Hochschule Worms
University of Applied Sciences

Network Information Hiding

COINS Summer School 2018

Steffen Wendzel

http://www.wendzel.de

RUSSI
ESTONIA
LATVIA
LITHUANIA
DENMARK
RUSSIA
BELARUS
Berlin
POLAND
UKRAIN
NETH.
BEL.
LUX.
Hochschule Worms
CZECH REPUBLIC
SLOVAKIA
MOLDOVA
AUSTRIA
HUNGARY
RANCE
SWITZ.
ROMANIA
SLOVENIA
CROATIA
BOSNIA AND HERZEGOVINA
SERBIA
BULGARIA
ITALY
GREECE

Fraunhofer
FKIE

Introducing myself

2016-now Prof. at Worms Univ. of Appl. Sciences

(since 2017: Scientific deputy head of ZTT unit)

2013-now: Researcher at Fraunhofer FKIE

(2013-16: Head of a research team on smart building security)

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

Photo: Elonicate Photography

Agenda

9:00 - Introduction, fundamentals, terminology, basic taxonomy

10:00 - Early covert channels and OS-level covert channels
- Fundamental countermeasures
- Network covert channels and hiding patterns

12:00 - Sophisticated (adaptive) hiding techniques
- Selected countermeasures

-break-

17:00 - Replications of experiments
- How to describe a hiding method (in case you find a new one)
- CPS steganography (over the network)

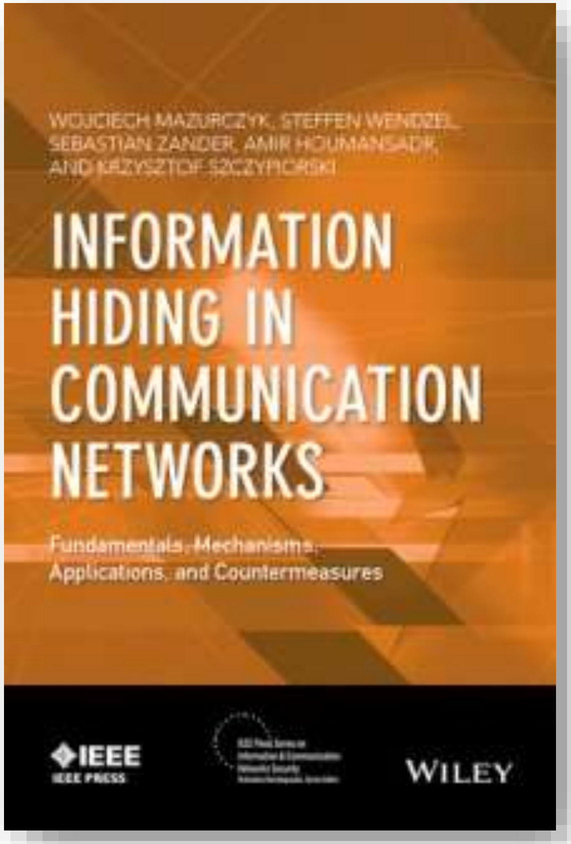
Addendum (slides):
- Analyzing citations in the domain (some more information regarding last evening’s discussion in the amphitheatre.

Introduction

About 60% of today’s lectures are based on our book on network information hiding.

- Community agreed on common understanding of many things to find a good basis for this book.
- Based on several years of research of the authors.
- Please note: the chapters on traffic obfuscation and network flow watermarking are not part of today‘s lectures.

In the remainder cited as **(Mazurczyk et al., 2016)**.



WOJCIECH MAZURCZYK, STEFFEN WENDZEL, SEBASTIAN ZANDER, AMIR HOUMANSADR, AND KRZYSZTOF SZCZYPIORSKI

INFORMATION HIDING IN COMMUNICATION NETWORKS

Fundamentals, Mechanisms, Applications, and Countermeasures

IEEE
IEEE PRESS

WILEY

INTRODUCTION

Information Hiding

What is „Information Hiding“? Two different examples:

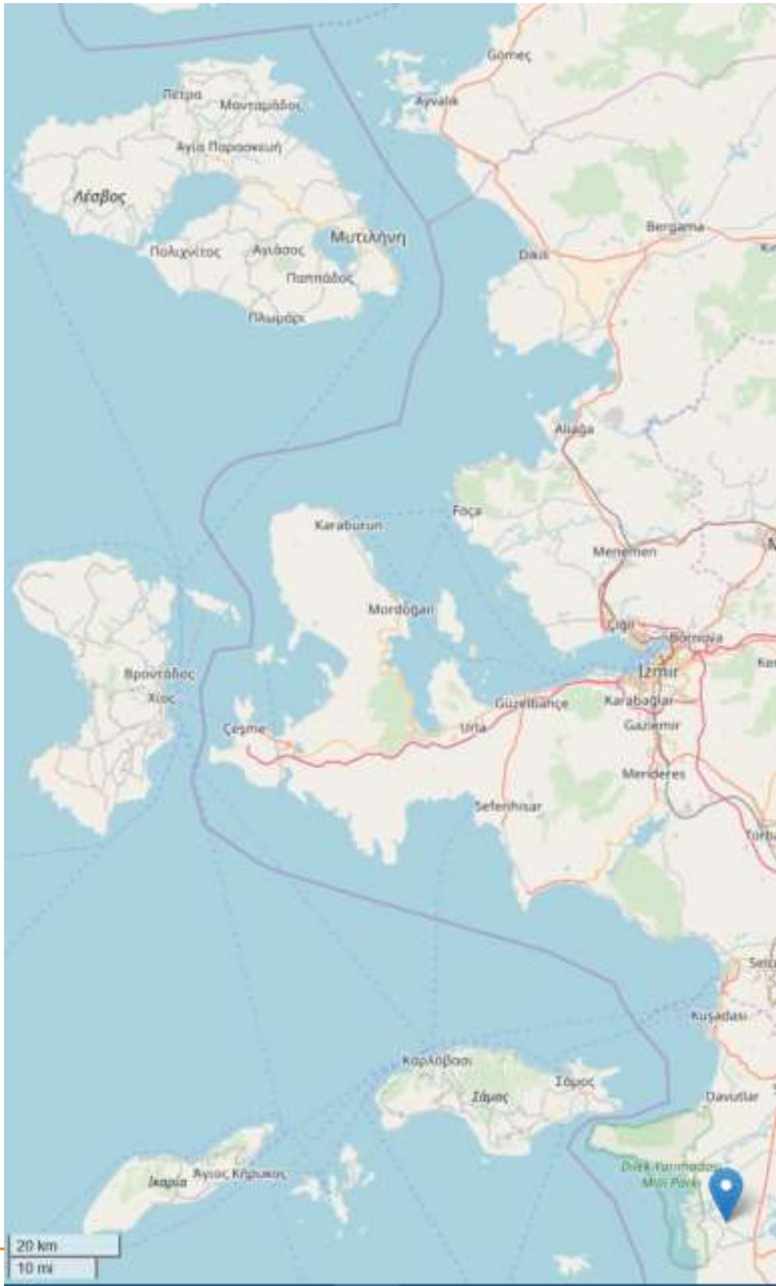
All figures taken from Wikipedia articles on ‚Steganography‘ and ‚Watermarking‘

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

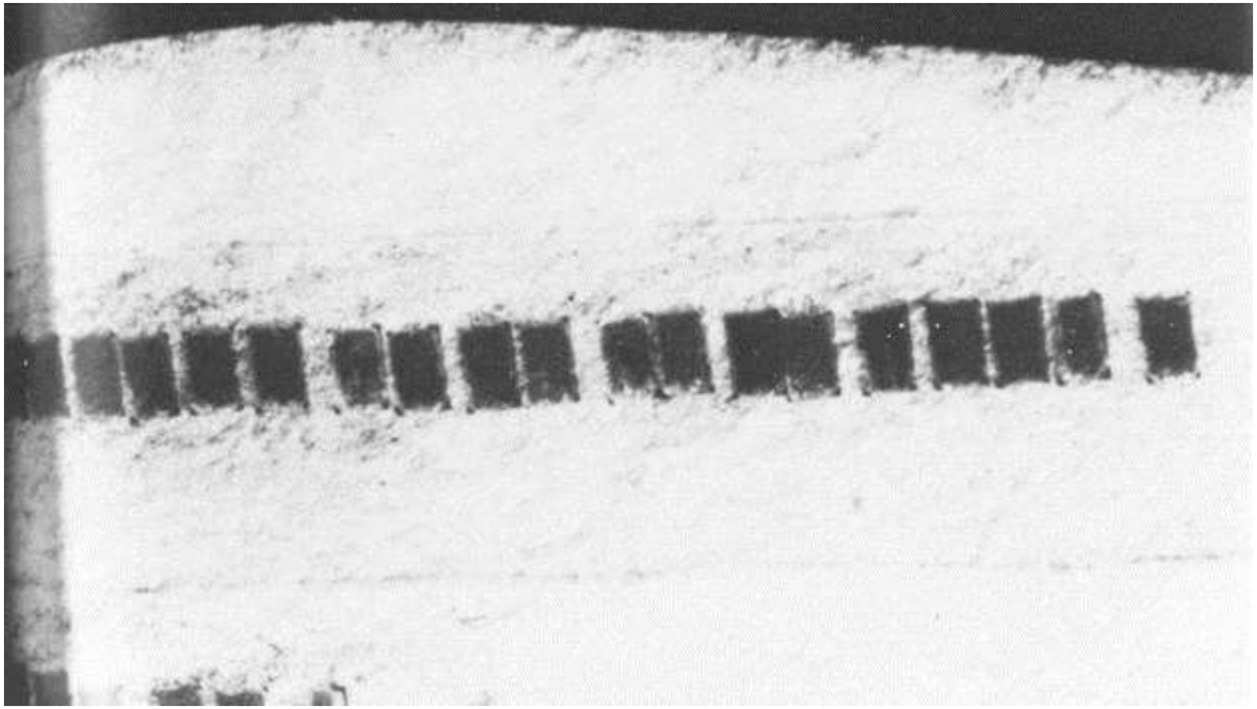
What is „Information Hiding“? Another example (from Fridrich, 2010):

- 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

Information Hiding

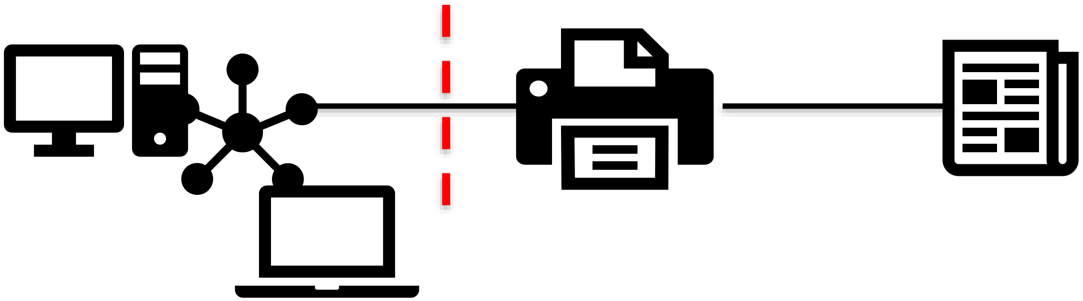
Another example: Microdots; used during WW2, e.g. by German spies in Mexico.

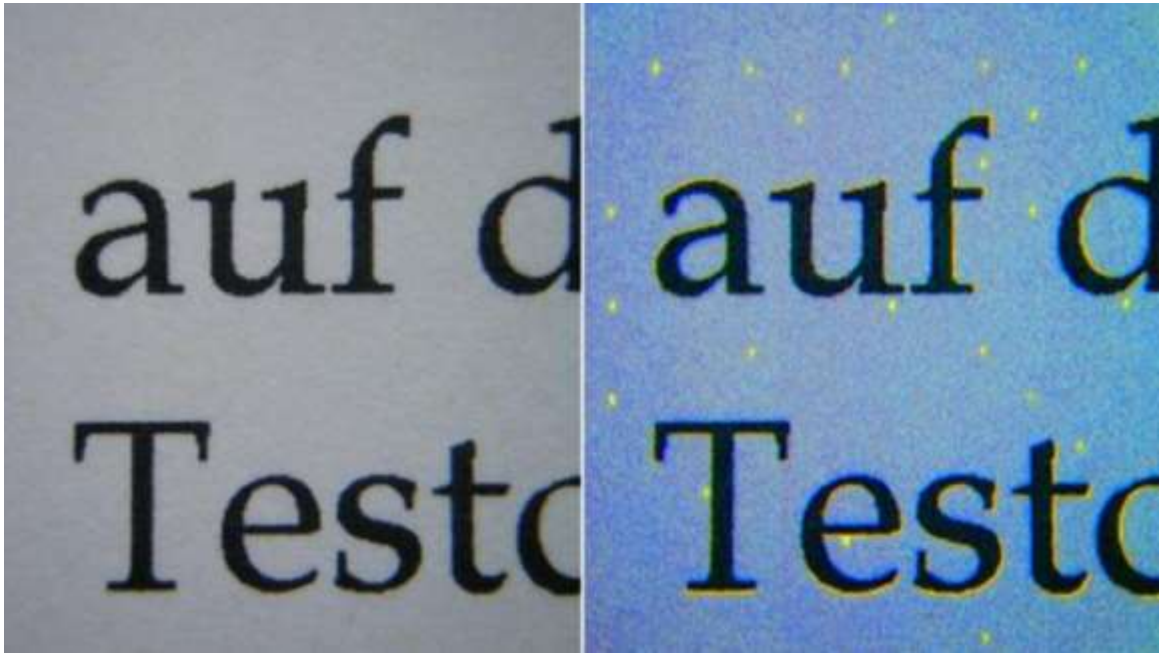


Microdots used by German spies, src: Wikipedia

Information Hiding

Another Example: Printer Watermarking

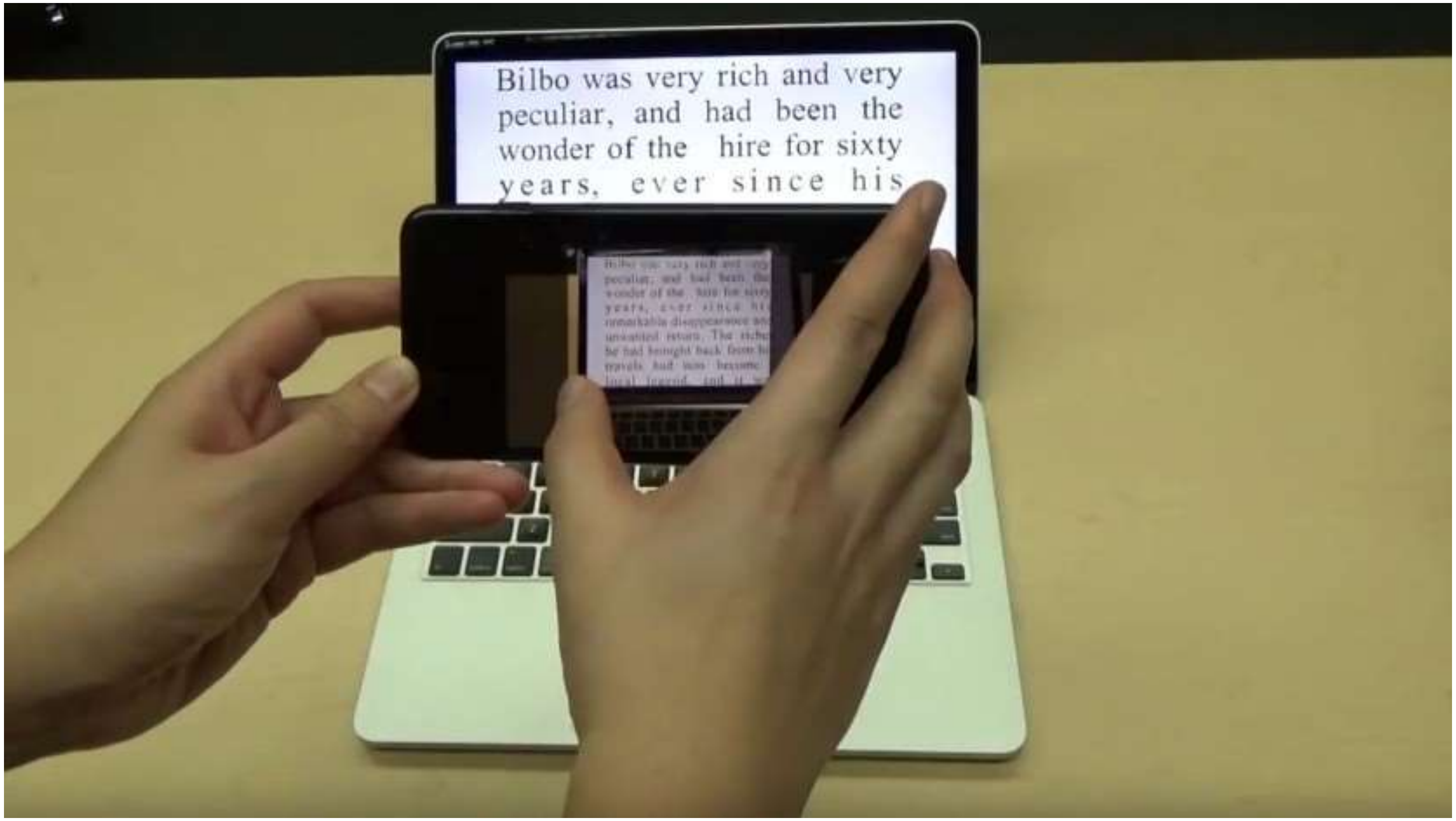




Src/Attribution: F. Heise/Wikipedia/BBC

Information Hiding

Final example: *fontcode* (works with digital and printed documents)


Bilbo was very rich and very
peculiar, and had been the
wonder of the hire for sixty
years, ever since his

