Network Covert Channel Patterns:
Current State & Methodology

Steffen Wendzel
Fraunhofer FKIE & Hochschule Worms

http://www.wendzel.de

Introducing myself

2016-now Prof. at Worms Univ. of Appl. Sciences
(since 2017: deputy scientific head of ZTT unit)

2013-now: Researcher at Fraunhofer FKIE

2009-2013: PhD student @University of Hagen

Primary research interests:

- Network Information Hiding/Covert Channels
  \- cleaning up the terminology, taxonomy, methodology
  \- developing countermeasures and new hiding techniques

- IoT/Smart Home/Smart Building Security
  \- network-level security, e.g. traffic normalization, anomaly detection, communication protocols

- Scientometrics for information security

Hochschule Worms
University of Applied Sciences

INTRODUCTION

Information Hiding

What is „Information Hiding“? Two different examples:

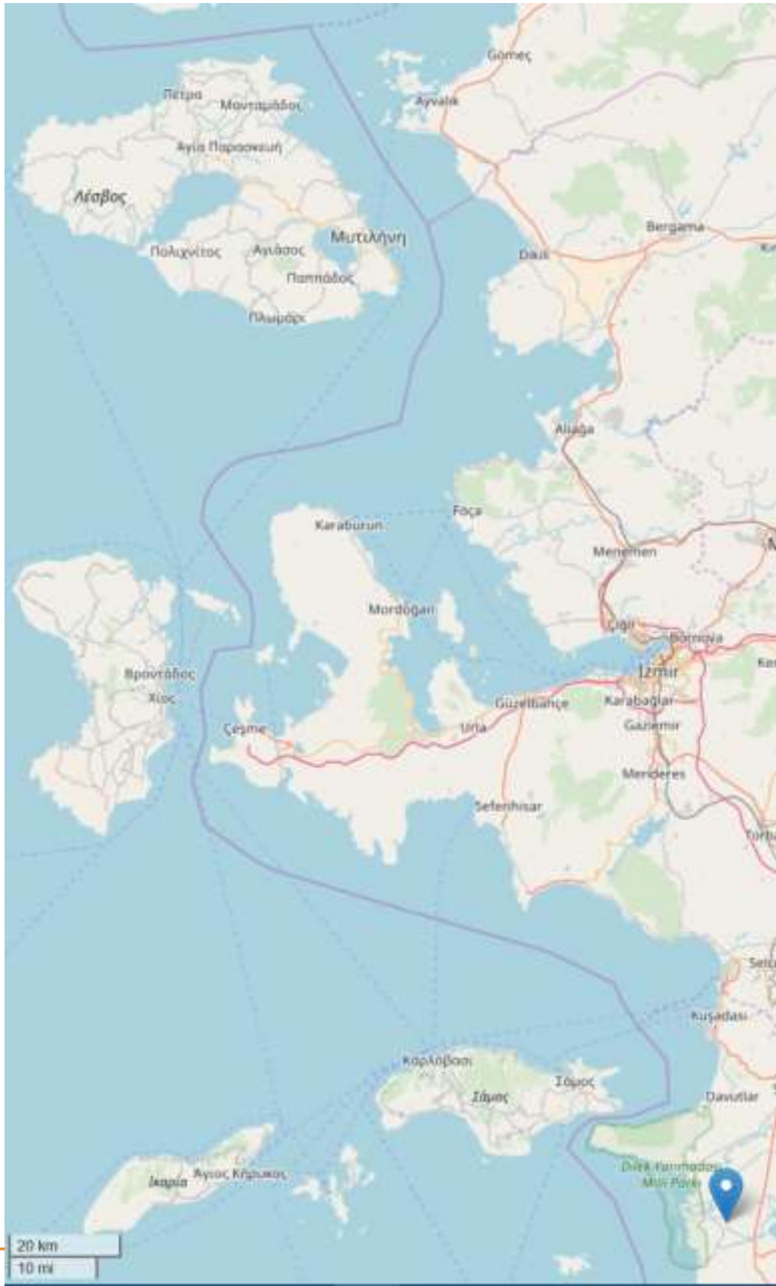
All figures taken from Wikipedia articles on ‚Steganography‘ and ‚Watermarking‘

Information Hiding

… it also appeared in ancient Greece.

499 BC: **Histiaeus** (ruler of Miletus) tattooed a message on the head of one of his slaves to send a message to Aristagoras (his son-in-law) to instruct him to revolt against the Persians.

(Several more cases of Steganography in ancient Greece are known.)



Information Hiding

What is „Information Hiding“? Another example (from Fridrich, 2010):

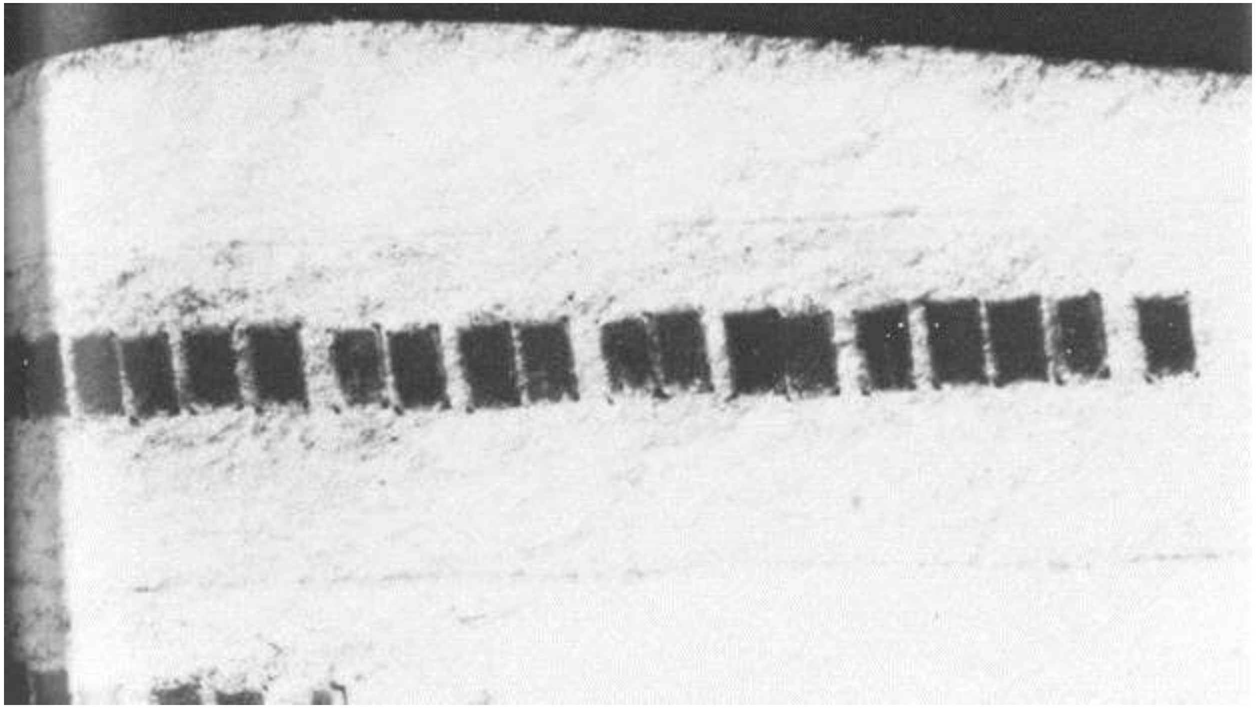
- 1978 World Championship in chess between Viktor Korchnoi (CH/RU) and Anatoly Karpov (RU)
  - Officials „limited Karpov to consumption of only one type of yogurt (violet) at a fixed time during the game.“ (Fridrich, 2010)

Fig.: private photo

Fridrich, J.: Steganography in Digital Media, Cambridge University Press, 2010.

Information Hiding

Another example: Microdots; used during WW2, e.g. by German spies in Mexico.



Microdots used by German spies, Fig.: Wikipedia

History of Information Hiding

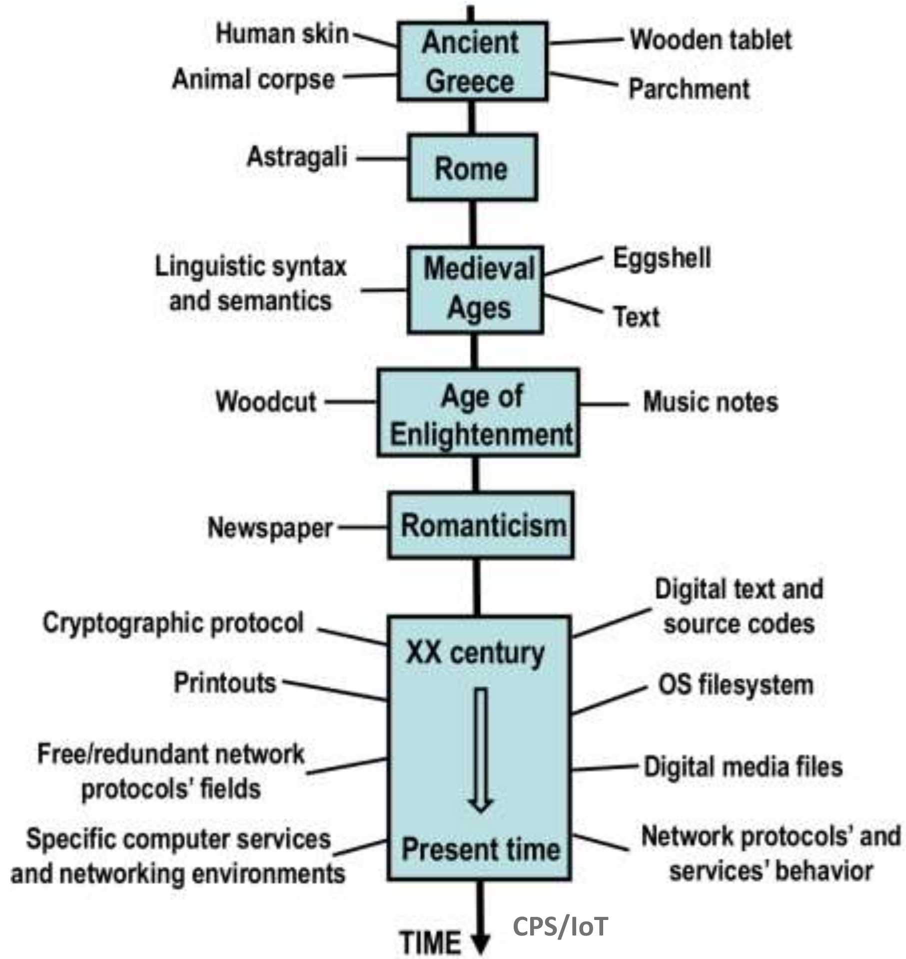


Fig.: W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Wiley-IEEE, 2016

History of Information Hiding

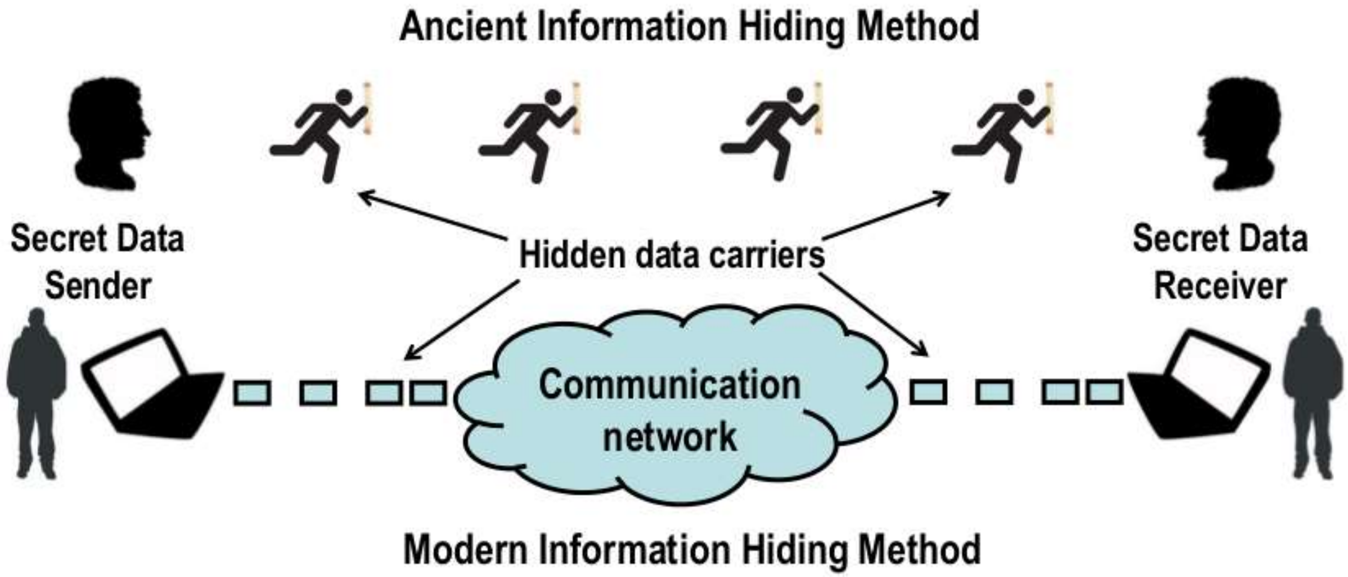


Fig.: W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Wiley-IEEE, 2016

Covert Data Storage & Communication

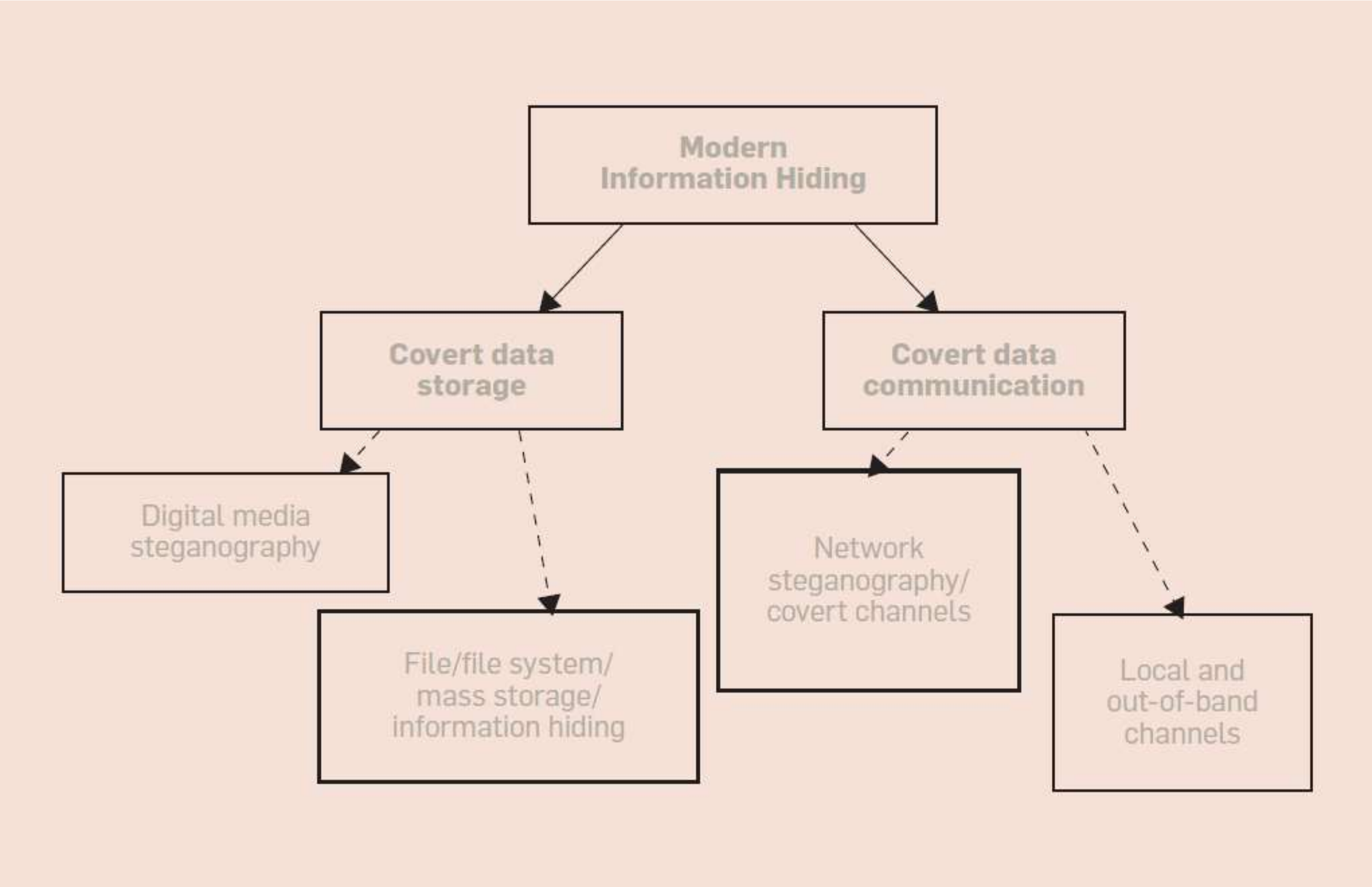


Fig.: W. Mazurczyk, S. Wendzel: Information Hiding: Challenges for Forensic Experts, Comm. ACM, 2018.

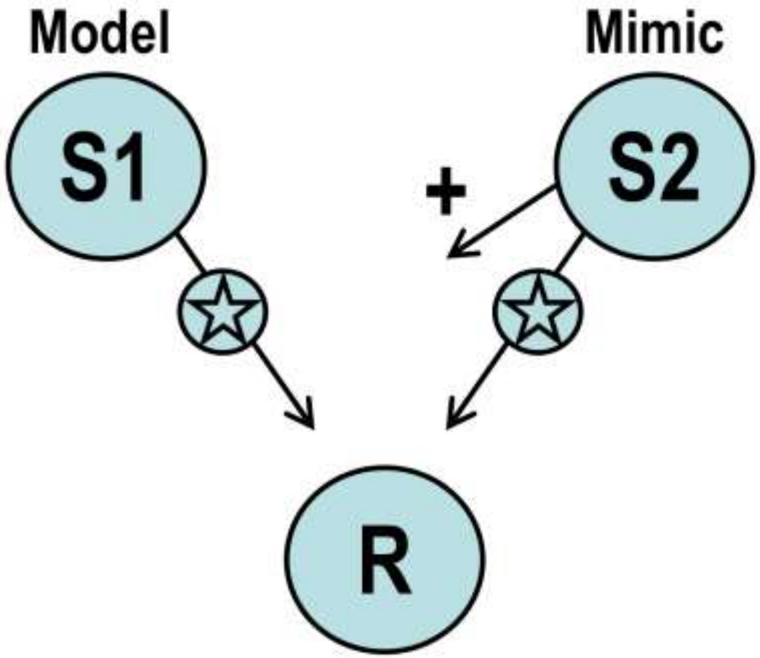
Application of Hiding Techniques

Okay, so what is the big difference between digital media and network carriers?

Feature/Type of the carrier	Digital media	Network traffic
Method's capacity/ bandwidth	Limited by the type of the digital media and the size of a file	Limited by the type of the traffic and the length of a transmission
Hidden data embedding	Cannot exceed file capacity	Can be slow but continuous over longer period of time
Data hiding application	Covert storage	Covert communication
Nature	Permanent	Ephemeral
Clues for forensic analysis	Can be available for forensic experts after transmission	Often not available when transmission ends
Method's detectability	Easy only if an original file is available	Hard due to different forms of acceptable traffic and varying network conditions
Cost of applying data hiding	Decrease in digital media quality	Increased delays, raised packet loss level, reduced feature set of protocols and/or affected user transmission quality
Robustness (secret data resistance to modifications)	Typically cannot survive conversion to another format	Typically vulnerable to dynamically changing network conditions

Fig.: W. Mazurczyk, S. Wendzel: Information Hiding: Challenges for Forensic Experts, Comm. ACM, 2018.

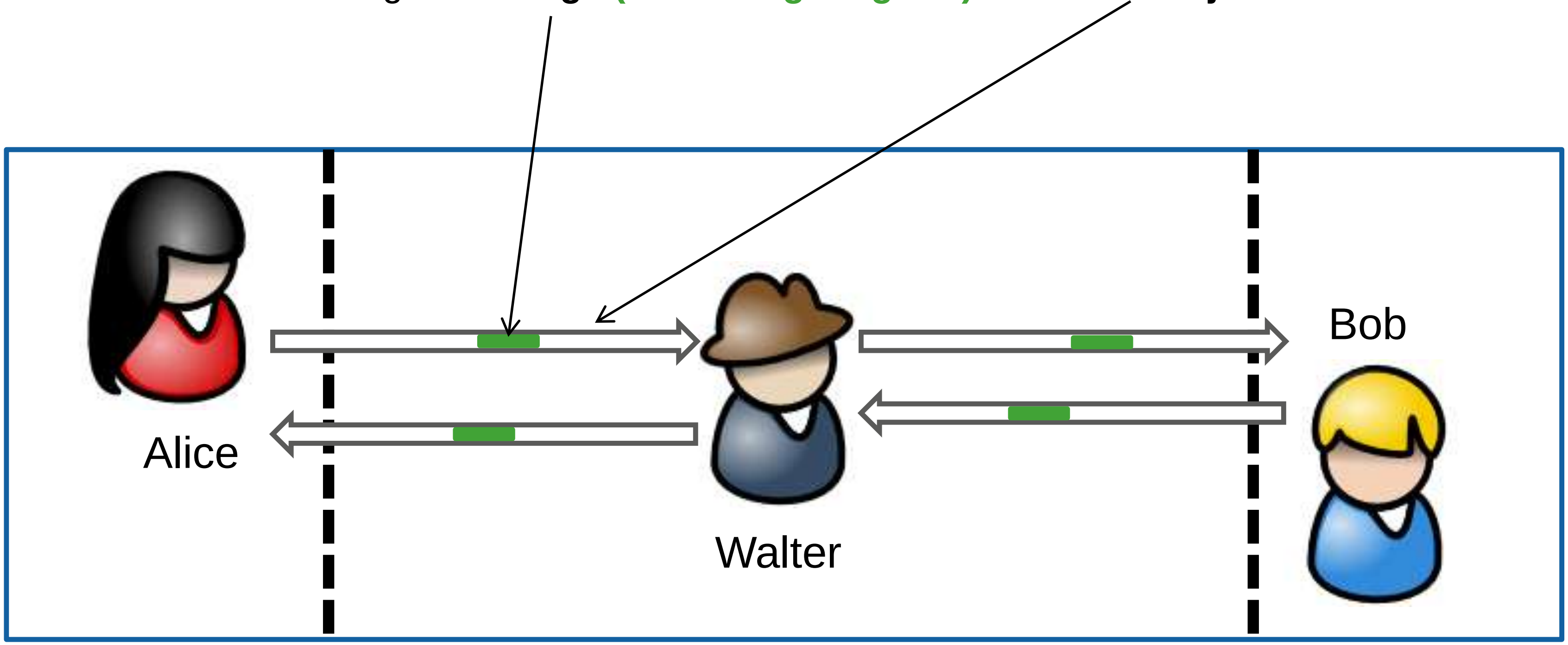
Basic Mimicry System



Basic mimicry system (Vane-Wright, 1976), Fig.: W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Wiley-IEEE, 2016

Terminology: Prisoner‘s Problem (Simmons, 1983)

- Covert Channel (Lampson, 1973): *“…not intended for information transfer at all”*
  - A covert channel without intention is a **side channel**
  - DoD defined it differently: CCs break a security policy (usually in MLS) (DoD, 1985).
- Steganography (Fridrich, 2010):
  - “Steganography can be informally defined as the practice of undetectably communicating a **message** **(a.k.a. steganogram)** in a **cover object**.“



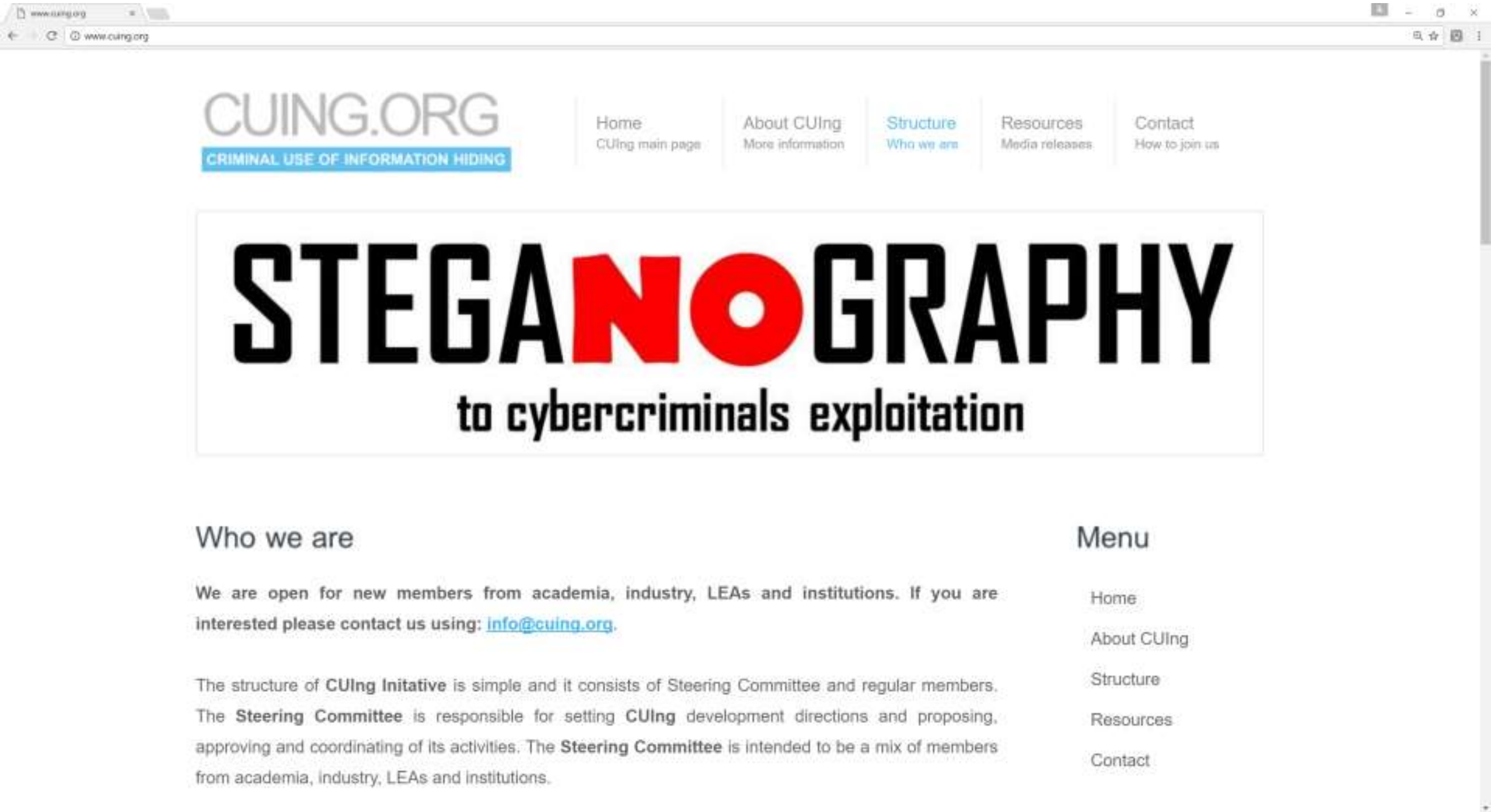
Fridrich, J.: Steganography in Digital Media, Cambridge University Press, 2010.
Lampson, B.W.: A Note on the Confinement Problem, Comm. ACM, 1973.

