional probability distribution of each word in the cover text generation process.
  Examples: *i*) This pattern can be realized by neural networks (e.g., Long Short-Term Memory - LSTM [106], Recurrent Neural Networks - RNN [107], Generative Adversarial Networks - GAN [108]), discrete random processes (e.g., Markov chain model [109]), and Context-Free-Grammars [110], just to mention some.

C. Digital Media Steganography Sub-Taxonomy

The whole field of media steganography is extremely wide with thousands of publications (scientific and technical) describing hiding methods and currently (mid 2022) more than 2,500 steganographic tools easily available via platforms such as GitHub or SouceForge. To provide a complete mapping of all methods encountered for all media formats is futile for this paper, instead it is shown that selected methods can actually be projected onto the proposed taxonomy, leaving the majority of the work that would be necessary to provide a full picture for follow-up publications.

As one of the most prominent textbooks in the field of media steganography, [3] groups the existing media steganography approaches in three different basic categories: steganography by cover selection, steganography by cover synthesis and steganography by cover modification. While all three approaches have their pros and cons, steganography by cover modification is currently by far the most used approach. It can be assumed that more than 95% of all steganographic tools for media objects that are currently freely available use this approach. To exemplify a projection of a media steganography approach onto the taxonomy, therefore a popular steganography by cover modification approach is used in the following descriptions: LSB (least-significant bit) steganography is one on the early (1990s) methods that is still popular in the information hiding community today [21]. In this method, the value of the LSB of a media object sample representation (an audio sample value in time domain, a pixel in an image, etc) is directly modified in media objects (e.g., PCM WAV encoded audio files or raster graphics image formats such as PNG, BMP, PGM, etc.). This can be done naively (modifying every value, which results in statistically easy to detect embedding) or in more sophisticated ways, using only a fraction of all available positions with an intelligent (i.e., key-dependent and content-aware) strategy (see, e.g., [3] for details). Independent of the strategy used for implementation, for the human auditory system (HAS) or the human visual system (HVS) the result is in most cases not identifiable as a stego object because the magnitude of the modifications performed lies way below the excitation threshold of HAS or HVS. Such a LSB embedding method would fit into the proposed taxonomy very well as a digital media value modulation. In coherence with the other sub-taxonomies and starting from this small example (and others briefly analysed by the authors), the following initial Digital Media Steganography Sub-Taxonomy with domain-specific patterns is proposed here:

- **E1d1. Digital Media State/Value Modulation**: The covert message is embedded by modulating the state or value of a digital media element, such as a pixel, video frame or sound sample.
  Examples: *i*) LSB embedding; *ii*) Echo hiding for audio

signals, utilizing (artificially created) echo components, where the data is usually hidden by varying characteristics of the echo such as the amplitude, decay cost and offset or delay; *iii*) Spread spectrum techniques; *iv*) Manipulating the phase coding for consecutive frames in an audio signal.
An early, but still good, reference providing details on most of the example techniques discussed below is [21]. From the available examples, the following set of initial sub-patterns can be defined for media steganography:

- **E1.1d1. Digital Media Reserved/Unused State/Value Modulation**: the covert message is embedded by modulating reserved/unused states/values of digital media elements.
  Examples: *i*) embedding message parts into media blocks that are not used/rendered in the playback/rendering process, such as non-referenced content chunks in PNG or PDF files.
- **E1.2d1. Digital Media Random State/Value Modulation**: a (pseudo-)random value or state of/in digital media data is replaced with a secret message (that is also following a pseudo-random appearance).
  Examples: *i*) in spread spectrum audio steganography the message is usually encoded into a pseudo-random bit-string which is in the embedding spread as much as applicable over the frequency spectrum, forming a low-amplitude 'background noise'.
- **E1.3d1. Digital Media LSB State/Value Modulation**: The LSB of digital media elements are modulated.
  Examples: *i*) low-bit encoding using the least significant bit plane of a media representation is a technique used in many different media formats, including raster images and audio files.
- **E1.4d1. Digital Media Character State/Value Modulation**: The features of characters in digital media elements are modulated.
  Examples: This pattern was rarely found to be represented in the literature. However, one notable example is the following: *i*) manipulation of the duration of notes in a MIDI audio file in [111].
- **E1.5d1. Digital Media Redundancy State/Value Modulation**: The redundancy of a digital media element's content is modulated (this is usually applied by a succeeding pattern that fills the gained space with covert data), e.g., by means of compression.
  Examples: *i*) Quantization index modulation in media compression operations, e.g., in MP3 compression (e.g., see [112]); *ii*) Exploiting the correlation/redundancy between both audio channels in stereo signals.

- **E2d1. Digital Media Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number of digital media elements appearing in a flow (succeeding messages *occur* shortly or long after previous ones).
  Examples: *i*) influence the rate of acoustic beeps appearing in an audio file; *ii*) influence the particular location of a pixel with a specific color in a digital image.
  This pattern has the following sub-patterns, which are special forms of element occurrences:
  - **E2.1d1. Digital Media Element Enumeration**: An attribute describing the quantity of digital media sub-elements is modulated. This pattern also applies if sub-elements are added to a digital media element to increase the overall length of the element (e.g., by adding more pixels to a digital image file).
    Examples: *i*) influence the number of succeeding beeps in an audio file; *ii*) influence the number of succeeding pixels with the same value or the overall number of pixels in a file.
  - **E2.2d1. Digital Media Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of a digital media element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5).
    Examples: This pattern is rarely represented through existing methods. However, one example could be *i*) the embedding of a blue screen frame at a specific location in a video file.

D. CPS Steganography Sub-Taxonomy

Analog to the previous domains, the following patterns have been identified for CPS steganography.

- **E1c1. CPS State/Value Modulation**: The covert message is embedded by modulating the state or value of a CPS element, such as an actuator state or values in unused elements.
  Examples: *i*) modulation of actuator states [33]; *ii*) modulation of a setpoint in a control command; *iii*) modulation of ICS configuration data (and its backups) for long-term storage of secret data [113]; *iv*) modulation of control action states in the controller logic [114]; *v*) modulation of control signal and state measurements [115].
  - **E1.1c1. CPS Reserved/Unused State/Value Modulation**: The covert message is embedded by modulating reserved/unused states/values of CPS elements.
    Examples: *i*) modulation of values in unused registers of embedded CPS equipment [33]; *ii*) modulation of values in unused bits of control commands
  - **E1.2c1. CPS Random State/Value Modulation**: A (pseudo-)random value or state of/in CPS or its data is replaced with a secret message (that is also following a pseudo-random appearance).
    Examples: *i*) overwrite a randomized order of colors in a (smart home) light show; *ii*) exploit MAC address randomization of an Android smartphone (here considered as a cyber-physical device due to its sensors and actuators) to overwrite it with a secret message; *iii)* an imaginable idea is to replace the randomized bits of the *ShadowAuth* CAN authentication message with encrypted covert data (cf. [116]); *iv*) one could potentially apply the idea of replacement of entries in a conditioner buffer from physical processes used in an

RNG similarly as described by a work of Evtyushkin et al. [117], where the buffer is first replaced by rseeds, followed by an entry that is replaced by the sender or not; *v*) modulation of high-resolution sensor values affected by random sensor- and/or process noise (c.f. [118], [119]).

  - **E1.3c1. CPS LSB State/Value Modulation**: The LSB of CPS elements are modulated.
    Examples: *i*) modulate the LSB for a setpoint of an actuator (e.g., a motor's RPM); *ii*) alter the history of stored LSBs of temperature sensor values to encode one bit per logged temperature value.
  - **E1.4c1. CPS Character State/Value Modulation**: The features of characters in CPS elements are modulated.
    This pattern is essentially an application of text steganography methods to the CPS context. It is kept for completeness. Examples: *i*) capitalizing every n^{th} character of a variable's, project's, or device's name; *ii*) changing font color or size of a comment in a code block in the TIA Portal[3] settings.
  - **E1.5c1. CPS Redundancy State/Value Modulation**: The redundancy of a CPS element's content is modulated (this is usually applied by a succeeding pattern that fills the gained space with covert data), e.g., by means of compression.
    Examples: *i*) renaming TIA projects, e.g., from "RTP020" to "soldering tip version RTP020"; *ii*) modulate the behavior of an ICS so that it either moves to its default position before proceeding with the next work piece or directly proceeds with that piece.

- **E2c1. CPS Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number of CPS elements appearing in a sequence (succeeding messages *occur* shortly or long after previous ones).
  Examples: *i*) influence the point in time at which a certain cyber-physical action is performed (e.g., opening/closing a window); *ii*) writing deduplication records into the ICSs' filesystem to exfiltrate secret data [120].
  This pattern has the following sub-patterns, which are special forms of element occurrences:
  - **E2.1c1. CPS Element Enumeration**: An attribute describing the quantity of CPS sub-elements is modulated. This pattern also applies if sub-elements are added to a CPS element to increase its overall size (e.g., by adding more configuration parameters to a CPS configuration).
    Examples: *i*) modulating the number of windows that are opened succeedingly; *ii*) modulating the number of setpoints accumulated within a single control command (e.g., using BACnet's *WriteMultiple* command).
  - **E2.2c1. CPS Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of a CPS element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5).
    Examples: *i*) influencing the location of a drone (e.g., so that its current location or a whole path represents a secret message); *ii*) modulate the position in time of a specific sensor reading operation (alternatively, succeeding read operations could be timed, e.g., to influence the time between commands which then can be observed by the CR); *iii*) raising the sampling frequency of sensor multiplexing in Android smartphones at a specific temporal position to transmit a secret message [121]; *iv*) switching between on and off state of a vibration motor in a smartphone in a specific temporal order [122]; *v*) modulation of actuator response time (required time to change its state) [123].

[3]TIA Portal is a configuration and management software for industrial control systems.

E. Filesystem Steganography Sub-Taxonomy

Compared to other fields, less work is available for filesystem steganography. Despite this, we identified the following patterns that can be used to describe the known hiding methods. Before discussing each mechanism in detail, we point out that the joint use of filesystem and steganography has been done for two different purposes. In the first case, many ideas are devoted to create a "steganographic filesystem", which is a way to store/organize data in a stealthy manner. To this aim, data is hidden in files by using carrier-specific techniques, such as LSB for images (see, e.g., [29]). From the perspective of creating a coherent organization, such a class of techniques will not be included in the filesystem steganography sub-taxonomy below. Instead, the reader might refer to text steganography for textual file content, digital media steganography for media content, and network steganography for network traces (pcap files, etc.). A second case concerns methods using filesystem-specific features for concealing data. Such methods are discussed below.

- **E1f1. Filesystem State/Value Modulation**: The covert message is embedded by modulating the state or value of a filesystem element, such as a file/inode attribute.
  Examples: *i*) similar to text steganography, replace a character of a textual filename with its homoglyph [124]; *ii*) embed secret data in unused/reserved or random-value meta-data fields [31]; *iii*) modulate timestamp values [125]; *iv*) case modulation of filename characters.
  - **E1.1f1. Filesystem Reserved/Unused State/Value Modulation**: The covert message is embedded by modulating reserved/unused states/values of filesystem elements.
    Examples: *i*) placing secret data in bytes of unused blocks [38], [126], [127]; *ii*) hide secret information in the metadata of deleted (now unused) files of the exFAT filesystem [128].
  - **E1.2f1. Filesystem Random State/Value Modulation**: A (pseudo-)random value or state of/in a filesystem or its sub-elements is replaced with a secret message (that is also following a pseudo-random appearance).
    Examples: *i*) in their seminal paper that introduced

filesystem steganography, [129] superimposed hidden files onto files with random (or randomly looking, e.g., encrypted) content; *ii*) similarly to the examples of **E1.1f1**, already allocated blocks that contain a high entropy (pseudo-random content) can be overwritten for steganographic storage [130] (instead of unused blocks).