Video: https://youtu.be/dejrBf9jW24

History of Information Hiding

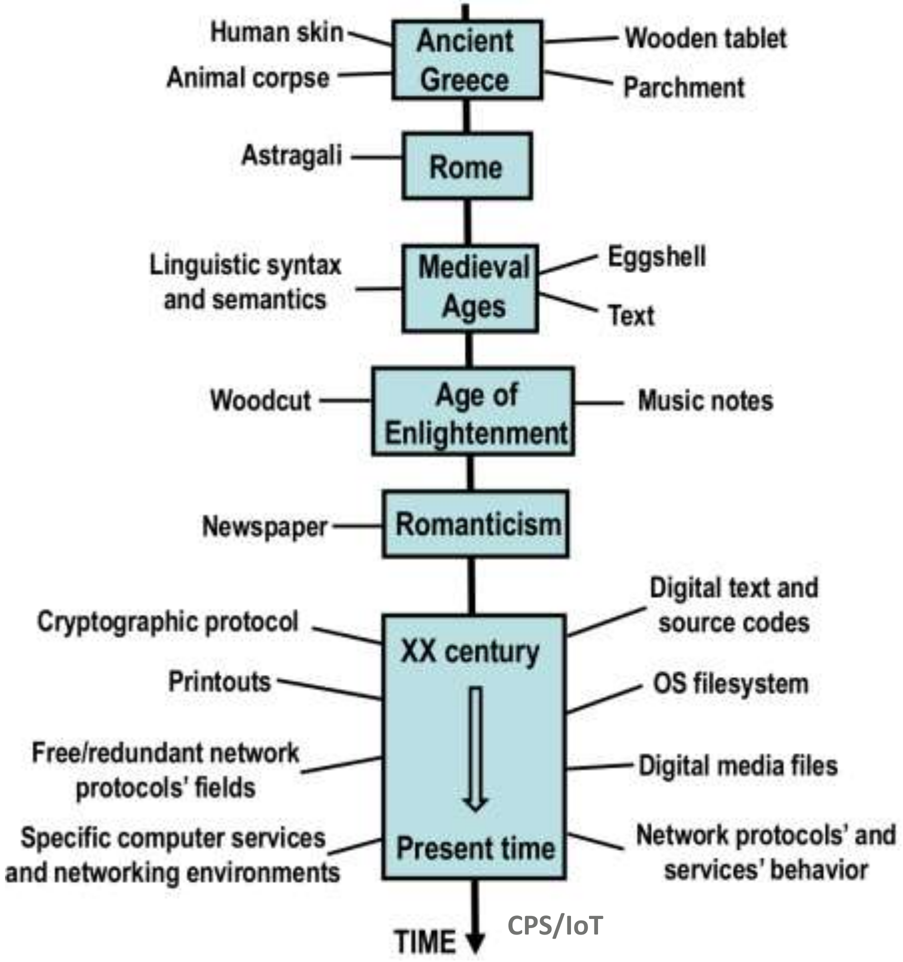


Fig. Information Hiding Methods During Time (Mazurczyk et al., 2016)

History of Information Hiding

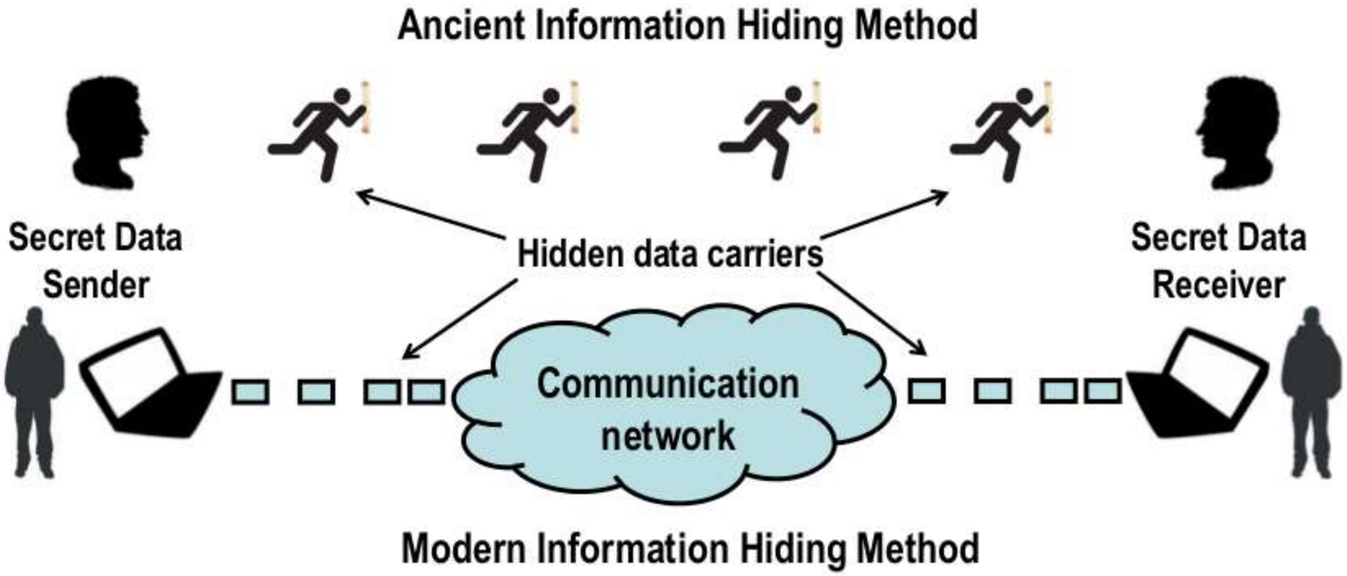


Fig. Difference between Ancient and Modern IH Methods (Mazurczyk et al., 2016)

Covert Data Storage & Communication

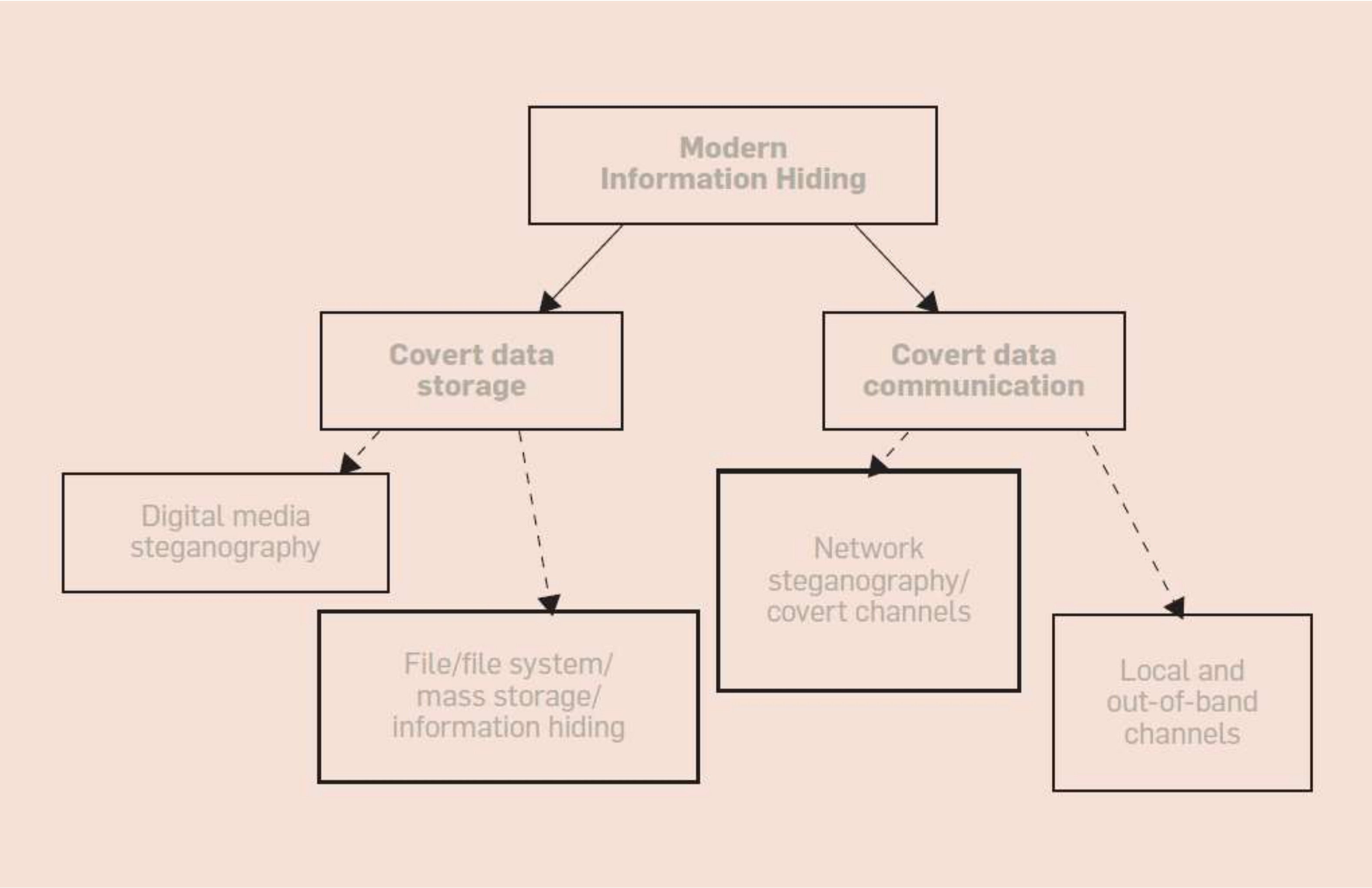
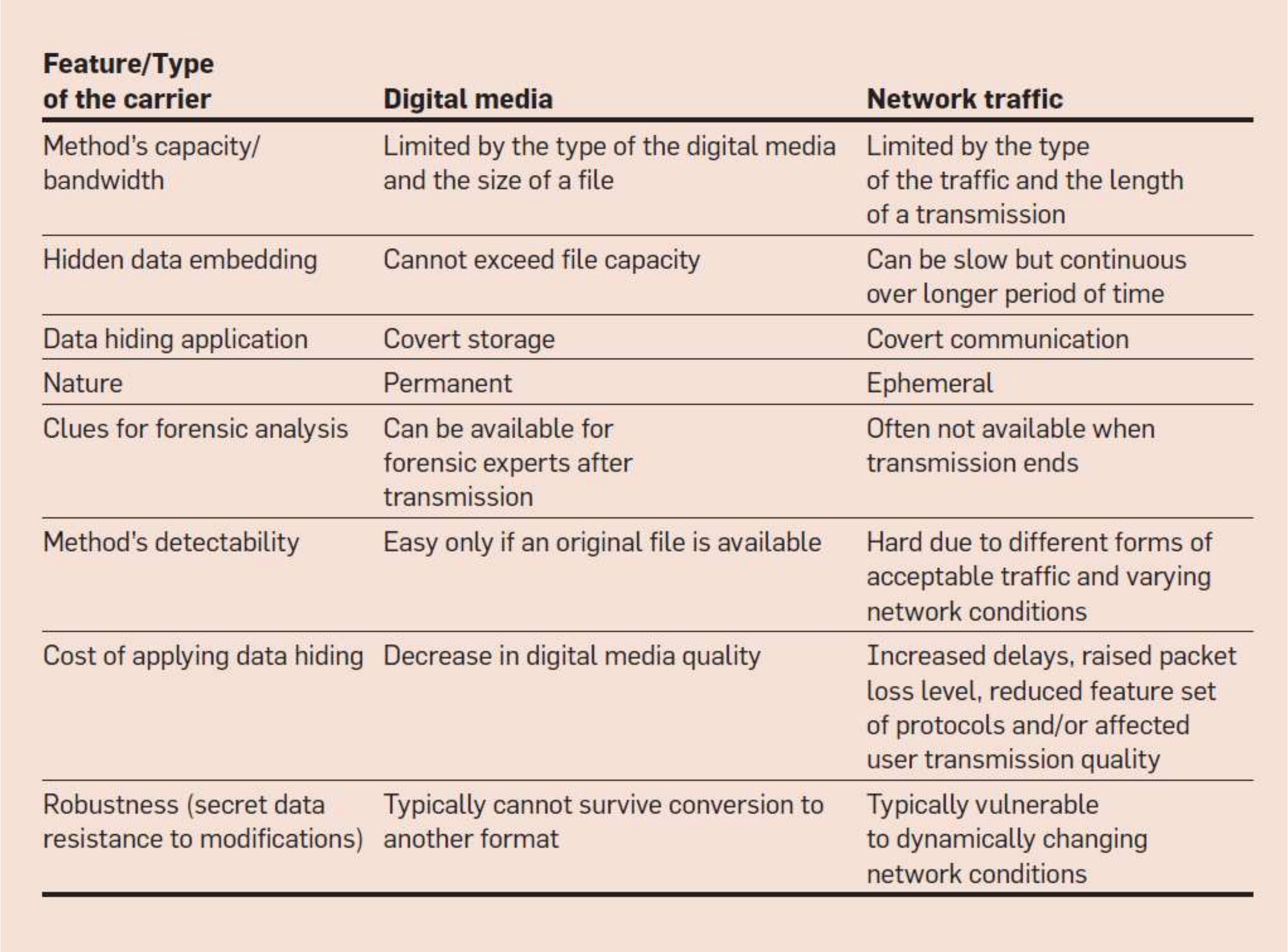

Modern Information Hiding
Covert data storage
Covert data communication
Digital media steganography
File/file system/ mass storage/ information hiding
Network steganography/ covert channels
Local and out-of-band channels

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

Application of Hiding Techniques

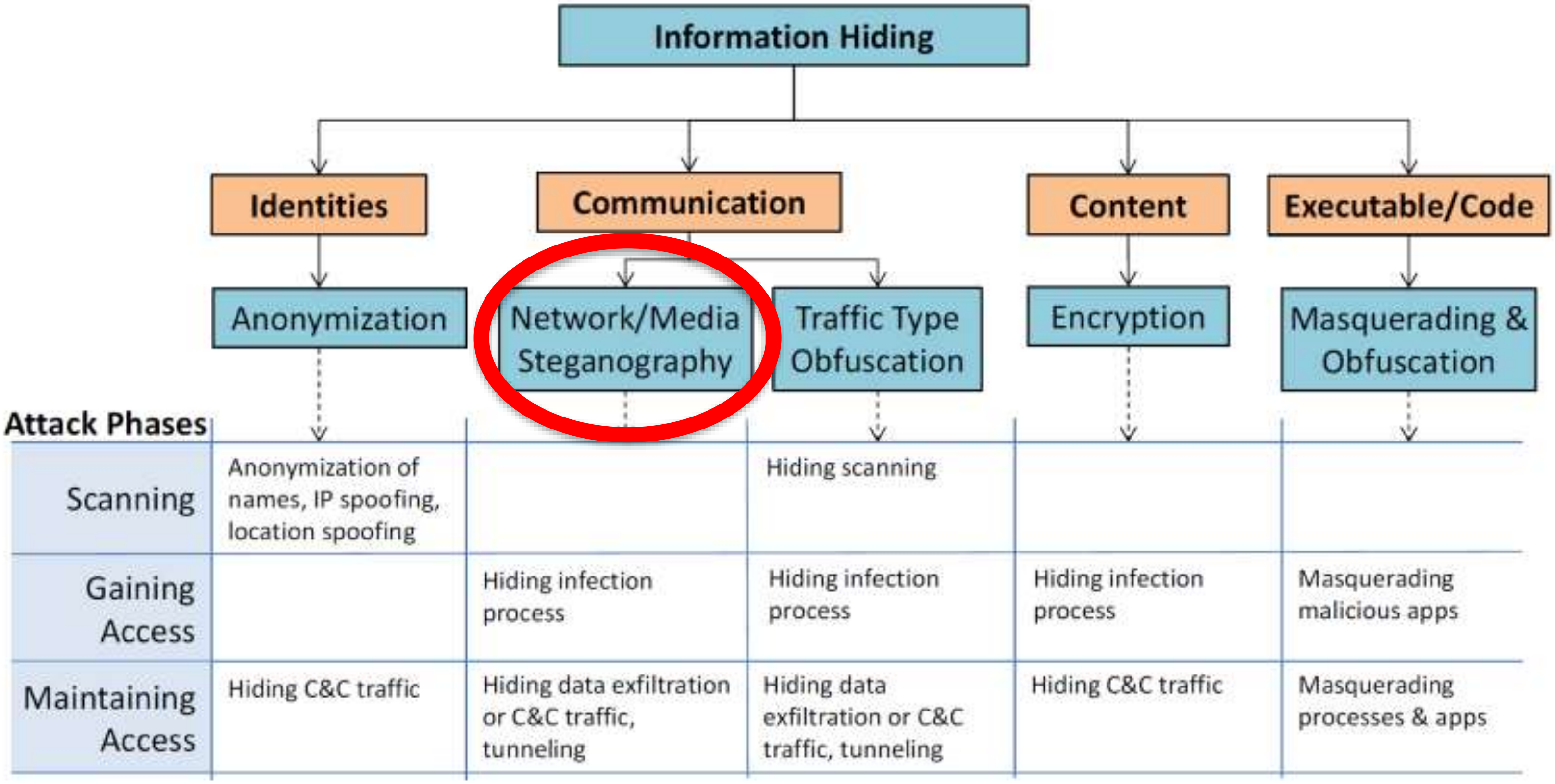
Okay, so what is the big difference between digital media and network carriers?



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.

Classification of IH techniques and their relation to basic attack phases



Attack Phases	Anonymization	Network/Media Steganography	Traffic Type Obfuscation	Encryption	Masquerading & Obfuscation
Scanning	Anonymization of names, IP spoofing, location spoofing		Hiding scanning		
Gaining Access		Hiding infection process	Hiding infection process	Hiding infection process	Masquerading malicious apps
Maintaining Access	Hiding C&C traffic	Hiding data exfiltration or C&C traffic, tunneling	Hiding data exfiltration or C&C traffic, tunneling	Hiding C&C traffic	Masquerading processes & apps

Figure 1: Classification of hiding techniques and how they are used by malware in the different attack phases

Fig.: K. Cabaj et al.: The New Threats of Information Hiding: the Road Ahead, IEEE IT Professional, 2018.