Is it applied in practice?

Several *recent* cases can be found in Kabaj et al.: [The new threats of information hiding: the road ahead](), IEEE IT Prof., Vol. 20(3), 2018 (Fig.).

Malware/exploit kit	Information-hiding method	Purpose
Vawtrak/Neverquest	Modification of the least-significant bits (LSBs) of favicons	Hiding URL to download a configuration file
Zbot	Appending data at the end of a JPG file	Hiding configuration data
Lurk/Stegoloader	Modification of the LSBs of BMP/PNG files	Hiding encrypted URL for downloading additional malware components
AdGholas	Data hiding in images, text, and HTML code	Hiding encrypted malicious JavaScript code
Android/Twitoor.A	Impersonating a pornography player or an MMS app	Tricking users into installing malicious apps and spreading infection
Fakem RAT	Mimicking MSN and Yahoo Messenger or HTTP conversation traffic	Hiding command and control (C&C) traffic
Carbanak/Anunak	Abusing Google cloud-based services	Hiding C&C traffic
SpyNote Trojan	Impersonating Netflix app	Tricking users into installing malicious app to gain access to confidential data
TeslaCrypt	Data hiding in HTML comments tag of the HTTP 404 error message page	Embedding C&C commands
Cerber	Image steganography	Embedding malicious executable
SyncCrypt	Image steganography	Embedding core components of ransomware
Stegano/Astrum	Modifying the color space of the used PNG image	Hiding malicious code within banner ads
DNSChanger	Modification of the LSBs of PNG files	Hiding malware AES encryption key
Sundown	Hiding data in white PNG files	Exfiltrating user data and hiding exploit code delivered to victims

Is it applied in practice?



Some potential scenarios

- Advanced Persistent Threats (APT): large-scale sophisticated data leakage, applying techniques such as `spear phishing'
- Malware: e.g. stealthy botnet C&C channels
- Military/secret service: Industrial espionage, stealthy communication
- Citizens: censorship circumvention
- Journalists: freedom of speech -> expression of opinions in networks with censorship

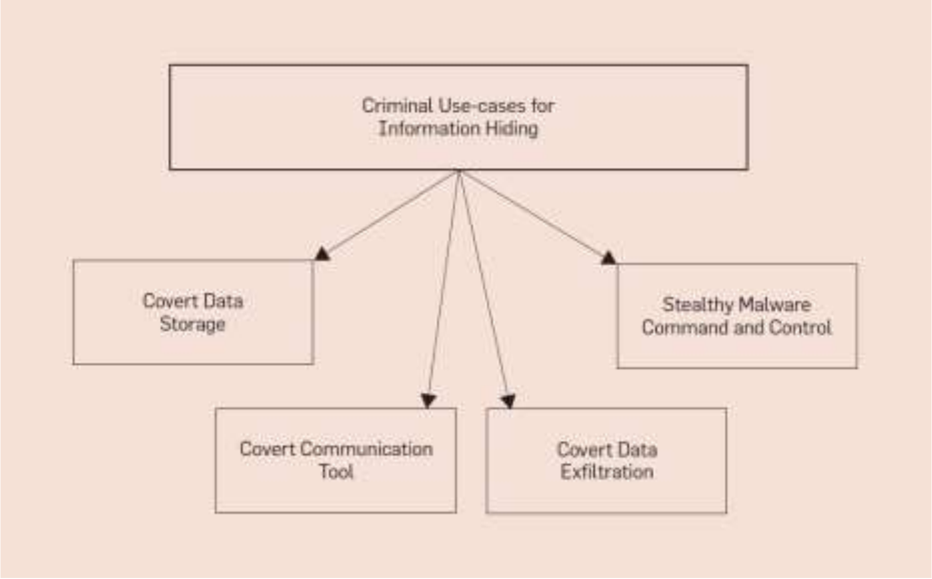
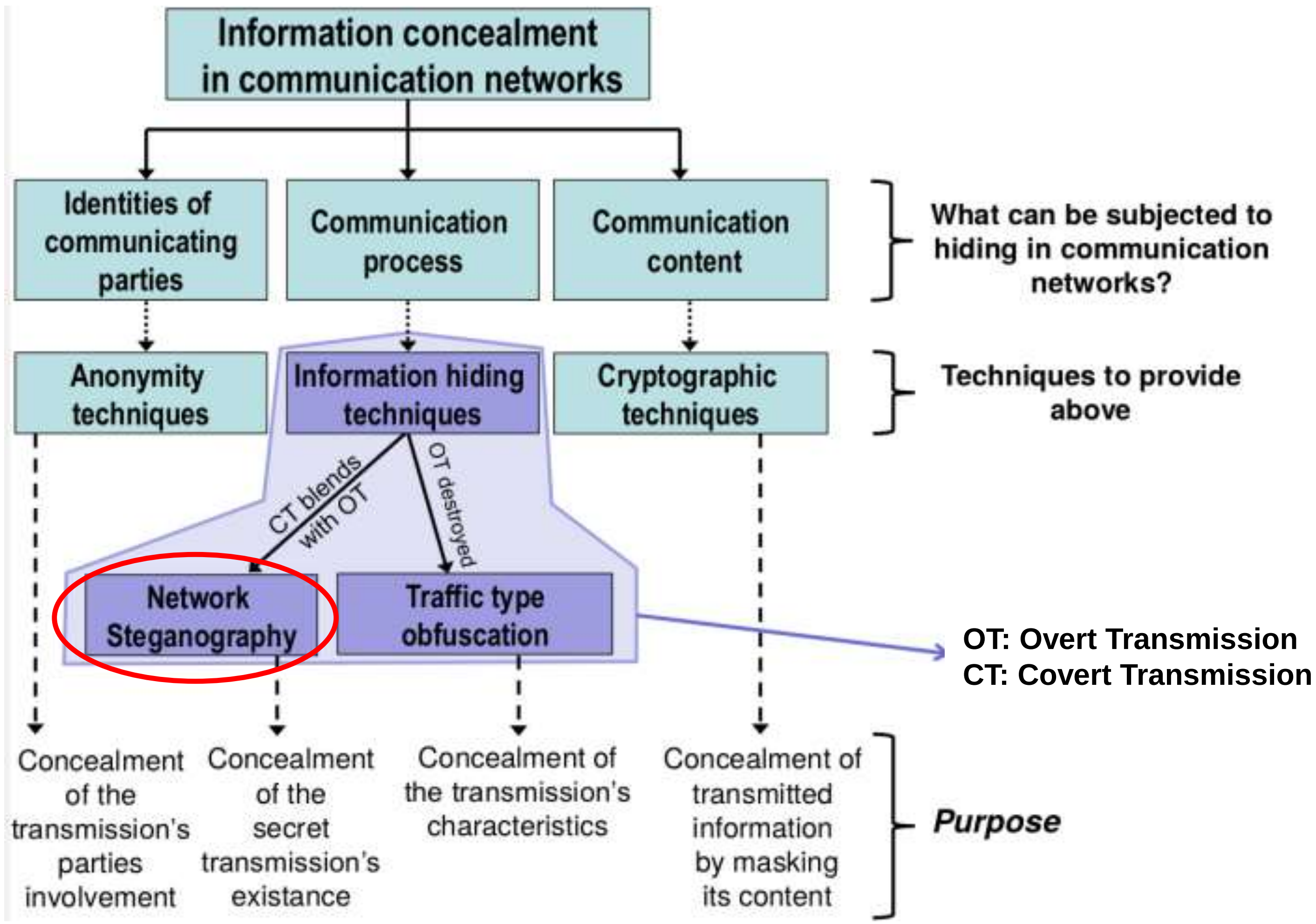


Fig.: Mazurczyk/Wendzel: Information Hiding: Challenges for Forensic Experts, Communications of the ACM, 2018. [link]

Network Information Hiding



OT: Overt Transmission
CT: Covert Transmission

Fig.: W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Wiley-IEEE, 2016

Section based mostly on S. Wendzel et al.: Pattern-based Survey of Network Covert Channel Techniques, ACM CSUR, 47(3), 2015.

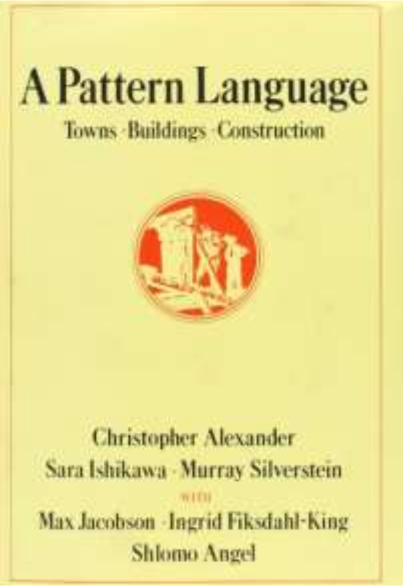
HIDING PATTERNS

(IMPROVING SCIENTIFIC FUNDAMENTALS OF NETWORK INFORMATION HIDING)

Patterns

- What are „**Patterns**“?
  - A solution to a re-occurring problem in a given context
  - They are re-usable and described in an abstract way

- Term introduced by Alexander *et al.* in 1977 for Architecture
- He presented a „pattern language“ comprising 253 patterns



- **Example:**
  - Problem: want to minimize artificial light
  - Context: saving energy
  - Solution: build a window into a building to receive as much sunlight as possible in that room.

Comments on Patterns

- A technique can only be a pattern **if it occurs multiple times**. In general, the scientific patterns community agrees on the minimal number of <u>three</u> occurrences.

- **Pattern collections** comprise patterns of a given domain. They can be understood as **pattern catalogs*** (but the latter is additionally searchable, e.g. by an index of patterns).
  - e.g., a collection of user interface patterns
  - Problematic aspect: the link-ability of patterns between collections differs due to non-unified structures in which the patterns are described.

* Terminology not unified in the literature. We can agree on collection==catalog for this lecture.

Pattern Languages

- **Pattern languages** were introduced to solve the mentioned problems of pattern collections:
  - they provide a unified description for patterns
  - allow to build links/hierarchies between patterns
  - introduce aliases to prevent redundancies

- **PLML** (Pattern Language Markup Language) is one dominating example of a pattern language.

PLML

- PLML allows the description of patterns (e.g. in XML); its development is ongoing.
- Patterns comprise various elements (attributes of PLML/1.1*):

Pattern Identifier	Name
Alias	Illustration
Description of the Problem	Description of the Context
Description of the Solution	Forces
Synopsis	Diagram
Evidence	Confidence
Literature	Implementation
Related Patterns	Pattern Links
Management Information	

* Newer version of PLML is available but the basic attributes remain. Not all attributes of the above table are used (are necessary) to describe hiding patterns.

Hiding Patterns

Hiding Patterns describe they **key idea of hiding techniques**. They are kept on an **abstract, non-detailed level**, help **cleaning up terminology**, and can **form a taxonomy**.

S. Wendzel, S. Zander et al.: Pattern-based Survey of Network Covert Channel Techniques, ACM CSUR, 47(3), 2015.

The following attributes were used

Table I. Used PLML/1.1 Attributes

Tag	Description
<pattern id>	Identifies a pattern within the particular catalog.
<name>	A correct assignment of a name for each pattern is important for the retrieval of a pattern when the pattern becomes part of a second catalog.
<alias>	Patterns can have different names, which are specified in the <alias> tag. The alias tag helps to find the same pattern when the pattern has different names in different catalogs.
<illustration>	An application scenario for the pattern.
<context>	Specifies the situations to which the pattern can be applied.
<solution>	Describes the solution for a problem to which the pattern can be applied. The attributes *problem* and *context* (cf. Fig. 1) are usually blurred but often not separated into two attributes.
<evidence>	Contains additional details about the pattern and its design. Moreover, the tag can contain examples for known uses of the pattern.
<literature>	Lists references to publications related to the pattern.
<implementation>	Introduces existing implementations, code fragments or implementational.

Image source: (Wendzel et al., 2015)

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

Patterns in Network Information Hiding

- Approx. 150 network hiding techniques exist; they hide secret information in meta data of network traffic.
  - Inconsistent terminology.
  - Re-inventions very common.

- Instead of dealing with all these hiding techniques separately, we only need to understand the few hiding patterns.

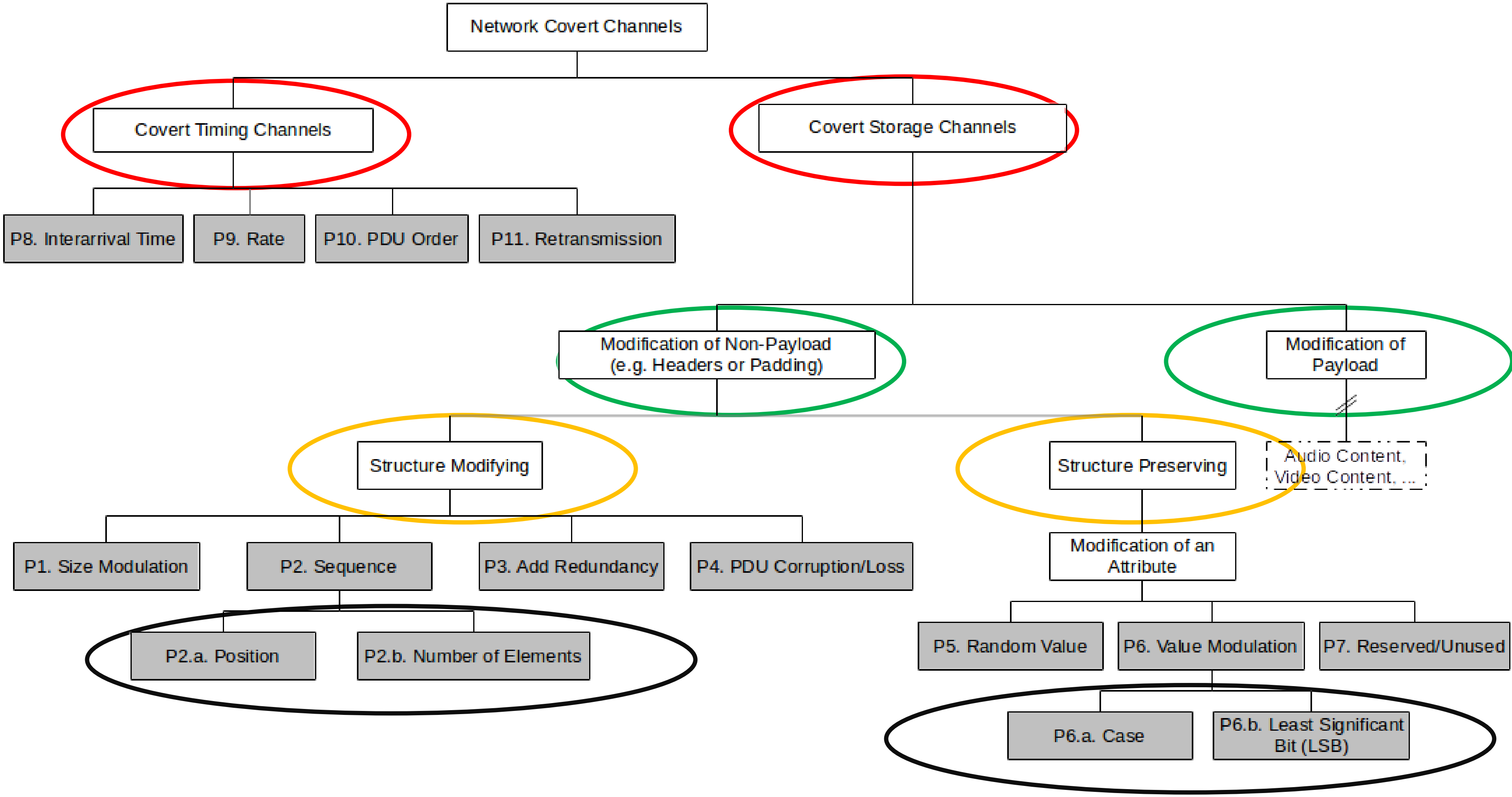
- **Eleven** (later a few more) patterns were found to describe all analyzed hiding techniques published between 1987 and 2015.

- Also, patterns provide better taxonomies due to their several features (links and child patterns, alias handling, unified attributes, …).

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

Patterns in Network Information Hiding

Patterns were set in relation to other patterns to introduce a **new taxonomy** of patterns. The 109 hiding techniques could be described by only 11 patterns.



S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P1. Size Modulation Pattern

- The overt channel uses the size of a header element or of a PDU* to encode the hidden message.

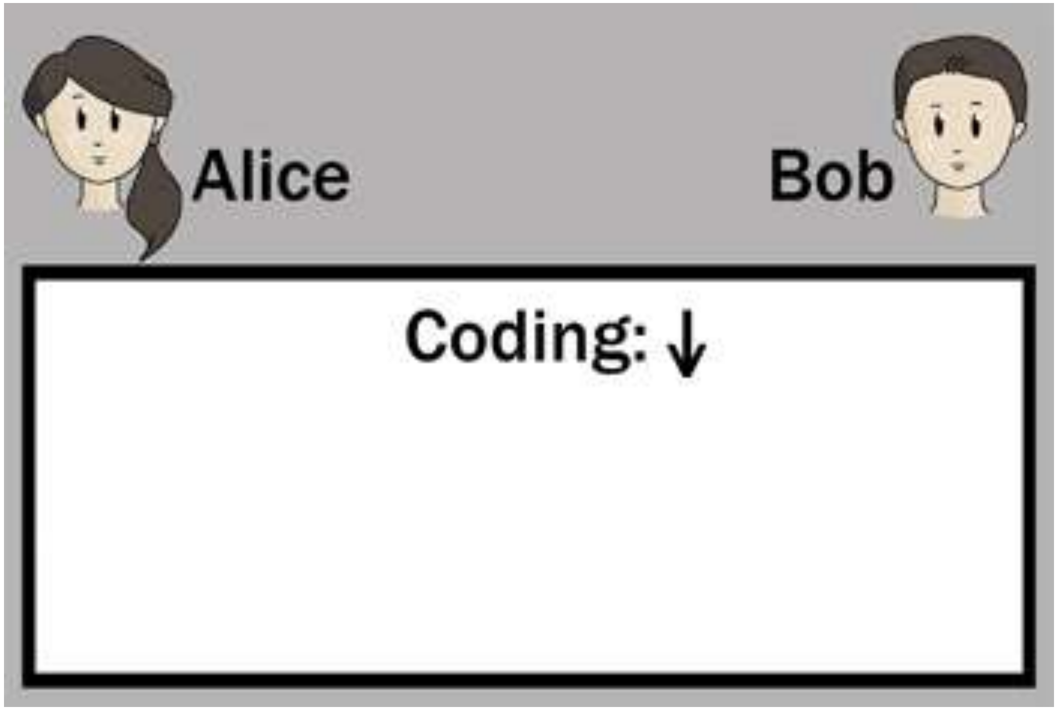


Image: J. Kammerlander.

*protocol data unit

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P1. Size Modulation Pattern

- Examples:
  - Modulation of data block length in LAN frames
  - Modulation of IP fragment sizes

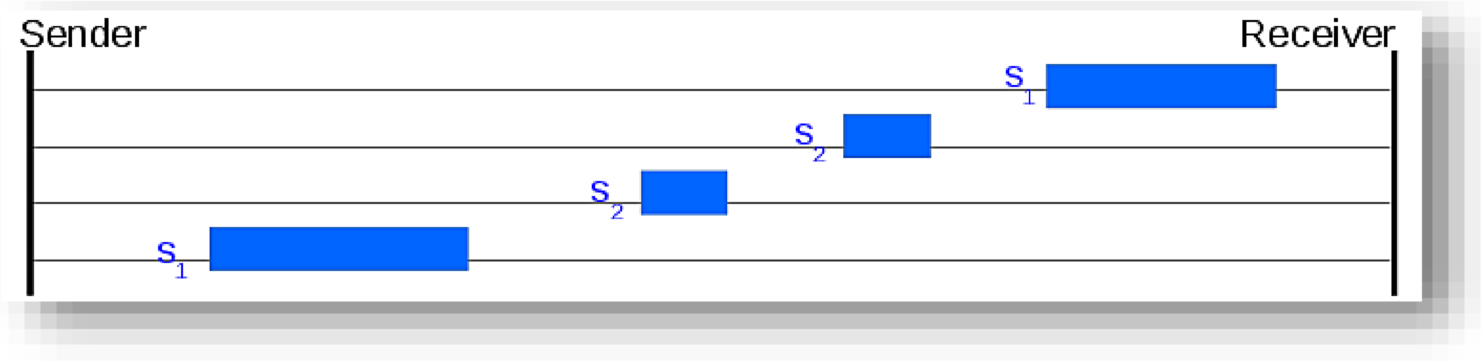


Image source: (Mazurczyk et al., 2016)

P2. Sequence Pattern

- The covert channel alters the sequence of header/PDU elements to encode hidden information.

- Examples:
  - Sequence of DHCP options
  - Sequence of FTP commands
  - Sequence of HTTP header fields

```
GET HTTP/1.1
Host: mywebsite.xyz
User-Agent: MyBrowser/1.2.3   } S1
Accept-Language: en-US
```

```
GET HTTP/1.1
Host: mywebsite.xyz
Accept-Language: en-US        } S2
User-Agent: MyBrowser/1.2.3
```

Image source: (Mazurczyk et al., 2016)

- Sub-patterns:
  - P2.a. Position Pattern (e.g. pos. of IPv4 option *x* in list of options)
  - P2.b. Number of Elements Pattern (e.g. # of IPv4 options)

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P3. Add Redundancy Pattern

- The covert channel creates new space within a given header element or within a PDU to hide data in it.

- Examples:
  - Extend HTTP headers with additional fields or extend values of existing fields

```
GET / HTTP/1.0          GET / HTTP/1.0
                        User-Agent: Mozilla/4.0
```

  - Create a new IPv6 destination option with embedded hidden data
  - Manipulate \`pointer‘ and \`length‘ values for IPv4 record route option to create space for data hiding

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P4. PDU Corruption

- The covert channel generates corrupted PDUs that contain hidden data or actively utilizes packet loss to signal hidden information.

- Examples:
  - Transfer corrupted frames in IEEE 802.11
  - MitM drops selected packets exchanged between two VPN sites to introduce covert information.

  E.g., sending a number of packets of which corrupted packets indicate hidden data:

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P5. Random Values

- The covert channel embeds hidden data in a header element containing a „random“ value.

- Examples:
  - Utilize IPv4 identifier field
  - Utilize the ISN of a TCP connection (cf. previous lecture on IH)
  - Utilize DHCP *xid* field

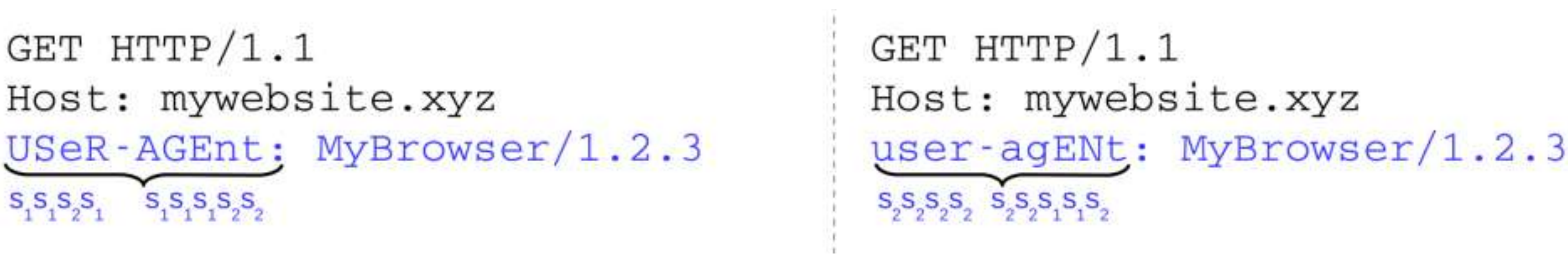
S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P6. Value Modulation Pattern

- The covert channel selects one of *n* values a header element can contain to encode a hidden message.

- Examples:
  - Send a frame to one of *n* available Ethernet addresses in a LAN
  - Encode information by the possible Time-to-live (TTL) values in IPv4 or in the Hop Limit values in IPv6

- Sub-patterns:
  - P6.a. Case pattern: case modification of letters in plaintext headers (e.g. SMTP command letter cases)
  - P6.b. LSB pattern: modify low order bits of header fields (e.g. TCP timestamp option)



S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P7. Reserved/Unused Pattern

- The covert channel encodes hidden data into a reserved or unused header/PDU element.
- Examples:
  - Utilize undefined/reserved bits in IEEE 802.5/data link layer frames
  - Utilize (currently) unused fields in IPv4, e.g. Identifier field, Don‘t Fragment (DF) flag or reserved flag or utilize unused fields in IP-IP encapsulation
  - Utilize the padding field of IEEE 802.3

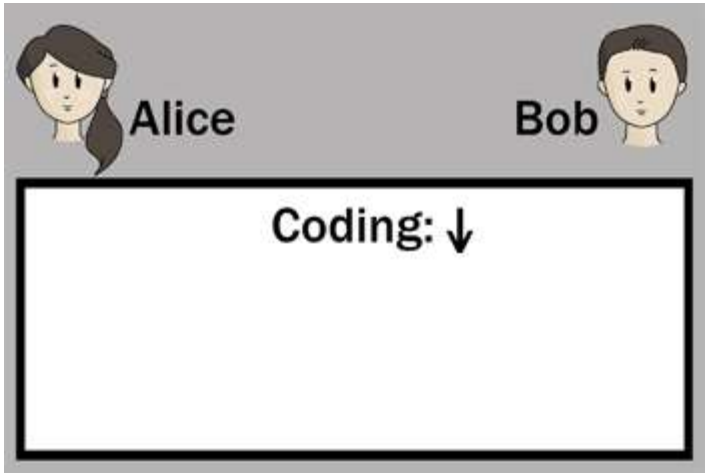


Image: J. Kammerlander.