- **E1.3f1. Filesystem LSB State/Value Modulation**: The LSB of filesystem elements are modulated.
  Examples: *i*) modulation of the nanoseconds bits in the NTFS access/creation time attribute of a file [125]; *ii*) modulation of the least significant bits of the exFAT fields *CreatE1msIncrement* and *LastModified10msIncrement* [128].
- **E1.4f1. Filesystem Character State/Value Modulation**: The features of characters in filesystem elements are modulated.
  Examples: *i*) given the support of a particular filesystem for upper- and lower-case characters, filename characters' cases can be modulated. This method can be applicable in all cases where both the covert sender and receiver have visibility over files. For instance, this approach has been used to transmit data remotely in Dropbox, e.g., modulation of names propagates through a shared folder to create a network covert channel [131]; *ii*) replacement of characters in filenames with homoglyphs.

- **E2f1. Filesystem Element Occurrence**: The covert message is encoded in the spatial or temporal location of elements, which can also, e.g., influence the rate or overall number of filesystem elements appearing in a sequence (succeeding messages *occur* shortly or long after previous ones).
  Examples: *i*) let a certain filename appear (or not) to encode a hidden message; *ii*) modulate the number of files or inodes in a filesystem; *iii*) position filesystem elements in a temporal or spatial order.
  This pattern has the following sub-patterns, which are special forms of element occurrences:
  - **E2.1f1. Filesystem Element Enumeration**: An attribute describing the quantity of filesystem sub-elements is modulated. This pattern also applies if sub-elements are added to a filesystem element to increase its overall size (e.g., by adding more files to a directory to increase the overall directory's length).
    Examples: *i*) modulate the number of inodes/files in a filesystem or the number of bytes within a file (influences file size/number of allocated blocks) to encode a secret message [132].
  - **E2.2f1. Filesystem Element Positioning**: The covert message is embedded by inserting or changing the temporal/spatial position of a filesystem element (this temporal/spatial position might be described through a virtual element, cf. Fig. 5).
    Examples: *i*) modulate the order of multiple file `create()` operations to influence their order of appearance in file managers (e.g., permutation steganography for FAT [133]); *ii*) write to even/odd FAT cluster[4] locations, depending on the bit value (and derivations of this approach) [134].

VII. Countermeasures

A plethora of measures exist to detect, limit and prevent steganography. For this reason, it is necessary to analyze which of these countermeasures can be used for protection against hiding patterns.

A. Detection of Steganography

a) Network Steganography: In network steganography, the observed values of protocol fields as modified by pattern **E1** are usually compared to the usual values of these fields, e.g., by comparing their entropy or compressibility scores [53], [54]. Another approach aims at observing the differences between legitimate and covert channel traffic's timing behavior (as it is unintentionally influenced by the embedding method), see, e.g., [68], in which the authors aim at detecting a covert channel-caused value modulation in a hash-based authentication by observing the timing of packets.

Until about 2010, the detection of pattern **E2** and its sub-patterns was mostly addressed by classical statistical approaches. Cabuk et al. utilized a metric based on regularity, compressibility and the so-called ϵ-similarity [84], [85], [135]. The regularity measure detects inter-packet delay based covert channels by analyzing whether a flow contains relatively constant delays [2], [85]:

$$\text{Regularity} = \text{STDEV}\left(\frac{|\sigma_i - \sigma_j|}{\sigma_i|}, \forall i, j < i\right).$$

In above formula, the normalized pairwise differences for the standard deviations σ_i of the inter-packet delays of a given set of packets, called *window*, is computed. A window has usually the size of 2,000 packets.

The compressibility score determines whether delays between packets of a flow with a given window size are composed of similar values. Therefore, the delays are rounded and brought into a string representation. The length of the string S and the compressed length of the same string $C = \Im(S)$ are then compared by calculating the compressibility score:

$$\kappa = \frac{|S|}{|C|}.$$

A high compressibility score indicates the presence of a covert channel (as it contains the same delays rather often) [85].

The third approach shown by Cabuk et al. is the *ϵ-similarity*, which quantifies the "similarity" of the inter-packet delays t_i in a flow that usually has a window of 2,000 packets as follows [85], [135]: for the packets in the window, the inter-packet delays are calculated. All inter-packet delays are afterwards sorted and their relative differences are calculated:

$$\lambda_i = \frac{|t_{i+1} - t_i|}{t_i}.$$

[4] In FAT, a cluster refers to a group of sectors.

Finally, the fraction of λ values below a threshold ϵ is calculated.

Similarly, Berk et al. have shown that channels of this pattern can be detected using a simple heuristic, in which the inter-packet delays of a flow are analyzed. Therefore, the number of packets $C(\mu)$ that are close to the mean of inter-packet delays are divided by the maximum number of packets with the same delay C_{max} to gain a probability P_{CC} indicating the presence of a covert channel [136]:

$$P_{CC} = \frac{C(\mu)}{C_{max}}.$$

Other traditional approaches for network covert channel detection use the Kolmogorov-Smnirnov test, the Kullback-Leibler distance, entropy, and entropy rate [2].

In addition to these statistical approaches, machine learning-based classifiers have been applied to detect channels belonging to pattern **E2**. The most prominent methods being decision trees [2], [137], [138], k-nearest neighbors (kNN) [139], support vector machines (SVM) [2], [139], [140] and several forms of neural networks [2], [141].

b) Text Steganography: In general, techniques manipulating the semantic of a text (or altering elements trying to preserve the semantic information) or some character features as done by pattern **E1** impact on the overall readability and/or the text appearance. Unfortunately, such a property is tightly-influenced by the context (e.g., prose versus technical writings) as well as by the skill of the reader. As a consequence, the design of countermeasures is hard and mitigation can only address a small sub-set of similar cases [17]. The majority of text steganography countermeasures attempts to first extract some characteristics belonging to the text. Then, they analyze the differences between a normal amount of text and the steganographic counterpart for detecting the presence of the hidden information. Literature showcases the use of many traditional statistical models based on simple characteristics, such as frequency of some text element feature (e.g., words, font type, color), like synonym frequency [142] for detection of synonym substitution methods. Different ML-based detectors can be used also, such as the SVM classifier for feature coding, which analyzes the font attributes and classifies the characteristic vector extracted from the distance of font attributes between every two adjacent characters [143]. At the same time, language-specific features (including grammar rules or stylistic conventions) could be further considered, mainly to improve the accuracy of the detection [144]. Summarizing, counteracting steganographic and hiding attempts targeting text/written carriers must face a very heterogeneous attack space. In fact, when the cloaking method is designed to prevent detection from a human entity, linguistic approaches could be not the best option. Hence, manipulating colors and fonts or making imperceptible alteration to the formatting could lead to visual artifacts difficult to spot. Instead, when the detection is supposed to be performed in an automatic manner, recurring to more culturally-driven and sophisticated approaches exploiting language features is usually the preferred choice. Therefore, engineering countermeasures always requires a good balance between the considered hiding patterns and their ability to generalize the detection process with a reasonable complexity.

Detection of some techniques for pattern **E2**, such as line and word shift coding or syntactic methods, can be performed visually by the human viewer in some cases [17]. Most often, high-level indicators, such as the occurrence of a word or the average length of the various sentences can be used to spot hidden data [145] or to train statistical models. Another example of statistical detectors involves neighbor difference feature for word shift methods [146]. Many ML-based detectors for generative linguistic steganography methods have been developed in the recent years, such as SVM classifier in [147], Softmax classifier in [148], TS-RNN [149], or CNN [150].

c) Digital Media Steganography: There is very few comprehensive literature existing which tries to map out the usage of steganalysis methods for the patterns **E1** and **E2** in digital media steganography. The most cited works in this field (e.g., [3] or [151]) point out that a vast number of different ML-based detection approaches are used here, ranging from early methods relying on tree-based classifiers or regression analysis to the always popular SVM. Since 2016 also digital media steganalysis experiences a rapid growth of neural network-based detection approaches, learning feature spaces suitable for the detection of a steganographic embedding rather than relying on traditional hand-crafted feature spaces.

d) CPS Steganography: When CPS states are manipulated by pattern **E1** to store secret data, classical approaches of anomaly detection can be performed to, e.g., determine the plausibility of the actuator values or the resulting sensor values (e.g., a temperature value might increase if the heating of a heater actuator is increased). Such anomaly detection can be achieved through entropy- and ML-based methods [152], [153]. This might be further enhanced using digital twins as references for anomaly detection. For pattern **E2**, a similar approach can be conducted: when the appearance of some element (such as a new node in a CPS environment) or the order of performed actions (elements) is manipulated by a steganographer, statistical tests, entropy tests and ML methods can be applied, too. See Luo et al. [154] for a recent overview on anomaly detection in CPS. Finally, for state/value modulation approaches by [114] and [115] (both belonging to **E1**) detection methods are currently not studied.

e) Filesystem Steganography: In contrast to fields like network steganography, where each element can be checked only once before it is gone, filesystems are long-living objects with continuous access by the user, both to the cover filesystem and to the hidden filesystem.

Hence, detection of pattern **E1** can be approached via long-term monitoring of access patterns, or analyzing access patterns from the past, as long as logs are available. These monitoring efforts are called traffic analysis and update analysis, and have been successfully applied [155] against StegFS [130], [156], despite intermediate attempts [157] of proposing anti-countermeasures against update analysis and traffic analysis of StegFS. However, [158] modify the log structure of a file system to hide data, which also might provide an anti-countermeasure against traffic analysis on log data. Moreover, when timestamps are used, a possible idea could be correlating

the installation date of the OS and various programs against files to spot inconsistencies [125]. Additionally, timestamps are not only stored in files, but also in various logs or in messages produced by drivers and kernels. If such a data is produced with a sufficient granularity/precision, it can be used as an effective starting point for computing metrics for spotting the presence of hidden information.

As described for pattern **E1**, detection of pattern **E2** largely relies on traffic or update analysis. Thus, the methods described there apply to pattern **E2** as well.

B. Limitation and Prevention of Steganography

a) Network Steganography: Traffic normalization can act in both, a blind and non-blind manner [2], [159]. Essentially, traffic normalization removes ambiguities in network traffic, including potential covert channels. Value modulation as performed by **E1** in network packets are usually limited or eliminated by a normalizer by either allowing only one specific value to be set (elimination) or by reducing the number of allowed values of a header field (limitation). However, there are also multi-level security approaches, like filter technology based on the Bell-LaPadula (BLP) model where a firewall system prevents the communication between nodes if it would violate the BLP's mandatory access control rules. An example for such a scenario was shown for network steganography in CPS environments running the BACnet protocol [62]. Further elimination protocols, such as the Blind Write-up [160] eliminate covert channels by completely allowing any policy-breaking communication.

Covert channels of pattern **E2** can be limited by artificially delaying network packets through a (usually blind) traffic normalizer [2], [159]. However, such delays rely on buffer space and computing capacity of a normalizer, enforcing limits on its performance. A similar case can be observed for the network pump [161], [162]. Additional limitation protocols, such as the store-and-forward-protocol (SAFP) limit policy-breaking communications [160]. Finally, the capacity of potential **E2**-based channels between virtual machines was already limited in the early 1990's: Hu applied fuzzing timing to slightly randomize the time at which virtual machines observe certain events of a host system [163]. However, a few non-blind limitation approaches also utilize traffic normalization: Wendzel and Keller have shown that protocol switching covert channels can be delayed on routers that monitor protocol switches [164].

b) Text Steganography: For limitation and prevention, different modifications on the text can be performed, which involve alteration of the text layout, such as formatting and encoding, known as structural attack; or modifications that involve semantic paraphrasing, which belongs to *Manipulation by Reader* (MBR) attacks [18].