Basic Mimicry System

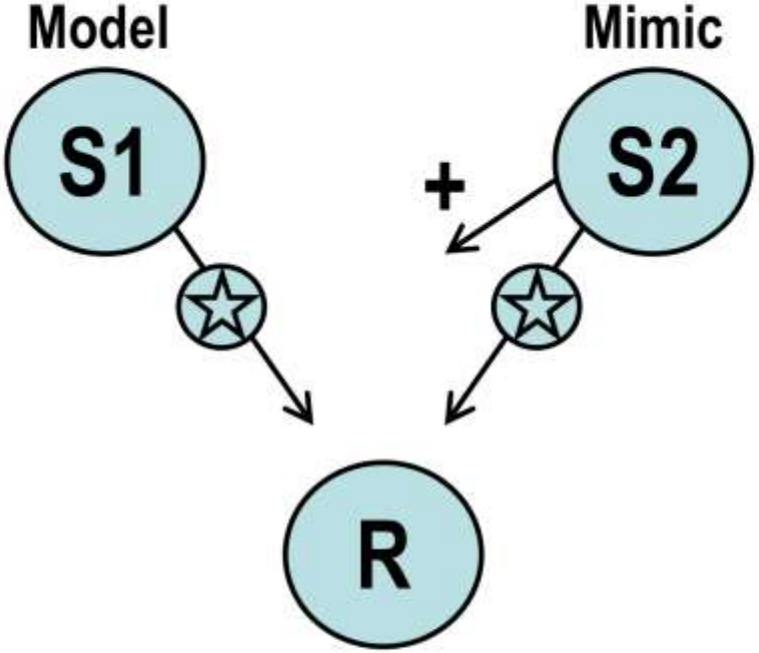
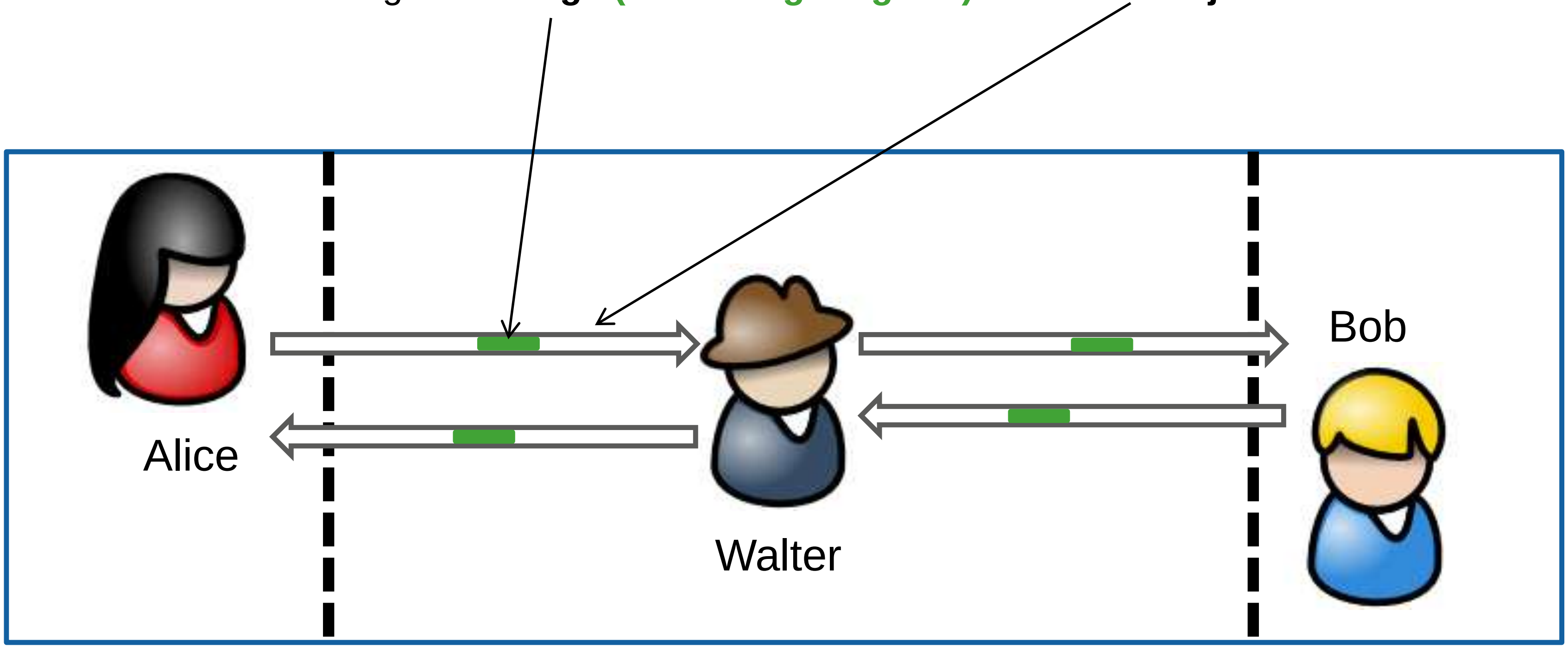


Fig. Basic mimicry system (Vane-Wright, 1976); figure from (Mazurczyk et al., 2016)

Terminology: Prisoner‘s Problem (Simmons, 1983)

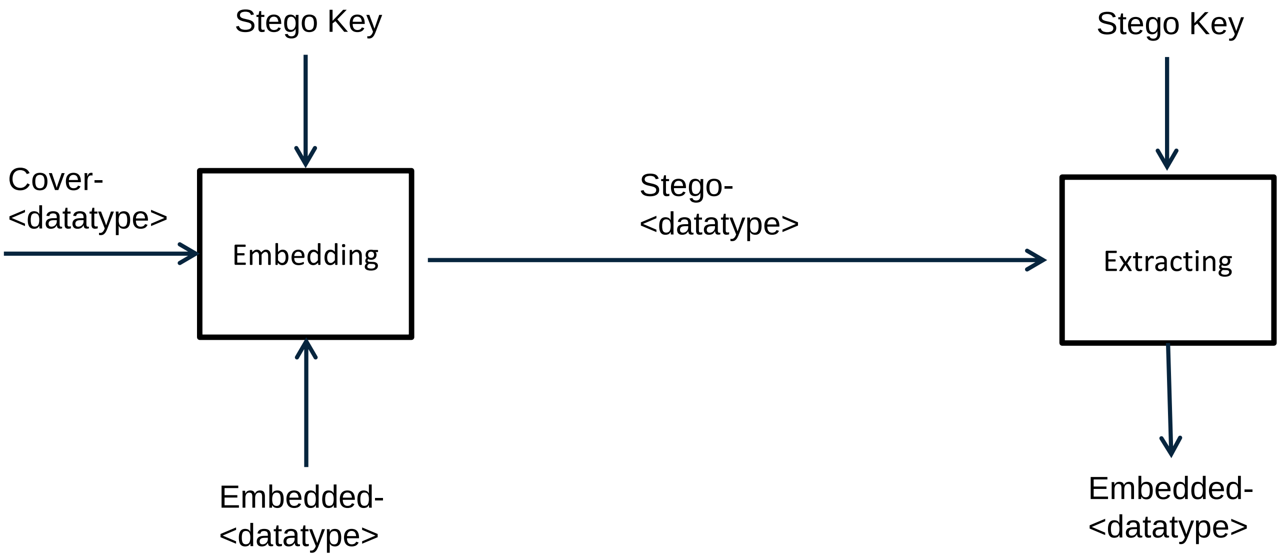
- Covert Channel (Lampson, 1973): *“…not intended for information transfer at all”*
  - A covert channel without intention is a **side channel**
  - DoD defined it differently: CCs break a security policy (usually in MLS) (DoD, 1985).

- Steganography (Fridrich, 2010):
  - “Steganography can be informally defined as the practice of undetectably communicating a **message** **(a.k.a. steganogram)** in a **cover object**.“



Terminology

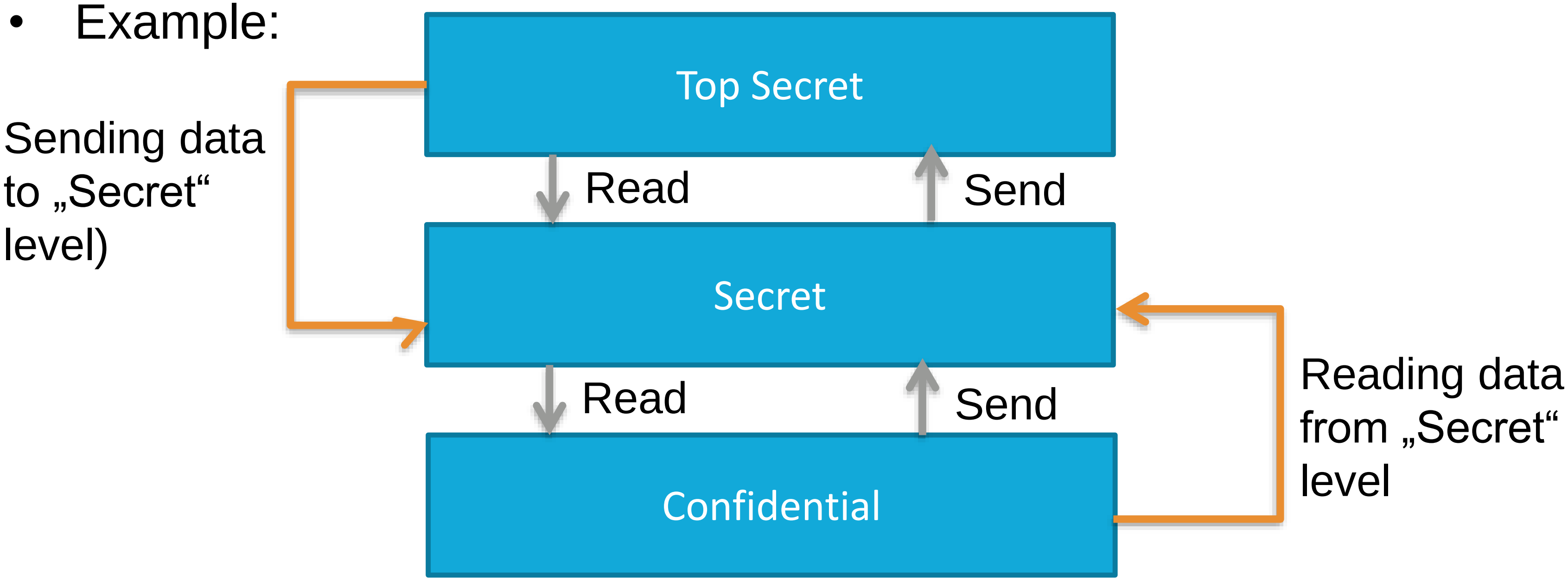
- Steganography (Fridrich, 2010):
  - "Steganography can be informally defined as the practice of undetectably communicating a **message** **(a.k.a. steganogram)** in a **cover object**."
- Terminology based on (Pfitzmann, 1996):



DoD Definition of Covert Channels in MLS Context

- In classical papers, a covert channel either violates the NRU (no read-up) or the NWD (no write-down) rule of the BLP.

- Example:



Top Secret

Read

Send

Secret

Read

Send

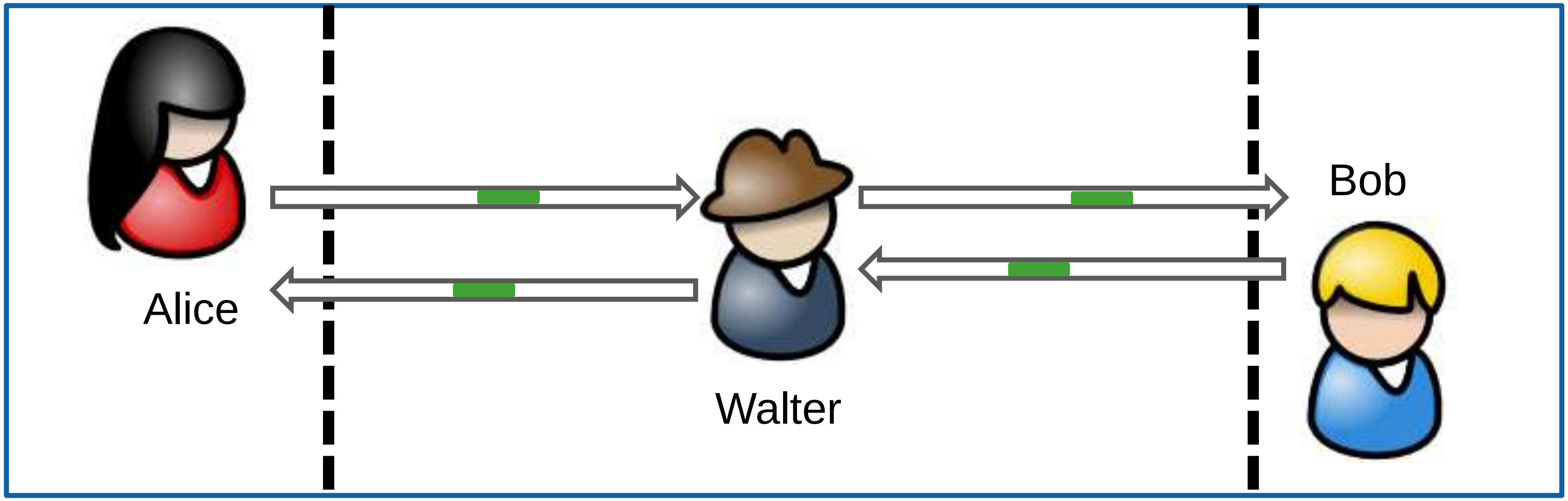
Confidential

Sending data to „Secret“ level)

Reading data from „Secret“ level

Definition

- Walter is referred to as a **warden**. He performs a so-called **steganalysis**.
- A warden can be
  - Passive
    - tries to detect the presence (and content) of a hidden message in a cover object and tries to determine who is involved in the steganographic communication
  - Active
    - Modifies the cover object (e.g. removes or replaces steganogram)
  - Malicious
    - Can introduce own messages to fool involved participants (e.g. message spoofing)



Is it applied in practice?

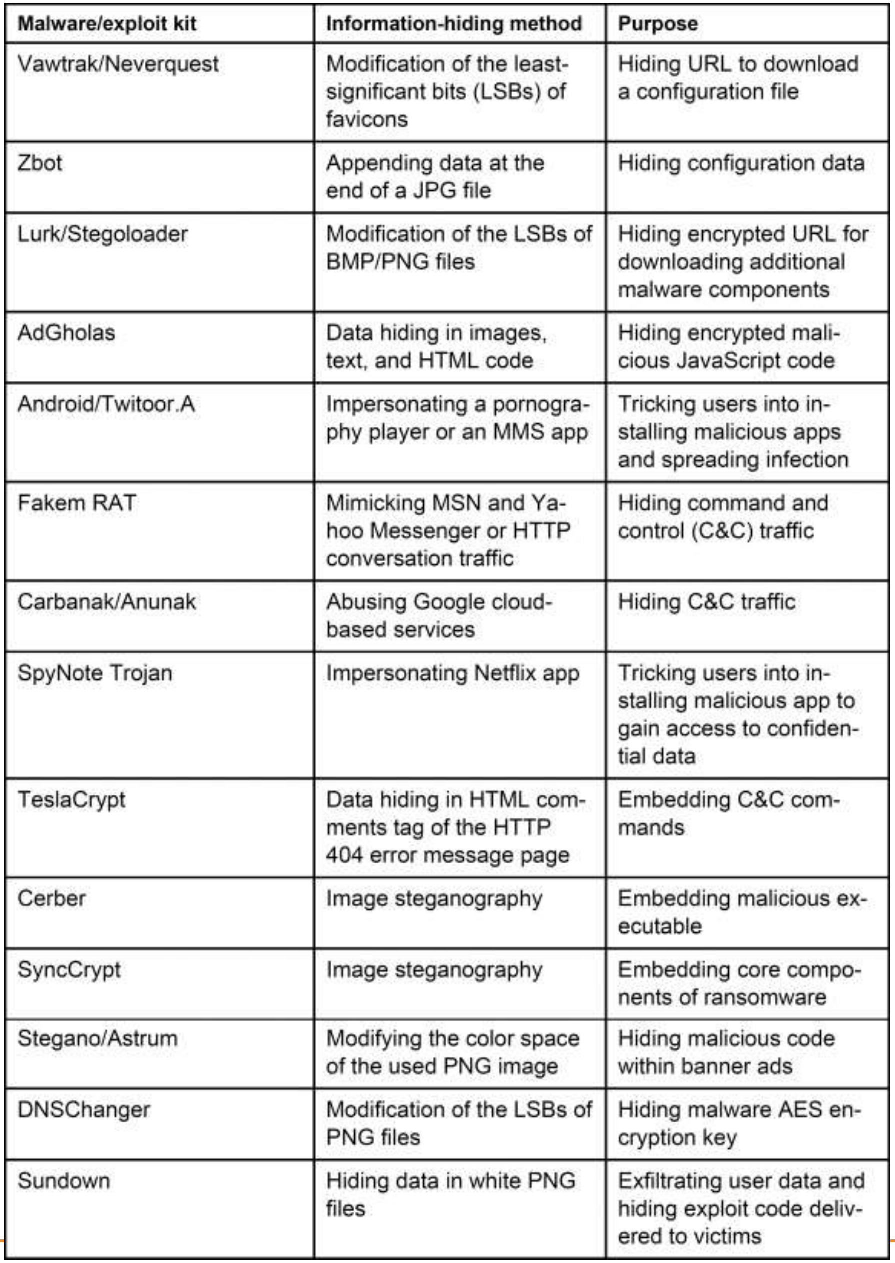
Early cases:

- 2002: „Operation Twins“ culminated in the capture of criminals associated with the „Shadowz Brotherhood“ group, a world-wide Internet pedophile organization.
  - Digital image steganography was used to hide a pornographic file within another innocent-looking one.
- 2008: Unknown person smuggled sensitive financial data out of U.S. Department of Justice using image steganography.
- 2010: Russian spy ring leaked classified information via image steganography from USA to Moscow.
- 2013: Linux Fokirtor malware hides traffic in SSH connections
- Since 2014: heaviy increase in NIH-capable malware

Sources: (1st-3rd case from Zielinska et al., 2014; 4th case from Schneier, 2013)

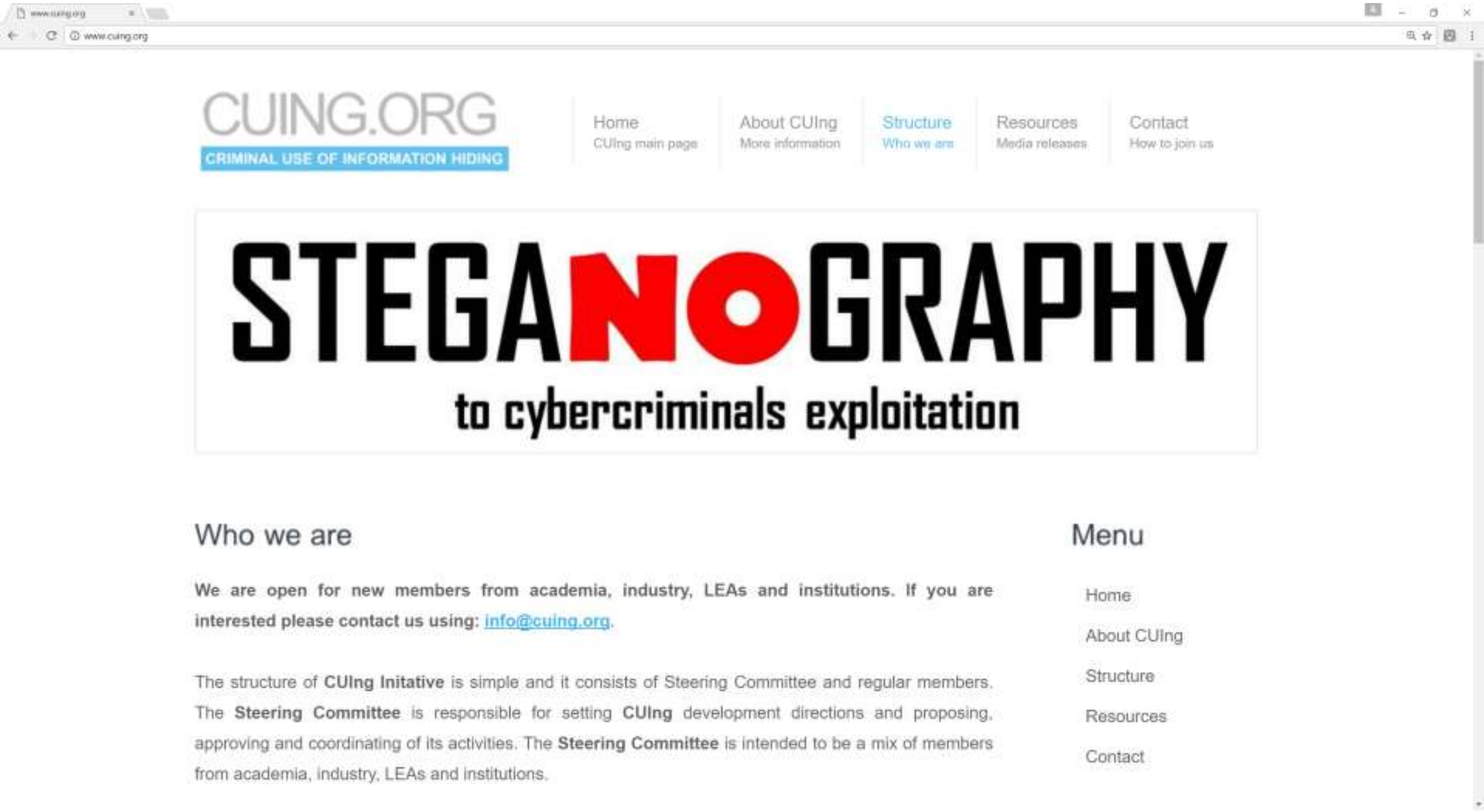
Is it applied in practice?