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P8. Inter-arrival Time Pattern

- The covert channel alters timing intervals between network PDUs (inter-arrival times) to encode hidden data.

- Examples:
  - Alter timings between LAN frames
  - Alter the response time of a HTTP server

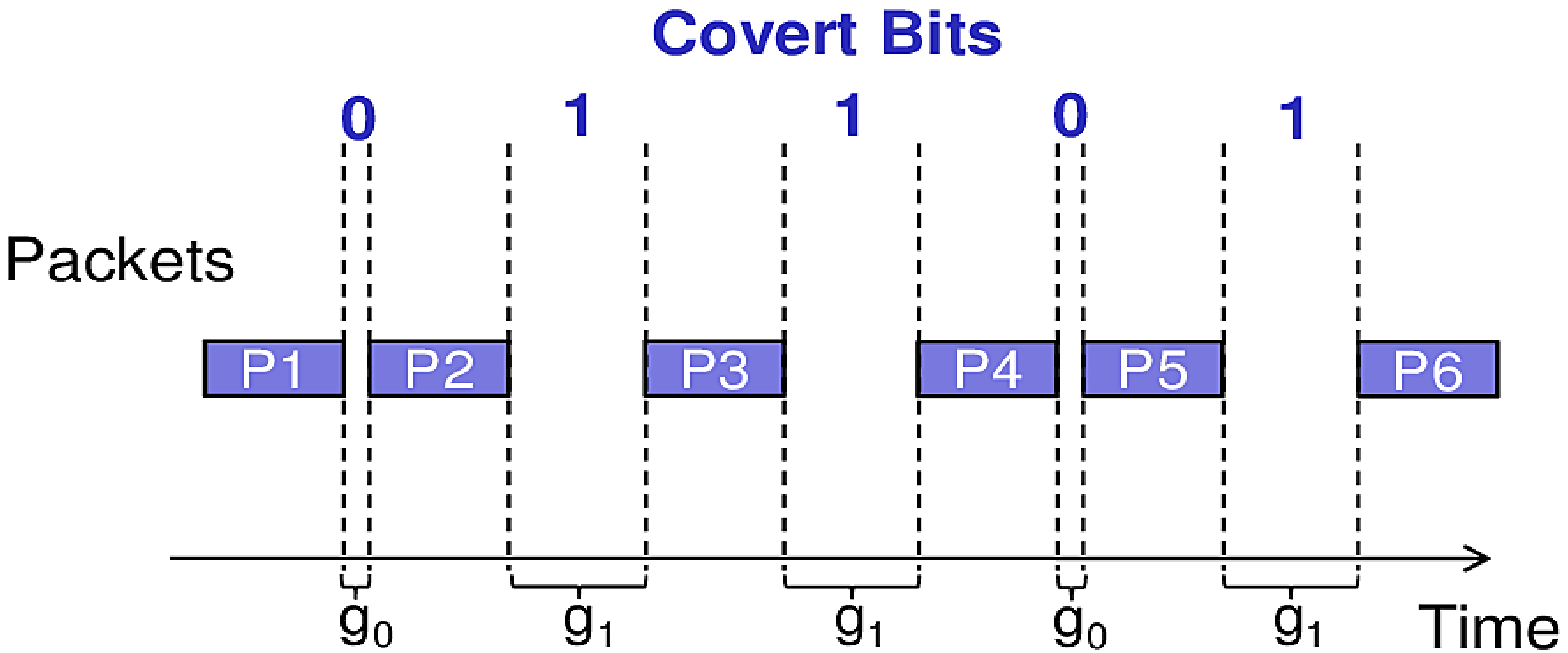


Image source: (Mazurczyk et al., 2016)

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

P9. Rate Pattern

- The covert channel sender alters the data rate of a traffic flow from itself or a third party to the covert channel receiver.
- Examples:
  - Exhaust the performance of a switch to affect the throughput of a connection from a third party to a covert channel receiver over time.
  - Directly alter the data rate of a legitimate channel between a covert channel sender and receiver.

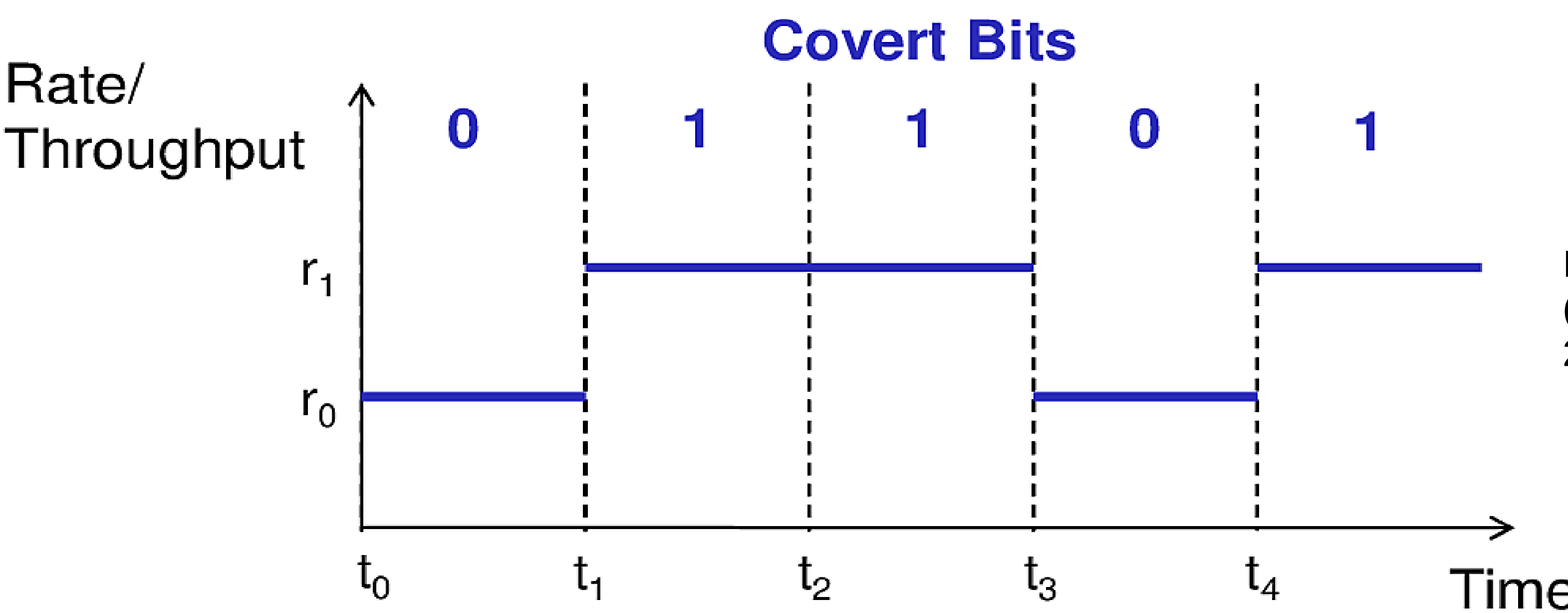


Image source: (Mazurczyk et al., 2016)

P10. PDU Order Pattern

- The covert channel encodes data using a synthetic PDU order for a given number of PDUs flowing between covert sender and receiver.

- Examples:
  - Modify the order of IPSec Authentication Header (AH) packets
  - Modify the order of TCP packets

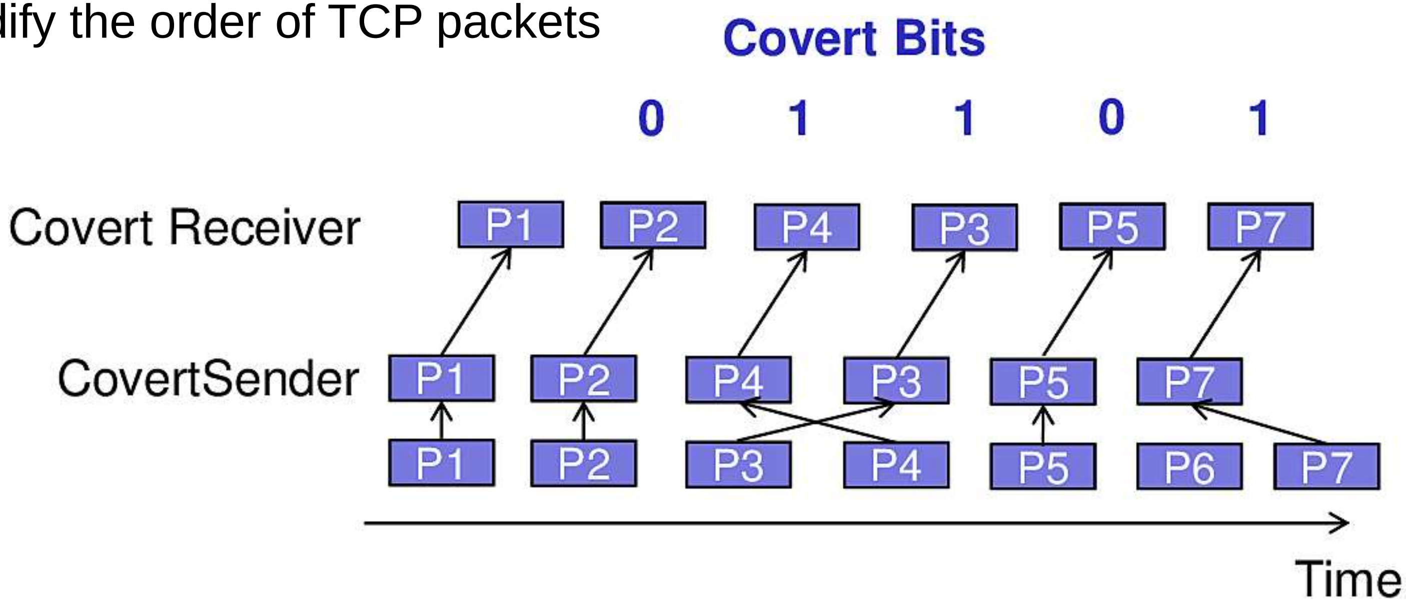


Image source: (Mazurczyk et al., 2016)

P11. Re-transmission Pattern

- A covert channel re-transmits previously sent or received PDUs.

- Examples:
  - Transfer selected DNS requests once/twice to encode a hidden bit per request.
  - Duplicate selected IEEE 802.11 packets
  - Do not acknowledge received packets to force the sender to re-transmit a packet.

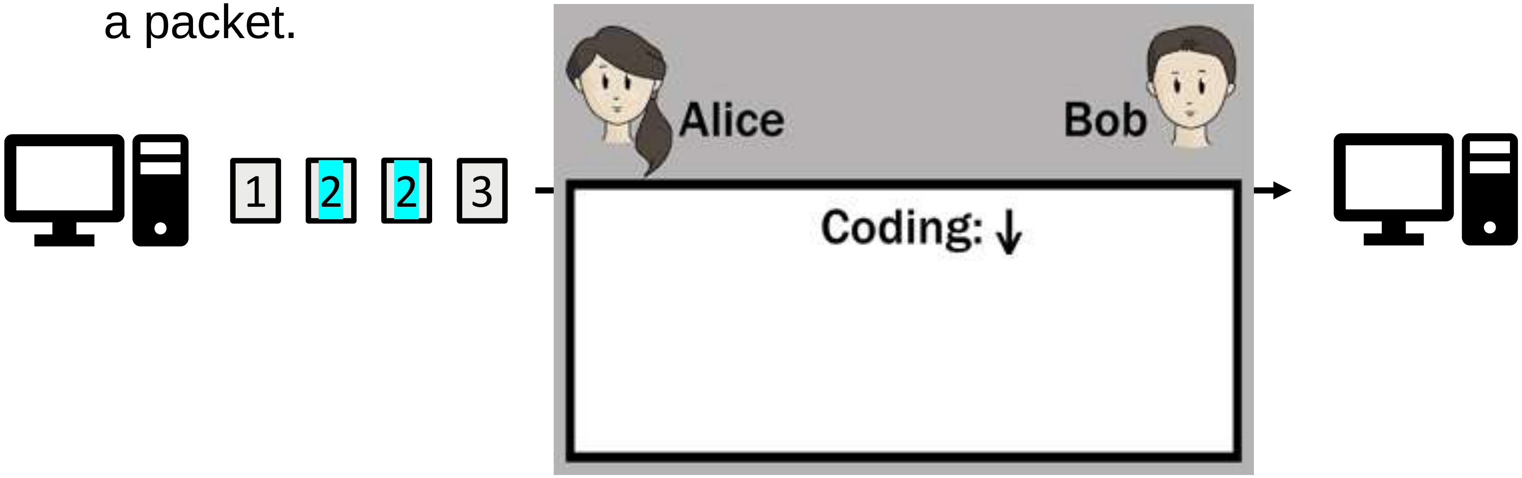


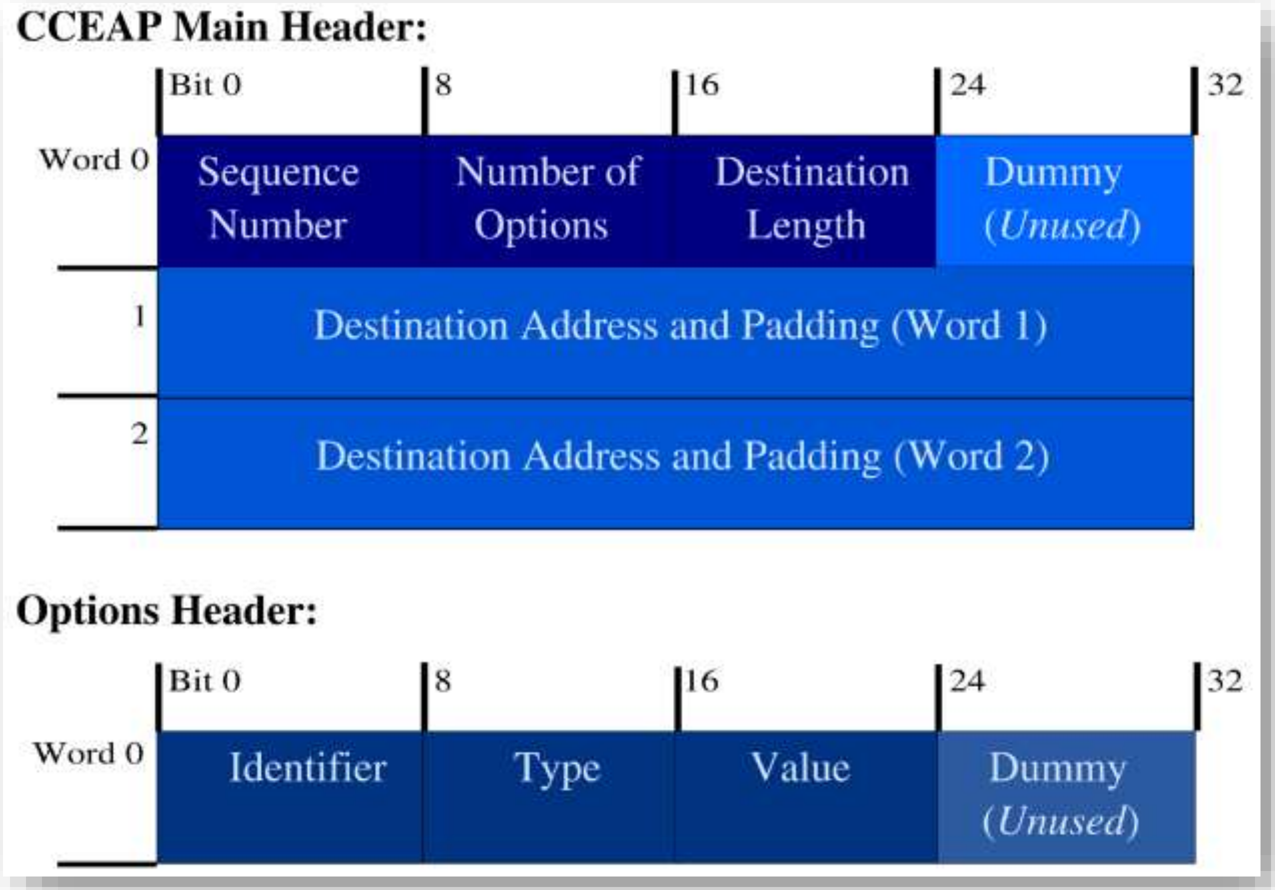
Image: J. Kammerlander.

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

CCEAP

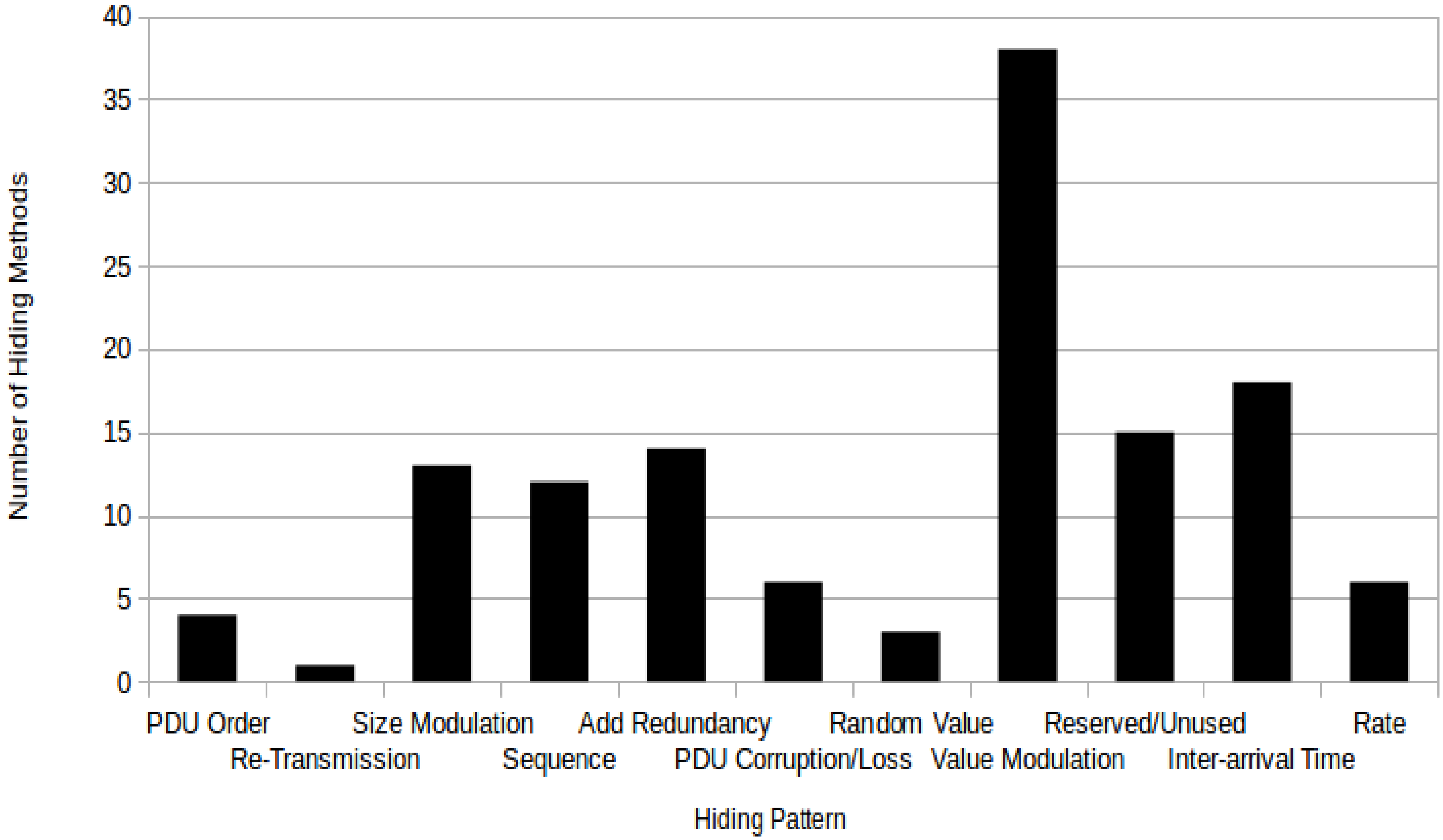
CCEAP is a tool for learning hiding patterns, available from Github.

- GUI is on the way.
- Sample exercises + solutions can be found here.
- There is also a poster.



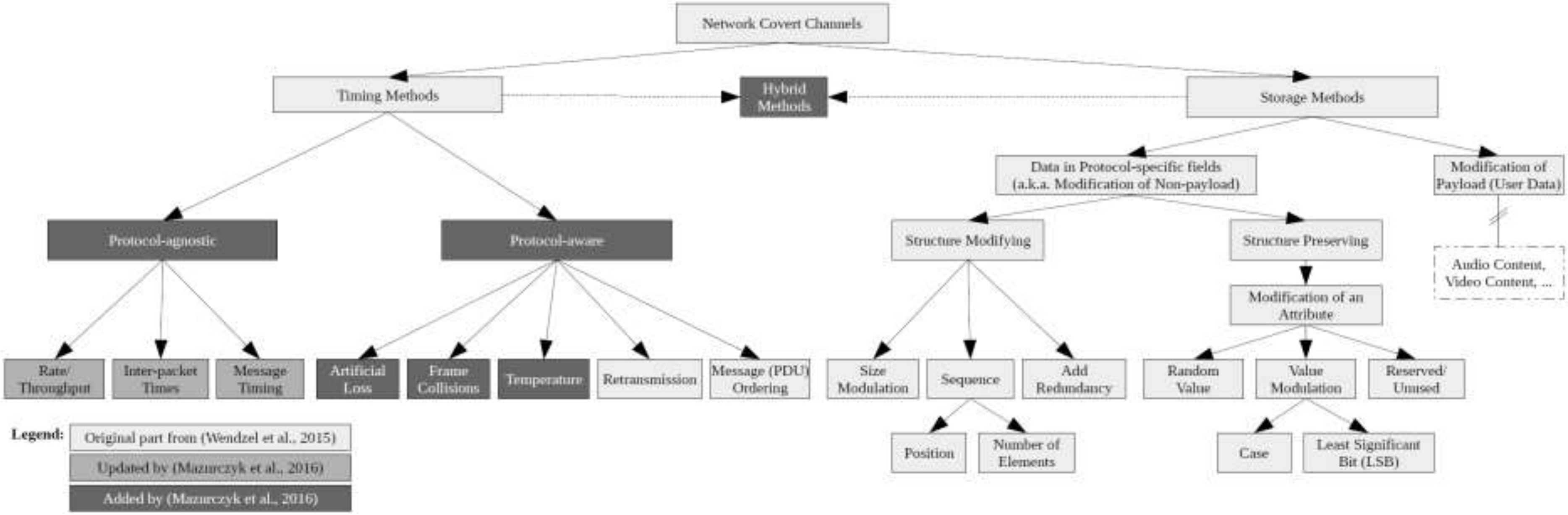
S. Wendzel, W. Mazurczyk: Poster: An Educational Network Protocol for Covert Channel Analysis Using Patterns, Proc. ACM CCS, 2016.

Published Hiding Techniques



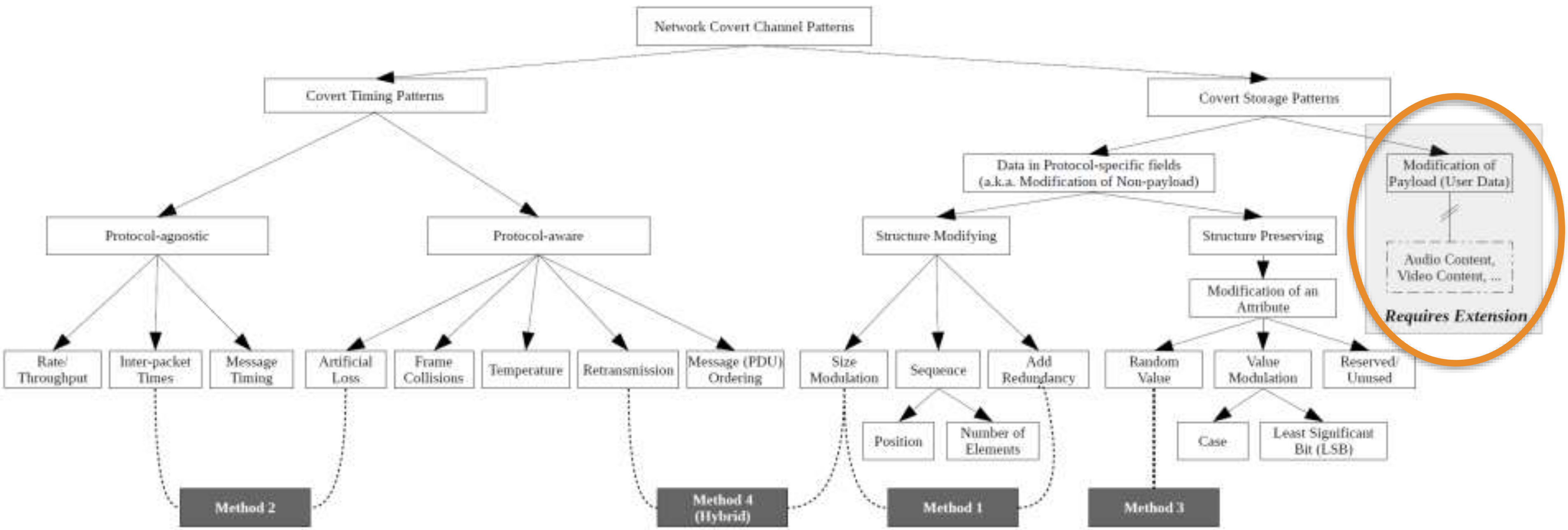
S. Wendzel et al. [Unified Description for Network Information Hiding Methods](), in: Journal of Universal Computer Science, 2016.