The MBR attacks [18] as a countermeasure involve also syntactic modification of the given text, modification of the number and type of used Unicode space characters, etc. Limitation and prevention of generative linguistic steganography methods are insufficiently studied.

c) Digital Media Steganography: Besides classical steganalysis methods, there exists the concept of an active warden (see [165]), actively modifying potential or suspected cover objects. Most media steganography patterns (in **E1** as well as **E2**) lack robustness against format conversions, noise additions and other active content modifications. Therefore, such active warden methods would provide technical countermeasures at the cost of degrading the media content and potentially harming other information hiding protocols, such as digital watermarks. For media formats such as PDF which allow for using the position of chunks to hide messages (**E2.2d1**) or allow misusing unused/not-rendered contend blocks, these files or streams could be sanitized to eliminate potential hiding locations.

d) CPS Steganography: A limitation of CPS steganography using patterns **E1** and **E2** can be achieved through several means, such as limiting the number of sensor value reads or actuator state manipulations during a timeframe. Such means can be applied similarly like limitation approaches in network scenarios (e.g., ICMP rate limiting [166]). However, CPS environments do not *always* allow such limitations: especially real-time environments require that deadlines are kept and limitations can render this impossible under targeted attacks because the attack consumes a fraction of the permitted requests of a timeframe. As CPS can also incorporate large parts of network steganography due their inherent (process-based) communication, traffic normalization e.g., using protocol converter, can be used as well to mitigate protocol-specific covert channels.

Additionally, above-mentioned BLP-based techniques [62] can be used to isolate certain CPS components from each other, thus preventing steganography. In the case of process data modulation, one potential mitigation is to decrease accuracy of measurements or sensor values by rounding or cutting unnecessary decimals. The core idea here is to decrease entropy of process data to reduce cover surface available to the steganographer and make detection more likely. Up to now, the analysis of steganography limitation and prevention in CPS is still in its infancy. It can be assumed (but must still be validated) that several general CPS defense strategies, such as those mentioned in [167], [168], can also prevent certain applications of patterns **E1** and **E2**. This is reasoned by the fact that steganography-application overlaps with classical *deception attacks*, where false data is injected into a CPS [168]. For the covert channel method proposed in [114] (belonging to **E1**), the authors mention the need to further investigate prevention methods. Another channel referenced in the CPS sub-taxonomy (Evtyushkin et al., also pattern **E1**) can be mitigated through different software- and hardware-based methods discussed by the authors in their original paper [117]. However, their work describes a channel that operates between CPU cores on the basis of PRNG manipulation and is not directly transferable to a typical CPS scenario.

e) Filesystem Steganography: In many scenarios, the owner and user of the cover filesystem and the machine it runs on are identical to the creator, owner and user of the hidden filesystem. In such a scenario, no authority can prevent the owner from installing a hidden filesystem. With regards to

limitation, countermeasures for detection might be applicable in an indirect way. If the countermeasure for detection such as traffic analysis with a particular window size [155] is known to the owner of the hidden filesystem, then frequency of accesses to the hidden filesystem might be reduced in order to stay under the radar and to escape detection. But this also means that the countermeasure has the potential to limit the frequency of accesses to the hidden filesystem, although it does not necessarily limit the capacity of the hidden filesystem.

Also, an approach used by [169] to limit database covert channels possibly could be applied to filesystems as well: a spurious process (SP) does either re-perform some previous action, or not. For instance, if CS creates a file then with some probability SP deletes this file and creates it again. If CR then checks for file existence, it cannot be sure whether the file was created by CS or the SP. Anti-countermeasures like redundancy to overcome the above countermeasure in turn increase the number of accesses to retrieve a hidden disk block, and thus lead to further limitation of the number of hidden disk blocks accessible while staying unnoticed.

Finally, approaches to filesystem steganography that rely on modification of timestamps might be limited with a method used in [163] for the VAX security kernel: timing behavior between virtual machines is becoming imprecise (through what Hu calls "fuzzy" timing). This limits covert timing channel capacity.

As described for pattern **E1**, prevention seems not achievable in filesystem steganography for pattern **E2**. Limitation approaches envisioned for pattern **E1** will largely also apply for pattern **E2**.

VIII. Discussion

This section covers the extendability of the presented taxonomy in Sect. VIII-A, followed by a discussion of its limitations in Sect. VIII-B.

A. Extendability of the Taxonomy

As our goal is the provision of an adaptive taxonomy that can emerge in the future, a crucial aspect of the taxonomy is its extendability.

As discussed in Sect. IV, we integrated several aspects that allow the extendability of our approach: (i) the PLML-based patterns description provides a clear format and structure that can be used for patterns discovered in the future; (ii) future patterns can be integrated into the taxonomy (generic patterns can be put into the generic taxonomy while domain-specific patterns can be derived and/or put into the domain-specific sub-taxonomies); (iii) if patterns cannot be integrated directly into existing branches of the taxonomy, there is no reason why new branches cannot be added to the generic taxonomy or to the sub-taxonomies; (iv) the naming conventions provide a clear guide on how future pattern numbers and names shall be composed. If absolutely necessary, new terms for actions and objects could be added.

1) Example - Steganography in Machine Learning Systems: The research field of machine learning has been and is still growing rapidly. This also offers room for new methods in steganography. Existing publications on this topic have shown interesting new approaches. These new methods often focus strongly on machine learning technicalities and require extensive involvement with the domain, as they are often deeply interwoven with the design and implementation of the training process of the machine learning models. We believe that this sub-topic needs a more extensive and deeper evaluation to be covered fully, than it would be possible in this publication. However, we want to use the opportunity and exemplify the integration of a newer steganographic domain into our patterns approach. Therefore, we added the preliminary domain *m* for machine learning-based steganography as well as a preliminary pattern for the domain (a new sub-taxonomy for this emerging domain requires excessive future work):

E2m1. Training Set Element Modulation: The training set, which is used to create a model, is manipulated by the covert sender to influence the training's outcome in a specific way.

Examples: *i*) A training set consists of examples from classification task A. The training set is however specifically chosen by the sender, in such a way that the resulting model will not only solve the classification task A, but also a hidden classification task B which has no obvious connection to task A [170]; *ii*) In a federated learning system, multiple clients use their local data to collaborate on a shared ML model without disclosing local data. In such a system, an attacker can choose specific examples to train and "poison" their local model. This will slightly change the behavior of the global model, which can then be detected and decoded by the receiver [171].

Other publications like [172] and [173] have focused on watermarking models to prove ownership. As we do not include watermarking techniques in this publication, further analysis and integration of these methods is left for future work.

B. Limitations

While our proposed taxonomy provides several advancements and addresses the limitations of previous taxonomies as discussed in Sect. II, it is also linked to some limitations:

No consideration of digital watermarking: Like steganography, digital watermarking is part of the information hiding discipline. As both steganography and digital watermarking share a certain methodology, it might be feasible to provide a taxonomy on an even broader basis that also allows for the inclusion of digital watermarking methods. However, our aim was to integrate steganography domains into one taxonomy, and the potential integration of digital watermarking is left for future work.

No proof of completeness: We conducted an excessive literature survey to discover hiding methods used in all domains of steganography. However, we cannot be certain that hiding methods published, e.g., in small national conferences not indexed in larger databases present ideas unknown to us, which would represent novel patterns. Similarly, we can expect

that future research proposes new patterns. As discussed in Sect. VIII-A, such upcoming patterns should be integrable into the taxonomy though.

Generalization: Clearly, there is no one-fits-all solution. Our taxonomy serves as an umbrella for a wide range of hiding methods. In this context, the patterns themselves do not provide a detail view. To remedy this aspect, we propose a unified description method for hiding methods described in scientific papers that is introduced in Sect. IX. The unified description method enforces the inclusion of additional details for hiding methods, such as the properties of cover objects and of the covert channel that is created by the hiding method.

No Consideration of reversibility and indirect hiding methods: As previous taxonomies, we did not cover the aspect of reversibility for hiding patterns. While some works have already analyzed reversibility for digital media steganography [174]–[176] as well as for network steganography [177], [178], we leave such a large analysis for future work as it requires excessive experimental evaluations which are not within the scope of a survey paper. Similarly, we do not present patterns or discussions on indirect hiding methods as a recent paper already provides a survey on these [16]. However, as our tutorial example in Sect. IX-C will show, the description of indirect methods is feasible with our methodology.

Lack of precise discussion of three-dimensional data: Recently, three-dimensional data is becoming important to enhance the user experience and interactivity of a vast arrays of immersive communication tools, entertainment and educational software, computer aided design platforms, and cultural heritage applications. Moreover, the expected evolution of many social media towards metaverse-based incarnations will increase the importance of three-dimensional information, especially in the medium term. Unfortunately, extending our taxonomy to consider also this type of data exhibits two major challenges. The first concerns with the major overlap with digital media. In fact, three-dimensional media often exploits image-based textures or is utilized to add immersive or realistic contents to classic multimedia products (e.g., for spatial audio). As a consequence, the majority of the approaches developed for hiding information are plain port of solutions already discussed in this work [179], [180]. The second challenge concerns the lack of a precise literature on steganography and three-dimensional artifacts. Indeed, the related corpus of work extends without clear boundaries and it is highly biased towards the concept of watermarking of meshes, i.e., a collection of points defining polygons, points, and other geometric entities [180], [181]. Within such a perimeter, the predominant concentration of works and techniques consider watermarking to enforce copyright, annotate shapes with semantic data, or prevent alterations, for instance malicious manipulations of clouds of points used in industrial or e-health applications [181], [182].

Therefore part of the ongoing research will be devoted to "align" the proposed taxonomy to the case of three-dimensional worlds. Owing to the design choices used to design the taxonomy (see, Sect. IV-A), this integration should be possible without the need of major alterations. Indeed, with the ubiquitous diffusion of IoT technologies and the Industry 4.0 revolution, the penetration of three-dimensional data in hardware/software ecosystems is expected to steadily grow. For instance, three-dimensional information is at the basis of new printing processes (e.g., 3D printing) or to capture accurate snapshots of natural environments, cultural heritage contents, and cyber-physical deployments. Accordingly, steganography is expected to penetrate as well, for instance to solve old and new data integrity problems [183]. Luckily, the used data structures are largely stored in standard files composed of array of characters, which can be the target of hiding mechanisms already discussed in this work (e.g., text methods or algorithm encoding data by altering values or attributes). This can witness the ability of our taxonomy of generalizing the hiding process by decoupling the pattern from the specific media.

IX. Tutorial: Using the Steganography Taxonomy in Academic Articles

In this tutorial section, we explain how the taxonomy can serve as a key tool regarding the unified description of hiding methods in scientific papers in Sect. IX-A, followed by example applications in Sect. IX-B and IX-C.

A. Using a Unified Description Method

A unified description aids the comparability of scientific experiments, results and methodology. Furthermore, the reproducibility of experimental results became increasingly problematic in recent years [184] and our unified description of hiding methods eases replication studies as the mandatory properties of the hiding methods are described in a pre-defined manner.

Our taxonomy can be applied by authors who describe hiding methods in papers. The described hiding methods must not necessarily represent new patterns, but they *can* present new patterns. We follow the example of Wendzel, Mazurczyk and Zander in [43] who proposed a unified description of *network* steganography methods to render them comparable and ensure that the hiding methods are described in a comparable manner.

Their original description foresees the attributes shown in Fig. 7. Descriptions must contain four major branches, which are all mandatory: an introduction, hiding method general information, hiding method process, and potential or tested countermeasures.

The first branch covers an essential introduction.

The second branch provides general attributes of the steganographic method, i.e., the author must state which hiding pattern is used, for which application scenario(s) the method is designed (e.g., for data exfiltration, command and control channels or in a general-purpose manner, see [43] for further examples) and which requirements the utilized carrier needs for the steganographic method (e.g., availability of certain network protocol features or characteristics, such as presence of legitimate packet losses).