Several newer 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?



Who we are

We are open for new members from academia, industry, LEAs and institutions. If you are interested please contact us using: info@cuing.org.

The structure of **CUIng Initative** is simple and it consists of Steering Committee and regular members. The **Steering Committee** is responsible for setting **CUIng** development directions and proposing, approving and coordinating of its activities. The **Steering Committee** is intended to be a mix of members from academia, industry, LEAs and institutions.

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

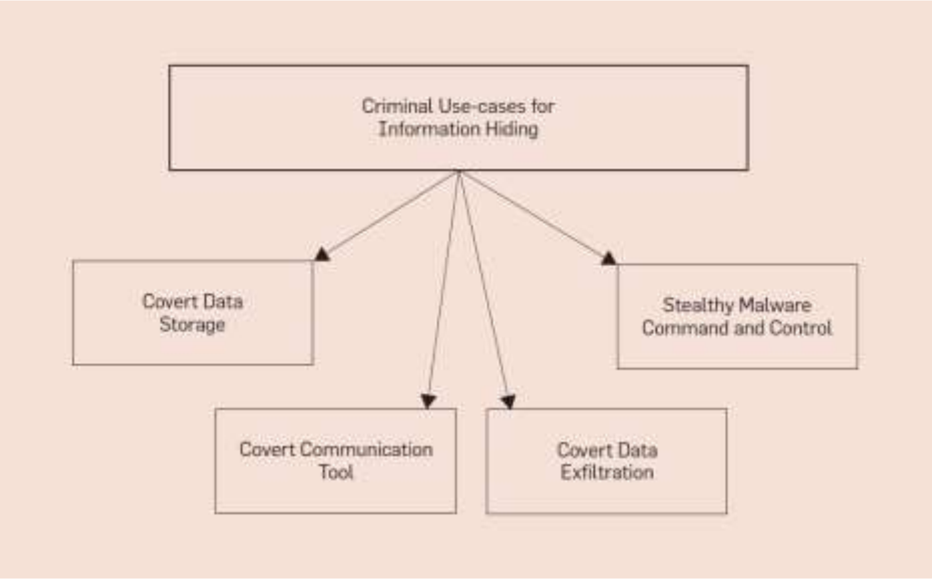

Criminal Use-cases for Information Hiding
Covert Data Storage
Stealthy Malware Command and Control
Covert Communication Tool
Covert Data Exfiltration

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

INTRODUCTION TO CLASSIC AND OS-LEVEL COVERT CHANNELS

Sample Covert Channel

- Consider two processes P_1 and P_2, running within the same environment. Some possible covert channels between the two are:

1. P_1 performs intensive computations to influence the system load (measured by P_2).
2. P_1 stops its operation at a given time t_1 or t_2 to signal a ‘0’ or a ‘1’ (P_2 monitors the process table)
3. P_1 either creates or does not create an entry in the file system known by P_2 (existence of the file signals hidden information)

These simple examples reveal that covert channels are usually not noise-free, need a protocol (when does a transmission start/end?) and need to detect errors in transmissions (e.g. using parity bits).

Covert Channels in Android (pt. 1)

- Plethora of research was conducted in recent years on covert channels in mobile phone environments.

- The goal is usually to establish a policy-breaking communication within two sandboxed apps.

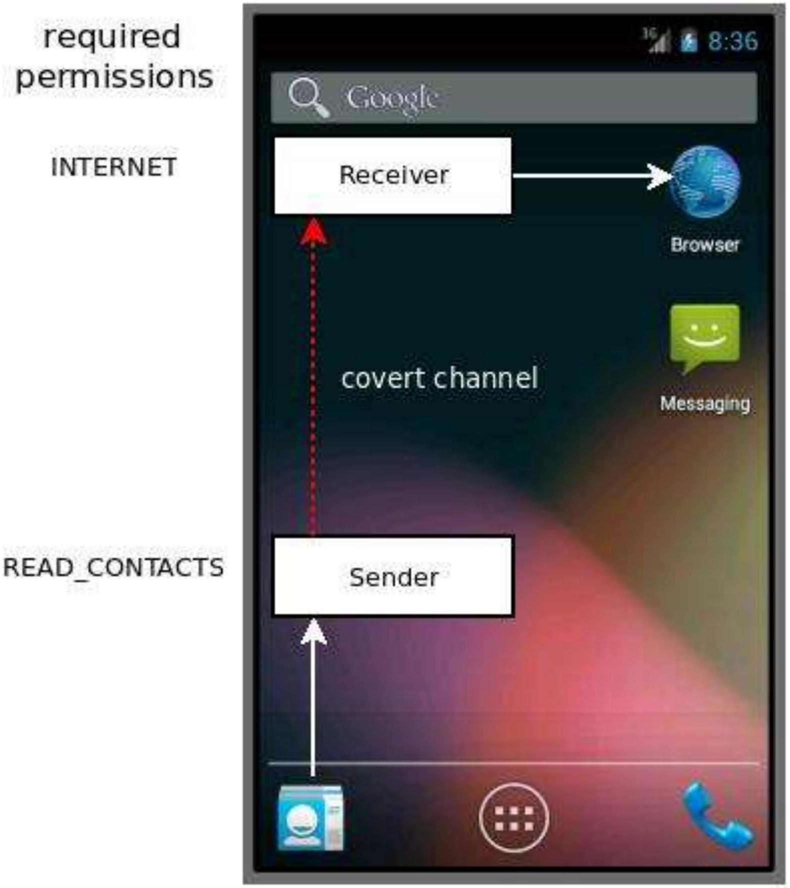
- In Android, apps have permissions, e.g. the permission to access the contacts or to use the Internet connection.

Covert Channels in Android (pt. 2)

- Example scenario using two apps (e.g. two smart home apps, one for monitoring energy consumption; one app is an energy advisor).

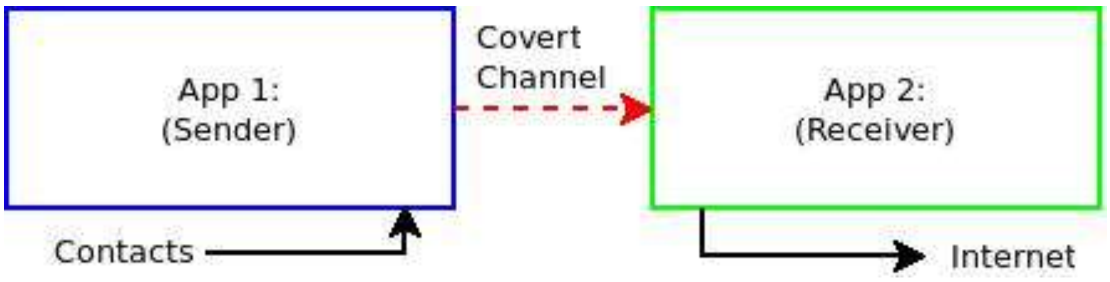
  cf. Lalande & Wendzel (2013) paper (pdf)

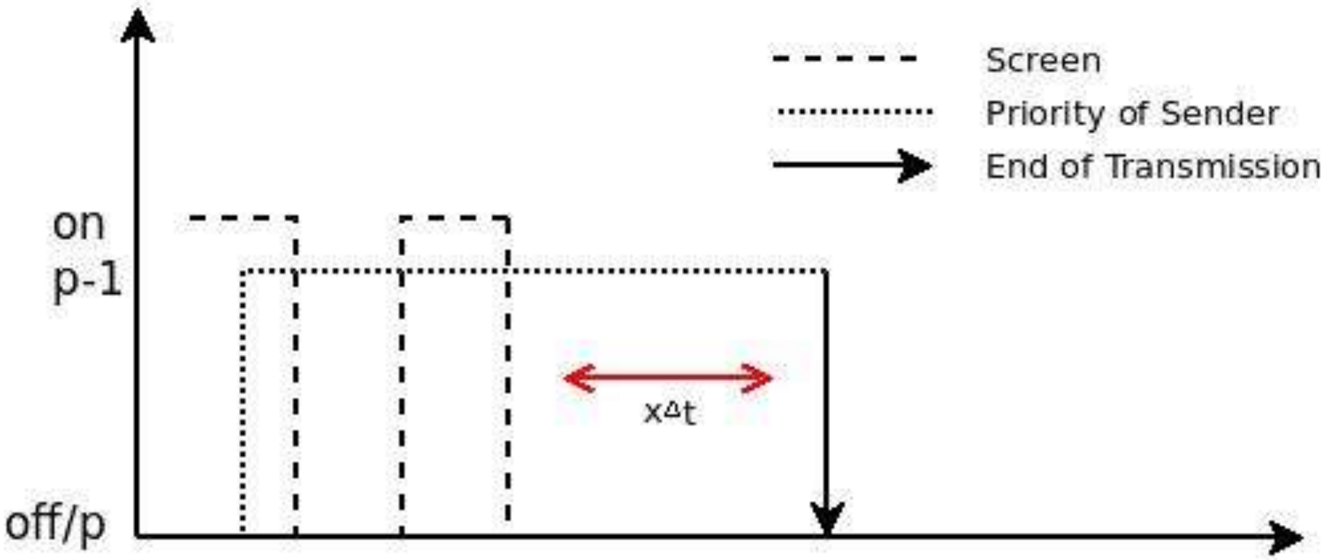
- Requirements for covert transmission:
  - Sender and receiver must run simultaneously
  - Transmission via process priority of ‚Sender‘
  - Transfer process starts when user turns of the screen



Covert Channels in Android

- How bits are transmitted (Lalande and Wendzel, 2013)





- Video:
  - http://www.dailymotion.com/video/x10lcyq_ectcm-2013-hiding-privacy-leaks-in-android-applications_tech

- Original slides:
  - http://www.wendzel.de/dr.org/files/Papers/ares13_slides.pdf

INTRODUCTION TO FUNDAMENTAL COUNTERMEASURES

SRM

- *Shared Resource Matrix* Methodology (SRM)

  (Kemmerer, 1983) and (Bishop, 2003)

  - General approach to detect covert storage channels
  - Can be applied at different steps of SDL
    - Covert channels can be detected within textual specifications of a software
    - … but also in source code
  - The idea was later improved by McHugh, but we focus only on the original version introduced by Kemmerer.
  - General assumption: A system is described by “operations” and “attributes”.

SRM

- Goal of the SRM is to determine whether an Operation X can modify (M) an attribute A under the condition that an Operation Y (w/ Y≠X) can read (R) attribute A.

- **Example:** Let us assume that:

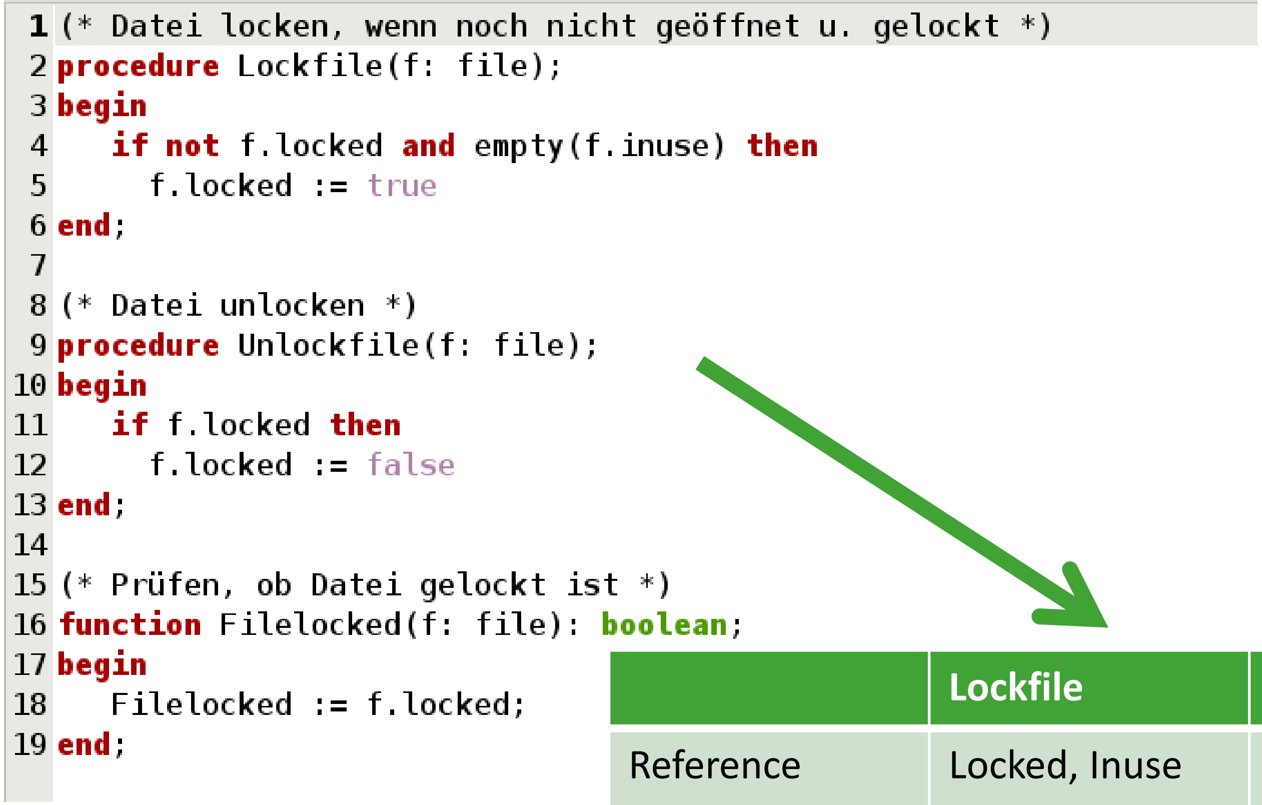
Attr. / Op.	Read	Write	Delete	Create
Existence of file	R	R	R, M	R, M
File owner	-	-	R	M
File name	R	R	R	M
File size	R	M	M	M

SRM

- Problems:
  - Some „covert channels“ can be false positives (e.g. if two operations could build an (R,M) pair but cannot be called by processes of different security levels).
  - The SRM supports no sequences of operations but a sequence of *n* operations may lead to an **indirect recognition** of a modified attribute (Bishop, 2003).
  - Kemmerer states that all storage and timing channels can be detected using the SRM. However, Bishop stated that this is wrong (see above).

Covert Flow Trees

- Covert Flow Trees (Kemmerer and Porras, 1991; Bishop, 2003)



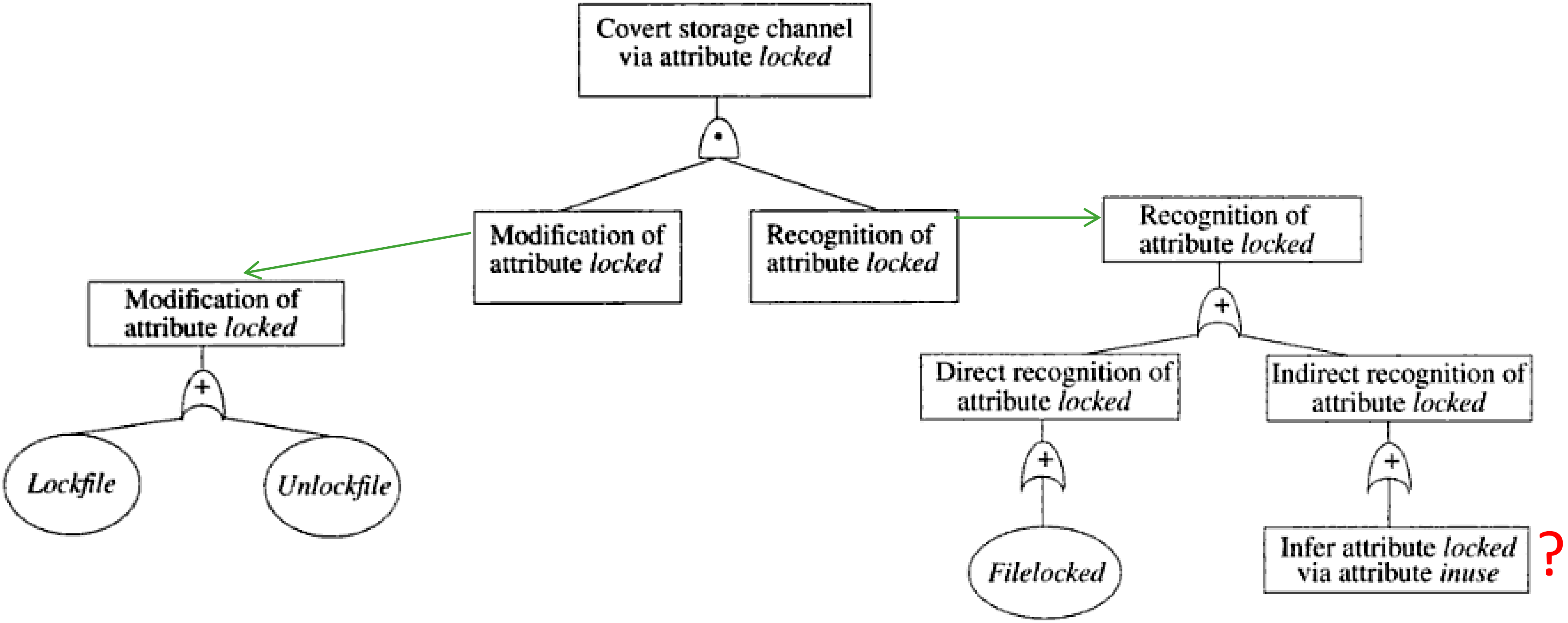
```
(* Datei locken, wenn noch nicht geöffnet u. gelockt *)
procedure Lockfile(f: file);
begin
   if not f.locked and empty(f.inuse) then
     f.locked := true
end;

(* Datei unlocken *)
procedure Unlockfile(f: file);
begin
   if f.locked then
     f.locked := false
end;

(* Prüfen, ob Datei gelockt ist *)
function Filelocked(f: file): boolean;
begin
   Filelocked := f.locked;
end;
```