2016 Taxonomy Add-on



W. Mazurczyk, S. Wendzel, K. Cabaj: [Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach](), ARES'18.

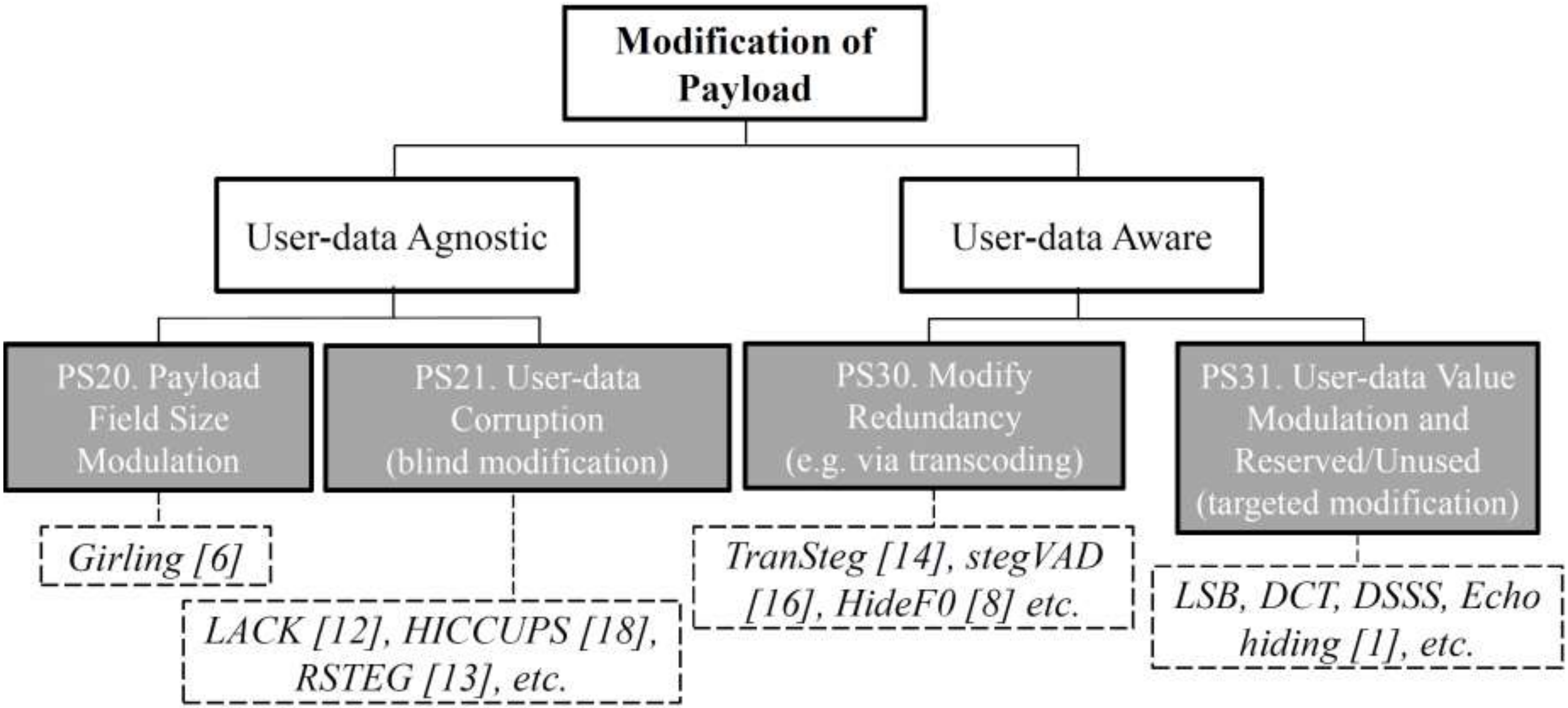
This Enables Hybrid Methods … but what about the payload?



W. Mazurczyk, S. Wendzel, K. Cabaj: [Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach](), ARES'18.

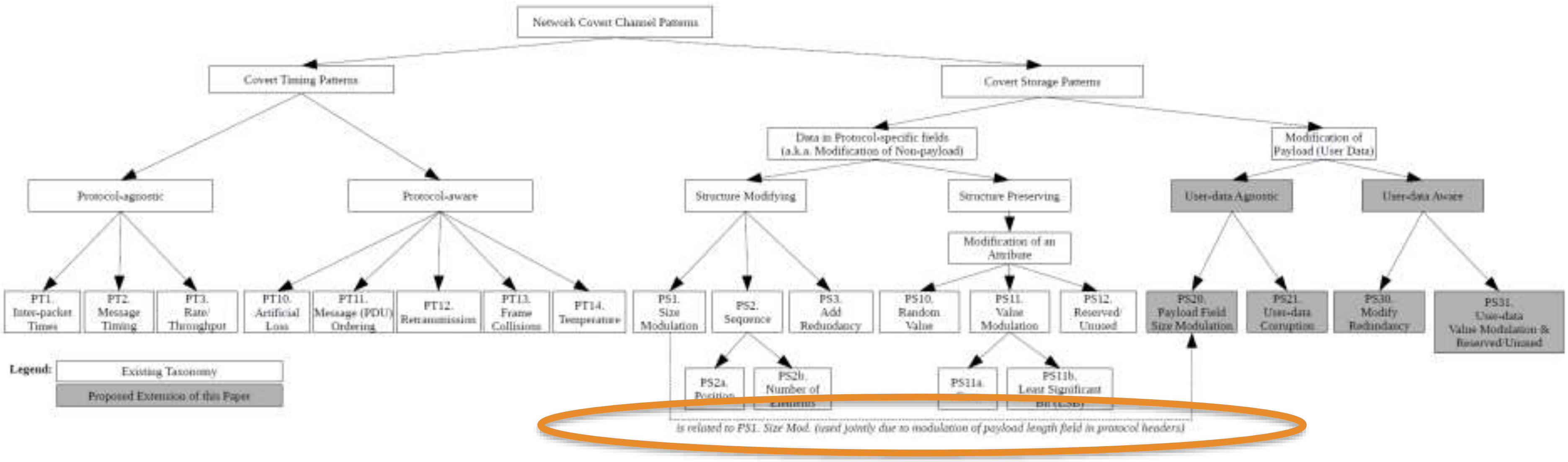
Patterns for Payload Modification

(Network-level View, not Digital Media Steganography)



W. Mazurczyk, S. Wendzel, K. Cabaj: Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach, ARES'18.

Proposal for Taxonomy Extension



PS20 is a derivate of PS2 (allowed by PLML).

W. Mazurczyk, S. Wendzel, K. Cabaj: [Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach](), ARES'18.

Pattern Variation

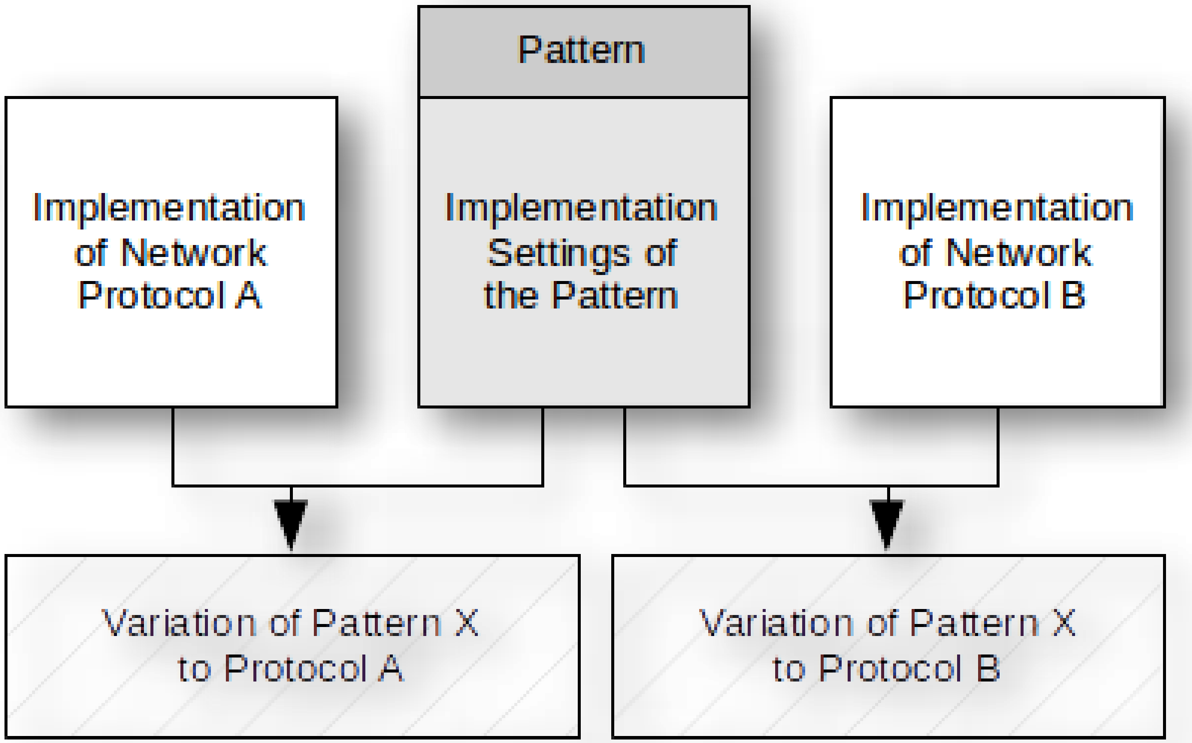
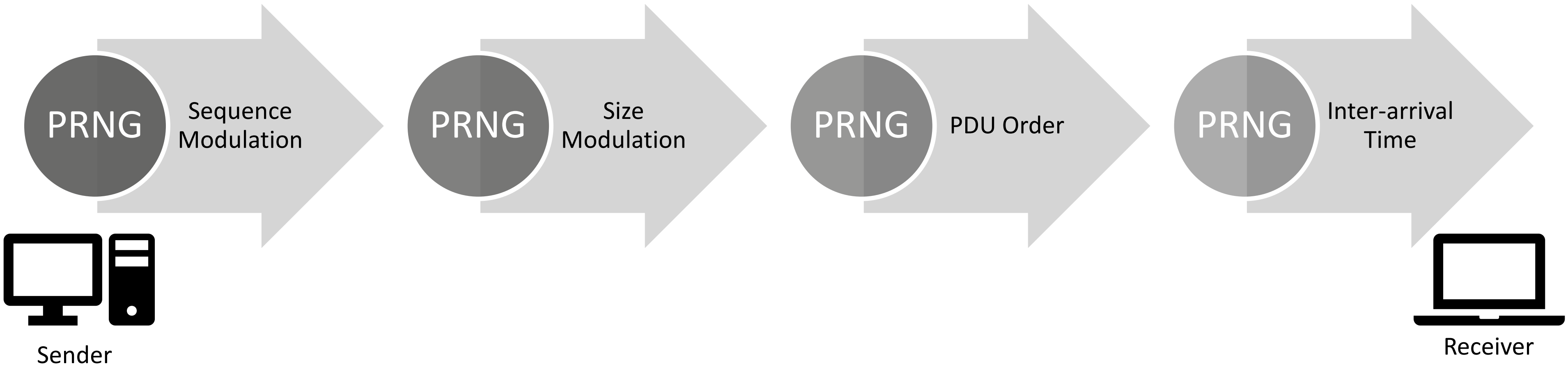


Image source: (Wendzel et al., 2015)

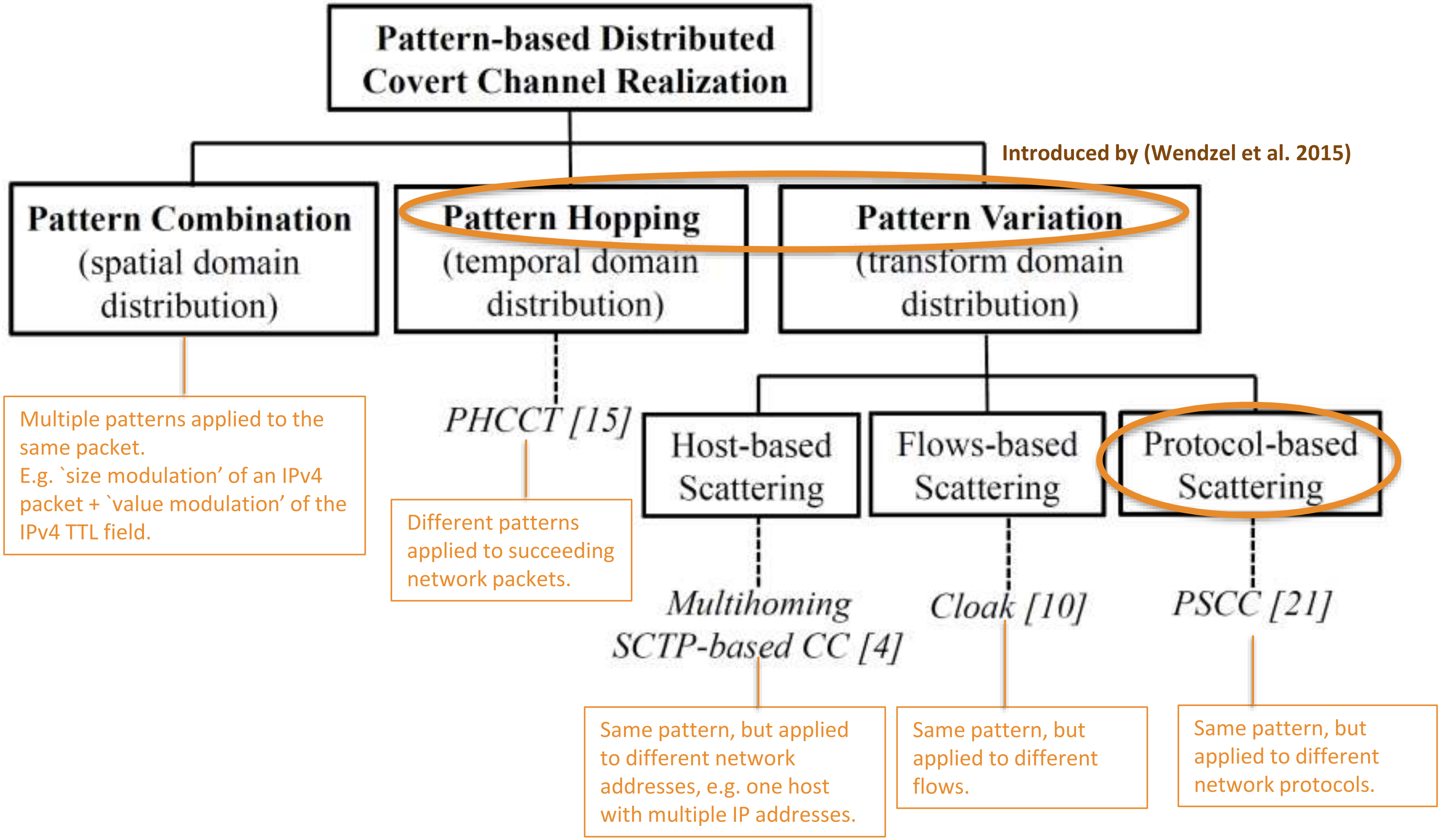
S. Wendzel et al.: [Pattern-based Survey and Taxonomy for Network Covert Channels](), ACM CSUR, Vol. 47(3), 2015.

Pattern Hopping



S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

Distributed Hiding Methods



W. Mazurczyk, S. Wendzel, K. Cabaj: Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach, ARES'18.

PATTERN-BASED COUNTERMEASURES

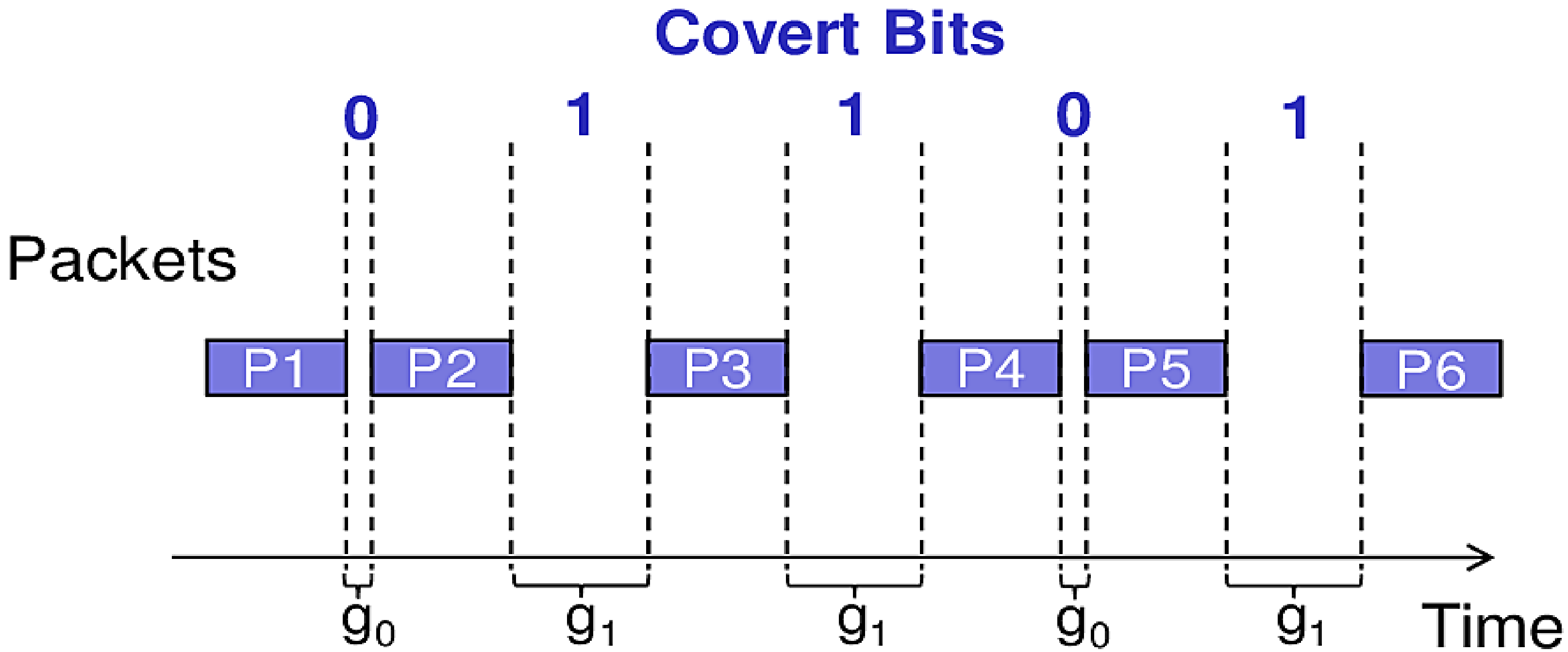
Prevention/Elimination

Limitation

Detection

Several methods exist …

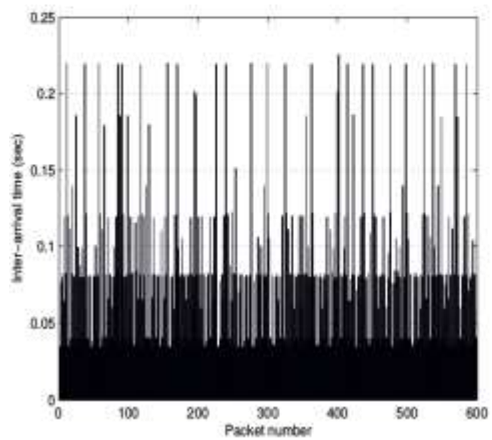
- cf. (Mazurczyk et al., 2016, Chapter 8) for an overview
- Today, we will consider only two of the **Inter-arrival Times Pattern**.

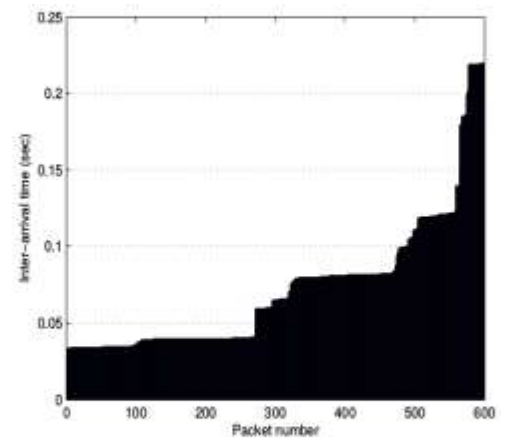


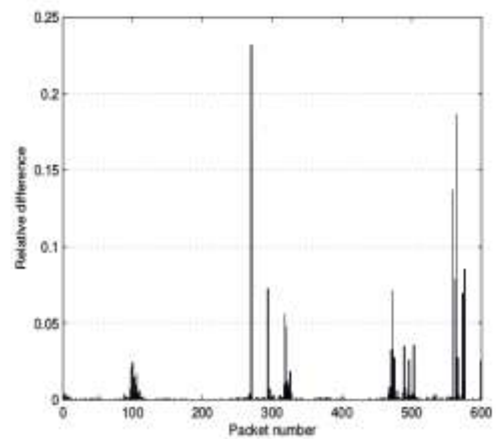
P8. Inter-arrival Time Pattern Detection: ε**-similarity**

In a nutshell:

1. Sort all inter-packet times of a flow.
2. For consecutive values T_i and T_{i+1}: calculate relative difference $\lambda_i = \frac{|T_{i+1} - T_i|}{T_i}$.
3. Calculate the percentage of λ values of a given flow that are below the threshold ε.





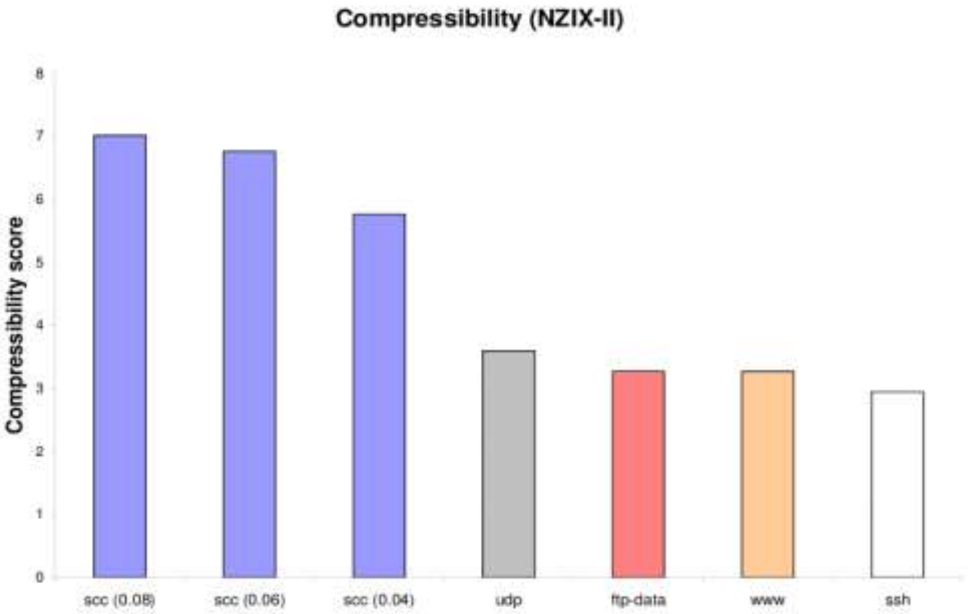


S. Cabuk et al.: IP Covert Channel Detection, in: Transactions on Information and System Security (TISSEC), ACM, 2009.

P8. Inter-arrival Time Pattern Detection: **Compressibility Score**

In a nutshell:

1. Record all inter-packet times of a flow $\Delta_{t_1}, \ldots, \Delta_{t_n}$.
2. Encode the inter-packet times in an ASCII string S, e.g. “A20A20A19B30B29A20...”.
3. Compress S with a compressor $\mathfrak{I}$ (e.g. *gzip*): $C = \mathfrak{I}(S)$.
4. Use $\kappa = \frac{|S|}{|C|}$ as an indicator for the presence of a covert channel.



S. Cabuk et al.: IP covert timing channels: design and detection, in Proc. 11th ACM CCS, 2004.
Fig.: S. Cabuk et al.: IP Covert Channel Detection, in: Transactions on Information and System Security (TISSEC), ACM, 2009.

2015-overview of potential counter-measures in combination with patterns

Table III. Application of Covert Channel Countermeasures to Patterns

	Elimination	Limitation	Detection
Storage Channel Patterns			
P1. Size Modulation			SA/ML
P2. Sequence	TN		SA/ML
P2.a. Position	TN		SA/ML
P2.b. Number of Elements	TN		SA/ML
P3. Add Redundancy	TN		SA/ML
P4. PDU Corruption/Loss	TN		SA/ML
P5. Random Value	TN		SA/ML
P6. Value Modulation		TN (limited), NPRC	SA/ML
P6.a. Case	TN		SA/ML
P6.b. LSB	TN		SA/ML
P7. Reserved/Unused	TN		SA/ML
Timing Channel Patterns			
P8. Interarrival Time		TN (limited), NPRC	SA/ML
P9. Rate		TN (limited), NPRC	SA/ML
P10. PDU Order		TN (limited) NPRC	SA/ML
P11. Retransmission			SA/ML

TN: Traffic Normalization
NPRC: Network Pump and Related Concepts
SA/ML: Statistical Approaches/Machine Learning

S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

Countermeasure Variation

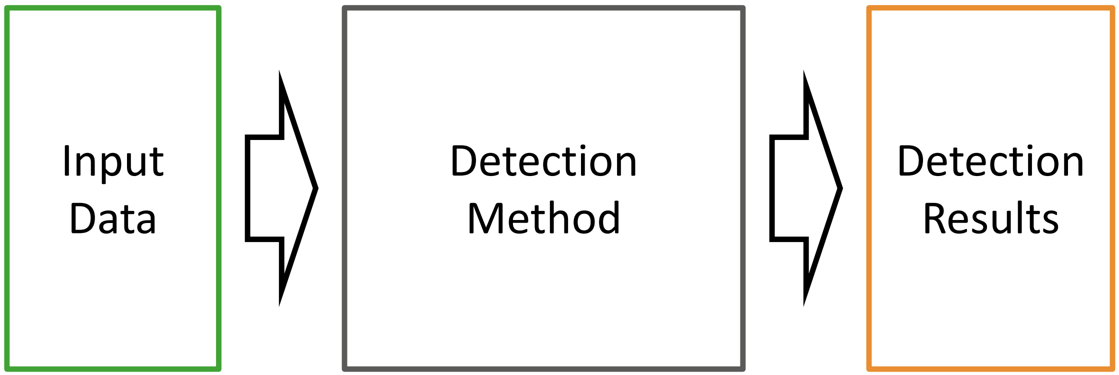
Problem: We lack countermeasures for several of the known patterns.