The third branch contains a description of the hiding method's process, which comprises a description of the sender-side steps to be undertaken to embed and transfer the secret information as well as the respective receiver-side, a description

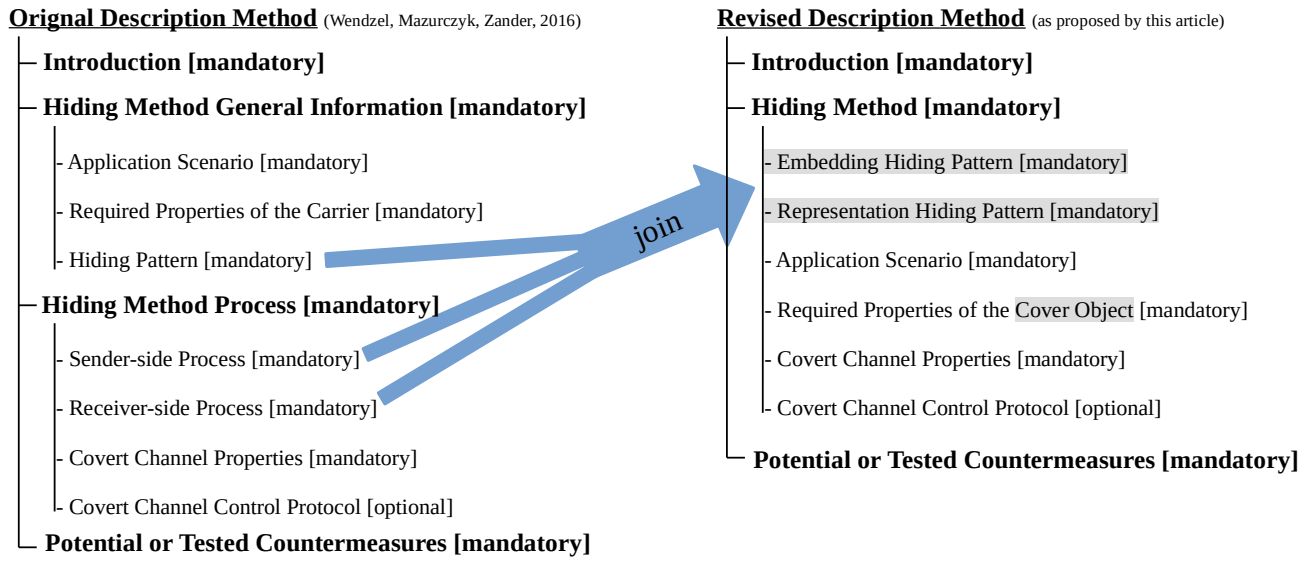

Fig. 7. Comparison of existing (left) and proposed (right) description methodology for hiding methods. Modifications are highlighted in grey.

of the covert channel's properties (channel capacity, robustness etc.) and the optional description of a steganographic (covert channel-internal) control protocol, which can enhance the functionality of a covert channel [26].

The fourth branch must at least mention the potential but optimally describes already evaluated countermeasures for detecting, limiting, and/or preventing the hiding method. This branch can optionally detail information about the experimental setup used to evaluate countermeasures. For instance, test data or test traffic might have been generated with particular tools, such as *CCgen* [185], *BroCCaDe* [186]–[188], *WoDiCoF* [189] or *CCHEF* [190], or under specific conditions of an operating environment.

To maximize the backward compatibility with works that utilize the original method, we propose to apply a derived structure, but with adjustments tailored to fit the new taxonomy. Our proposed description structure is shown on the right side of Fig. 7 and renders the description structure more compact and thus easier to apply in page-restricted short papers and posters.

Modification of the Previous Description Method: As mentioned, we decided to keep the characteristics of the original description method — also for backward compatibility. However, we simplified the methodology and added generalizations as follows (cf. Fig. 7).

First of all, the original description method already tried to address the problem of a decoupled sending and receiving process in combination with hiding patterns but failed to join these three aspects. For this reason, the original description method had the *hiding pattern* in the first branch and the *sender-side process* and *receiver-side process* described in the second branch. Due to the clear distinction in *embedding* and *representation* patterns as proposed in the article at hand, we can merge these three aspects into two (as indicated by the arrows in Fig. 7). This merge also eliminates the need for two separate categories (*hiding method general information* and *hiding method process*). For this reason, we ended up with only three branches. This modification also eliminates a redundancy of the previous work where the pattern's functioning overlaps with the sender- and receiver-side processes. Another tiny adjustment is that we renamed *carrier* into *cover object* to better reflect all domains of steganography. We kept the word *covert channel* as all steganographic hiding methods establish a covert channel in the end.

The following two examples illustrate the application of the proposed description method.

B. Example for Text Steganography

Our first example comes from the domain of text steganography and is kept simple.

Introduction: Alice and Bob exchange secret messages using the modification of letters so that they are kept slightly italic to represent a hidden '0' (otherwise hidden '1'). They perform the modifications using ink on a paper, i.e., a non-digital form is applied.

Embedding Hiding Pattern: To embed a secret message, Alice produces a cover text[5]. While she writes every letter element separately to the document, she modulates a letter's characteristic of being italic. Therefore, the embedding pattern is **E1.4t1. Text Character State/Value Modulation**.

Representation Hiding Pattern: Since Bob needs to interpret the (non-)italic character for every letter as well, the representation pattern matches the embedding pattern: **R1.4t1. Text Character State/Value Modulation**.

Application Scenario: Alice and Bob aim to exchange secret letters in a prisoner's problem scenario [191]: Alice and Bob reside in isolated prison cells and can only exchange messages through the warden Walter. Both need to find a way that does not raise the suspicion of Walter when a secret message is embedded into an innocent looking one. When Walter investigates the piece of paper exchanged between Alice and

[5]The creation of elements is neglected in the pattern specification due to the *Rule of Mandatory Occurrence of Elements*, see Sect. V-E.

Bob, he is only able to recognize the cover story, and not the hidden adjustments that represent the secret symbols.

Required Properties of the Cover Object: The cover object (in this case a piece of paper) needs to be writable with ink and messages with italic letters must not be discarded by Walter.

Covert Channel Properties: Given that Alice can embed n modifiable letters within a page (punctuation, whitespaces etc. cannot be used) and each of these modifiable letters carries 1 bit of secret data, she can embed n bits per page.

In general, the method is rather robust unless messages are discarded or lost. However, the clearer the italics are emphasized, the more robust the method as the modifications are easier to recognize by Bob.

Covert Channel Control Protocol: No control protocol is applied. However, one could define a simple error-detecting protocol, in which the last letter of each line represents a parity bit. A more sophisticated control protocol could additionally use some of the letters to indicate the start/end of a message.

Potential or Tested Countermeasures: While clear italics increase the robustness of a message, they decrease the stealthiness as the italics become more perceptible by Walter. Walter could actively aim at spotting such italics to detect the method. Furthermore, if – for security reasons – messages are jumbled or re-typed by Walter before being delivered to Bob, the message will become corrupted.

C. Example for Network Steganography with an Indirect and Hybrid Hiding Method

In this second example, we describe a steganographic method that implements its hidden information in the *position of events* in conjunction with the *modulation of a value* in an indirect manner. We use this example to demonstrate that complex scenarios can be described by our method.

Introduction: Mileva et al. [54] implemented an MQTT 5.0-based covert channel in which third-party node's connections are disrupted, causing a *re*connection. Sender and receiver agree on an encoding in advance, where one ID represents a secret symbol. To embed a secret message, the CS lets *connect* packets appear by duplicating an existing node's ID by connecting another node to an MQTT broker. This leads to reconnection packets from the legitimate nodes. The reconnection is then observed by the CR. The secret message is encoded by a tuple {time of disconnect; ID of disconnected node}, i.e., the order of appearance of disconnects is a key for sorting the IDs of disconnected nodes. Each ID is linked to a secret symbol.

Embedding Hiding Pattern: As explained in Tab. II, the exploitation of reconnects for network steganography can be expressed through hybrid patterns. They require certain header bits to be set (e.g., in TCP this would be the RST flag) or commands to be sent (**E1n1. Network State/Value Modulation**) while they also require a precise timing (element positioning in the temporal sense, **E2.2n1 Network Element Positioning**) as the time of reconnection is used to encode a secret message together with a node's ID.

Please note that one might additionally list the related indirect covert channel pattern in this category, see Fig. 2 in [16] for an overview.

Representation Hiding Pattern: The CR observes the occurrence of reconnections (recognition of modulated values via **R1n1. Network State/Value Modulation**), which also means that elements representing new connections must be tracked in a temporal manner (**R2.21n. Network Element Positioning**).

Application Scenario: The method can be used for covert communication between one or more covert senders and receivers. In [55], the authors propose a data exfiltration scenario in an Operational Technology environment, where the covert sender has no access to the internet and transfers sensitive metadata to the receiver, who is connected to the internet and, thus, is able to exfiltrate the covert message to an external server. This scenario provides an uni-directional communication, but the channel can also be used in a bi-directional manner.

Required Properties of the Cover Object: This method requires a connection between nodes and the broker that can be reconnected and where the nodes have the option to reconnect themselves, e.g., in case they lose their connection due to an unstable network environment. To transmit messages, the CS needs to connect nodes with duplicated IDs, the CR has to read the traffic of the broker. The method requires sufficient noise in the number of legitimate reconnections to not be immediately suspicious.

Covert Channel Properties: The method operates in a MitM scenario. It cannot be used in an end-to-end scenario as other nodes will not receive the reconnection information of the reconnected node. The channel is an uni-directional and indirect one.

The bandwidth depends on the number of reconnection packets n that can be sent per second and a single node ID contains 16 bits. For this reason, the channel offers $16 \cdot n$ bps [54].

The robustness can be influenced by a higher number of caused reconnections which interfere with another as well as general traffic noise that might influence the reliability of the network. However, noisy legitimate reconnections can be filtered by the receiver by regarding the source address of the sent packet. Furthermore, the QoS feature of MQTT influences the reliability of messages reaching the CR. For a timing interval of 1 s and QoS=1, approximately 72.4% of the secret symbols reached the CR, while increasing the interval to 8 s resulted in 94.1% successfully transferred symbols.

Covert Channel Control Protocol: While it was not the case in the original work, one could use a fraction of the hidden bits, e.g., every second bit, to transfer error correction/detection or control information, e.g., to indicate the start or end of a transmission. See [26] for examples on control protocols in network steganography.

Potential or Tested Countermeasures: The more symbols are sent per unit of time, the easier the channel can be detected. For instance, to limit detectablility, one can use ternary encoding, for example, to limit the symbols sent.

Mileva et al. have shown that the channel can be distinguished from legitimate traffic with high detectability [54] using a modified version of the *compressibility score* of Cabuk et al. [85]. Detailed experimental results for the detectability are provided by the original paper.

Mileva et al. also proposed one countermeasure to limit the capacity of this channel: an adversary could send randomly chosen node IDs to disrupt the interpretation of a covert message.

X. Conclusion & Future Work

In this article, we presented a taxonomy for steganography hiding methods able to capture the nuances of an almost unbounded number of domains related to digital contents and hardware/software entities. To elaborate such a concept, we used a pattern-based approach and we took advantage of objects and actions. As a result, the proposed taxonomy should be considered an *object*-oriented organization for grouping and classifying steganographic actions that can target specific objects. To prove the flexibility of our approach, we applied the taxonomy to several domains of steganography, including network entities, textual contents, digital media, CPS and industrial settings, and the cloaking of data within filesystems.

In order to design and refine our idea, we revised previous attempts and we performed major efforts to unify good practices and elaborate suitable workarounds to major shortcomings. In more detail, state-of-the-art taxonomies dealing with steganography and data hiding exhibited various limitations in handling embedding patterns, i.e., they lacked the "representation pattern" concept. Another important advancement concerns the ability of dropping the unnecessary artificial distinction between temporal and non-temporal hiding methods as well as the need of using inconsistent pattern naming conventions. As a result, our taxonomy is very expressive and provides a more solid foundation for the rapidly-emerging corpus of research on the use of steganography both as an attack and defense technique. Lastly, our taxonomy and the related unified description template can be used to explain hybrid hiding methods. In this case, multiple patterns can be combined to capture sophisticated cloaking mechanisms.

Future works aim at enriching the taxonomy with a deeper mapping of methods belonging to the field of media steganography, especially to consider the required adjustments to handle media streams (including variable-bit-rate audio/video flows) and protocols devoted to guarantee the interaction of users and contents (e.g., in the context of games and AR/VR applications). Part of our ongoing research is devoted to analyze the feasibility of integrating methods for watermarking digital and network traffic into our general framework. At the same time, an important amount of effort will be used to support and advocate the adoption of the presented taxonomy to the scientific community. Hence, this article (including the tutorial part) should be considered both as a sort of "textbook" and manifesto to gather other researches and spawn an iterative process to enrich and refine our vision. To support such a process, we plan to maintain the taxonomy over the next decade with bi-annual meetings and propose updates via the ad-hoc website: https://patterns.ztt.hs-worms.de.