	Lockfile	Unlockfile	Filelocked
Reference	Locked, Inuse	Locked	Locked
Modify	Locked	Locked	-
Return	-	-	Locked

Covert Flow Trees

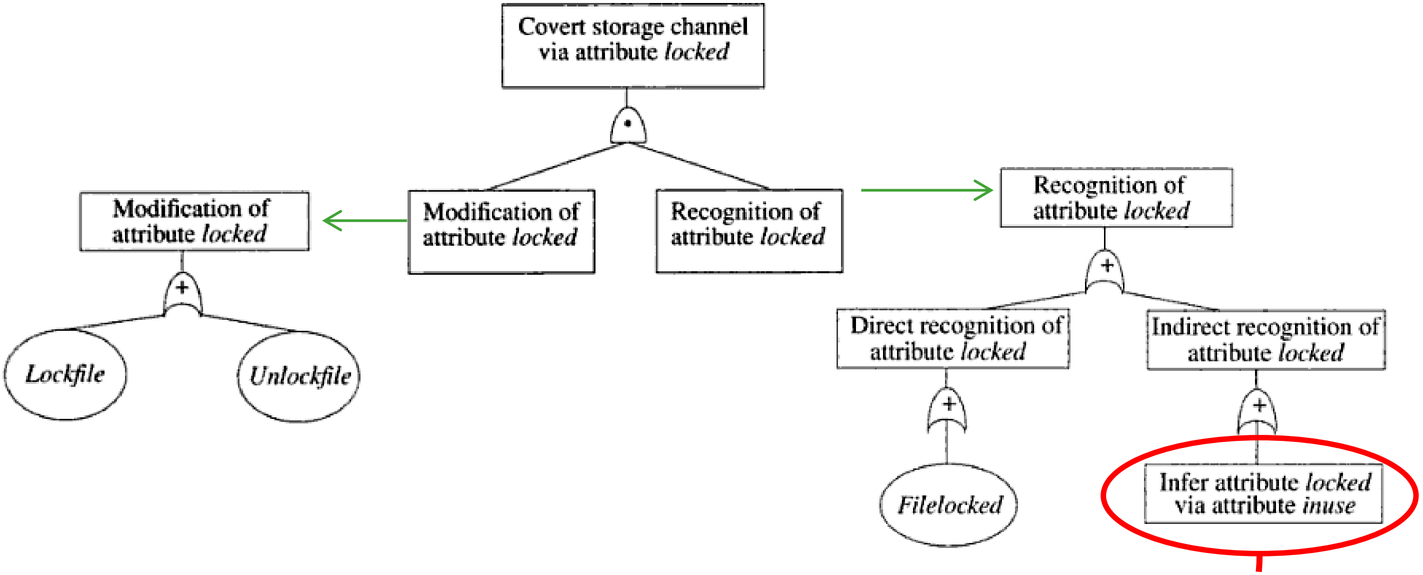
- Covert Flow Trees (Kemmerer and Porras, 1991; Bishop, 2003)
  - In comparison to SRM, CFTs can only be applied at the code level.

	Lockfile	Unlockfile	Filelocked
Reference	Locked, Inuse	Locked	Locked
Modify	Locked	Locked	-
Return	-	-	Locked


Covert storage channel via attribute *locked*
Modification of attribute *locked*
Recognition of attribute *locked*
Modification of attribute *locked*
Lockfile
Unlockfile
Recognition of attribute *locked*
Direct recognition of attribute *locked*
Indirect recognition of attribute *locked*
Filelocked
Infer attribute *locked* via attribute *inuse*
?

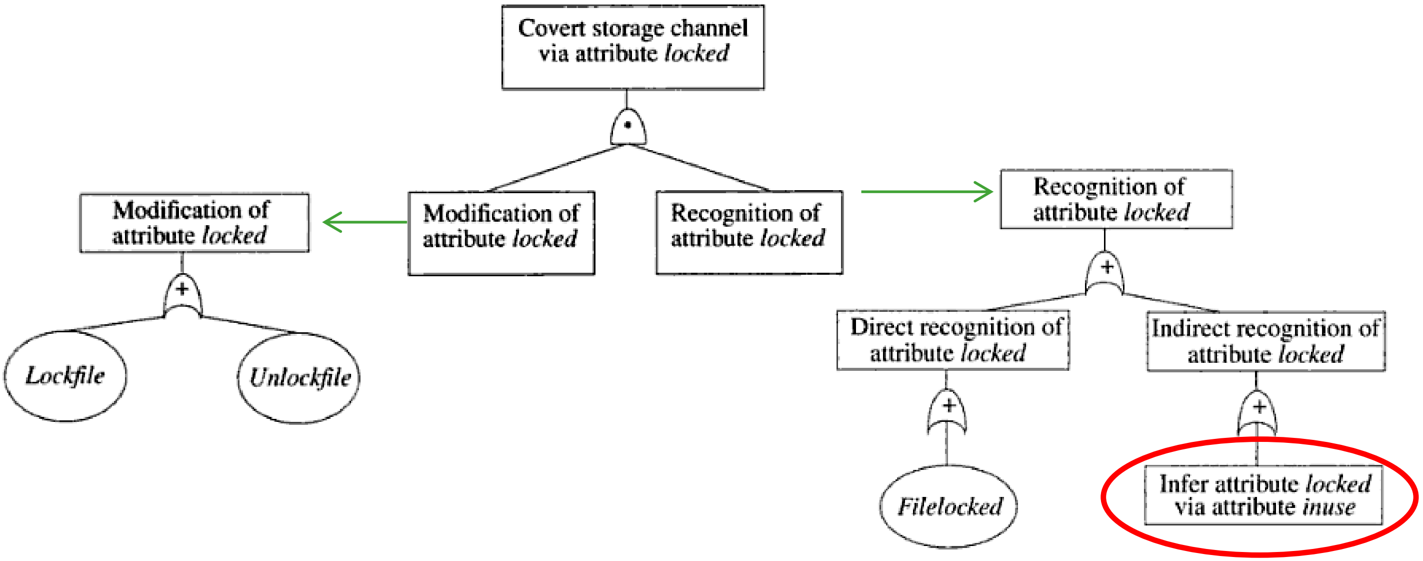
Covert Flow Trees

- Covert Flow Trees (Kemmerer and Porras, 1991; Bishop, 2003)


Covert storage channel via attribute *locked*
Modification of attribute *locked*
Modification of attribute *locked*
Recognition of attribute *locked*
Recognition of attribute *locked*
Lockfile
Unlockfile
Direct recognition of attribute *locked*
Indirect recognition of attribute *locked*
Filelocked
Infer attribute *locked* via attribute *inuse*

- Generation of CFT lists: They represent sequences of operations that represent a potential covert channel.
  - List 1: Operations capable of modifying an attribute
    List 2: Operations capable of reading an attribute
  - List 1: ((Lockfile), (Unlockfile))
    List 2: ((Filelocked), (false))

Covert Flow Trees


Covert storage channel via attribute *locked*
Modification of attribute *locked*
Recognition of attribute *locked*
Modification of attribute *locked*
Lockfile
Unlockfile
Recognition of attribute *locked*
Direct recognition of attribute *locked*
Indirect recognition of attribute *locked*
Filelocked
Infer attribute *locked* via attribute *inuse*

- Finally, we combine both lists to determine the potential covert channel‘s flows:

  Lockfile -> Filelocked

  Unlockfile -> Filelocked

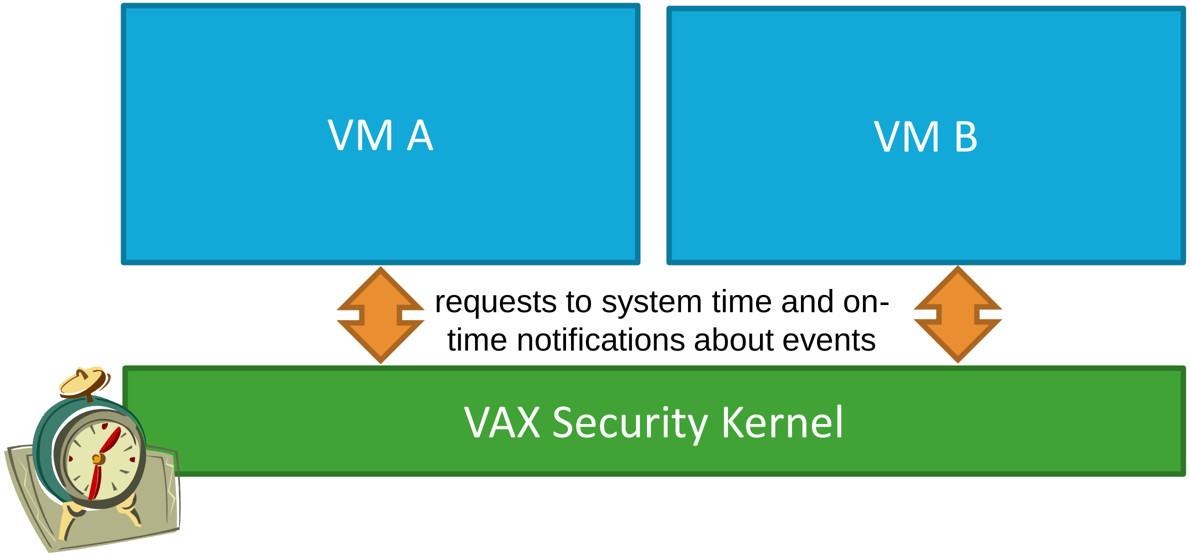
  These two flows can also be used together to form a covert storage channel with two states.

Covert Flow Trees

- Covert Flow Trees (Kemmerer and Porras, 1991; Bishop, 2003)
- Discussion:
  - CFTs can only be applied at the source code level (drawback in comparison to the SRM)
  - Nobody has published work on timing channel detection; so far, CFTs can only be applied to detect storage channels
  - Visual representation of flows and automatic CFT generation supported by tools
  - Support for indirect information flows

Fuzzy Time

- Fuzzy Time (Lu, 1991)
  - Approach to limit the channel capacity of covert timing channels between virtual machines; already in 1991 (VAX security kernel).
  - The more precise a time measurement is, the higher is the channel capacity (finer distinction of elapsed time possible).
  - No detection or prevention of timing channels.



VM A

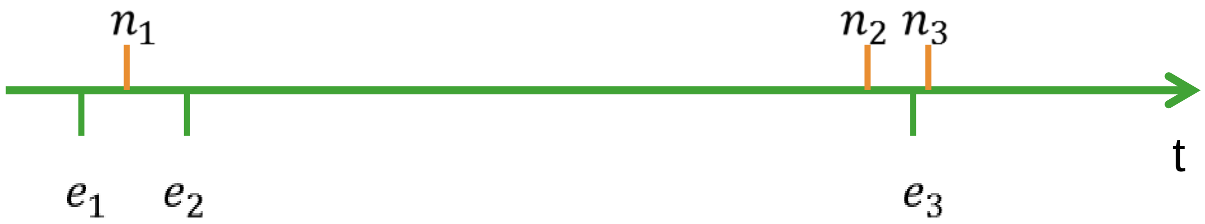
VM B

requests to system time and on-time notifications about events

VAX Security Kernel

Fuzzy Time

Notification Time (Upticks)



Event Time (Downticks)

NETWORK INFORMATION HIDING

Definition

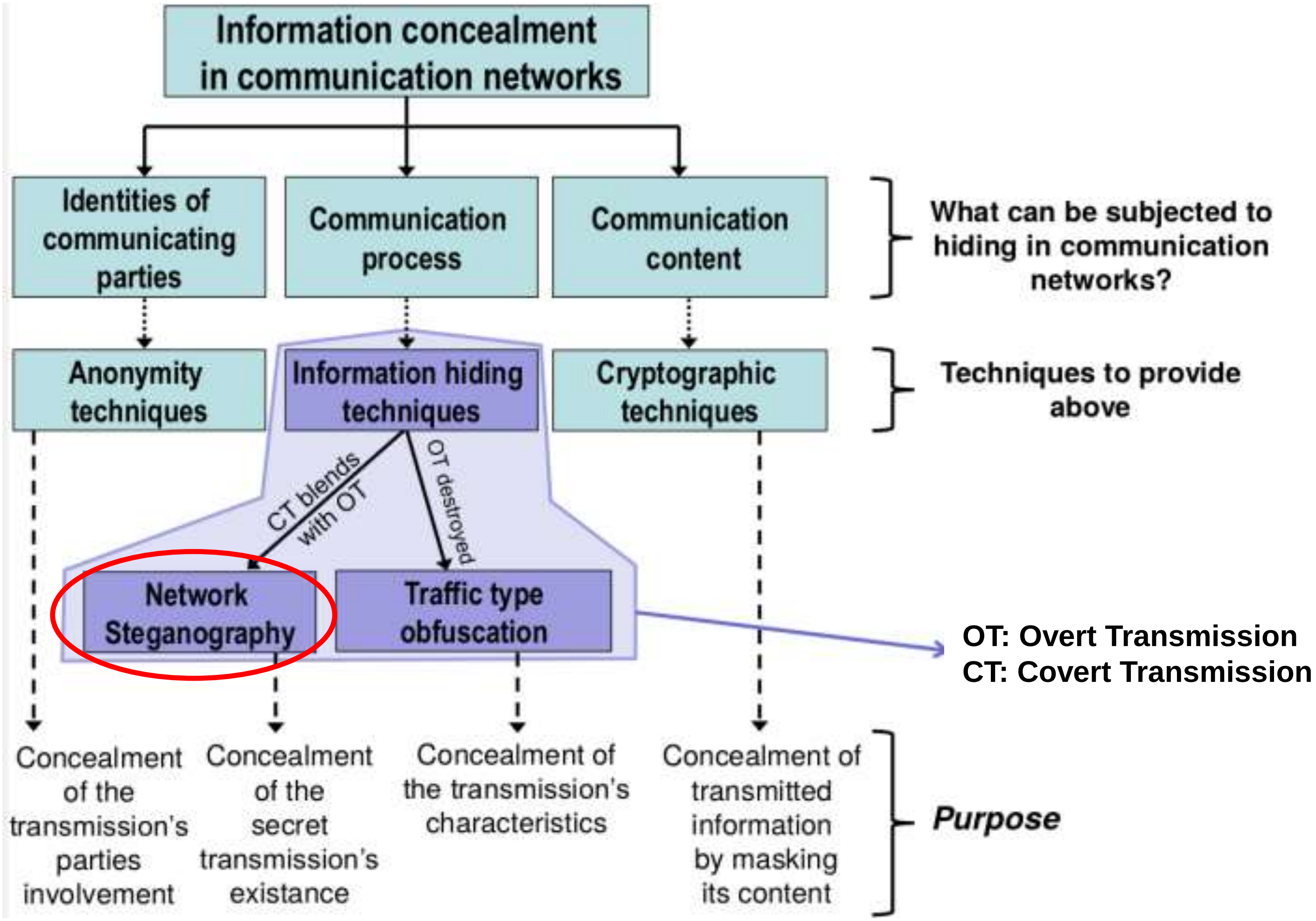
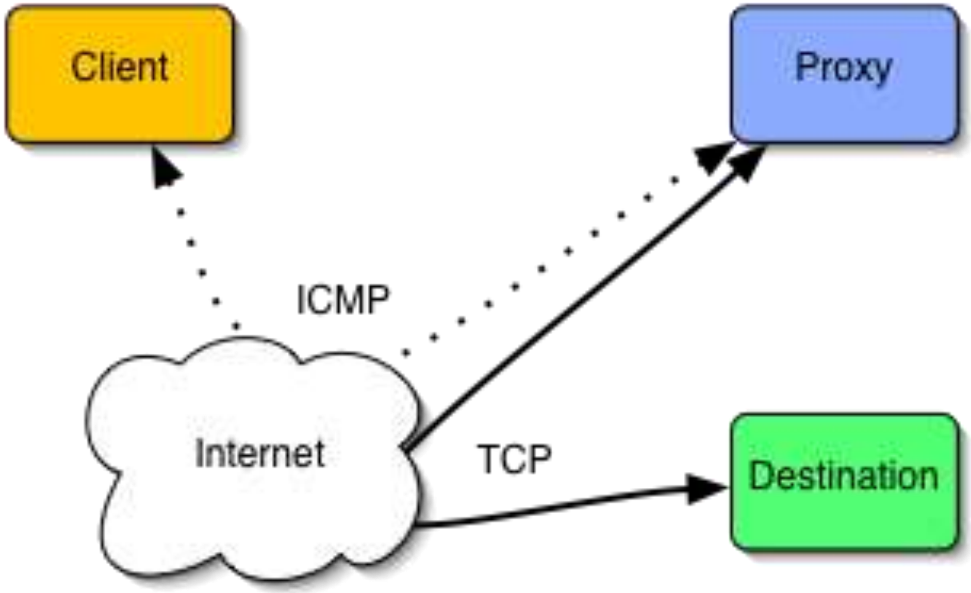

Information concealment in communication networks
Identities of communicating parties
Communication process
Communication content
What can be subjected to hiding in communication networks?
Anonymity techniques
Information hiding techniques
Cryptographic techniques
Techniques to provide above
CT blends with OT
OT destroyed
Network Steganography
Traffic type obfuscation
OT: Overt Transmission
CT: Covert Transmission
Concealment of the transmission's parties involvement
Concealment of the secret transmission's existance
Concealment of the transmission's characteristics
Concealment of transmitted information by masking its content
Purpose

Fig. Classification of Information Hiding Techniques (Mazurczyk et al., 2016)

Differences to **traditional** digital media steganography

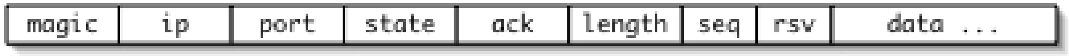
- No clear distinction between **steganography** and **covert channel**
  - Instead: **network covert channel** or **network steganographic channel handled separately**
  - Unified: a steganographic **method** creates such a **covert channel** (Mazurczyk, 2016)
- Covert data is hidden in overt network transmissions
- The „cover object“ is now called „carrier“
- Advantage of a constant transmission (e.g. permanent data leakage)
- Difficult to analyze **all** network data
- Smaller delay
- With the growth of the Internet, the options for network IH grew and grow, too.

Simple Example: Ping Tunnel



Analysis and improvements: 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.

Secret data is embedded into the ICMP echo payload.
In addition, a small protocol of the following format is used:



magic	ip	port	state	ack	length	seq	rsv	data ...