Solution: Introduction of **countermeasure variation**.

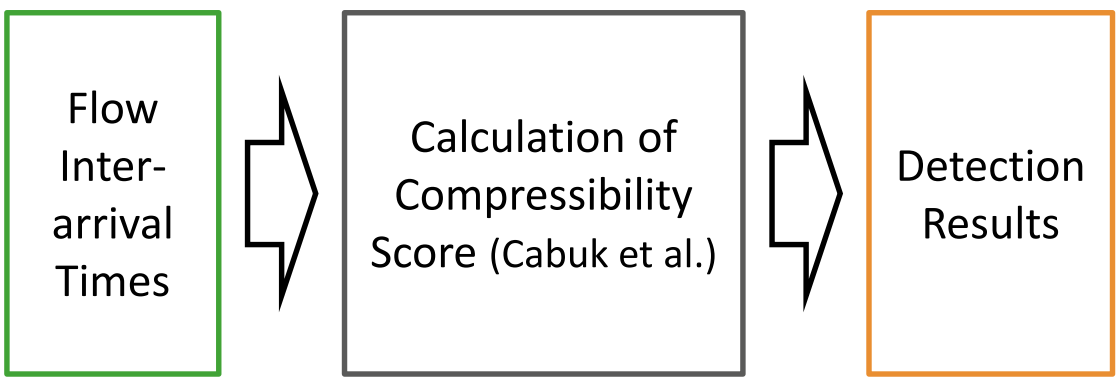
S. Wendzel, D. Eller, W. Mazurczyk: *One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels*, in Proc. CECC'18, ACM, 2018.

Countermeasure Variation

Classic covert channel countermeasures look like this:



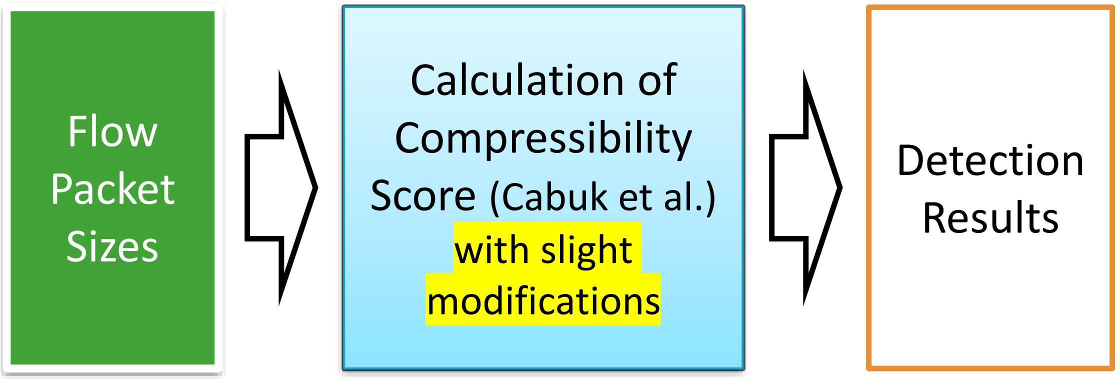
For instance:



S. Wendzel, D. Eller, W. Mazurczyk: *One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels*, in Proc. CECC'18, ACM, 2018.

Countermeasure Variation

Countermeasure Variation modifies the input to the detection method and alters the detection method as little as possible.



S. Wendzel, D. Eller, W. Mazurczyk: *One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels*, in Proc. CECC'18, ACM, 2018.

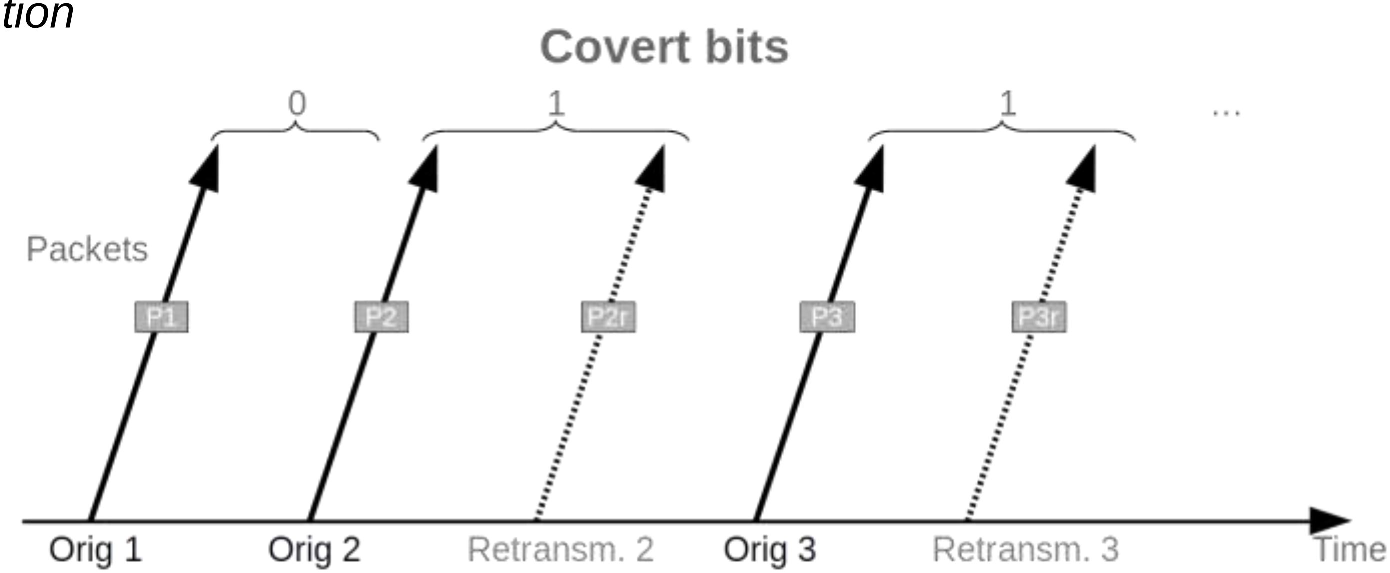
Countermeasure Variation

So far, we performed countermeasure variation for

- Compressibility Score
- ε-similarity
- *Regularity*

Each in combination with the following patterns:

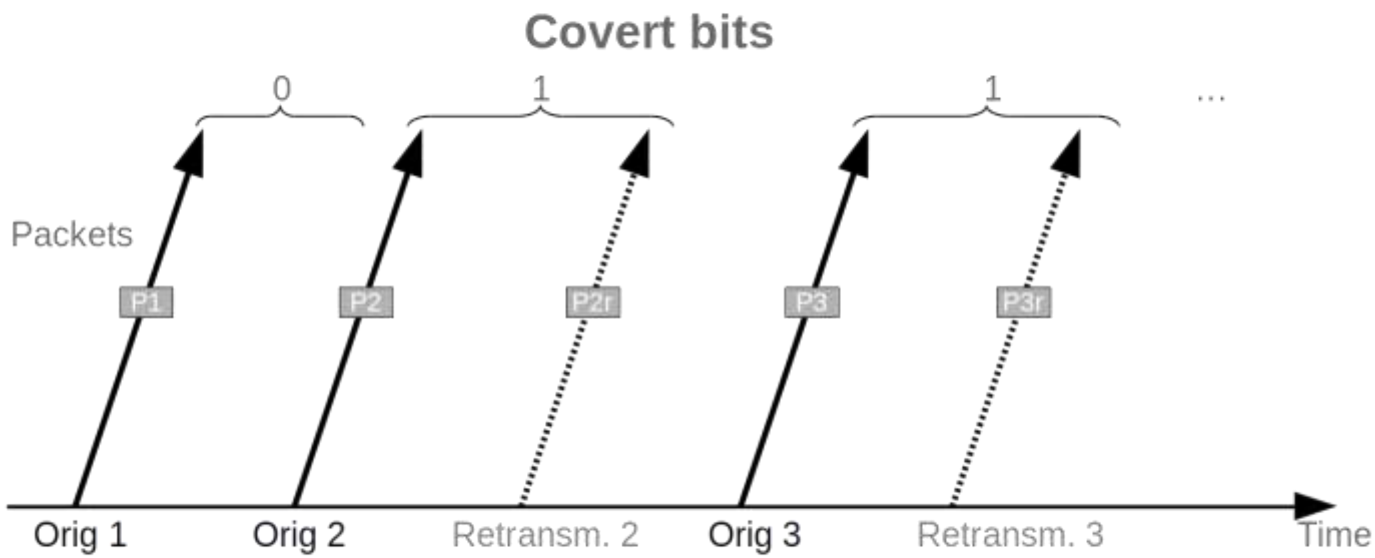
- Size Modulation
- Artificial Re-transmission
- *Sequence Modulation*
- *Value Modulation*



S. Wendzel, D. Eller, W. Mazurczyk: *One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels*, in Proc. CECC'18, ACM, 2018.
S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

- Using TCP re-transmissions
- To match traffic patterns, we
  - studied typical re-transmissions of Internet traffic (different routes; repeated measurements several times for each route; at different days/hours), and
  - adjusted and optimized our CC to legitimate traffic's characteristics (very low transmission rate to increase covertness; robust coding).



S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

ε-similarity

Input modifications:

Succeeding retransmission's sequence numbers

Modification of detection algorithm:

- Adjust thresholds for detection.

Compressibility

Input modifications:

Succeeding retransmission's sequence numbers

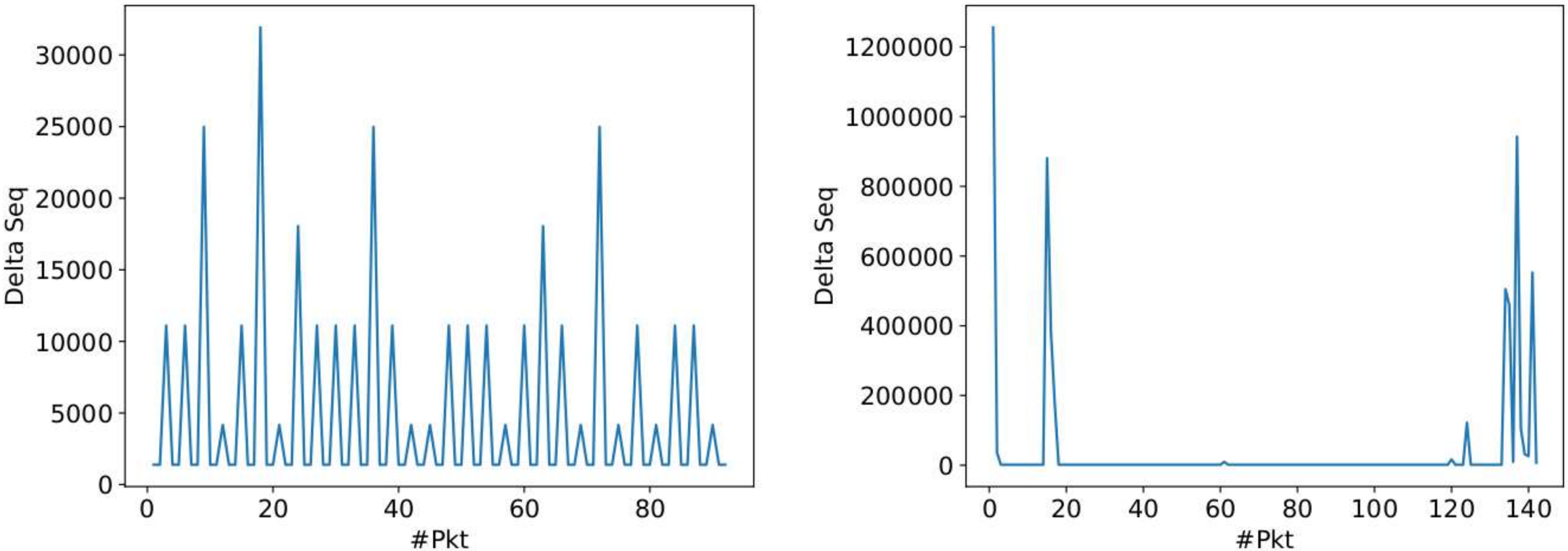
Modification of detection algorithm:

- Replace IAT-to-ASCII string conversion with new algorithm so that it can deal with 32-bit unsigned int.
- Adjust thresholds for detection.

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for ε-similarity (figures created by S. Zillien):



(a) Typical Covert channel traffic

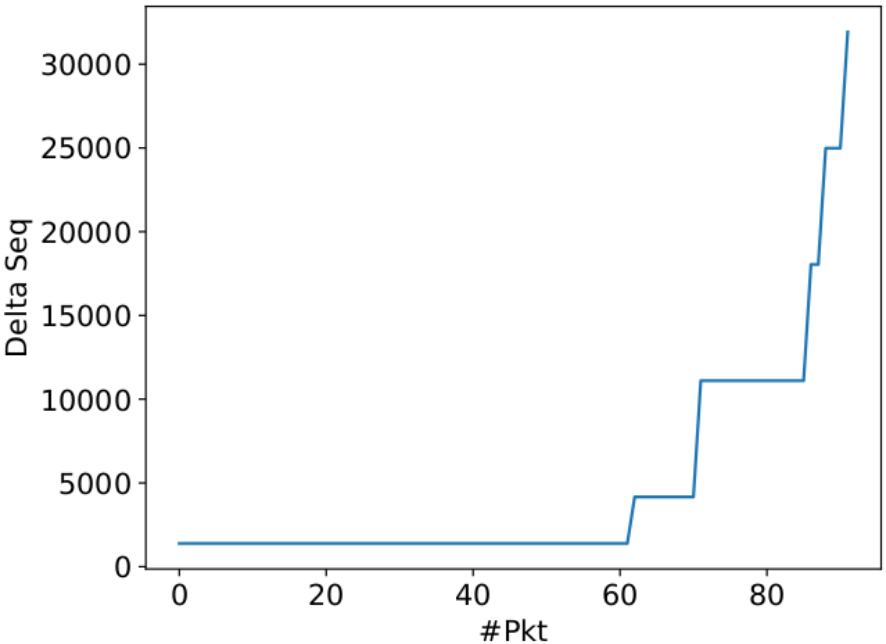
(b) Regular traffic (Germany 2)

Comparison: covert - regular: Δ values between retransmissions

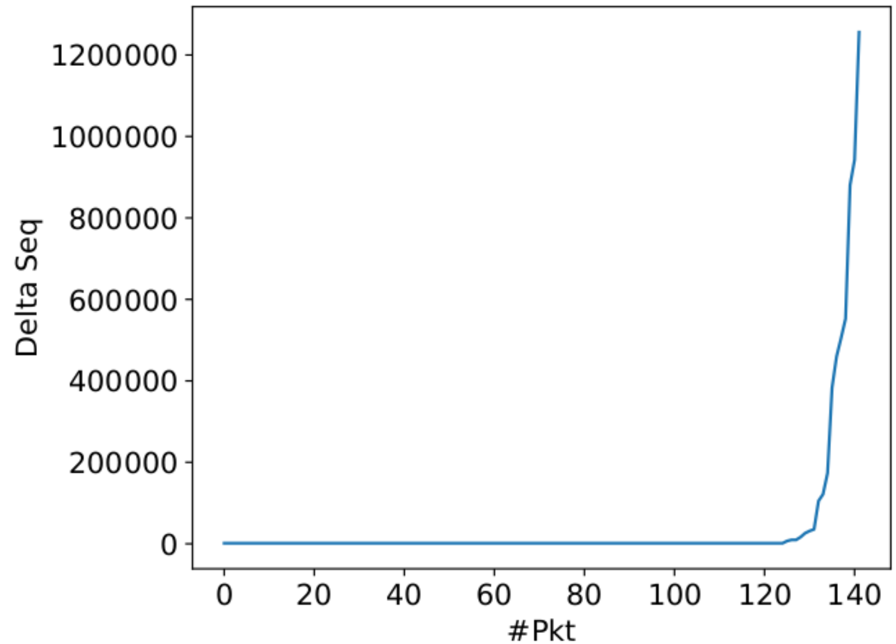
S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for ε-similarity (figures created by S. Zillien):



(a) Typical Covert channel traffic



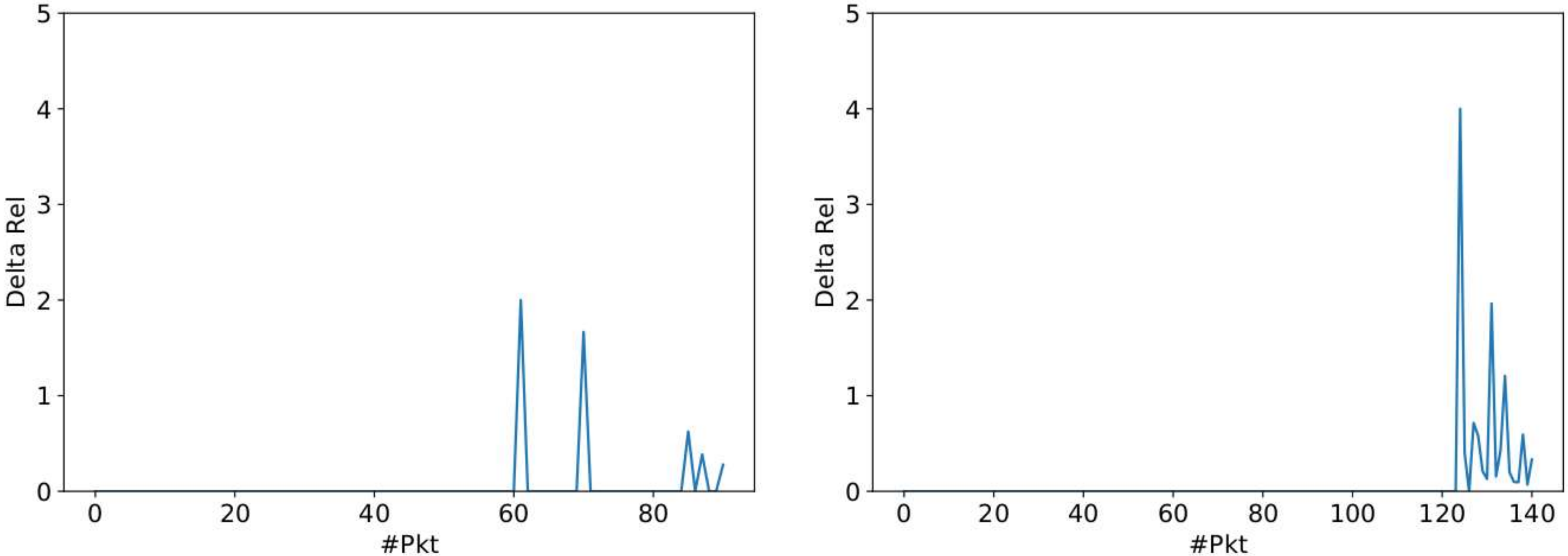
(b) Regular traffic (Germany 2)

Comparison covert - regular: sorted Δ values between retransmissions

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for ε-similarity (figures created by S. Zillien):



(a) Typical Covert channel traffic

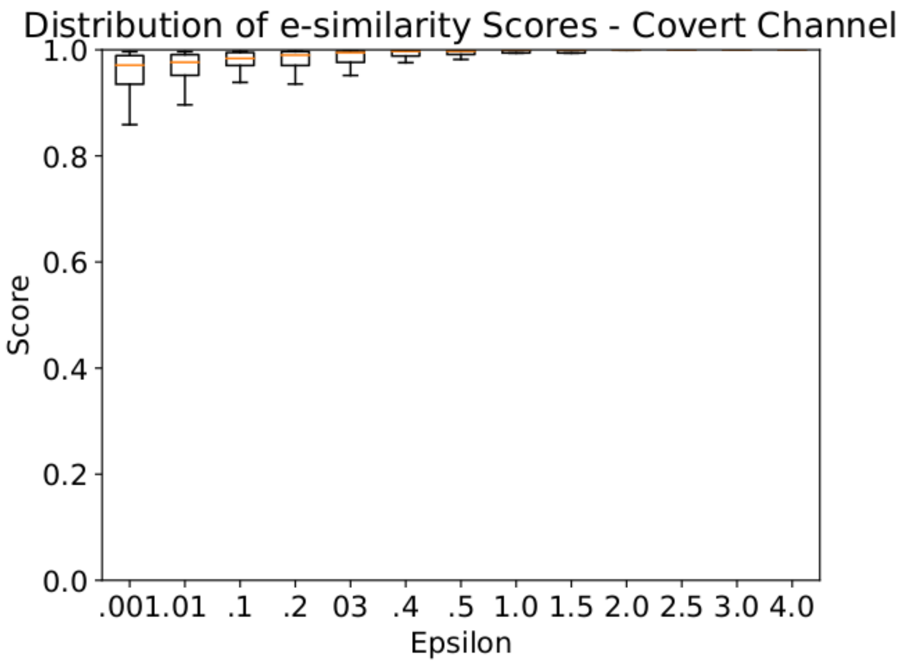
(b) Regular traffic (Germany 2)

Comparison covert - regular: relative differences of λ values

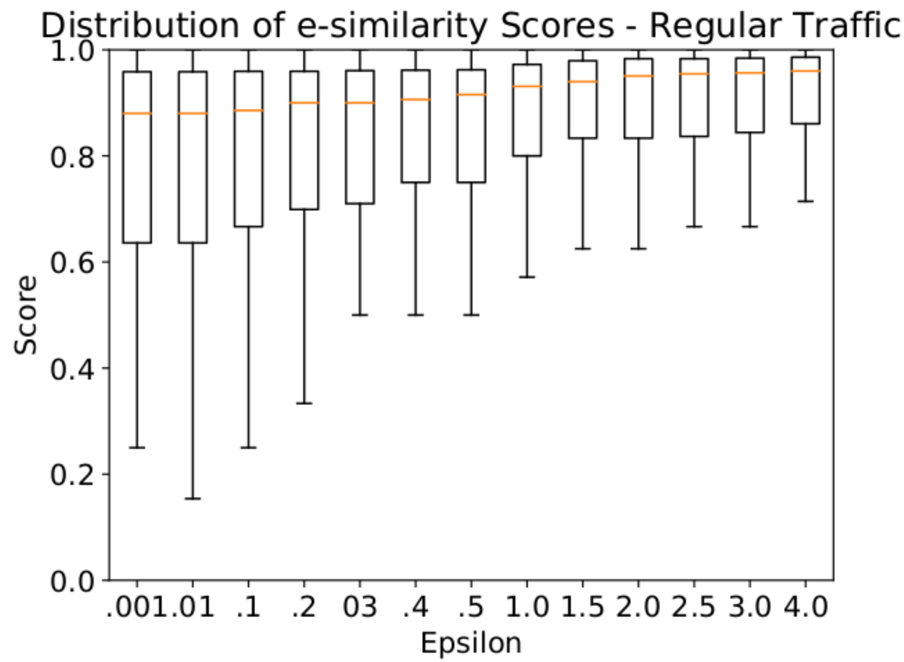
S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for ε-similarity (figures created by S. Zillien):



(a) Covert channel traffic



(b) Regular traffic

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec’18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for ε-similarity:

Results (mixed covert channels vs. mixed regular traffic): We chose $\epsilon = 0.01$ with an upper threshold of 0.997 (no lower threshold), $\epsilon = 0.2$ with a lower threshold of 0.95 and $\epsilon = 2.5$ with a lower threshold of 1.0 (both no upper threshold).

Detection results - ϵ-similarity

		Actual Class	
		Covert Channel	Regular Traffic
Detected Class	Covert Channel	154	1
	Regular Traffic	6	130

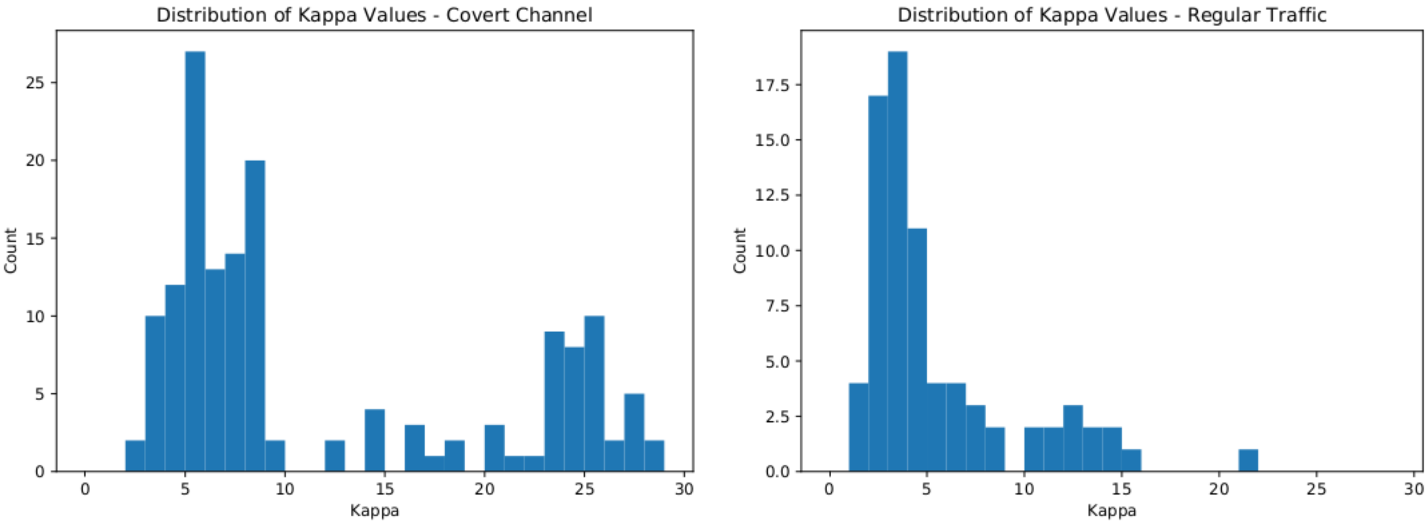
Please note that we focused solely on the detection of an optimized covert channel.
Also, the remaining undetectable channels were those configured using large gaps $D \geq 500$ between retransmissions combined with extremely few retransmissions (≤ 27) (resulting anyway in a short transmission and low transmission rate).

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for compressibility (figures created by S. Zillien):

Compressibility worked not so well (values of legitimate and covert traffic are quite overlapping; performs better with longer input data, i.e. more retransmissions)



(a) Covert channel traffic (b) Regular traffic

However, channel was an optimized one. Better results for trivial retransmission channels.