Acknowledgments

Parts of the work of Laura Hartmann have been funded by the European Union from the European Regional Development Fund (EFRE) and the State of Rhineland-Palatinate (MWWK), Germany. Funding content: P1-SZ2-7 F&E: Wissens- und Technologietransfer (WTT), Application number: 84003751, project MADISA.

Parts of the work from Brandenburg and Magdeburg authors in this paper (i.e., on definitions and general discussions) have been funded by the German Federal Ministry for Economic Affairs and Energy (BMWi, Stealth-Szenarien, Grant No. 1501589A and 1501589C) within the scope of the German Reactor-Safety-Research-Program.

Parts of the work of Luca Caviglione, Jörg Keller and Wojciech Mazurczyk have been supported by the SIMARGL Project - *Secure Intelligent Methods for Advanced RecoGnition of maLware and stegomalware*, with the support of the European Commission and the Horizon 2020 Program, under Grant Agreement No. 833042.

Parts of the work of Sebastian Zillien have been funded by the research training group SIVERT of the German federal state of Rhineland-Palatinate.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," Proceedings of the IEEE, vol. 87, no. 7, 1999, pp. 1062–1078.

[2] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski, Information Hiding in Communication Networks: Fundamentals, Mechanisms, and Applications, ser. IEEE Series on Information and Communication Networks Security. Wiley, 2016.

[3] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2009.

[4] J. F. DeFranko and D. Serpanos, "The 12 flavors of cyberphysical systems," Computer, vol. 54, no. 12, 2021, pp. 104–108.

[5] S. Wendzel, S. Zander, B. Fechner, and C. Herdin, "Pattern-based survey and categorization of network covert channel techniques," Computing Surveys (CSUR), vol. 47, no. 3, 2015.

[6] S. Fincher, "PLML: Pattern language markup language / perspectives on HCI patterns: Concepts and tools," 2004, CHI 2003 summary document, https://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html.

[7] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, and T. Neubert, "A revised taxonomy of steganography embedding patterns," in The 16th International Conference on Availability, Reliability and Security, ser. ARES 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3465481.3470069

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE transactions on dependable and secure computing, vol. 1, no. 1, 2004, pp. 11–33.

[9] B. Pfitzmann, "Information hiding terminology - results of an informal plenary meeting and additional proposals," in Proc. First International Workshop on Information Hiding, ser. LNCS, vol. 1174. Springer, 1996, pp. 347–350.

[10] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity—a proposal for terminology," in Designing privacy enhancing technologies. Springer, 2001, pp. 1–9.

[11] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," Communications of the ACM, vol. 57, no. 3, 2014, pp. 86–95.

[12] S. Zander, G. J. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," IEEE Communications Surveys and Tutorials, vol. 9, 2007, pp. 44–57.

[13] C. Zhiyong and Z. Yong, "Entropy based taxonomy of network convert channels," in Proc. 2nd Int. Conf. on Power Electronics and Intelligent Transportation System (PEITS), 2009, pp. 451–455.

[14] A. Mileva and B. Panajotov, "Covert channels in TCP/IP protocol stack-extended version," Open Computer Science, vol. 4, no. 2, 2014, pp. 45–66.

[15] J. Lubacz, W. Mazurczyk, and K. Szczypiorski, "Principles and overview of network steganography," IEEE Communications Magazine, vol. 52, no. 5, 2014, pp. 225–229.

[16] T. Schmidbauer and S. Wendzel, "SoK: A survey of indirect network-level covert channels," in Proc. 17th Asia Conference on Computer and Communications Security (ASIACCS 2022). ACM, 2022, pp. 546–560.

[17] K. Bennett, "Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text," CERIAS Tech Report, Purdue University, Tech. Rep. 2004-13, 2004.

[18] M. T. Ahvanooey, Q. Li, J. Hou, A. R. Rajput, and Y. Chen, "Modern text hiding, text steganalysis, and applications: A comparative analysis," Entropy, vol. 21, no. 4, 2019, p. 355.

[19] R. Bergmair, "A comprehensive bibliography of linguistic steganography," 2009, http://www.semantilog.org/biblingsteg/.

[20] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis." J. Inf. Hiding Multim. Signal Process., vol. 2, no. 2, 2011, pp. 142–172.

[21] N. F. Johnson and S. Katzenbeisser, "A survey of steganographic techniques," in Information hiding, 2000, pp. 43–78.

[22] S. Gianvecchio and H. Wang, "Detecting covert timing channels: An entropy-based approach," in Proceedings of 14th ACM Conference on Computer and Communication Security (CCS), November 2007.

[23] S. Z. Goher, B. Javed, and N. A. Saqib, "Covert channel detection: A survey based analysis," in High Capacity Optical Networks and Emerging/Enabling Technologies. IEEE, 2012, pp. 057–065.

[24] L. Caviglione, "Trends and challenges in network covert channels countermeasures," Applied Sciences, vol. 11, no. 4, 2021, p. 1641.

[25] A. Houmansadr, C. Brubaker, and V. Shmatikov, "The parrot is dead: Observing unobservable network communications," in 2013 IEEE Symposium on Security and Privacy. IEEE, 2013, pp. 65–79.

[26] S. Wendzel and J. Keller, "Hidden and under control," annals of telecommunications-annales des télécommunications, vol. 69, no. 7, 2014, pp. 417–430.

[27] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," IEEE Communications Surveys & Tutorials, vol. 9, no. 3, 2007, pp. 44–57.

[28] H. Khan, M. Javed, S. A. Khayam, and F. Mirza, "Designing a cluster-based covert channel to evade disk investigation and forensics," Computers & Security, vol. 30 (1), 2011, pp. 35–49.

[29] A. Baliga, J. Kilian, and L. Iftode, "A web based covert file system," in Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems, ser. HOTOS'07. USA: USENIX Association, 2007.

[30] C. Sosa, B. C. Sutton, and H. H. Huang, "PicFS: The privacy-enhancing image-based collaborative file system," in 16th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2010, Shanghai, China, December 8-10, 2010. IEEE Computer Society, 2010, pp. 99–106. [Online]. Available: https://doi.org/10.1109/ICPADS.2010.13

[31] E. Huebner, D. Bem, and C. K. Wee, "Data hiding in the NTFS file system," Digit. Investig., vol. 3, no. 4, 2006, pp. 211–226. [Online]. Available: https://doi.org/10.1016/j.diin.2006.10.005

[32] W. Mazurczyk and L. Caviglione, "Steganography in modern smartphones and mitigation techniques," IEEE Communications Surveys & Tutorials, vol. 17, no. 1, 2014, pp. 334–357.

[33] S. Wendzel, W. Mazurczyk, and G. Haas, "Don't you touch my nuts: Information hiding in cyber physical systems," in IEEE Security and Privacy Workshops (SPW'17). IEEE, 2017, pp. 29–34.

[34] T. Schmidbauer, S. Wendzel, A. Mileva, and W. Mazurczyk, "Introducing dead drops to network steganography using ARP-caches and SNMP-walks," in Proceedings of the 14th International Conference on Availability, Reliability and Security, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3339252.3341488

[35] W. Mazurczyk and S. Wendzel, "Information hiding: Challenges for forensic experts," Commun. ACM, vol. 61, no. 1, Dec. 2017, p. 86–94. [Online]. Available: https://doi.org/10.1145/3158416

[36] A. Hadida, J. Lampel, W. Walls, and A. Joshi, "Hollywood studio filmmaking in the age of Netflix: a tale of two institutional logics," Journal of Cultural Economics, vol. 45, 06 2021.

[37] S. Gupta and D. Gupta, "Text-steganography: Review study & comparative analysis," International Journal of Computer Science and Information Technologies (IJCSIT), vol. 2, no. 5, 2011, pp. 2060–2062.

[38] J. Han, M. Pan, D. Gao, and H. Pang, "A multi-user steganographic file system on untrusted shared storage," in Twenty-Sixth Annual Computer Security Applications Conference, ACSAC 2010, Austin, Texas, USA, 6-10 December 2010, C. Gates, M. Franz, and J. P. McDermott, Eds. ACM, 2010, pp. 317–326. [Online]. Available: https://doi.org/10.1145/1920261.1920309

[39] K. Eckstein and M. Jahnke, "Data hiding in journaling file systems," in Proceedings of 5th Digital Forensic Research Workshop, 2005.

[40] T. Ulz, M. Feldbacher, T. Pieber, and C. Steger, "Sensing danger: exploiting sensors to build covert channels," in Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019), Prague, Czech Republic, 2019, pp. 100–113.

[41] P. Krishnamurthy, F. Khorrami, R. Karri, D. Paul-Pena, and H. Salehghaffari, "Process-aware covert channels using physical instrumentation in cyber-physical systems," IEEE Transactions on Information Forensics and Security, vol. 13 (11), 2018, pp. 2761–2771.

[42] M. Hildebrandt, R. Altschaffel, K. Lamshöft, M. Lange, M. Szemkus, T. Neubert, C. Vielhauer, Y. Ding, and J. Dittmann, "Threat analysis of steganographic and covert communication in nuclear I&C systems," in International Conference on Nuclear Security: Sustaining and Strengthening Efforts, 2020.

[43] S. Wendzel, W. Mazurczyk, and S. Zander, "A unified description method for network information hiding methods," Journal of Universal Computer Science (J.UCS), vol. 22, no. 11, 2016, pp. 1456–1486. [Online]. Available: http://dx.doi.org/10.3217/jucs-022-11-1456

[44] S. Wendzel and C. Palmer, "Creativity in mind: Evaluating and maintaining advances in network steganographic research." Journal of Universal Computer Science (J.UCS), vol. 21, no. 12, 2015, pp. 1684–1705, https://dx.doi.org/10.3217/jucs-021-12-1684.

[45] D. Spiekermann, J. Keller, and T. Eggendorfer, "Towards covert channels in cloud environments: a study of implementations in virtual networks," in International Workshop on Digital Watermarking. Springer, 2017, pp. 248–262.

[46] S. Zander, G. Armitage, and P. Branch, "Covert channels in multiplayer first person shooter online games," in 2008 33rd IEEE Conference on Local Computer Networks (LCN). IEEE, 2008, pp. 215–222.

[47] C. Xiao, C. Zhang, and C. Zheng, "FontCode: embedding information in text documents using glyph perturbation," ACM Transactions on Graphics, vol. 1, no. 1, December 2017.

[48] M. Khairullah, "A novel text steganography system using font color of the invisible characters in microsoft word documents," in Proceedings of the Second International Conference on Computer and Electrical Engineering (ICCEE '09), December 2009, pp. 482–484.

[49] I. Stojanov, A. Mileva, and I. Stojanovic, "A new property coding in text steganography of microsoft word documents," in Proceedings of SECURWARE 2014: The Eighth International Conference on Emerging Security Information, Systems and Technologies, November 2014, pp. 25–30.

[50] L. Ji, Y. Fan, and C. Ma, "Covert channel for local area network," in IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS'10), 2010, pp. 316–319.

[51] N. B. Lucena, G. Lewandowski, and S. J. Chapin, "Covert channels in IPv6," in International Workshop on Privacy Enhancing Technologies. Springer, 2005, pp. 147–166.

[52] W. Mazurczyk, S. Wendzel, and K. Cabaj, "Towards deriving insights into data hiding methods using pattern-based approach," in Proc. Second International Workshop on Criminal Use of Information Hiding (CUING 2018), part of Proc. ARES'18. ACM, 2018, pp. 10:1–10:10.

[53] A. Velinov, A. Mileva, S. Wendzel, and W. Mazurczyk, "Covert channels in MQTT-based internet of things," ACCESS, vol. 7, 2019, pp. 161 899–161 915.

[54] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, and W. Mazurczyk, "Comprehensive analysis of MQTT 5.0 susceptibility to network covert channels," Computers & Security (COSE), vol. 104, no. 102207, 2021.