Figs.: http://www.cs.uit.no/%7Edaniels/PingTunnel/

Types of (Network) Covert Channels

- Local and network covert channels
- Storage and timing channels
- Active and passive covert channels
  - e.g. Rutkowska‘s work (and earlier academic work)
- Intentional (covert) and unintentional (side) channels
  - e.g. side channels in web applications
- Noisy and noise-free covert channels
- Direct and indirect covert channels
  - e.g. via web page + server load

References

- Bishop, M.: Computer Security, Art and Science, Addison-Wesley Professional, 2003, Chapter 17.
- DoD: Trusted Computer System Evaluation Criteria (TCSEC), Department of Defense, 1985.
- Fadlalla, Y. A. H.: Approaches to Resolving Covert Storage Channels in Multilevel Secure Systems, PhD thesis, University of New Brunswick, 1996.
- Fridrich, J.: Steganography in Digital Media, Cambridge University Press, 2010.
- Hu, W.-M.: Reducing Timing Channels with Fuzzy Time, Symp. Security and Privacy, IEEE, 1991.
- Kemmerer, R.: Shared resource matrix methodology: an approach to identifying storage and timing channels, Trans. Comp. Systems, ACM, 1983.
- Kemmerer, R., Porras, P.: Covert Flow Trees: A Visual Approach to Analyzing Covert Storage Channels, Trans. Software Engineering, IEEE, 1991.
- Lalande, J.F., Wendzel, S.: Hiding privacy leaks in Android applications using low-attention raising covert channels, Proc. 8th ARES, Regensburg, DE, IEEE, 2013.
- Lampson, B.W.: A Note on the Confinement Problem, Comm. ACM, 1973.
- Mazurczyk, W., Wendzel, S., Zander, S. et al.: Information Hiding in Communication Networks, Wiley / IEEE Comp. Soc. Press, 2016.
- Mazurczyk, W., Wendzel, S.: Information Hiding: Challenges for Forensic Experts, Communications of the ACM, 2017 (in press).
- Petitcolas, F.A.P., Anderson, R., Kuhn, M.G.: Information Hiding – A Survey, Proc. IEEE, 1999.
- Pfitzmann, B.: Information Hiding Terminology, Proc. 1st Information Hiding Workshop, Springer, 1996.
- Schneier, B.: Fokirtor, https://www.schneier.com/blog/archives/2013/11/fokirtor.html, Nov. 2013.
- Simmons, G.: The Prisoner‘s Problem and the Subliminal Channel, Symp. Security and Privacy, IEEE, 1983.
- Vane-Wright, R. I.: A unified classification of mimetic resemblances, Biological Journal of the Linnean Society, 1976.
- Wendzel, S., Mazurczyk, W., Zander, S.: Unified Description for Network Information Hiding Methods, Journal of Universal Computer Science, 2016. https://www.researchgate.net/profile/Steffen_Wendzel/publication/288059908_Unified_Description_for_Network_Information_Hiding_Methods/links/56a3812608ae232fb2057942.pdf?origin=publication_list
- Zielinska, E., Mazurczyk, W., Szczypiorski, K: Trends in steganography, Comm. ACM, 2014.

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

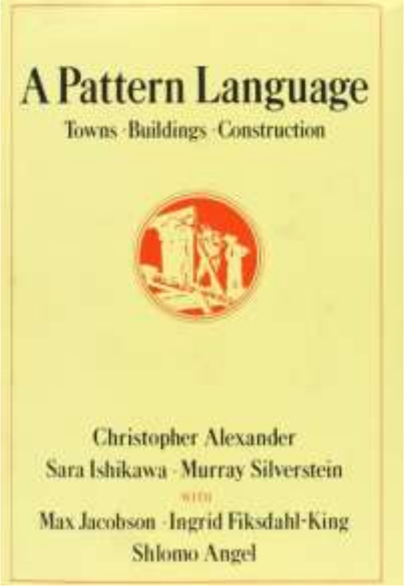
HIDING PATTERNS

(CLEANING UP NETWORK IH)

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



A Pattern Language

Towns · Buildings · Construction

Christopher Alexander

Sara Ishikawa · Murray Silverstein

with

Max Jacobson · Ingrid Fiksdahl-King

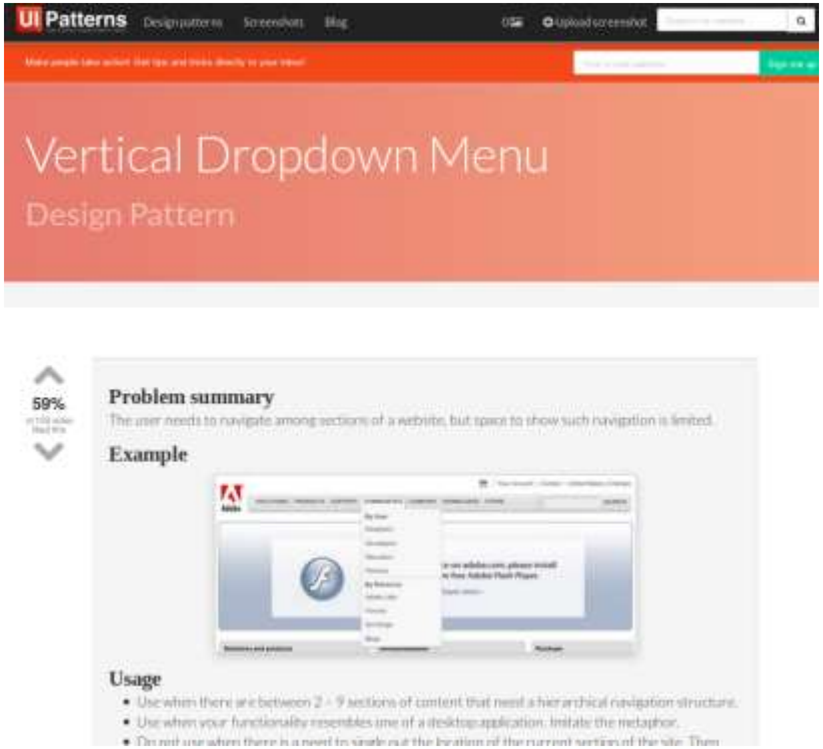
Shlomo Angel

- **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.

Patterns

- „Architectural Patterns“ discussed in Informatics are rooted in *Software Engineering* (introduced by the „Gang of Four“, GoF).
- Well-known are UI design patterns, cf. www.ui-patterns.com.
  - Example „Vertical Dropdown Menu“:

- Note: Patterns can even be used to generate user interfaces in a semi-automated manner (cf. Engel et al., 2013).



Vertical Dropdown Menu

Design Pattern

Problem summary

The user needs to navigate among sections of a website, but space to show such navigation is limited.

Example

Usage

- Use when there are between 2 - 9 sections of content that need a hierarchical navigation structure.
- Use when your functionality resembles one of a desktop application. Imitate the metaphor.
- Do not use when there is a need to single out the location of the current section of the site. Then

Comments on Patterns

- A technique can only be a pattern **if it occurs multiple times**. In general, the scientific patterns community agrees on a minimal number of three 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 (latest version PLML/1.2).
- 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; bold font indicates importance of an attribute within related work e.g. (Wendzel et al., 2015).

Patterns in Network Information Hiding

Patterns are an abstract construct. In the following, we will exemplify their practical application in the context of network information hiding.

Patterns in Network Information Hiding

- Idea of using patterns in network information hiding was first introduced in (Wendzel et al., 2015). Please cf. http://www.ih-patterns.blogspot.com where you can also download the paper.

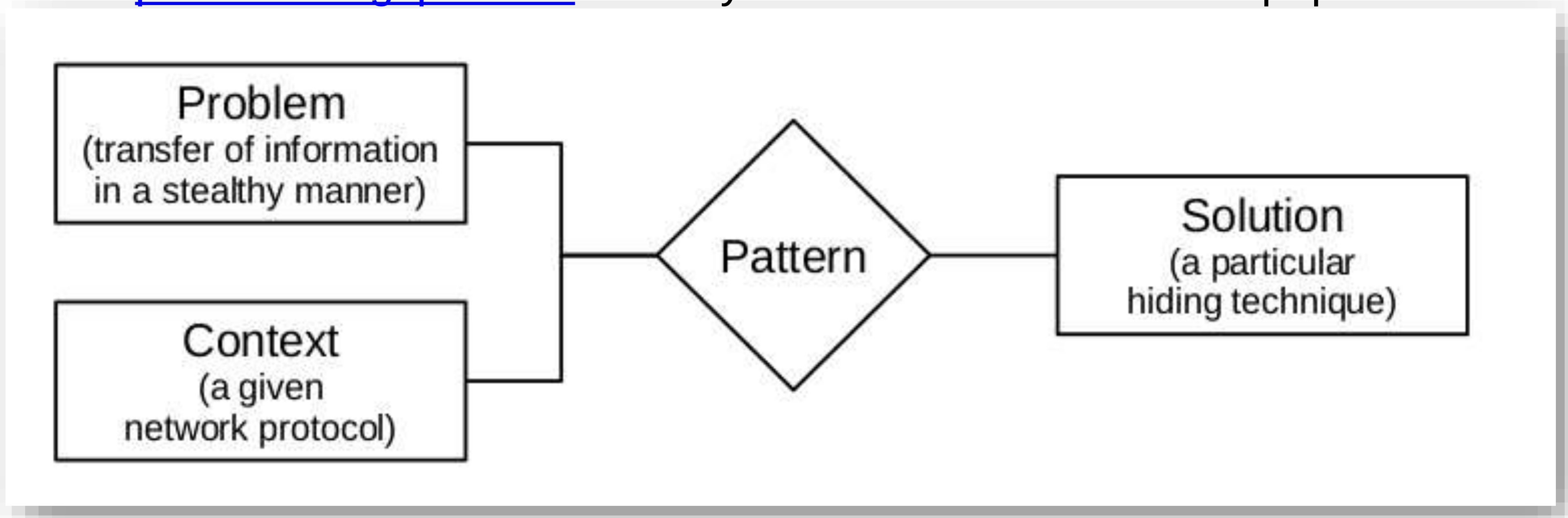

Problem
(transfer of information in a stealthy manner)
Context
(a given network protocol)
Pattern
Solution
(a particular hiding technique)

Image source: (Wendzel et al., 2015)

The following attributes were used

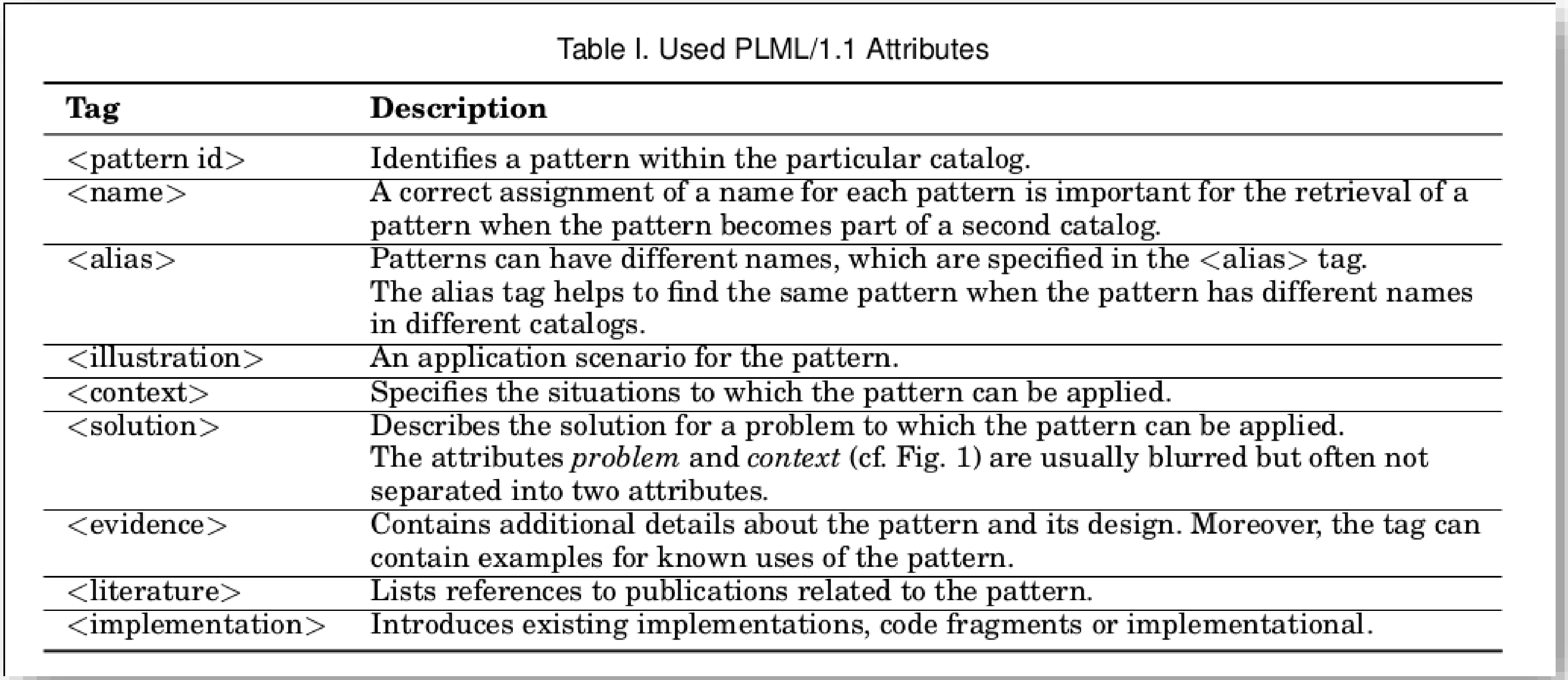


Table I. Used PLML/1.1 Attributes

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

- 130+ hiding techniques exist; they hide secret information in meta data of network traffic. A common description of all these methods in patterns can provide a better basis for the analysis of this mass of hiding techniques: instead of dealing with all these hiding techniques separately, we only need to understand the few hiding patterns.

- **Eleven** patterns were found to describe all analyzed hiding techniques published between 1987 and 2015.

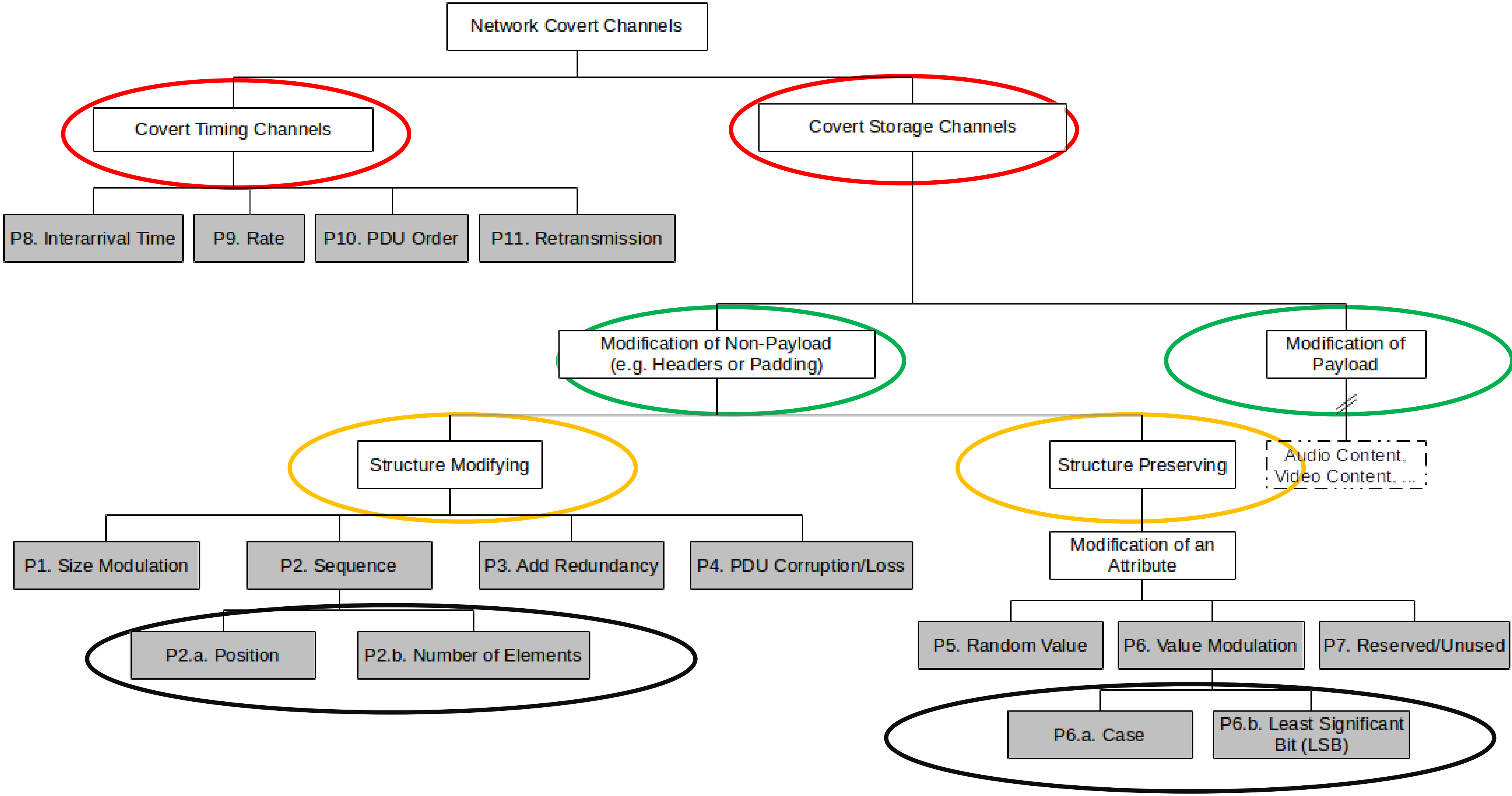
- Teaching own research just because it is the SotA and IMHO the easiest way to understand the discipline, i.e. lowest workload for you.

- 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.

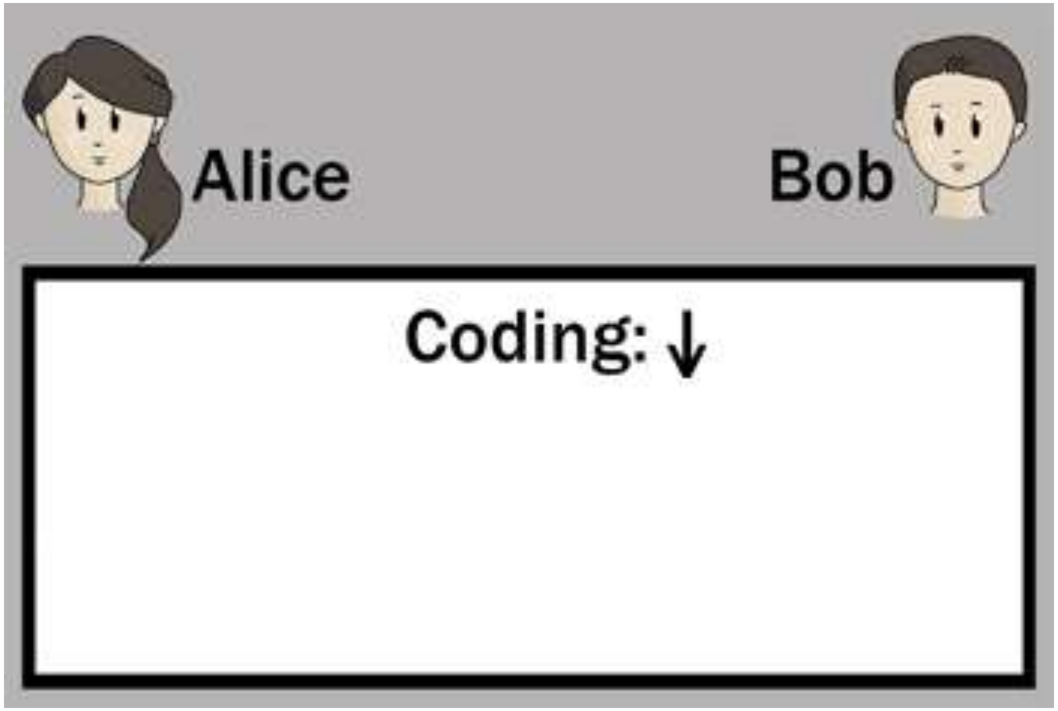


Image: J. Kammerlander.