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

Countermeasure Variation for the Artificial Re-transmission Pattern

Results for compressibility:

Using an exemplary threshold $\kappa = 6$, we obtained the following detection results:

Detection results - compressibility

		Actual Class	
		Covert Channel	Regular Traffic
Detected Class	Covert Channel	136	26
	Regular Traffic	24	51

S. Zillien, S. Wendzel: Detection of Covert Channels in TCP Re-transmissions, in Proc. NordSec'18, Springer, 2018.

DYNAMIC WARDENS

Dynamic Wardens

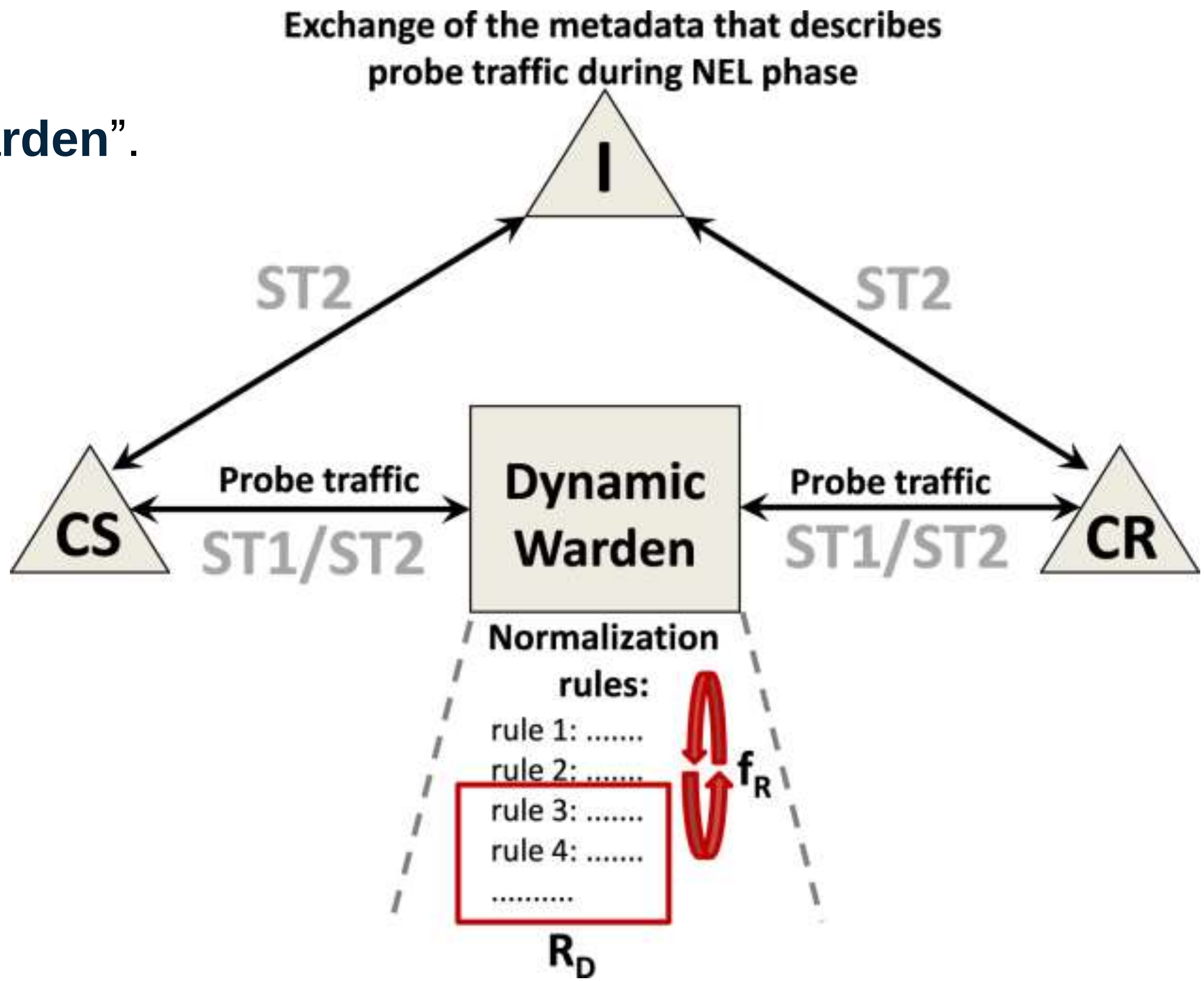
- **Problem:**
  Adaptive covert channels determine blocked covert channels by continuously checking connectivity. This allows them to circumvent filter technology.

- For instance, new firewall will be determined soon, followed by the utilization of different covert channels by the covert channel. This is called *Network Environment Learning* (NEL).

W. Mazurczyk, S. Wendzel et al.: Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

Dynamic Wardens

- **Solution:** Introducing a "**Dynamic Warden**".

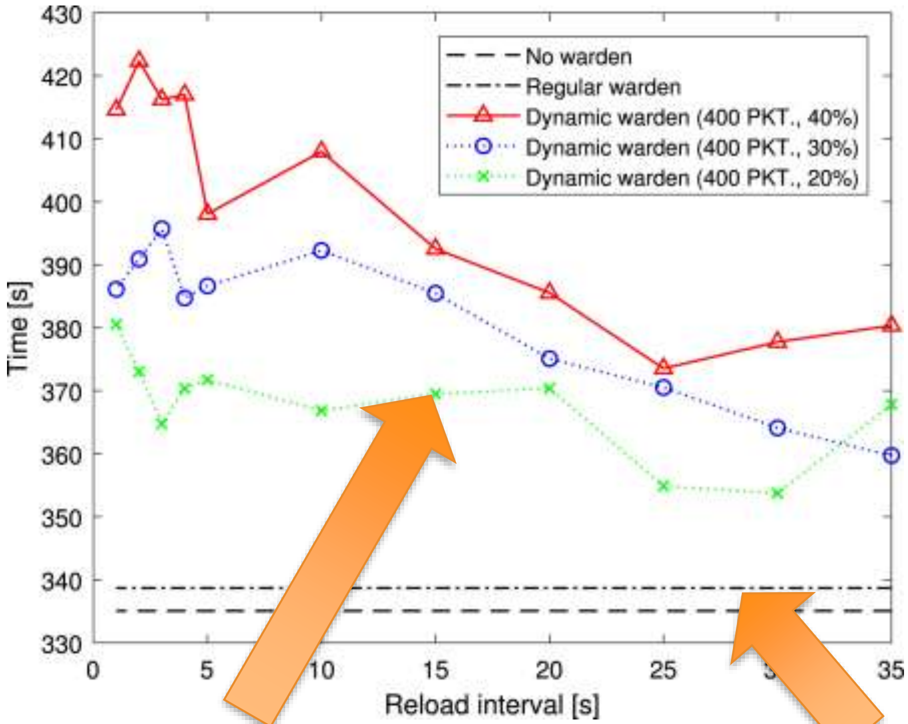


W. Mazurczyk, S. Wendzel et al.: [Countering adaptive network covert communication with dynamic wardens](), Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

Dynamic Wardens: Results

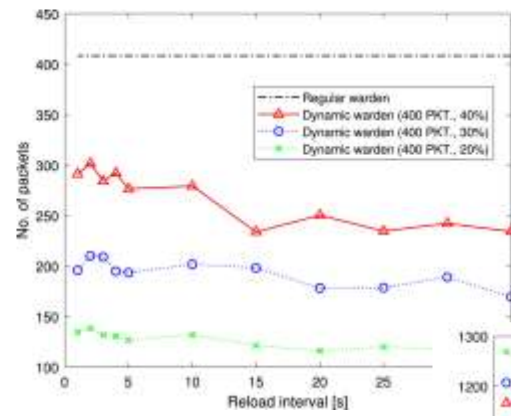
=> Results obtained from static configuration, each test repeated 20 times. Figures show average results.

Influence of the reload frequency on the **time needed to complete the transfer of 400 covert packets** for different types of wardens.

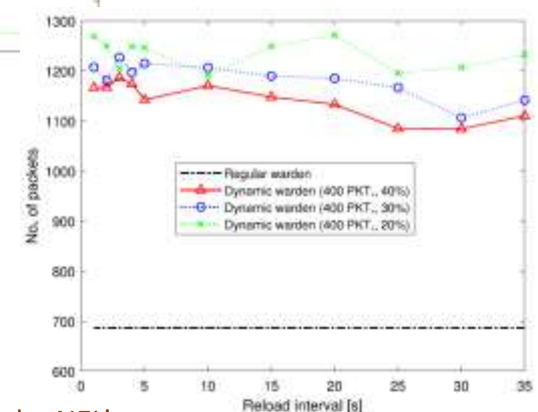


Dynamic warden with only 20% rule-set.

Influence of the reload frequency the on **the number of *normalized* packets** from CS to CR for different types of wardens.



Influence of the reload frequency the on **the number of *forwarded* packets** from CS to CR for different types of wardens. => more probe traffic

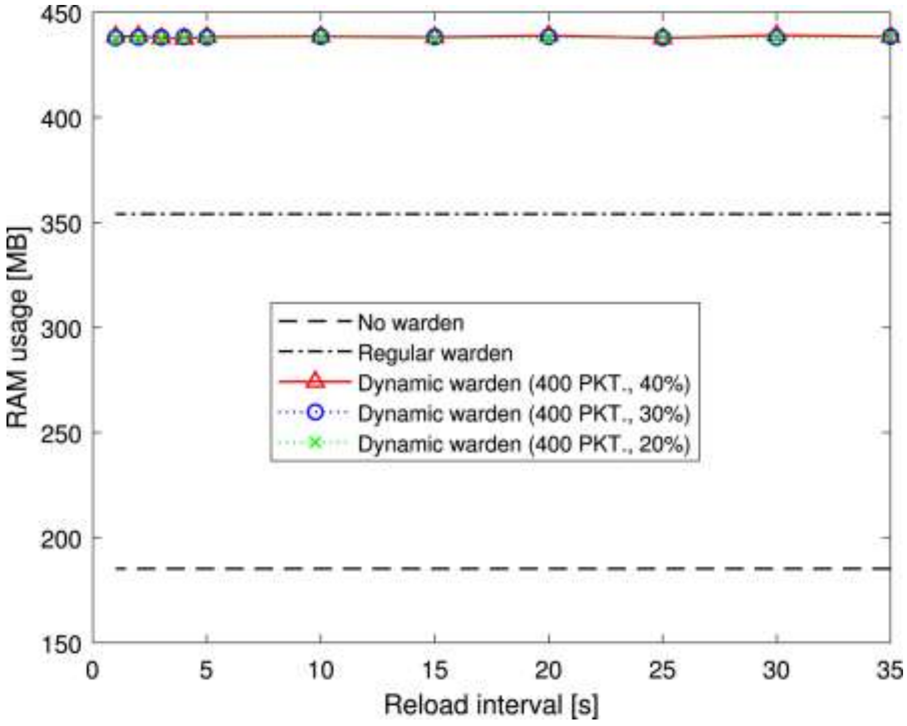


Regular warden with 80% rule-set (i.e. 80% of all the NEL's covert channels can be blocked)!

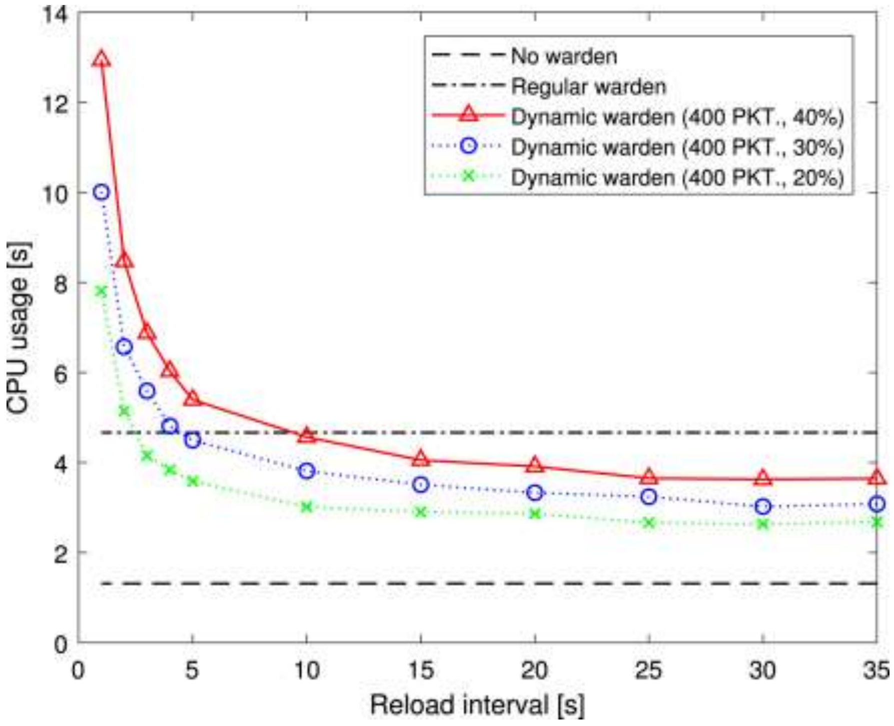
W. Mazurczyk, S. Wendzel et al.: Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

Dynamic Wardens: Results

Influence of the reload frequency on the **RAM usage** for different types of wardens (all wardens based on same Python code basis).



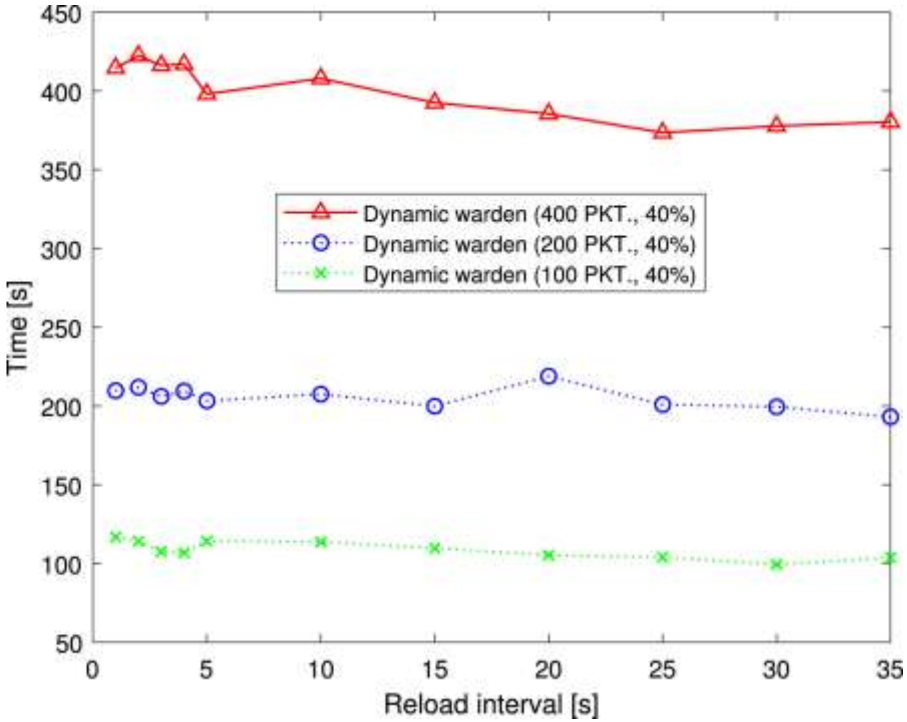
Influence of the reload frequency the on the **CPU usage** from CS to CR for different types of wardens.



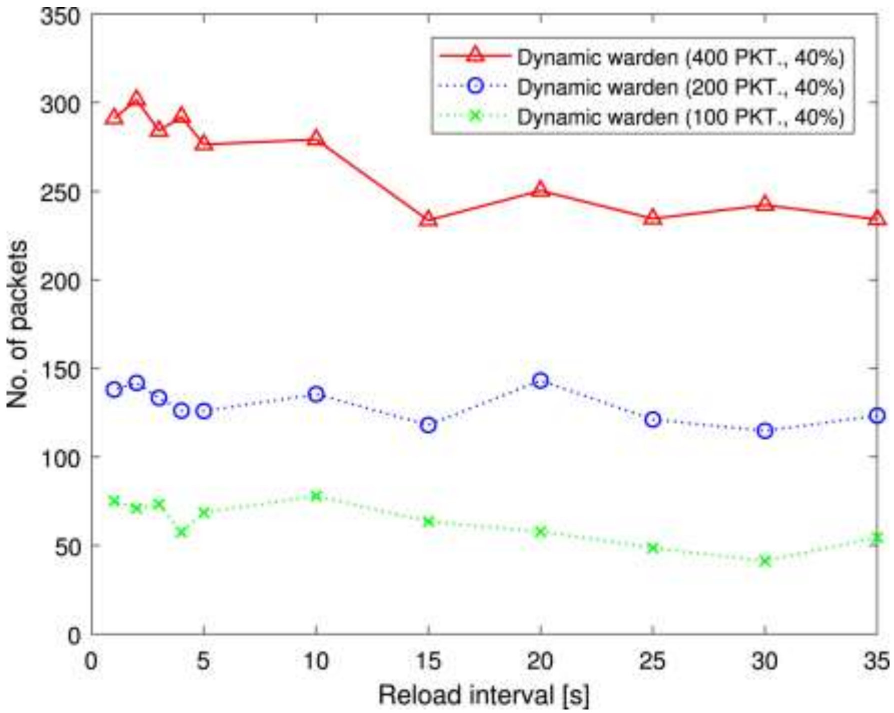
W. Mazurczyk, S. Wendzel et al.: Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

Dynamic Wardens: Results

Influence of the reload frequency on the **time needed to complete the transfer** of covert packets **for different lengths of the covert transmissions** (R_D=40%).



Influence of the reload frequency the on **the number of normalized packets** of covert packets **for different lengths of the covert transmissions** (R_D=40%).



W. Mazurczyk, S. Wendzel et al.: Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

Randomized Dynamic Warden: Results

Is it possible to load less filter rules on average by randomizing the number of loaded rules and the reload frequency?

- V1: $f_R \in \langle 1\text{ s}; 35\text{ s}\rangle$ and $R_D \in \langle 2\%; 100\%\rangle$ (i.e. between 1 and 50 rules). This means that the reload interval and the size of an active ruleset are selected randomly for the typical values investigated for the dynamic warden in the previous experiments.
- V2: $f_R \in \langle 1\text{ s}; 35\text{ s}\rangle$ and $R_D \in \langle 20\%; 40\%\rangle$ (i.e. the size of the active ruleset is randomly selected between 10 and 20 rules). Such values were tested for the dynamic warden in the previous sections.
- V3: $f_R \in \langle 1\text{ s}; 10\text{ s}\rangle$ and $R_D \in \langle 20\%; 100\%\rangle$ (i.e. between 10 and 50 rules). This means that the reload interval is selected from the values for which the best results have been achieved for the dynamic warden in the previous experiments.
- V4: $f_R \in \langle 1\text{ s}; 10\text{ s}\rangle$ and $R_D \in \langle 20\%; 40\%\rangle$ (i.e. between 10 and 20 rules) – both the reload interval and the size of the active ruleset are selected randomly in the ranges for which the best experimental results have been obtained for the dynamic warden investigated in the previous experiments.

Variant V3 offers the best results in terms of the

- **time needed to complete the covert transfer** and
- the **volume of traffic generated by the adaptive covert channel** parties (which is comparable with the best results obtained by the static setup for the dyn. warden)
- While offering **lower CPU and RAM consumption** (than static setup for the dyn. warden).

W. Mazurczyk, S. Wendzel et al.: Countering adaptive network covert communication with dynamic wardens, Future Generation Computer Systems, Vol. 94, pp. 712-725, Elsevier, 2019.

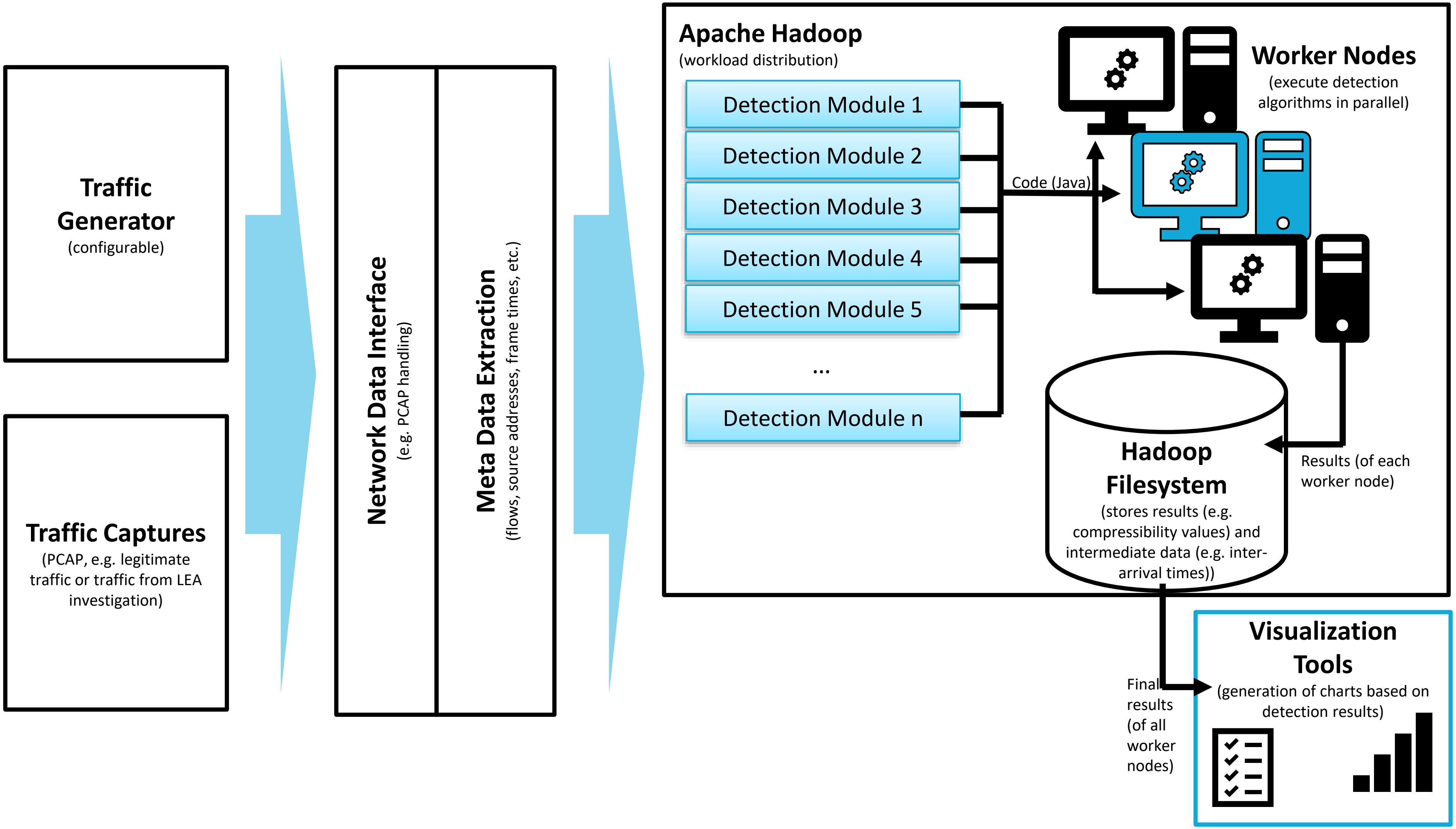
REPLICATING EXPERIMENTS

Replicating Experiments

- Almost nobody seems to replicate experimental results of other researchers in the covert channel domain.
  - Manifold reasons, e.g. it is difficult to publish replication studies.
- But: How trustworthy are provided results?

Replicating Experiments

WoDiCoF (*Worms Distributed Covert Channel Detection Framework*)

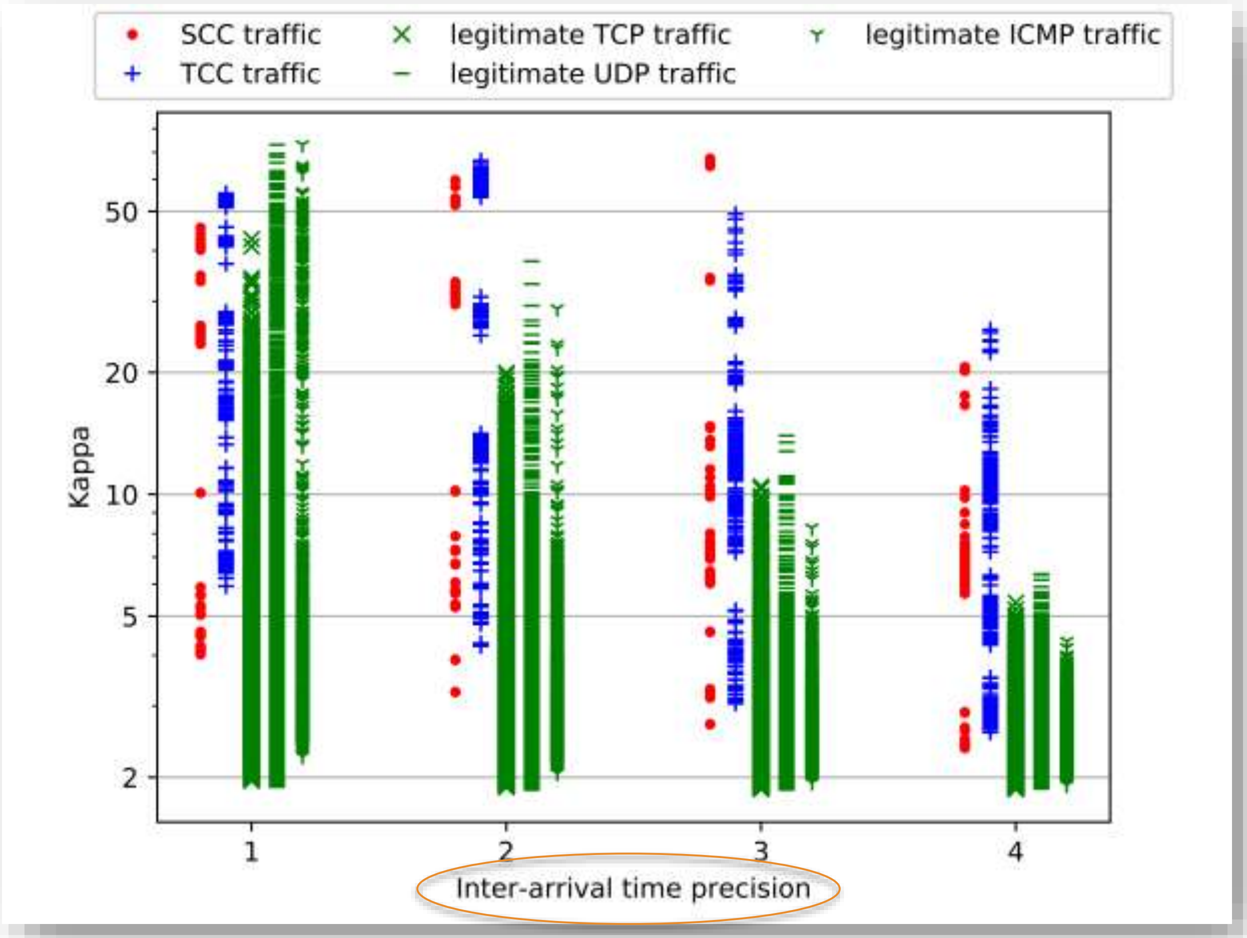


Replication Study: Compressibility of Cabuk et al.