[55] L. Hartmann, S. Zillien, and S. Wendzel, "Reset- and reconnection-based covert channels in CoAP," in European Interdisciplinary Cybersecurity Conference, ser. EICC. New York, NY, USA: Association for Computing Machinery, 2021, p. 66–71. [Online]. Available: https://doi.org/10.1145/3487405.3487660

[56] T. G. Handel and M. T. Sandford II., "Hiding data in the OSI network model," in Proceedings of the 1st International Workshop on Information Hiding, 1996, pp. 23–38.

[57] M. Wolf, "Covert channels in LAN protocols," in Proc. Local Area Network Security, ser. LNCS. Springer, 1989, vol. 396, pp. 89–101.

[58] B. Jankowski, W. Mazurczyk, and K. Szczypiorski, "Information hiding using improper frame padding," eprint arXiv:1005.1925, 2010.

[59] A.-R. Sadeghi, S. Schulz, and V. Varadharajan, "The silence of the lans: Efficient leakage resilience for IPsec VPNs," in Computer Security – ESORICS 2012, ser. LNCS, vol. 7459. Springer Berlin Heidelberg, 2012, pp. 253–270.

[60] D. Stødle, "Ping tunnel – for those times when everything else is blocked," 2009, http://www.cs.uit.no/~daniels/PingTunnel/.

[61] daemon9, "LOKI2 (the implementation)," Phrack Magazine, vol. 7, no. 51, 1997, http://www.phrack.org/issues.html?issue=51&id=6.
[62] S. Wendzel, B. Kahler, and T. Rist, "Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet," in Proc. 2012 Int. Conf. Green Computing and Communications (GreenCom). IEEE, 2012, pp. 731–736.
[63] R. Rios, J. Onieva, and J. Lopez, "HIDE_DHCP: Covert communications through network configuration messages," in Proc. IFIP TC 11 27th International Information Security Conference. Springer, 2012.
[64] W. Mazurczyk and K. Szczypiorski, Covert Channels in SIP for VoIP signalling. Springer, 2008, pp. 65–72.
[65] C. H. Rowland, "Covert channels in the TCP/IP protocol suite," First Monday, vol. 2, no. 5, May 1997, https://firstmonday.org/ojs/index.php/fm/article/view/528/449.
[66] J. Rutkowska, "The implementation of passive covert channels in the Linux kernel," 2004, speech held at the 21st Chaos Communication Congress, Berlin, Germany, http://events.ccc.de/congress/2004/fahrplan/files/319-passive-covert-channels-slides.pdf.
[67] N. Lucena, J. Pease, P. Yadollahpour, and S. J. Chapin, "Syntax and semantics-preserving application-layer protocol steganography," in Proceedings of 6th Information Hiding Workshop, May 2004.
[68] T. Schmidbauer, S. Wendzel, and J. Keller, "Challenging channels: Encrypted covert channels within challenge-response authentication," in Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES), 2022, in press.
[69] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert messaging through TCP timestamps," in Proc. 2nd International Conference on Privacy Enhancing Technologies. Springer, 2003, pp. 194–208.
[70] R. Patuck and J. Hernandez-Castro, "Steganography using the extensible messaging and presence protocol (XMPP)," CoRR, vol. abs/1310.0524, 2013.
[71] G. Bernieri, S. Cecconello, M. Conti, and G. Lain, "TAMBUS: A novel authentication method through covert channels for securing industrial networks," Computer Networks, vol. 183, 2020, p. 107583.
[72] A. Dyatlov and S. Castro, "Exploitation of data streams authorized by a network access control system for arbitrary data transfers: tunneling and covert channels over the HTTP protocol," Gray-World.net, Tech. Rep., 2005.
[73] W. Mazurczyk, P. Szaga, and K. Szczypiorski, "Using transcoding for hidden communication in IP telephony," Multimedia Tools Appl., vol. 70, no. 3, 2014, pp. 2139–2165.
[74] S. Schmidt, W. Mazurczyk, J. Keller, and L. Caviglione, "A new data-hiding approach for IP telephony applications with silence suppression," in Proceedings of the 12th International Conference on Availability, Reliability and Security, 2017, pp. 1–6.
[75] B. Jankowski, W. Mazurczyk, and K. Szczypiorski, "PadSteg: Introducing inter-protocol steganography," Telecommunication Systems, vol. 52, no. 2, 2013, pp. 1101–1111.
[76] C. Kraetzer, J. Dittmann, A. Lang, and T. Kuehne, "WLAN steganography: A first practical review," in Proc. 8th Workshop on Multimedia and Security (MMSEC'06), 2006, pp. 17–22.
[77] S. Zillien and S. Wendzel, "Detection of covert channels in TCP retransmissions," in Secure IT Systems, N. Gruschka, Ed. Cham: Springer International Publishing, 2018, pp. 203–218.
[78] X. Li, Y. Zhang, F. Chong, and B. Zhao, "A covert channel analysis of a real switch," Dep. of Computer Science, University of California, Santa Barbara, Tech. Rep., 2011.
[79] W. Mazurczyk, M. Smolarczyk, and K. Szczypiorski, "Retransmission steganography and its detection," Soft Computing, vol. 15, no. 3, 2011, pp. 505–515.
[80] W. Mazurczyk and K. Szczypiorski, "Evaluation of steganographic methods for oversized IP packets," Telecommunication Systems, vol. 49, no. 2, 2012, pp. 207–217.
[81] X.-g. Zou, Q. Li, S.-H. Sun, and X. Niu, "The research on information hiding based on command sequence of FTP protocol," in Proc. 9th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES 2005), Part III, ser. LNCS, vol. 3683. Springer Berlin Heidelberg, 2005, pp. 1079–1085.
[82] T. Graf, "Messaging over IPv6 destination options," 2003, swiss Unix User Group.
[83] A. Getchell, "Re: For those interested in covert channels," 2008, a posting on the securityfocus penetration testing mailinglist, https://seclists.org/pen-test/2008/Dec/292.
[84] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert timing channels: design and detection," in Proceedings of the 11th ACM conference on Computer and communications security, ser. CCS '04. New York, NY, USA: ACM, 2004, pp. 178–187. [Online]. Available: http://doi.acm.org/10.1145/1030083.1030108
[85] ——, "IP covert channel detection," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 4, April 2009, pp. 22:1–22:29.
[86] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in Proc. 15th USENIX Security Symposium. USENIX Association, 2006, pp. 59–75.
[87] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-based covert timing channels: Automated modeling and evasion," in Proceedings of Recent Advances in Intrusion Detection (RAID) Symposium, September 2008. [Online]. Available: http://www.cs.wm.edu/~srgian/paper/raid08.pdf
[88] S. Zander, G. Armitage, and P. Branch, "Stealthier inter-packet timing covert channels," in IFIP Networking. Springer Berlin Heidelberg, May 2011, pp. 458–470.
[89] V. Ravi, M. Alazab, S. Srinivasan, A. Arunachalam, and K. Soman, "Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning," IEEE Transactions on Engineering Management, 2021.
[90] H. Alsaadi, M. Al-Anni, R. Almuttairi, O. Bayat, and O. Ucan, "Text steganography in font color of MS Excel sheet," in DATA '18: Proceedings of the First International Conference on Data Science, E-learning and Information Systems, 2018, pp. 1–7.
[91] W. Bhaya, A. M. Rahma, and D. Al-Nasrawi, "Text steganography based on font type in MS-Word documents," Journal of Computer Science, vol. 9, no. 7, 2013, pp. 898—-904.
[92] R. Villán, S. Voloshynovskiy, O. Koval, J. Vila, E. Topak, F. Deguillaume, Y. Rytsar, and T. Pun, "Text data-hiding for digital and printed documents: Theoretical and practical considerations," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, 2006.
[93] M. Chapman, G. Davida, and M. Rennhard, "A practical and effective approach to largescale automated linguistic steganography," in Proceedings of the Information Security Conference (ISC '01), 2001, pp. 156–156.
[94] C. Jin, D. Zhang, and M. Pan, "Chinese text information hiding based on paraphrasing technology," in Proceedings of the International Conference on Information Science and Management Engineering (ISME2010), vol. 1, 2010.
[95] M. Shirali-Shahreza, "Text steganography by changing words spelling," in Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT 2008), vol. 3, 2008, pp. 1912–1913.
[96] M. Topkara, U. Taskiran, and M. J. Atallah, "Information Hiding Through Errors: A Confusing Approach," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, 2007.
[97] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35 (Nos3&4), 1996, pp. 313–336.
[98] B. Murphy and C. Vogel, "The syntax of concealment: reliable methods for plain text information hiding," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, 2007.
[99] A. Gutub and M. Fattani, "A novel arabic text steganography method using letter points and extensions," in Proceedings of the WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), vol. 21, 2007, pp. 28—-31.
[100] Z.-H. Wang, T. D. Kieu, C.-C. Chang, and M.-C. Li, "Emoticon-based text steganography in chat," in Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA), 2009, pp. 457–460.
[101] V. Iranmanesh, H. J. Wei, S. L. Dao-Ming, and O. A. Arigbabu, "On using emoticons and lingoes for hiding data in SMS," in International Symposium on Technology Management and Emerging Technologies (ISTMET), 2015, pp. 103–107.
[102] L. Y. Por, K. Wong, and K. O. Chee, "UniSpaCh: A text-based data hiding method using unicode space characters," Journal of Systems and Software, vol. 85, 2012, pp. 1075–1082.
[103] T. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, 1995, pp. 1495–1504.
[104] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in Proceedings of the 14th Annual Joint Conference of the