*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

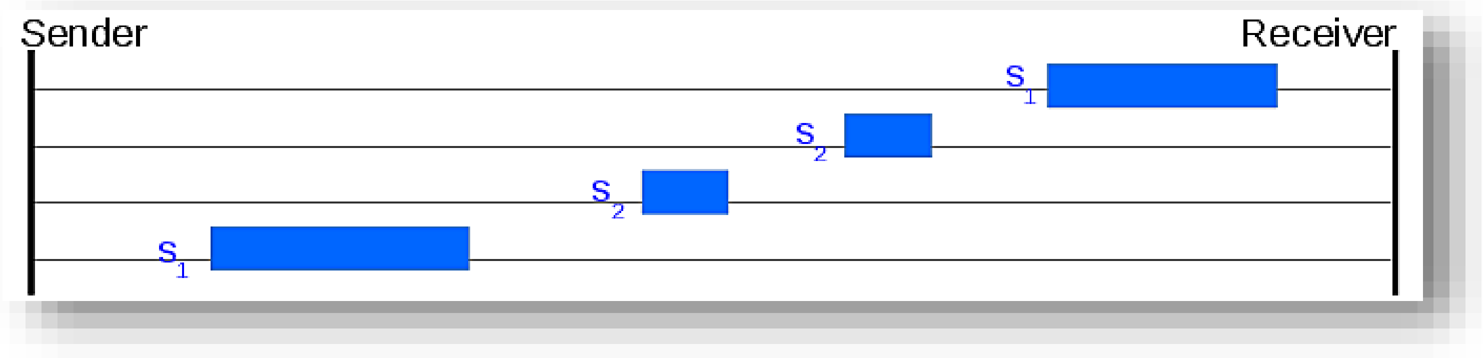


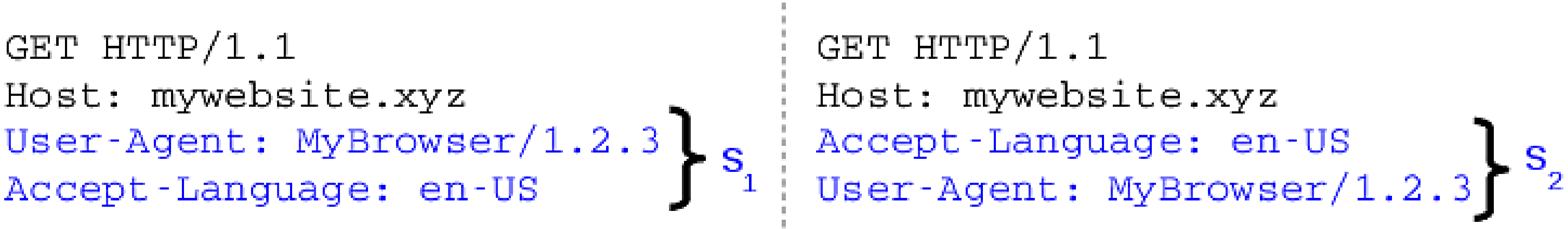
Image source: (Mazurczyk et al., 2016)

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

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
Accept-Language: en-US
```

```
GET HTTP/1.1
Host: mywebsite.xyz
Accept-Language: en-US
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

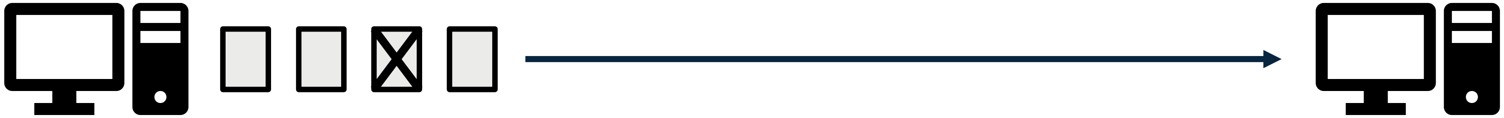
S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

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 in 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 first ISN of a TCP connection (cf. previous lecture on IH)
  - Utilize DHCP *xid* field

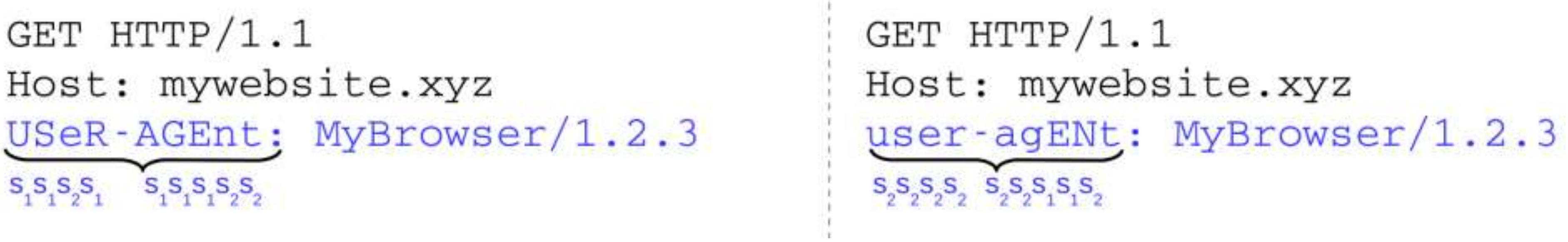
S. Wendzel et al.: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

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)



```
GET HTTP/1.1
Host: mywebsite.xyz
USeR-AGEnt: MyBrowser/1.2.3
```

$s_1 s_1 s_2 s_1 \quad s_1 s_1 s_1 s_2 s_2$

```
GET HTTP/1.1
Host: mywebsite.xyz
user-agENt: MyBrowser/1.2.3
```

$s_2 s_2 s_2 s_2 \quad s_2 s_2 s_1 s_1 s_2$

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

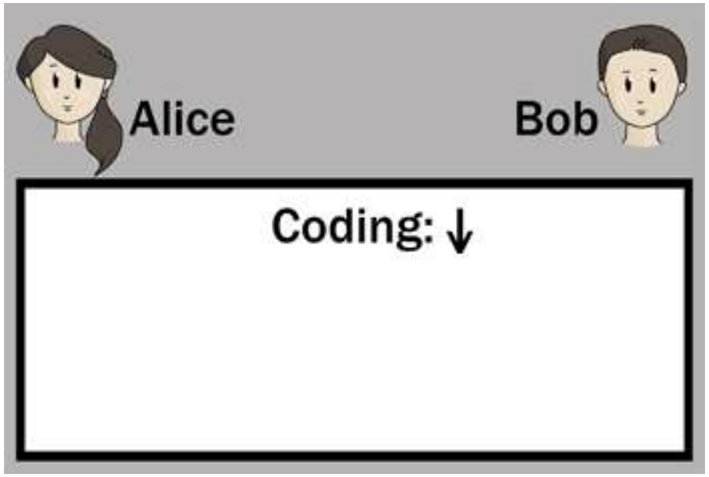


Image: J. Kammerlander.

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

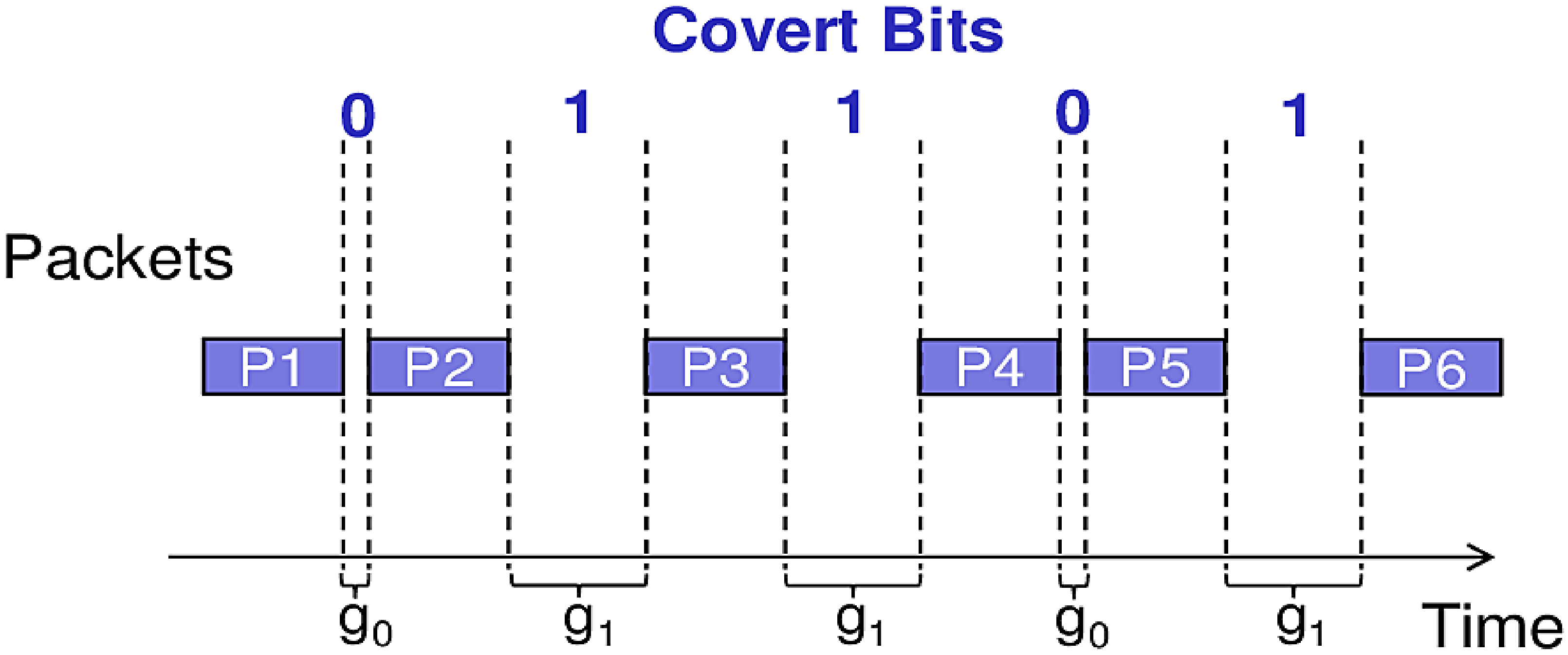


Image source: (Mazurczyk et al., 2016)

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.

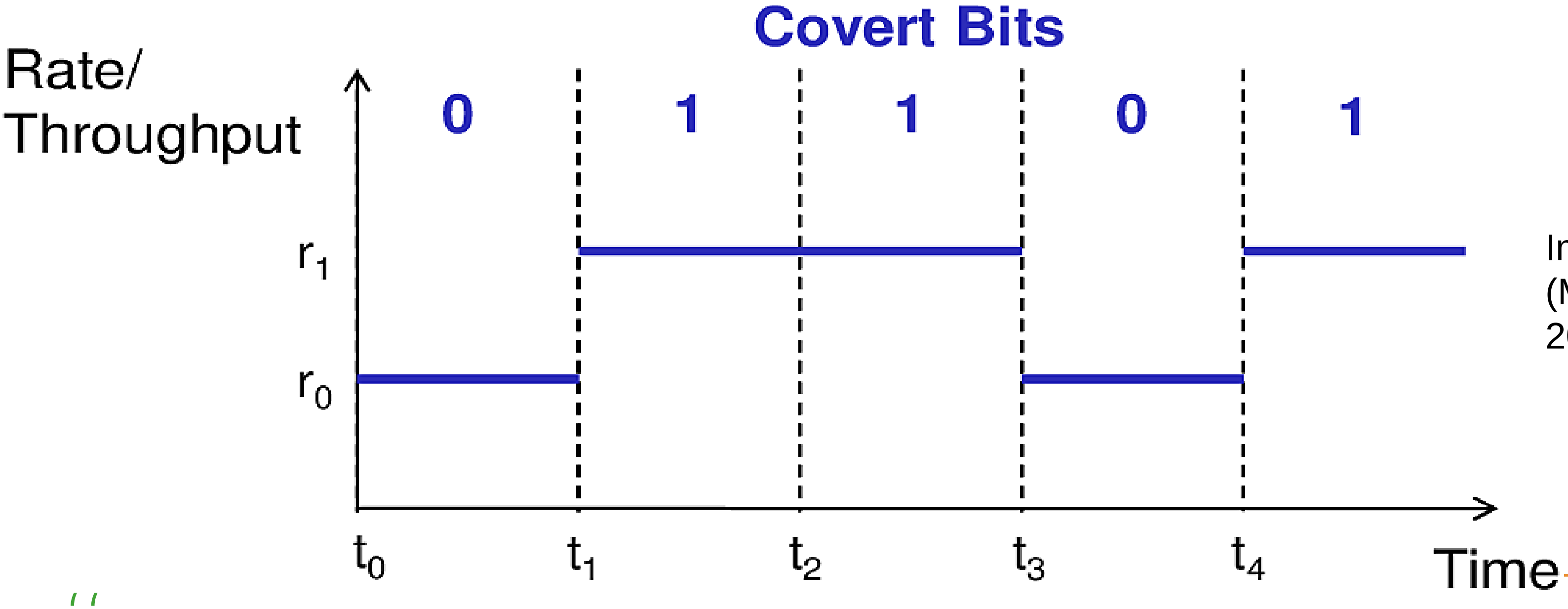


Image source: (Mazurczyk et al., 2016)

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

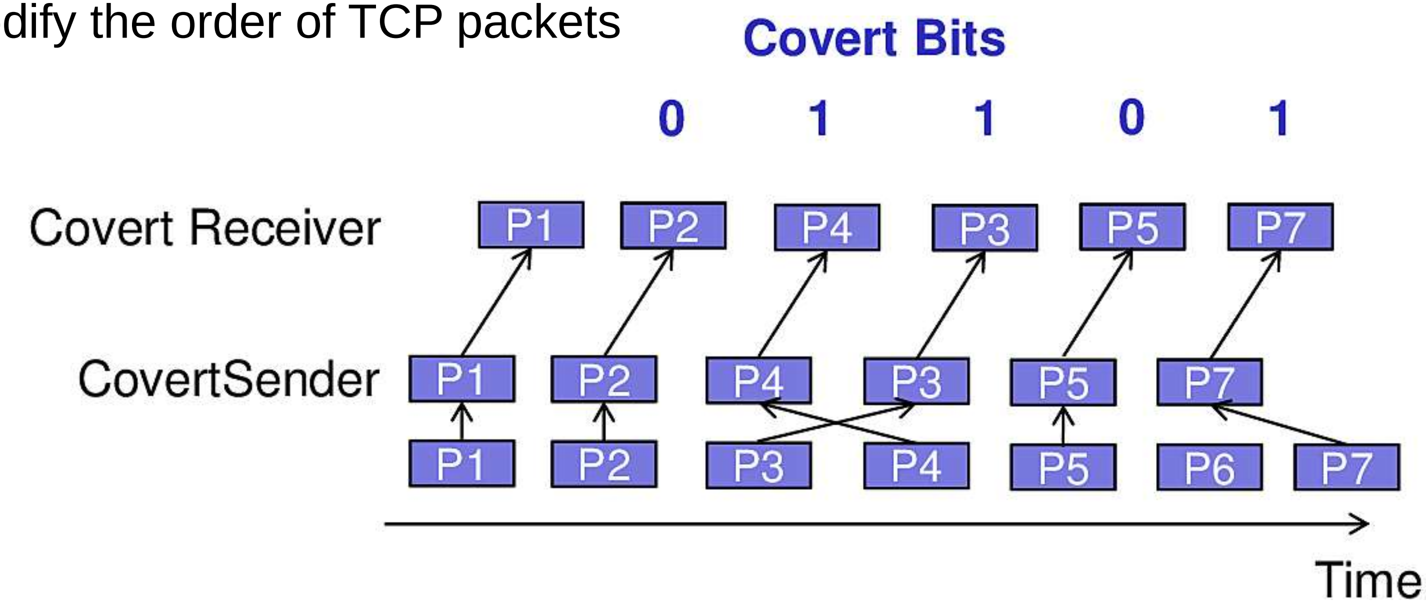


Image source: (Mazurczyk et al., 2016)

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

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.