- Published in ACM Transactions on Information and System Security (TISSEC), as an extended version of an ACM CCS paper.
- 137/469 citations *(Jan-16-2019, src: Google Scholar)*
- However, compressibility was only covered in the journal version.

Replication Study: Compressibility of Cabuk et al.

Let's see how the precision of the measured IAT values influences κ…



Replication Study: Compressibility of Cabuk et al.

Let's see what happens if we transfer slightly different data over the covert channel …

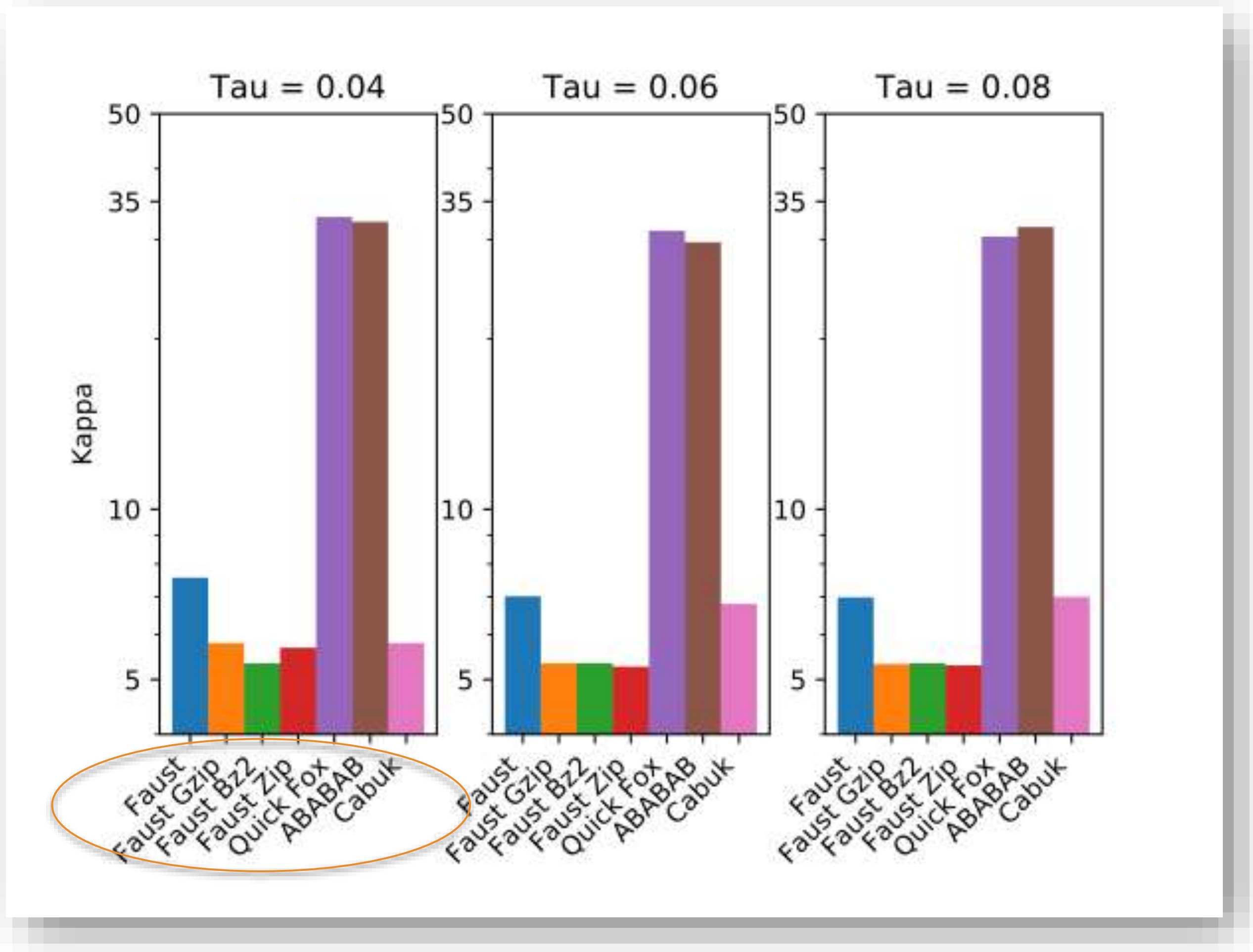


Fig.: R. Keidel, S. Wendzel, S. Zillien et al.: [WoDiCoF - A Testbed for the Evaluation of (Parallel) Covert Channel Detection Algorithms](), J.UCS, Vol. 24(5), 2018.

Replication Study: Compressibility of Cabuk et al.

Let's see how Kappa differs when we utilize a different network connection …

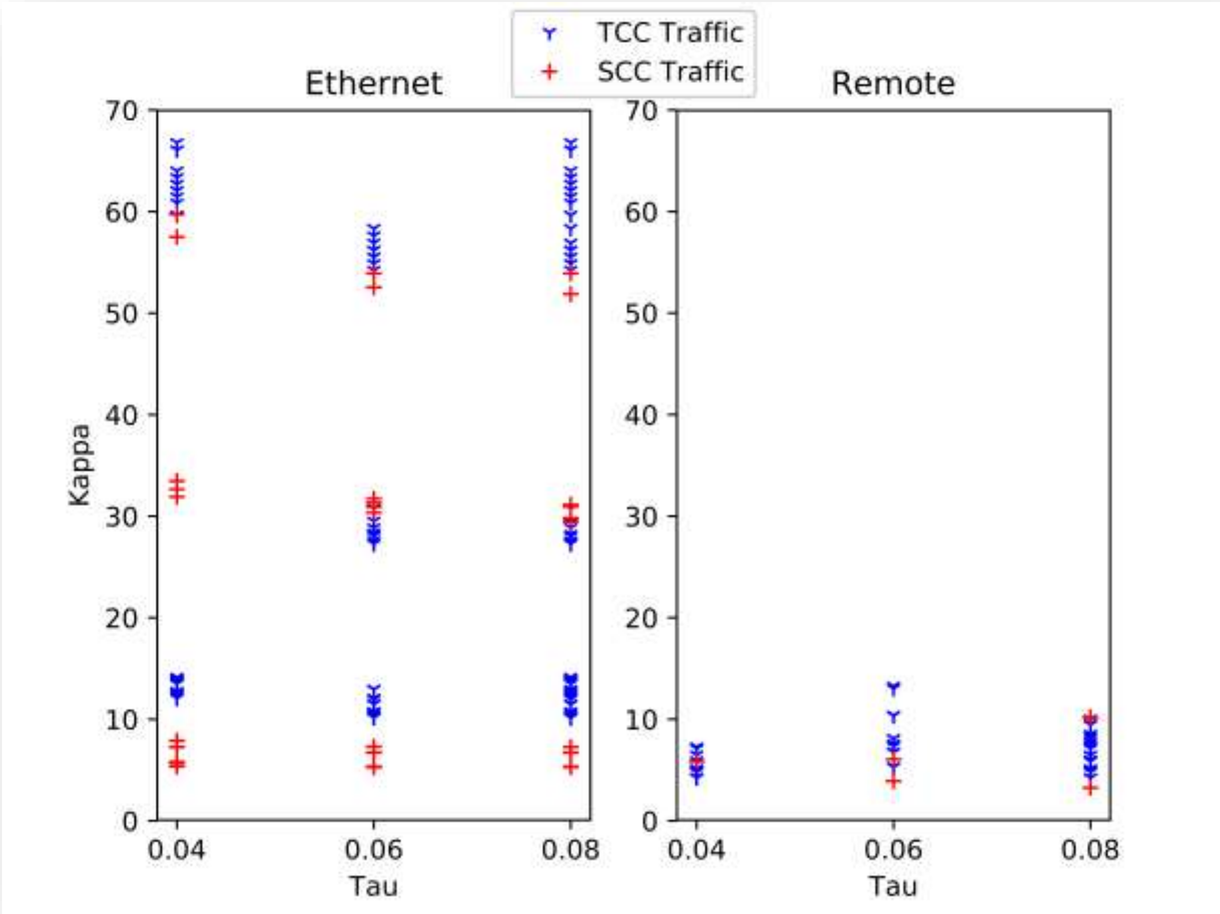


Fig.: R. Keidel, S. Wendzel, S. Zillien et al.: [WoDiCoF - A Testbed for the Evaluation of (Parallel) Covert Channel Detection Algorithms](), J.UCS, Vol. 24(5), 2018.

Finally: Testing Parallel Performance

Parallelization using Apache Hadoop with several gigabytes of PCAP recordings.

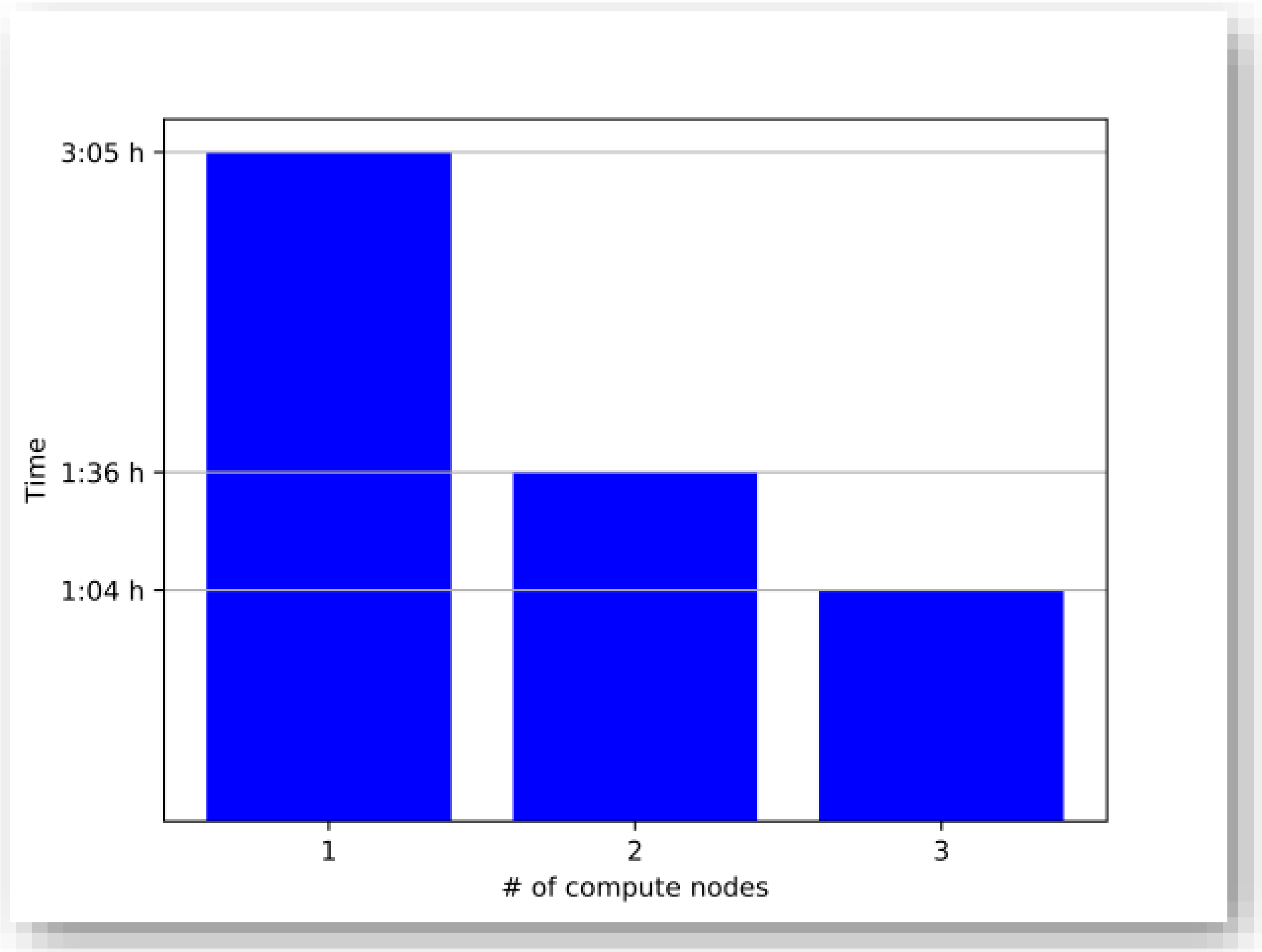


Fig.: R. Keidel, S. Wendzel, S. Zillien et al.: WoDiCoF - A Testbed for the Evaluation of (Parallel) Covert Channel Detection Algorithms, J.UCS, Vol. 24(5), 2018.

Summary

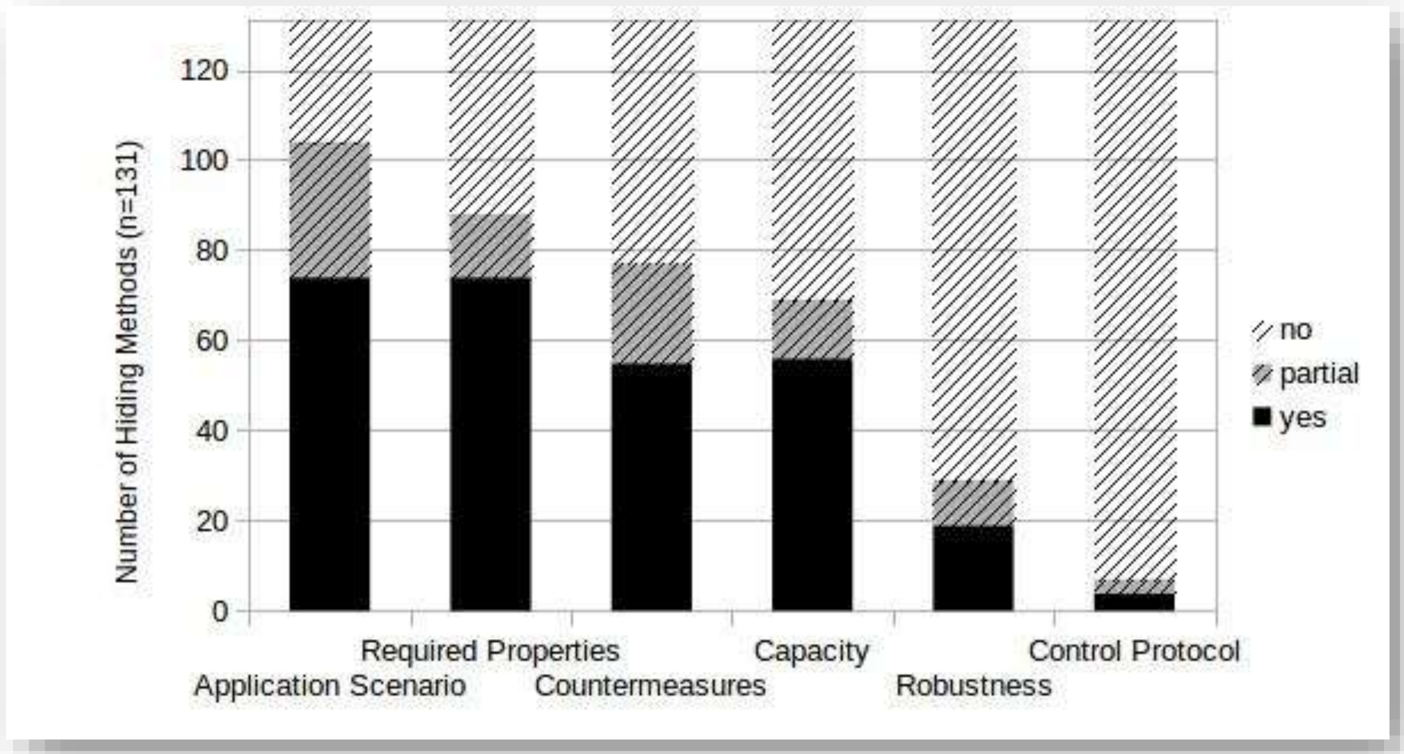
- Replication can lead to new insights:

  Even if previous work (such as in case of Cabuk et al.) is not "wrong", replication studies can extend our understanding of how a method performs under changing circumstances.

HOW TO DESCRIBE A NEW HIDING METHOD?

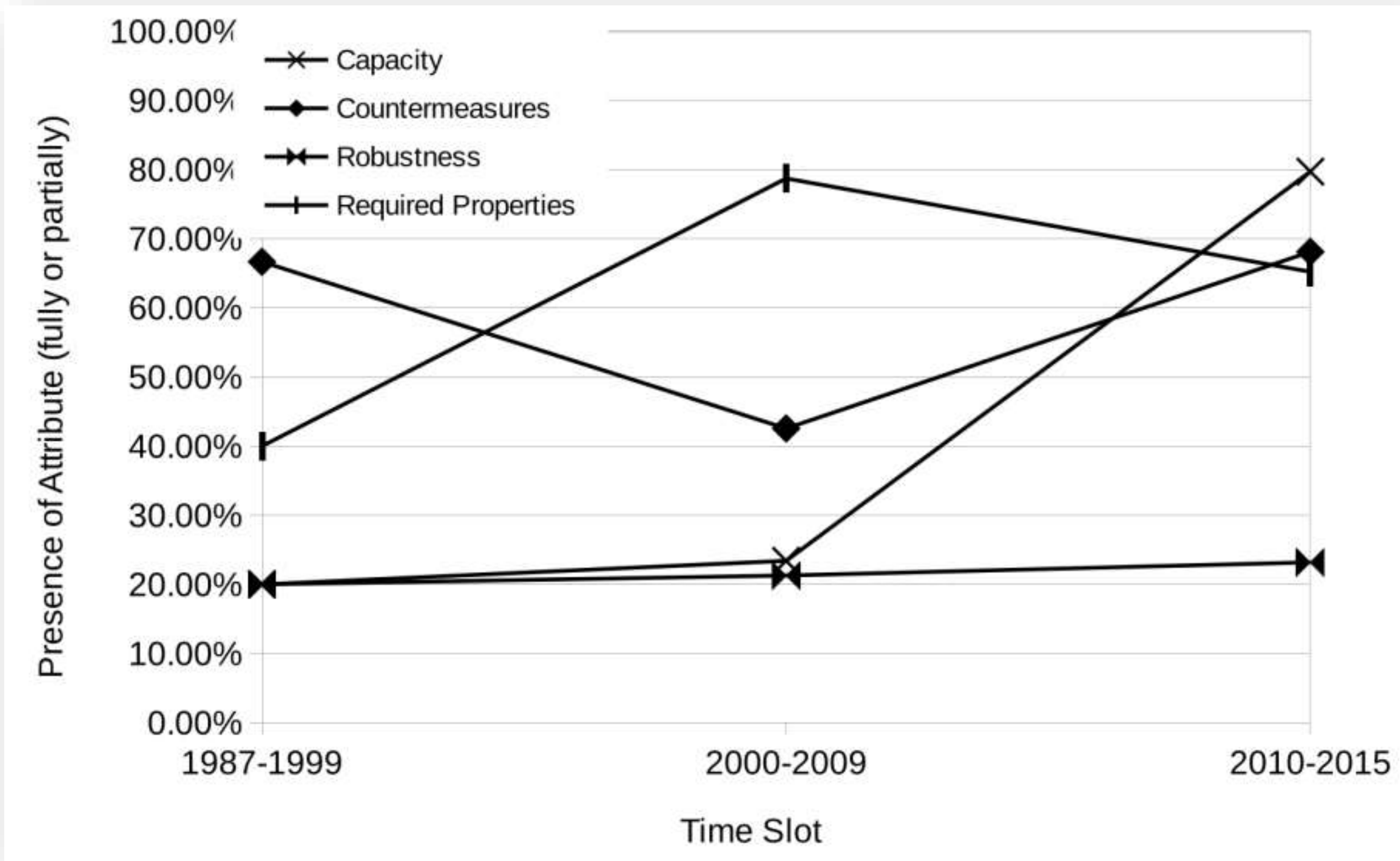
Analysis of 131 Hiding Techniques

The descriptions of hiding techniques in scientific papers highly vary, rendering it very difficult to compare them.



S. Wendzel et al. Unified Description for Network Information Hiding Methods, in: Journal of Universal Computer Science, 24(5), 2018.

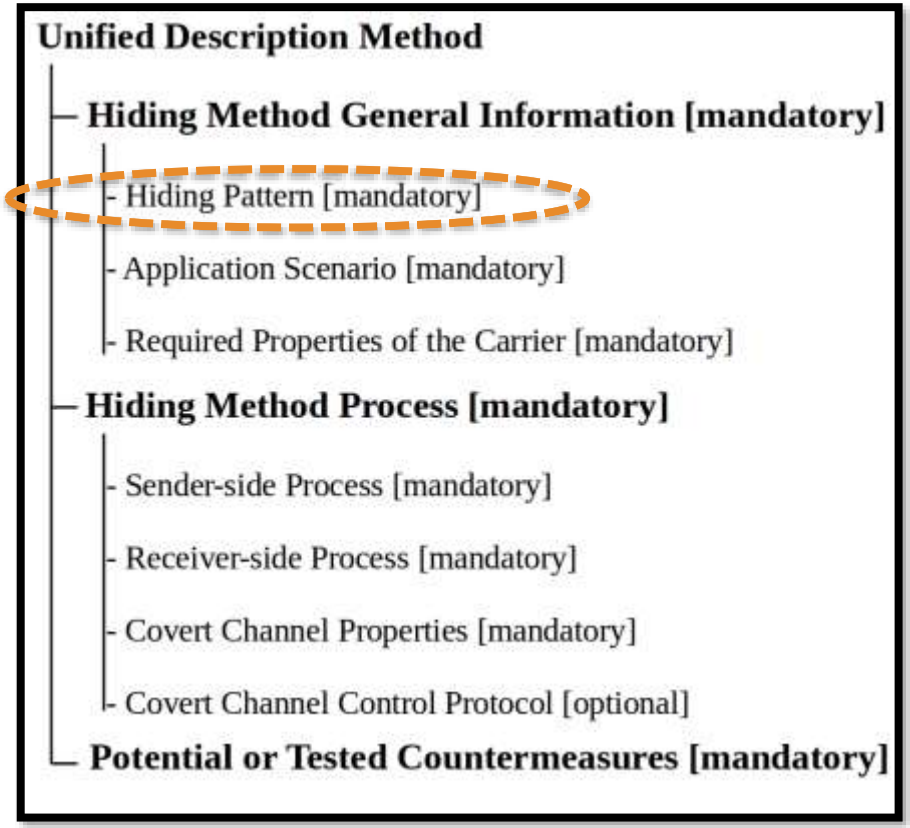
Analysis of 131 Hiding Techniques



S. Wendzel et al. Unified Description for Network Information Hiding Methods, in: Journal of Universal Computer Science, 24(5), 2018.

Describing Hiding Methods Using Patterns

- We proposed a method to unify the descriptions within new publications. Our method is simply called a **unified description method**.
- Detailed description of the attributes + examples can be found in the paper.



S. Wendzel et al. Unified Description for Network Information Hiding Methods, in: Journal of Universal Computer Science, 24(5), 2018.

Examples for Applying the Unified Description Method …

… can be found here: http://www.jucs.org/jucs_24_5/wodicof_a_testbed_for.

Or in the work of others. e.g.

- Graniszewski, Waldemar, Jacek Krupski, and Krzysztof Szczypiorski. "SOMSteg-Framework for Covert Channel, and its Detection, within HTTP." *Journal of Universal Computer Science* 24(7), 2018: 864-891.
- Mileva, Aleksandra, Aleksandar Velinov, and Done Stojanov. "New Covert Channels in Internet of Things." in Proc. *SECURWARE 2018*, 2018: 30-36.
  - … and follow-up paper at Int. Journal Adv. Sec., in press.

Summary

- Information Hiding faces inconsistency in its experimental methodology and its terminology/taxonomy.
  - **Patterns** and the **Unified Description Method** are means to improve the situation.
  - Results of **Experimental Replication** underpins the need for better experimental testing.
    - Both approaches (especially patterns) increasingly applied by the research community

- There is a lack of countermeasures when it comes to certain patterns.
  - Solution: Introduced **Countermeasure Variation**.

- When dealing with adaptive covert channels (NEL), current countermeasures such as static traffic normalizers do not perform well.
  - Solution: Introduced **Dynamic Wardens**.

Are there any questions?

THANK YOU FOR YOUR KIND ATTENTION.

PS. Patterns can also help preventing scientific re-inventions,
cf. *S. Wendzel, C. Palmer: Creativity in Mind: Evaluating […], J.UCS, Vol. 21(12), 2015.*
My publications are available here.

Call for Papers!

IEEE *Transactions on Industrial Informatics* (IF 5.43)

Special Issue on Cyber-Physical Security in Industrial Environments

Deadline: May 1, **2019**

Elsevier *Future Generation Computer Systems* (IF 4.64)

Special Issue on Emerging Topics in Defending Networked Systems

Deadline: Jan 25, **2020**

Upcoming finalized SI:

IEEE *Security & Privacy*

Special Issue on Digital Forensics, pt. II
(probably out by end of the month?)

----BACKUP SLIDES---

SOPHISTICATED HIDING METHODS

Reliability & Control (Micro) Protocols

Control (or micro) protocols are embedded into a covert channel.