IEEE Computer and Communications Societies (INFOCOM '95), April 1995, pp. 853—-860.
[105] A. Odeh, K. Elleithy, and M. Faezipour, "Steganography in text by using MS Word symbols," in Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education, 2014.
[106] T. Fang, M. Jaggi, and K. Argyraki, "Generating steganographic text with LSTMs," in Proceedings of ACL 2017, Student Research Workshop, 2017, pp. 100–106.
[107] Z.-L. Yang, X.-Q. Guo, Z.-M. Chen, Y.-F. Huang, and Z. Y.-J., "RNN-Stega: Linguistic steganography based on recurrent neural networks," IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, 2018, pp. 1280–1295.
[108] Z. Yang, N. Wei, Q. Liu, Y. Huang, and Y. Zhang, "GAN-TStega: text steganography based on generative adversarial networks," in Digital Forensics and Watermarking (IWDW 2019), LNCS, vol. 12022. Springer, Cham, 2019.
[109] N. Wu, Z. Yang, Y. Yang, L. Li, P. Shang, W. Ma, and Z. Liu, "STBS-Stega: Coverless text steganography based on state transition-binary sequence," International Journal of Distributed Sensor Networks, vol. 16, no. 3, 2020.
[110] M. Chapman and G. Davida, "Hiding the hidden: A software system for concealing ciphertext as innocuous text," in International Conference on Information and Communications Security, 1997, pp. 335–345.
[111] L. Guan, Y. Jing, S. Li, and R. Zhang, "A standard MIDI file steganography based on music perception in note duration," in Proceedings of the 6th Conference on Sound and Music Technology (CSMT), W. Li, S. Li, X. Shao, and Z. Li, Eds. Singapore: Springer Singapore, 2019, pp. 99–107.
[112] M. Qiao, A. Sung, and Q. Liu, "Steganalysis of MP3Stego," 06 2009, pp. 2566–2571.
[113] L. Hartmann and S. Wendzel, "How feasible are steganographic and stealth attacks on TIA project meta-data of ICS: A case study with real-world data," in Proc. European Interdisciplinary Cybersecurity Conference (EICC 2021). ACM, 2021.
[114] A. Abdelwahab, W. Lucia, and A. Youssef, "Covert channels in cyber-physical systems," IEEE Control Systems Letters, vol. 5, no. 4, 2021, pp. 1273–1278.
[115] K. Gheitasi and W. Lucia, "Undetectable finite-time covert attack on constrained cyber-physical systems," IEEE Transactions on Control of Network Systems, 2022, in press.
[116] S. Kim, G. Yeo, T. Kim, J. J. Rhee, Y. Jeon, A. Bianchi, D. Xu, and D. J. Tian, "ShadowAuth: Backward-compatible automatic CAN authentication for legacy ECUs," in Proc. 2022 ACM on Asia Conference on Computer and Communications Security, ser. ASIA CCS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 534–545. [Online]. Available: https://doi.org/10.1145/3488932.3523263
[117] D. Evtyushkin and D. Ponomarev, "Covert channels through random number generator: Mechanisms, capacity estimation and mitigations," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 843–857. [Online]. Available: https://doi.org/10.1145/2976749.2978374
[118] K. Lamshöft, T. Neubert, C. Krätzer, C. Vielhauer, and J. Dittmann, "Information hiding in cyber physical systems: Challenges for embedding, retrieval and detection using sensor data of the SWAT dataset," in Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, ser. IH&MMSec '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 113–124. [Online]. Available: https://doi.org/10.1145/3437880.3460413
[119] A. Herzberg and Y. Kfir, "The chatty-sensor: A provably-covert channel in cyber physical systems," in Proceedings of the 35th Annual Computer Security Applications Conference, ser. ACSAC '19. New York, NY, USA: Association for Computing Machinery, Dec. 2019, pp. 638–649.
[120] A. Bacs, S. Musaev, K. Razavi, C. Giuffrida, and H. Bos, "DUPEFS: Leaking data over the network with filesystem deduplication side channels," in 20th USENIX Conference on File and Storage Technologies (FAST 22). Santa Clara, CA: USENIX Association, Feb. 2022, pp. 281–296. [Online]. Available: https://www.usenix.org/conference/fast22/presentation/bacs
[121] C. Shepherd, J. Kalbantner, B. Semal, and K. Markantonakis, "A side-channel analysis of sensor multiplexing for covert channels and application fingerprinting on mobile devices," 2021. [Online]. Available: https://arxiv.org/abs/2110.06363
[122] A. Al-Haiqi, M. Ismail, and R. Nordin, "A new sensors-based covert channel on Android," vol. 2014, no. 969628, 2014.
[123] A. Herzberg and Y. Kfir, "The Leaky Actuator: A Provably-covert Channel in Cyber Physical Systems," in Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy, ser. CPS-SPC'19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 87–98.
[124] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Detecting homoglyph attacks with a siamese neural network," in 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018, pp. 22–28.
[125] S. Neuner, A. G. Voyiatzis, M. Schmiedecker, S. Brunthaler, S. Katzenbeisser, and E. R. Weippl, "Time is on my side: Steganography in filesystem metadata," Digital Investigation, vol. 18, 2016, pp. S76–S86. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742287616300433
[126] A. D. McDonald and M. G. Kuhn, "StegFS: A steganographic file system for Linux," in International Workshop on Information Hiding. Springer, 1999, pp. 463–477.
[127] A. Barker, S. Sample, Y. Gupta, A. McTaggart, E. L. Miller, and D. D. E. Long, "Artifice: A deniable steganographic file system," in 9th USENIX Workshop on Free and Open Communications on the Internet, FOCI 2019, Santa Clara, CA, USA, August 13, 2019, S. E. McGregor and M. C. Tschantz, Eds. USENIX Association, 2019. [Online]. Available: https://www.usenix.org/conference/foci19/presentation/barker
[128] J. Heeger, Y. Yannikos, and M. Steinebach, "ExHide: Hiding data within the ExFAT file system," in The 16th International Conference on Availability, Reliability and Security, ser. ARES 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3465481.3470117
[129] R. J. Anderson, R. M. Needham, and A. Shamir, "The steganographic file system," in Information Hiding, Second International Workshop, Portland, Oregon, USA, April 14-17, 1998, Proceedings, ser. Lecture Notes in Computer Science, D. Aucsmith, Ed., vol. 1525. Springer, 1998, pp. 73–82. [Online]. Available: https://doi.org/10.1007/3-540-49380-8_6
[130] H. Pang, K.-L. Tan, and X. Zhou, "StegFS: A steganographic file system," in Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405). IEEE, 2003, pp. 657–667.
[131] L. Caviglione, M. Podolski, W. Mazurczyk, and M. Ianigro, "Covert channels in personal cloud storage services: The case of Dropbox," IEEE Transactions on Industrial Informatics, vol. 13, no. 4, 2017, pp. 1921–1931.
[132] B. Chang, Z. Wang, B. Chen, and F. Zhang, "MobiPluto: File system friendly deniable storage for mobile devices," in Proceedings of the 31st Annual Computer Security Applications Conference, Los Angeles, CA, USA, December 7-11, 2015. ACM, 2015, pp. 381–390. [Online]. Available: https://doi.org/10.1145/2818000.2818046
[133] J. Aycock and D. M. N. de Castro, "Permutation steganography in FAT filesystems," Transactions on Data Hiding and Multimedia Security X, vol. 8948, 2015, pp. 92–105.
[134] H. Khan, M. Javed, S. A. Khayam, and F. Mirza, "Designing a cluster-based covert channel to evade disk investigation and forensics," Computers & Security (COAS), vol. 30, no. 1, 2011, pp. 35–49.
[135] S. Cabuk, "Network covert channels: Design, analysis, detection, and elimination," Ph.D. dissertation, Purdue University, 2006.
[136] V. Berk, A. Giani, and G. Cybenko, "Detection of covert channel encoding in network packet delays," Department of Computer Science - Dartmouth College, Tech. Rep., 2005.
[137] S. Wendzel and S. Zander, "Detecting protocol switching covert channels," in Proc. 37th IEEE Conference on Local Computer Networks (LCN), 2012, pp. 280–283.
[138] F. Iglesias, V. Bernhardt, R. Annessi, and T. Zseby, "Decision tree rule induction for detecting covert timing channels in TCP/IP traffic," in International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer, 2017, pp. 105–122.
[139] J. Han, C. Huang, F. Shi, and J. Liu, "Covert timing channel detection method based on time interval and payload length analysis," Computers & Security, vol. 97, 2020, p. 101952. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820302285
[140] P. L. Shrestha, M. Hempel, F. Rezaei, and H. Sharif, "A support vector machine-based framework for detection of covert timing channels," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, 2015, pp. 274–283.
[141] O. Darwish, A. Al-Fuqaha, G. B. Brahim, I. Jenhani, and A. Vasilakos, "Using hierarchical statistical analysis and deep neural networks to detect covert timing channels," Applied Soft Computing, vol. 82, 2019, p. 105546.

[142] L. Xiang, X. Sun, G. Luo, and B. Xia, "Linguistic steganalysis using the features derived from synonym frequency," Multimedia Tools and Applications, vol. 71, 2014, pp. 1893–1922.

[143] L. Xiang, X. Sun, G. Luo, and C. Gan, "Research on steganalysis for text steganography based on font format," in Third International Symposium on Information Assurance and Security, 2007, pp. 490–495.

[144] S. Changder, D. Ghosh, and N. Debnath, "Linguistic approach for text steganography through Indian text," in 2010 2nd international conference on computer technology and development. IEEE, 2010, pp. 318–322.

[145] P. Meng, L. Hang, W. Yang, Z. Chen, and H. Zheng, "Linguistic steganography detection algorithm using statistical language model," in 2009 international conference on information technology and computer science, vol. 2. IEEE, 2009, pp. 540–543.

[146] L. Li, L. Huang, X. Zhao, W. Yang, and Z. Chen, "A statistical attack on a kind of word-shift textsteganography," in International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008, pp. 1503–1507.

[147] Z. Chen, L. Huang, Z. Yu, W. Yang, L. Li, X. Zheng, and X. Zhao, "Linguistic steganography detection using statistical characteristics of correlations between words," in The 11th In-ternational Workshop on Information Hiding, Darmstadt, Germany, 2008, pp. 224–235.

[148] Z. Yang, Y. Huang, and Y.-J. Zhang, "A fast and efficient text steganalysis method," IEEE Signal Processing Letters, vol. 26, 2019.

[149] Z. Yang, K. Wang, J. Li, Y. Huang, and Y.-J. Zhang, "TS-RNN: Text steganalysis based on recurrent neural networks," IEEE Signal Processing Letters, vol. 26, 2019.

[150] J. Wen, X. Zhou, P. Zhong, and Y. Xuen, "Convolutional neural network based text steganalysis," IEEE Signal Processing Letters, vol. 26, 2019.

[151] R. Böhme, Advanced Statistical Steganalysis, ser. Information Security and Cryptography. Springer Berlin Heidelberg, 2010.

[152] J. Tonejc, J. Kaur, A. Karsten, and S. Wendzel, "Visualizing BACnet data to facilitate humans in building-security decision-making," in Human Aspects of Information Security, Privacy, and Trust, T. Tryfonas and I. Askoxylakis, Eds. Cham: Springer International Publishing, 2015, pp. 693–704.

[153] J. Tonejc, S. Güttes, A. Kobekova, and J. Kaur, "Machine learning methods for anomaly detection in BACnet networks." J. Univers. Comput. Sci., vol. 22, no. 9, 2016, pp. 1203–1224.

[154] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities," ACM Computing Surveys (CSUR), vol. 54, no. 5, 2021, pp. 1–36.

[155] C. Troncoso, C. Díaz, O. Dunkelman, and B. Preneel, "Traffic analysis attacks on a continuously-observable steganographic file system," in Information Hiding, 9th International Workshop, IH 2007, Saint Malo, France, June 11-13, 2007, Revised Selected Papers, ser. Lecture Notes in Computer Science, T. Furon, F. Cayre, G. J. Doërr, and P. Bas, Eds., vol. 4567. Springer, 2007, pp. 220–236. [Online]. Available: https://doi.org/10.1007/978-3-540-77370-2_15

[156] H. Pang, K. Tan, and X. Zhou, "Steganographic schemes for file system and b-tree," IEEE Trans. Knowl. Data Eng., vol. 16, no. 6, 2004, pp. 701–713. [Online]. Available: https://doi.org/10.1109/TKDE.2004.15

[157] X. Zhou, H. Pang, and K. Tan, "Hiding data accesses in steganographic file system," in Proceedings of the 20th International Conference on Data Engineering, ICDE 2004, 30 March - 2 April 2004, Boston, MA, USA, Z. M. Özsoyoglu and S. B. Zdonik, Eds. IEEE Computer Society, 2004, pp. 572–583. [Online]. Available: https://doi.org/10.1109/ICDE.2004.1320028

[158] T. Peters, M. A. Gondree, and Z. N. J. Peterson, "DEFY: A deniable, encrypted file system for log-structured storage," in 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015. The Internet Society, 2015. [Online]. Available: https://www.ndss-symposium.org/ndss2015/defy-deniable-encrypted-file-system-log-structured-storage

[159] M. Handley, V. Paxson, and C. Kreibich, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics," in Proc. 10th USENIX Security Symposium, vol. 10, 2001, pp. 115–131.

[160] N. Ogurtsov, H. Orman, R. Schroeppel et al., "Covert channel elimination protocols," Department of Computer Science, University of Arizona, Tech. Rep., 1996.

[161] M. H. Kang and I. Moskowitz, "A pump for rapid, reliable, secure communication," in Proc. 1st ACM Conference on Computer and Communication Security (CCS'93), November 1993, pp. 119–129.

[162] M. H. Kang, I. S. Moskowitz, and D. C. Lee, "A network pump," IEEE Transactions on Software Engineering, vol. 22, no. 5, 1996, pp. 329–338.

[163] W.-M. Hu, "Reducing timing channels with fuzzy time," in Proc. 1991 Symposium on Security and Privacy. IEEE, 1991, pp. 8–20.

[164] S. Wendzel and J. Keller, "Preventing protocol switching covert channels," International Journal On Advances in Security, vol. 5, no. 3 and 4, 2012, pp. 81–93.

[165] M. Zawawi, R. Mahmod, N. Udzir, F. Ahmad, and J. Desa, "Active warden as the main hindrance for steganography information retrieval," in 2012 International Conference on Information Retrieval & Knowledge Management. IEEE, 2012, pp. 277–280.