Benefits:

- Reliable data transfer
- Session management for covert transactions
- Covert overlay network addressing schemes
- Dynamic routing for covert channel overlays
- Upgrades of a covert channel overlay infrastructure
- Peer discovery within a covert channel overlay
- Switching of utilized network protocols
- Adaptiveness to network configuration changes

S. Wendzel, J. Keller: Hidden and Under Control, Annals of Telecommunications (ANTE), Springer, 2014.

Reliability & Control (Micro) Protocols

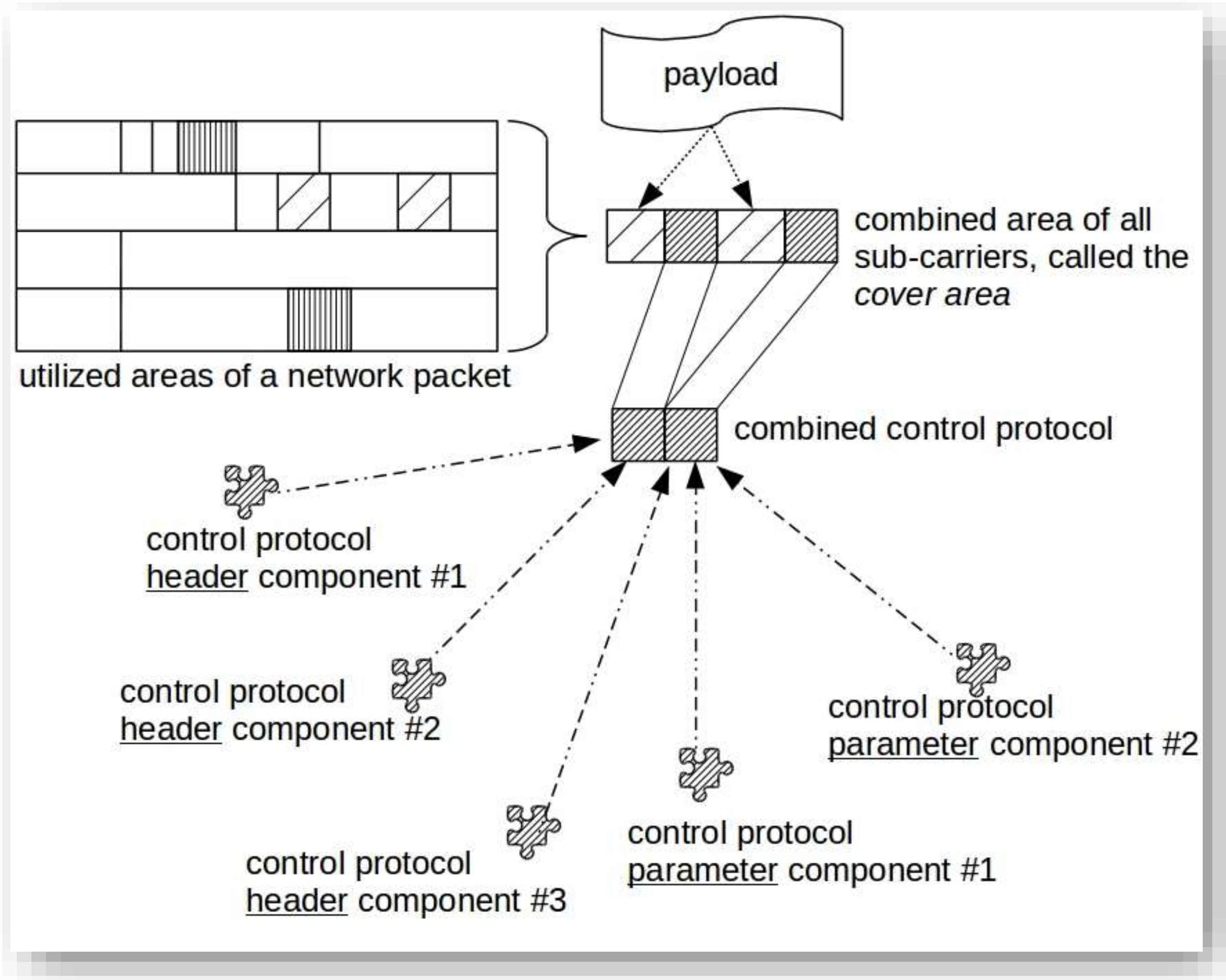
- (Formal) approaches for designing control protocols are available.

- … and so are optimization methods.

… and countermeasures, cf.
Jaspreet Kaur, Steffen Wendzel, Omar Eissa, Jernej Tonejc, Michael Meier: Covert Channel-internal Control Protocols: Attacks and Defense, *Security and Communication Networks (SCN)*, Vol. 9(15), Wiley, 2016.

S. Wendzel, J. Keller: Hidden and Under Control, Annals of Telecommunications (ANTE), Springer, 2014.

Reliability & Control (Micro) Protocols



Source: (Mazurczyk et al., 2016)

Network Environment Learning

- NEL allows covert channel nodes to determine how filters in their network environment are configured by probing several covert channel techniques.

- NEL is a constant process.

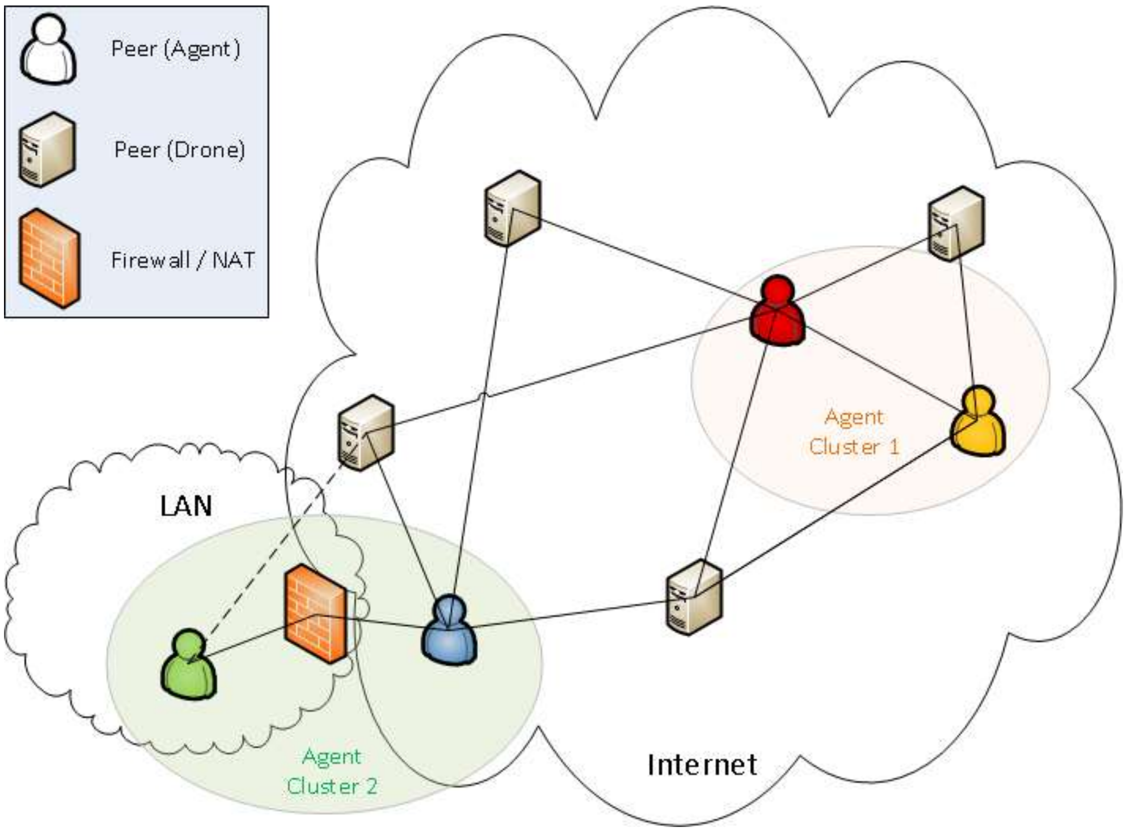
- Originally introduced by Yarochkin et al.
  - Circumvention-method improved a few years later by myself.

Yarochkin, Fedor V., et al. "Towards adaptive covert communication system." *Dependable Computing, 2008. PRDC'08. 14th IEEE Pacific Rim International Symposium on*. IEEE, 2008.

Wendzel, Steffen. "The Problem of Traffic Normalization Within a Covert Channel's Network Environment Learning Phase." *Sicherheit*. Vol. 12. 2012.

Dynamic Overlay Routing for Covert Channels

- Building overlays provides several advantages, such as …
  - Bypassing firewalls
  - Utilizing third-party nodes
  - QoS
- Based on control (micro) protocols
- Prototype with OSPF-like protocol in 2012.



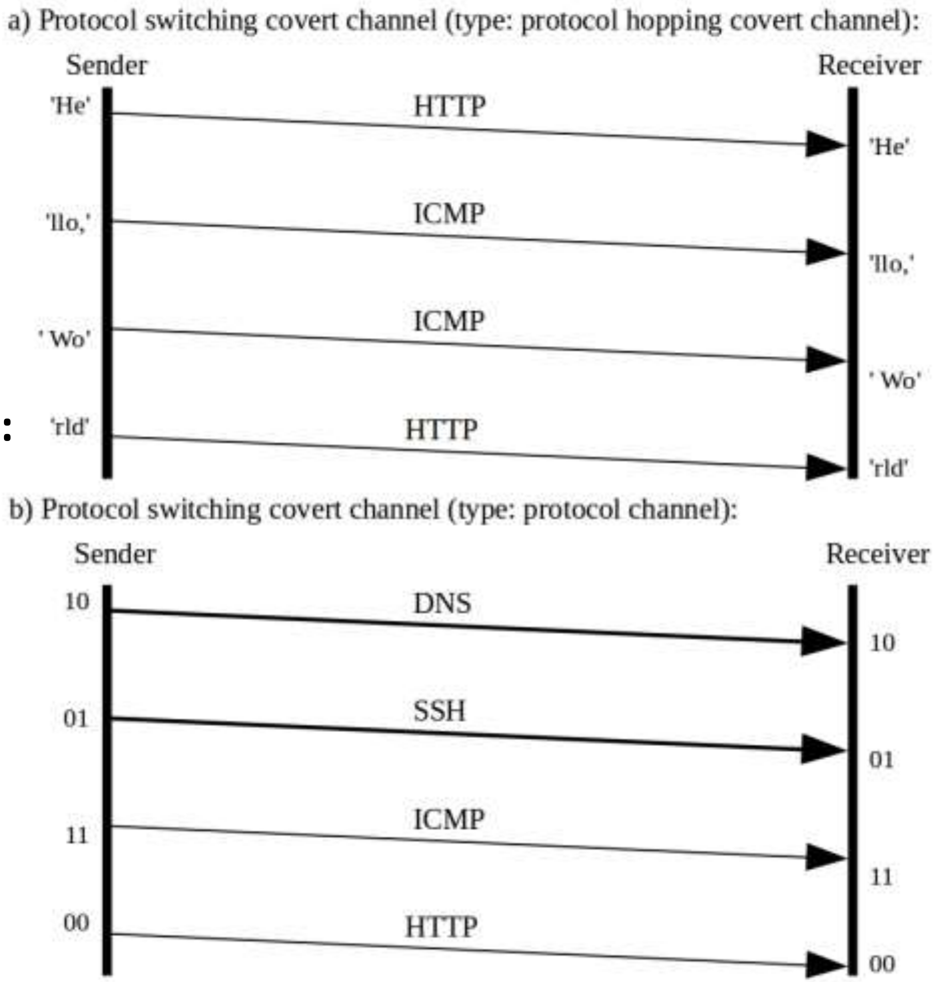
Backs, P., Wendzel, S., Keller, J.: Dynamic Routing in Covert Channel Overlays Based on Control Protocols, Proc. ISTP, IEEE, 2012.

Protocol Switching, Protocol Hopping, Pattern Hopping

Protocol Hopping Covert Channel (PHCC):
Secret information is split over multiple network protocols to increase hurdles for a forensic traffic analysis.

Protocol Switching Covert Channel (PSCC):
Secret information is represented by the protocol itself.

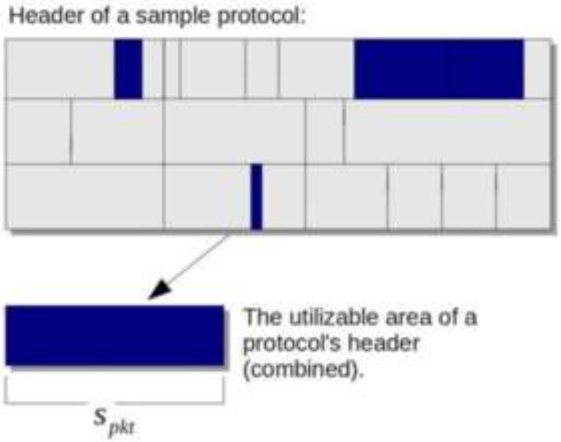
Pattern Hopping:
For every new piece of secret information a PRNG selects one of the patterns (+variation) to transfer the data.



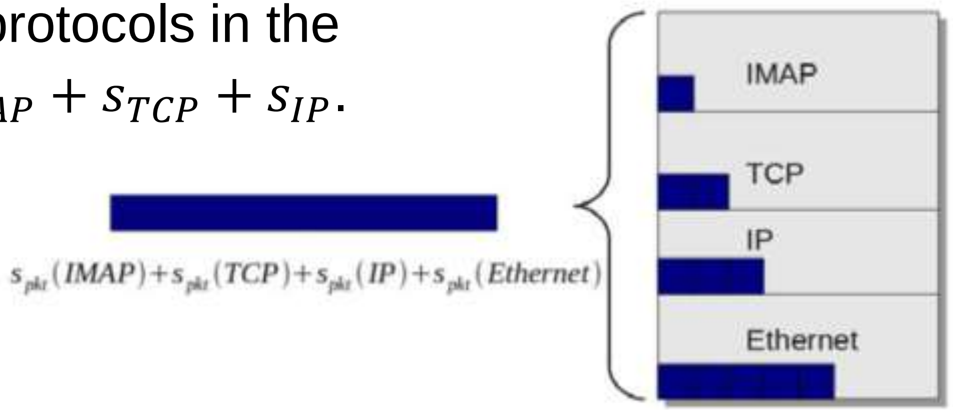
S. Wendzel, S. Zander: Detecting protocol switching covert channels, Proc. *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*. IEEE, 2012.
S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, Proc. Communications and Multimedia Security (CMS), 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- Let us assume a covert channel could utilize an area of s_{pkt} bits in a protocol header. To transfer a message of size $s_{overall}$, we would thus need $N = \left\lceil \frac{s_{overall}}{s_{pkt}} \right\rceil$ packets, and $2N$ packets if every packet would require an acknowledgement from the CR.



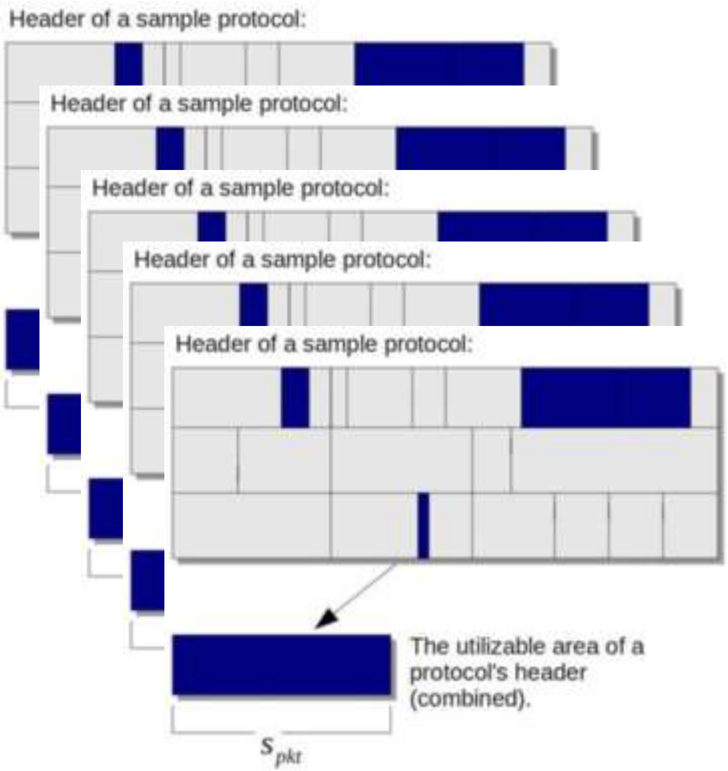
- For a multi-layered protocol, a CC could combine the s_{pkt}values of the protocols in the selected layers, e.g. $s_{\text{pkt}} = s_{IMAP} + s_{TCP} + s_{IP}$.



S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- For a PHCC using n Protocols $P_1 \dots P_n$, we can calculate the average amount of data transferrable per packet as $\overline{s_{pkt}} = \sum_{i=1}^{n} p_i s_i$, where P_i is chosen with probability p_i and provides s_i bits of covert storage per packet.



S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- Now, we can optimize a PHCC for different purposes (**QoS**), e.g.
  - A password cracking program needs to transfer a short password string out of a network (e.g. one password/hour).

    => keep a low profile (transfer only few packets: minimize overhead)
  - Urgently (but still covertly) leak videos of harmed protesters in a country with Internet censorship to the press.

    => still keep a low profile, BUT transfer data rather quickly (high throughput).

S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- If **high throughput** is required, we can maximize f_1:

$$\mathrm{f}_1 = \sum_{i=1}^{n} p_i s_i.$$

- We do this under the set of constraints that $\sum_i p_i = 1$ and that an m with $0 < \mathrm{m} \leq p_i \leq 1$ is used as a minimum threshold for selecting protocol P_i so that every protocol has a chance for selection and **render forensic analysis more difficult**.

- We suggest to chose a low value $m = c/n$, with $c < 1$, e.g. for $n = 20$ protocols, and $c = 0.2$, every protocol would be selected with at least 1% probability.

S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- If the goal is to **generate little overhead** and optimize covertness this way, we first need to introduce

$$q_i = \frac{sizeof(P_i)}{s_{pkt}(P_i)}$$

… to indicate how many bits are transferred to send a single covert bit using a protocol P_i.

- Now, we can minimize f_2 (again, we consider the inclusion of all protocols using some threshold value m):

$$f_2 = \sum_{i=1}^{n} p_i q_i.$$

S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

Optimizing PHCC (Wendzel & Keller, 2011)

- One could also optimize covertness if each protocol (or better: each covert channel technique) is assigned a covertness level, e.g. $c_i \in \mathbb{N}$.

- One could then maximize f_3 (again, we consider the inclusion of all protocols using some threshold value m):

$$f_3 = \sum_{i=1}^{n} p_i c_i.$$

S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, in Proc. CMS 2011, LNCS 7025, Springer 2011.

- Optimizing protocol utilization for PHCCs is already nice to have, but can we also optimize the micro protocol so that we raise even fewer attention?

[Would I raise this question if the answer would be no?]

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

- We skip the part on micro protocol size minimization using protocol engineering, cf. some of my papers on this topic if you are interested.

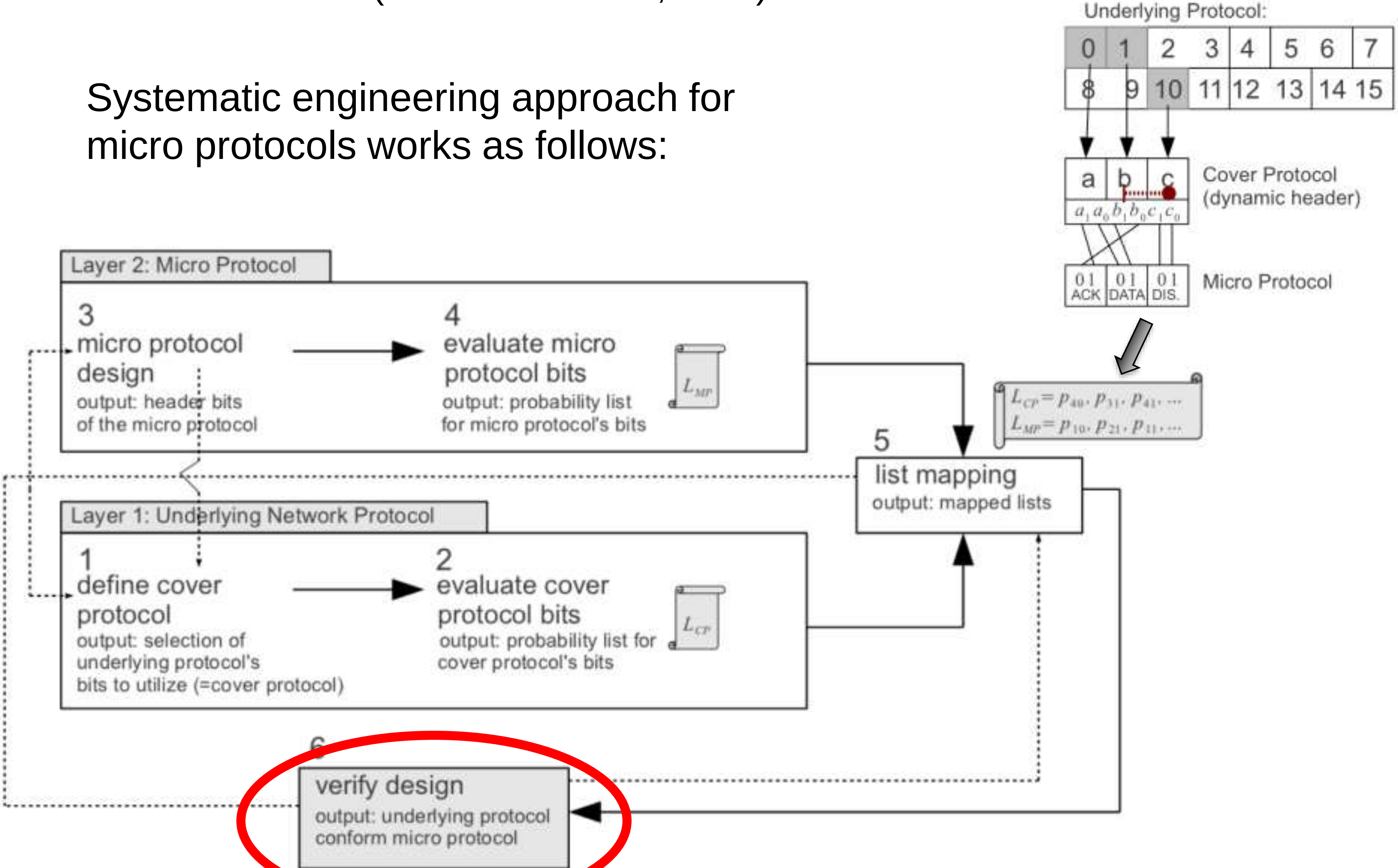
- **Goal:** Embed the micro protocol in a low-attention raising manner.

- **Answer:** We use a tailored protocol engineering approach (we will cover this at least in a nutshell).

S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

Systematic engineering approach for micro protocols works as follows:



S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

- For step 6 (**design verification**), one needs to **make sure that there are no undesired bit-combinations set in the underlying protocol through the micro protocol operation** (e.g. protocol header flags that would break a standard).

- **Solution:** model both protocols using formal grammar of Chomsky type 2 (regular) or 3 (context-free) and perform a language inclusion test to test compatibility, i.e. the language of the micro protocol must be equal (or a sub-set) of the cover protocol's language.

S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

- First, we define the rules of the **cover protocol** as $G_{CP} = (V, \Sigma, P, S)$, where V is the set of non-terminals, Σ is the set of terminals, P the set of productions, and $S \in V$ the start symbol.
- Next, we define the formal grammar for the micro protocol G_{MP} in the same manner.
- We also perform a mapping of terminal symbols in Σ, e.g., $\mathrm{a}_0 \equiv \neg ACK,\ a_1 \equiv ACK$.

Example:

$$\begin{gathered}\boldsymbol{G_{CP} = (V, \Sigma, P, S)},\\ \boldsymbol{V = \{S, A, B, C\}},\\ \boldsymbol{\Sigma = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{b}_0, \mathrm{b}_1, \mathrm{c}_0, \mathrm{c}_1\}},\\ \text{and } \mathbf{P = \{S \rightarrow AB|AC},\\ \mathbf{A \rightarrow a_1|a_0},\\ \mathbf{B \rightarrow b_1|b_0},\\ \mathbf{C \rightarrow b_1c_1|Bc_0\}}\end{gathered}$$

$$\begin{gathered}\boldsymbol{G_{MP} = (V, \Sigma, P, S)},\\ \boldsymbol{V = \{S, B, C_A, C_B\}},\\ \boldsymbol{\Sigma = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{b}_0, \mathrm{b}_1, \mathrm{c}_0, \mathrm{c}_1\}},\\ \text{and } \mathbf{P = \{S \rightarrow a_0B|a_1B},\\ \mathbf{B} \rightarrow \boldsymbol{b_0C_A|b_1C_B},\\ \mathbf{C_A} \rightarrow \boldsymbol{c_0},\\ \mathbf{C_B \rightarrow c_0|c_1\}}\end{gathered}$$

S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

Finally, we test whether $L(G_{MP}) \subseteq \mathrm{L}(\mathrm{G}_{\mathrm{CP}})$, i.e. we perform a language inclusion test. This can be done either by hand for small languages or automatized (*for conditions, cf. our paper*).

Illustration:

Therefore, it is required to build sentences for all possible conditions of the micro protocol (e.g. setting flag X and flag Y within the same packet). For instance, to test whether the "ACK" flag and the "DIS" flag can be set within the same micro protocol header without breaking the standard conform behavior of the underlying protocol, we have to verify, if the following sentence of G_{MP} within G_{CP} is possible:

$$\{ACK, \neg DATA, DIS\} \equiv a_1 b_0 c_1 \quad (9)$$

However, the production rules do not allow to create the sentence "$a_1 b_0 c_1$" (only similar results are possible: "$a_1 b_0 c_0$" (AC), "$a_1 b_1 c_1$" (AC) and "$a_0 b_1 c_1$" (AC)). Thus acknowledging data and introducing a disconnect at the same time within the covert channel connection is not feasible with the provided configuration due to the conflict of setting the bits "a" and "c" without setting the bit "b" (DATA flag). We discuss solutions for this problem in Sect. 2.6.

S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Optimizing Micro Protocol Embedding
(Wendzel & Keller, 2012)

But what if …?

- **… the language inclusion test fails**? -> modify CP selection or MP design.

- **… we need to model connection-oriented protocols?** Can be done as described in the paper or potentially with I/O automata composition as described by *Lynch, N. A.: Distributed Algorithms. Morgan Kaufmann (1996).*

S. Wendzel, J. Keller: Systematic engineering of control protocols for covert channels, in Proc. CMS 2012, LNCS 7394, Springer 2012.

Video Summary of the Patterns and Sophisticated Hiding Techniques