[166] R. Ravaioli, G. Urvoy-Keller, and C. Barakat, "Characterizing icmp rate limitation on routers," in 2015 IEEE International Conference on Communications (ICC). IEEE, 2015, pp. 6043–6049.

[167] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," IEEE Internet of Things Journal, vol. 4, no. 6, 2017, pp. 1802–1831.

[168] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," Annual reviews in control, vol. 47, 2019, pp. 394–411.

[169] Y. A. H. Fadlalla, "Approaches to resolving covert storage channels in multilevel secure systems," Ph.D. dissertation, University of New Brunswick, 1996.

[170] A. Sen, S. Alfeld, X. Zhang, A. Vartanian, Y. Ma, and X. Zhu, "Training set camouflage," in Decision and Game Theory for Security, L. Bushnell, R. Poovendran, and T. Başar, Eds. Cham: Springer International Publishing, 2018, pp. 59–79.

[171] G. Costa, F. Pinelli, S. Soderi, and G. Tolomei, "Covert channel attack to federated learning systems," 2021.

[172] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in 27th USENIX Security Symposium (USENIX Security 18). Baltimore, MD: USENIX Association, Aug. 2018, pp. 1615–1631. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/adi

[173] F. Boenisch, "A systematic review on model watermarking for neural networks," Frontiers in Big Data, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fdata.2021.729663

[174] C.-C. Chang and C.-Y. Lin, "Reversible steganographic method using SMVQ approach based on declustering," Information Sciences, vol. 177, no. 8, 2007, pp. 1796–1805.

[175] ——, "Reversible steganography for VQ-compressed images using side matching and relocation," IEEE Transactions on Information Forensics and Security, vol. 1, no. 4, 2006, pp. 493–501.

[176] C. Song, Y. Zhang, and G. Lu, "Reversible data hiding in encrypted images based on image partition and spatial correlation," in Proc. International Workshop On Digital Watermarking (IWDW), 2018, pp. 180–194.

[177] W. Mazurczyk, P. Szary, S. Wendzel, and L. Caviglione, "Towards reversible storage network covert channels," in Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019, pp. 1–8.

[178] P. Szary, W. Mazurczyk, S. Wendzel, and L. Caviglione, "Analysis of reversible network covert channels," IEEE Access, vol. 10, 2022, pp. 41 226–41 238.

[179] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV – a survey," IEEE transactions on circuits and systems for video technology, vol. 17, no. 11, 2007, pp. 1606–1621.

[180] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," IEEE Transactions on Multimedia, vol. 10, no. 8, 2008, pp. 1513–1527.

[181] P. R. Alface and B. Macq, "From 3D mesh data hiding to 3D shape blind and robust watermarking: a survey," in Transactions on data hiding and multimedia security II. Springer, 2007, pp. 91–115.

[182] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "Three-dimensional meshes watermarking: Review and attack-centric investigation," in International Workshop on Information Hiding. Springer, 2007, pp. 50–64.

[183] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici, "Security of additive manufacturing: Attack taxonomy and survey," Additive Manufacturing, 2018.

[184] J. F. DeFranko and J. Voas, "Reproducibility, fabrication, and falsification," Computer, vol. 54, no. 12, 2021, pp. 24–26.

[185] F. Iglesias, F. Meghdouri, R. Annessi, and T. Zseby, "CCgen: Injecting covert channels into network traffic," Security and Communication Networks, vol. 2022, 2022.
[186] H. Gunadi and S. Zander, "Bro covert channel detection (broccade) framework: scope and background," Murdoch University, Tech. Rep., 2017.
[187] ——, "Bro covert channel detection (broccade) framework: design and implementation," Murdoch University, Tech. Rep., 2017.
[188] ——, "Performance evaluation of the bro covert channel detection (broccade) framework," Murdoch University, Tech. Rep., 2018.
[189] R. Keidel, S. Wendzel, S. Zillien, E. S. Conner, and G. Haas, "WoDiCoF – a testbed for the evaluation of (parallel) covert channel detection algorithms." J. Univers. Comput. Sci., vol. 24, no. 5, 2018, pp. 556–576.
[190] S. Zander and G. Armitage, "CCHEF – covert channels evaluation framework design and implementation," Swinburne University of Technology. Centre for Advanced Internet Architectures, Tech. Rep., 2008.
[191] G. J. Simmons, "The prisoners' problem and the subliminal channel," in Proc. CRYPTO, 1983, pp. 51–67.

XI. Biographies

Steffen Wendzel is a professor of information security and computer networks at Hochschule Worms, Germany, where he is also the scientific director of the Center for Technology and Transfer (ZTT). In addition, he is a lecturer at the Faculty of Mathematics & Computer Science at the FernUniversität in Hagen, Germany, from which he also received his Ph.D. (2013) and Habilitation (2020). Before joining Hochschule Worms, he led a smart building security research team at Fraunhofer FKIE in Bonn, Germany. Steffen (co-)authored more than 170 publications and (co-)organized several conferences and workshops (e.g., EICC'21, Sicherheit'16, IWSMR'19-'22) and special issues for major journals, such as IEEE Security & Privacy (S&P), Elsevier Future Generation Computer Systems (FGCS), Journal of Universal Computer Science (J.UCS), Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), and IEEE Transactions Industrial Informatics (TII). He is editorial board member of J.UCS, Journal of Cybersecurity & Mobility (JCSM) and Frontiers in Computer Science. His major research focus is on covert channels, network steganography, scientific taxonomy, and IoT security. Website: https://www.wendzel.de.

Luca Caviglione is a Senior Research Scientist at the Institute for Applied Mathematics and Information Technologies of the National Research Council of Italy. He holds a Ph.D. in Electronic and Computer Engineering from the University of Genoa, Italy. His research interests include optimization of large-scale computing frameworks, traffic analysis, wireless and heterogeneous communication architectures, and network security. He is an author and co-author of more than 150 academic publications, and several patents in the field of p2p and energy-aware computing. He has been involved in Research Projects and Network of Excellences funded by the ESA, the EU and the MIUR. He is a work group leader of the Italian IPv6 Task Force, a contract professor in the field of networking/security and a professional engineer. He is the head of the IMATI Research Unit of the National Inter-University Consortium for Telecommunications and part of the Steering Committee of the Criminal Use of Information Hiding initiative.

Wojciech Mazurczyk is a University Professor with Institute of Computer Science, Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT), Poland. He received his B.Sc. (2003), M.Sc. (2004), Ph.D. (2009, with honours) and D.Sc. (habilitation, 2014) all in Telecommunications from WUT. He also is an author or co-author of 2 books, over 200 papers, 2 patent applications and over 35 invited talks. He has been involved in many international and domestic research projects as a principal investigator or as a senior researcher. He served as a guest editor of many special issues devoted to network security (among others: IEEE TDSC, IEEE S&P, IEEE Commag). He has been serving as Technical Program Committee Member of (among others): RAID, IEEE GLOBECOM, IEEE ICC, IEEE LCN, IEEE CNS, ACSAC, ARES and ACM IH&MMSec. From 2016 Editor-in-Chief of an open access Journal of Cyber Security and Mobility. Between 2018 and 2020 he was an Associate Editor of the IEEE TIFS and MCN Series Editor for the IEEE ComMag.

Aleksandra Mileva received the B.Sc., M.Sc., and Ph.D. degrees from Ss. Cyril and Methodius University in Skopje, Macedonia. She is currently a Full Professor with the Faculty of Computer Science, University Goce Delcev, Štip, Macedonia, where she is also the Head of the Laboratory of Computer Security and Computer Forensics. Her research interests include computer and network security, digital steganography, the IoT protocols and security, cryptography, computer forensics, and quasigroups theory. Since 2019, she has been a member of the EURASIP Data Forensics and Security TAC. She was with the management committee of two COST actions IC1201: BETTY and IC1306: Cryptography for Secure Digital Interaction. She is also member of the editorial boards of the Journal of Cyber Security and Mobility, and Mathematics, Computer Science and Education.

Jana Dittmann Jana Dittmann is a University Professor and the leader of the Advanced Multimedia and Security Laboratory (AMSL) at the Otto-von-Guericke University of Magdeburg, Germany. In addition, she is a Professorial Research Fellow with the University of Buckingham, UK. She is an internationally renowned academic and a leading researcher in the field of security of multimedia with a strong research record in the areas of steganography, digital watermarking, biometrics, and forensics from the technical and computer science perspectives as well as the user perception, user interaction, and legal perspectives. She is a member of the Association for Computing Machinery (ACM), and the German Gesellschaft für Informatik (GI). She has an excellent record in attracting EU funding, has experience as an Associated Editor of a number of prestigious journals, including the ACM Multimedia Systems and the IEEE Transactions on Image Processing.

Christian Krätzer is post-doc staff and lecturer at the Advanced Multimedia and Security Lab (AMSL) at the department of computer science at the Otto-von-Guericke University of Magdeburg, Germany. His research interests focus on steganography & steganalysis, digital watermarking, media forencics, biometrics, and cryptography. Website: https://omen.cs.uni-magdeburg.de/itiamsl/cms/front_content.php?idart=237

Kevin Lamshöft is a Ph.D. student at the Otto-von-Guericke University Magdeburg, Germany. His research topics are located in the field of Information Hiding (Network-Steganography, Covert Channels and Side Channels) as well as Cyber Security (especially IT/OT Security and Forensics regarding Cyber Physicals Systems, Industry 4.0 and the Internet of Things).

Claus Vielhauer is a full professor for IT Security at Brandenburg University of Applied Sciences in Germany and has an additional affiliation as the leader of the biometrics research group as part of the Advanced Multimedia and Security Lab (AMSL) at Otto-von-Guericke University of Magdeburg, Germany. His research interests are in biometrics and IT forensics with specialization in multimodal and behavioral-based recognition, biometric cryptography and applications of biometrics & forensics to multimedia, as well as Human-to-Computer Interaction (HCI). Additionally, he is elaborating methods from biometric user authentication to related IT security problems such as digital watermarking, steganography or IT forensics. He has a large number of book, journal and conference publications in the areas of biometric signal processing, forensics, pattern recognition, HCI and multimedia security. Furthermore, he serves as editor for Springer EURASIP Journal on Information Security and IET Biometrics. Prof. Vielhauer is member of the European Association for Signal Processing (EURASIP) and has been nominated as deputy national delegate for Germany to a number of EU ICT COST Actions on topics of biometrics and forensics.

Laura Hartmann Since 2019, Laura Hartmann is a Ph.D. student at FernUniversität in Hagen, Germany. She worked for the project MADISA (*Machine learning for attack detection using data of industrial control systems*) (till 2021) and is now a researcher at the Center for Technology and Transfer (ZTT). Her research interests include network steganography and anomaly detection for data of industrial control systems. Laura (co-)authored seven publications so far. Website: https://madisa.ztt.hs-worms.de

Jörg Keller Jörg Keller is a professor of computer engineering at FernUniversität in Hagen, Germany, where he leads the Parallelism & VLSI research group since 1996. He received MSc and PhD degrees and habilitation from Saarland University, Saarbrücken, in 1989, 1992, and 1996, respectively. His research interests include network steganography, cryptographic primitives for embedded systems, energy-efficient and fault-tolerant parallel computation, and blended and virtual laboratories. He is author or co-author of 2 books and more than 170 refereed articles in journals and conference proceedings. He has co-organized numerous conferences and workshops, and special issues for Journal of Universal Computer Science. He is a member of the editorial boards of Journal of Cybersecurity & Mobility and Journal of Universal Computer Science. His research group website: https://feu.de/pv

Tom Neubert is a research assistant and Ph.D. student at the Brandenburg University of Applied Science, Germany. His research topics are located in the field of Information Hiding (Network-Steganography, Covert Channels and Side Channels) as well as Machine Learning and IT-Forensics.

Sebastian Zillien is a researcher and Ph.D. student at the Hochschule Worms, Germany, where he works for the project SIVERT (Secure & Intelligent Visualization- & Real-time Reconstruction Methods (for pCT)). His field of research includes network & protocol security, covert channels and reliability. He has worked on multiple projects in his field and has had publications, among others, at IFIP SEC, NordSec and in J.UCS.