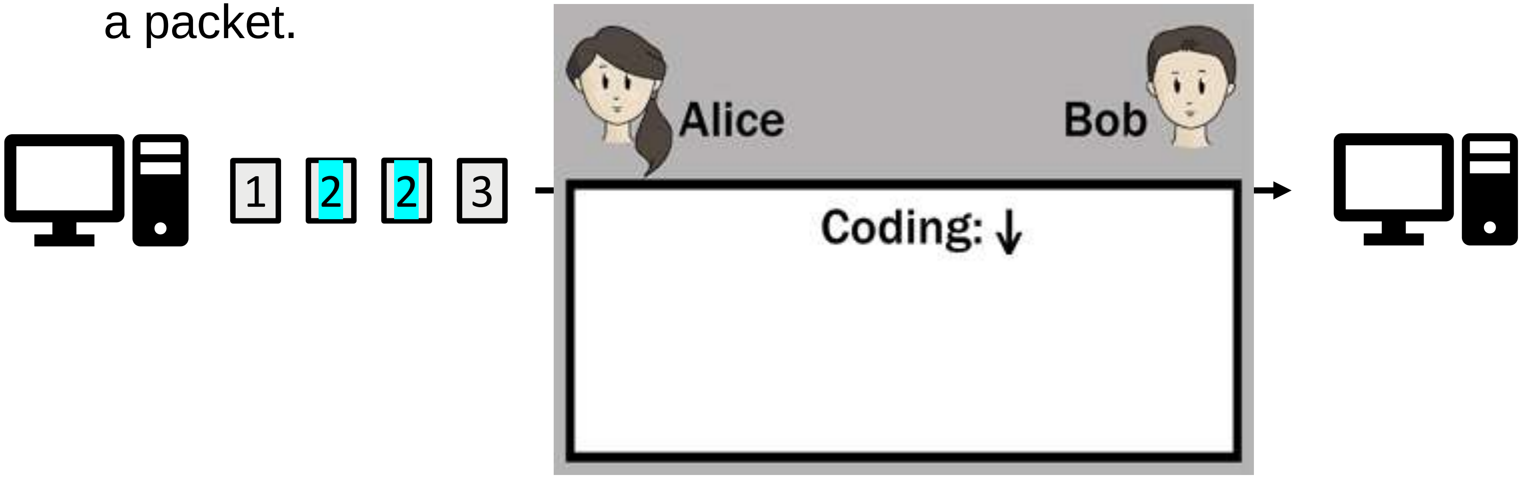


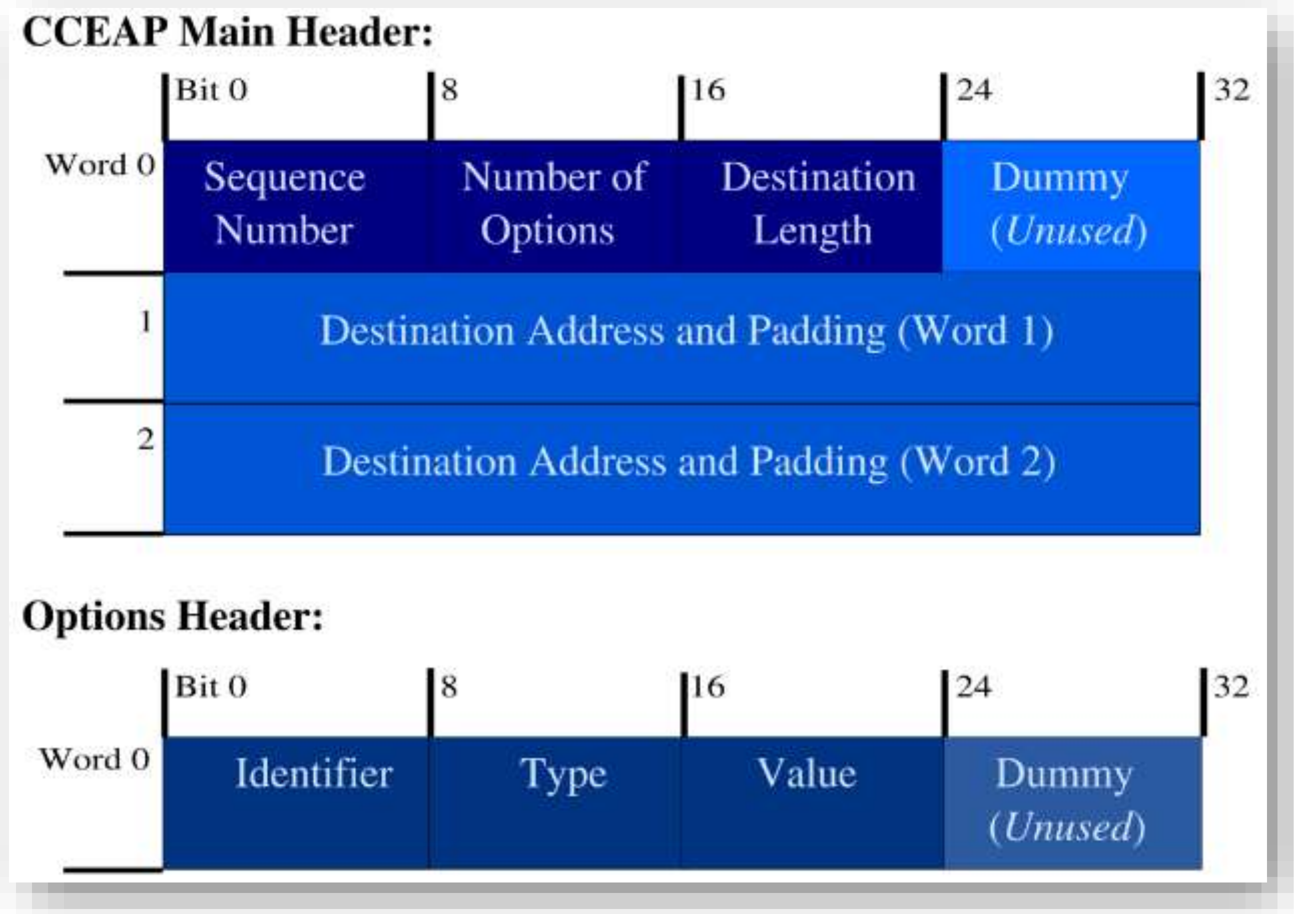
Image: J. Kammerlander.

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.



CCEAP Main Header:

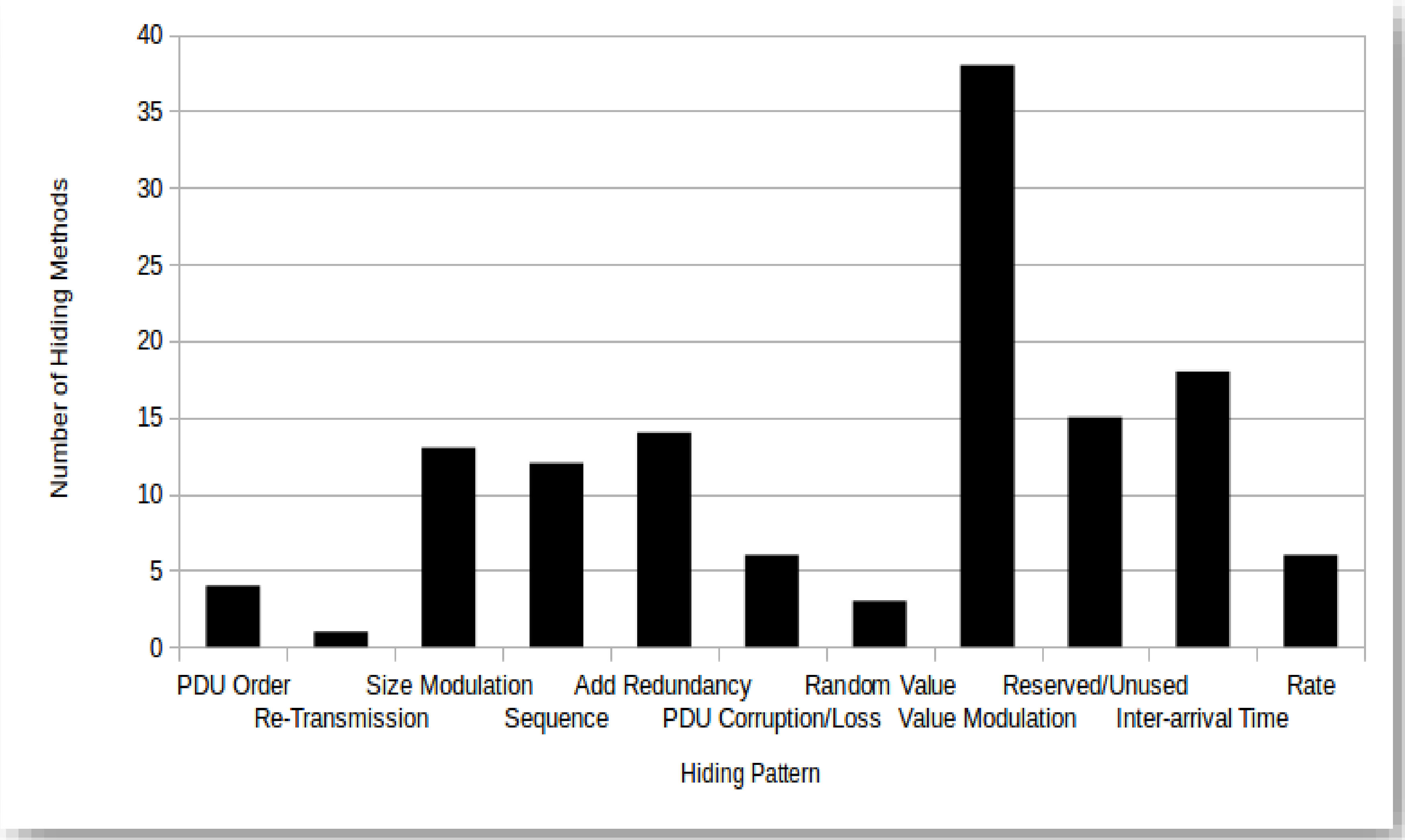
	Bit 0	8	16	24–32
Word 0	Sequence Number	Number of Options	Destination Length	Dummy (*Unused*)
1	Destination Address and Padding (Word 1)			
2	Destination Address and Padding (Word 2)			

Options Header:

	Bit 0	8	16	24–32
Word 0	Identifier	Type	Value	Dummy (*Unused*)

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.

Pattern Variation

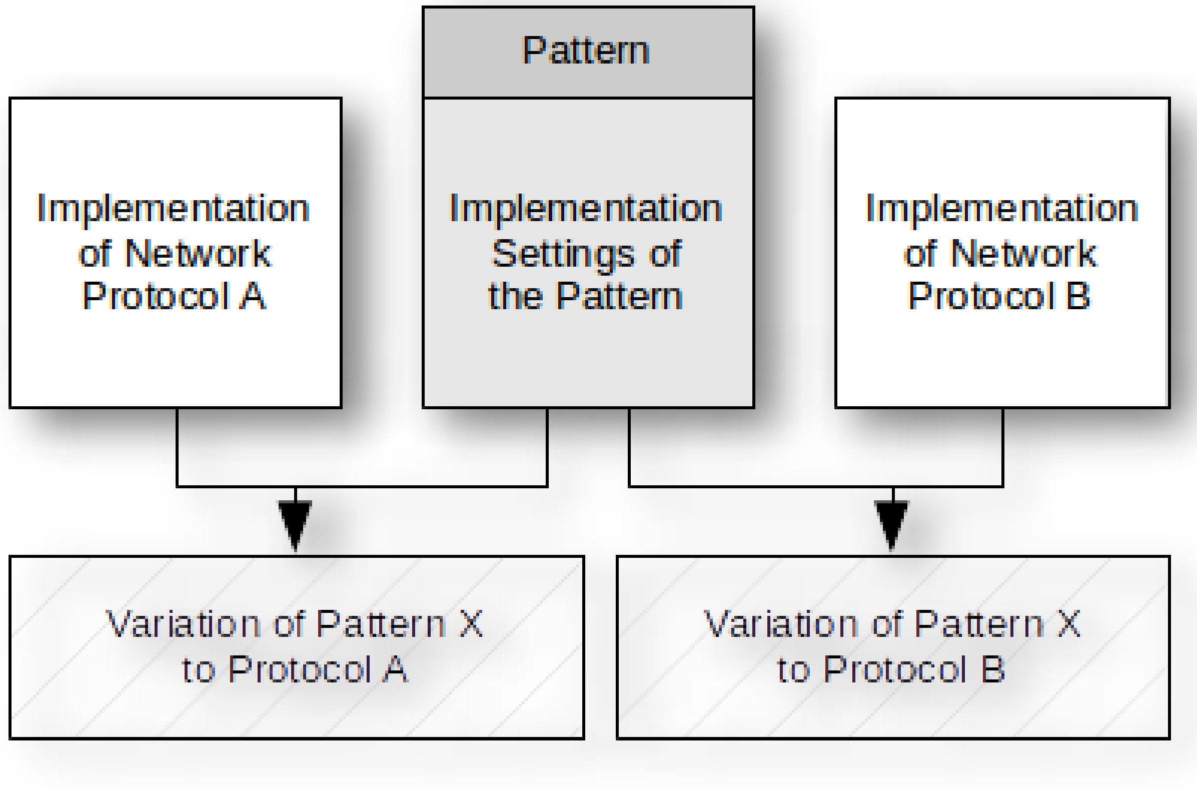

Pattern
Implementation of Network Protocol A
Implementation Settings of the Pattern
Implementation of Network Protocol B
Variation of Pattern X to Protocol A
Variation of Pattern X to Protocol B

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.

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

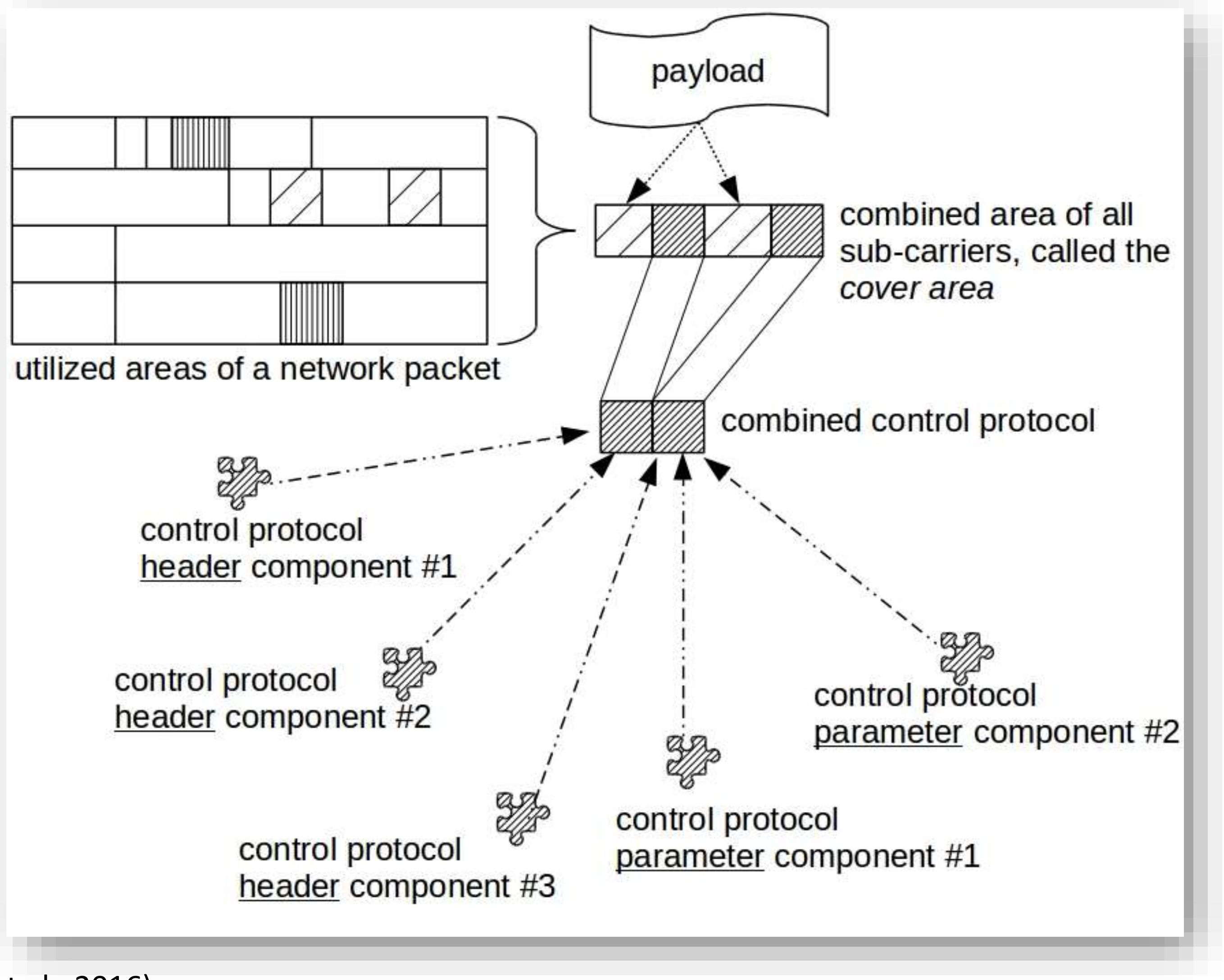
- (Formal) approaches for designing control protocols are available.

- … 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


payload
combined area of all sub-carriers, called the *cover area*
utilized areas of a network packet
combined control protocol
control protocol header component #1
control protocol header component #2
control protocol header component #3
control protocol parameter component #1
control protocol parameter component #2

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.

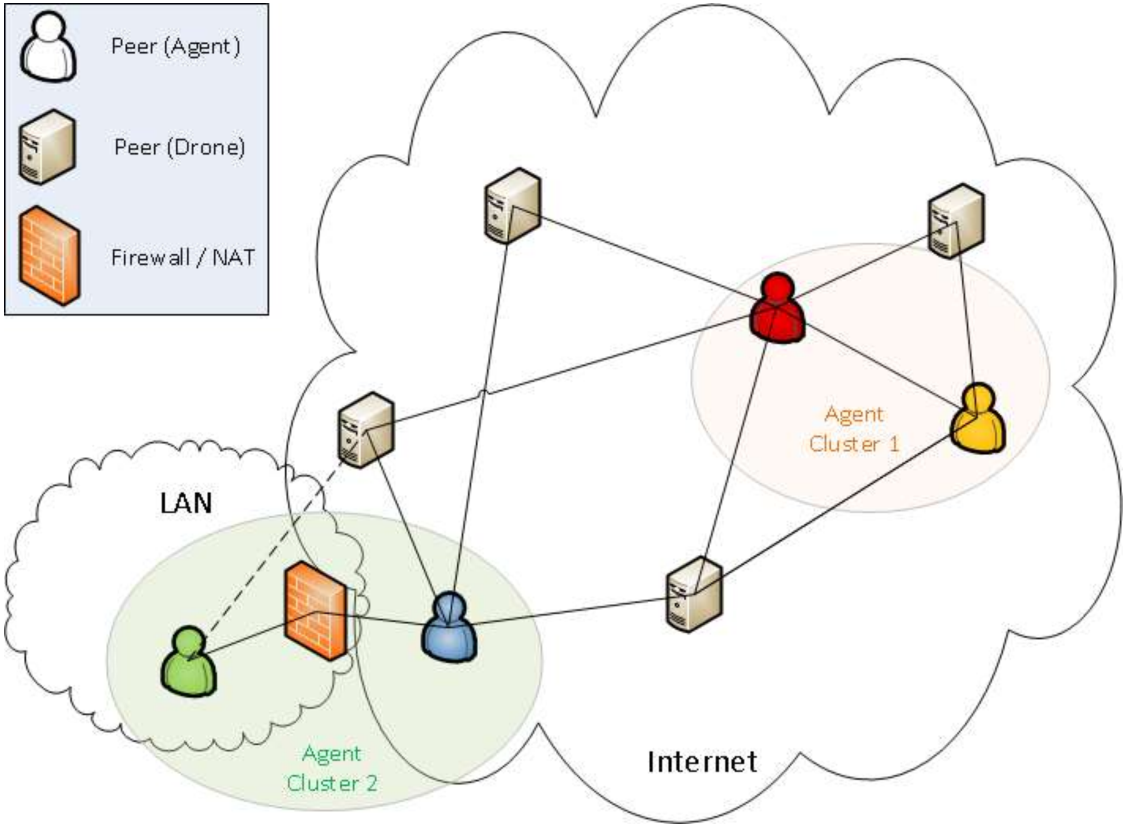
- Originally introduced by Yarochkin et al.
    - Circumvention-method improved a few years later by myself.

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.



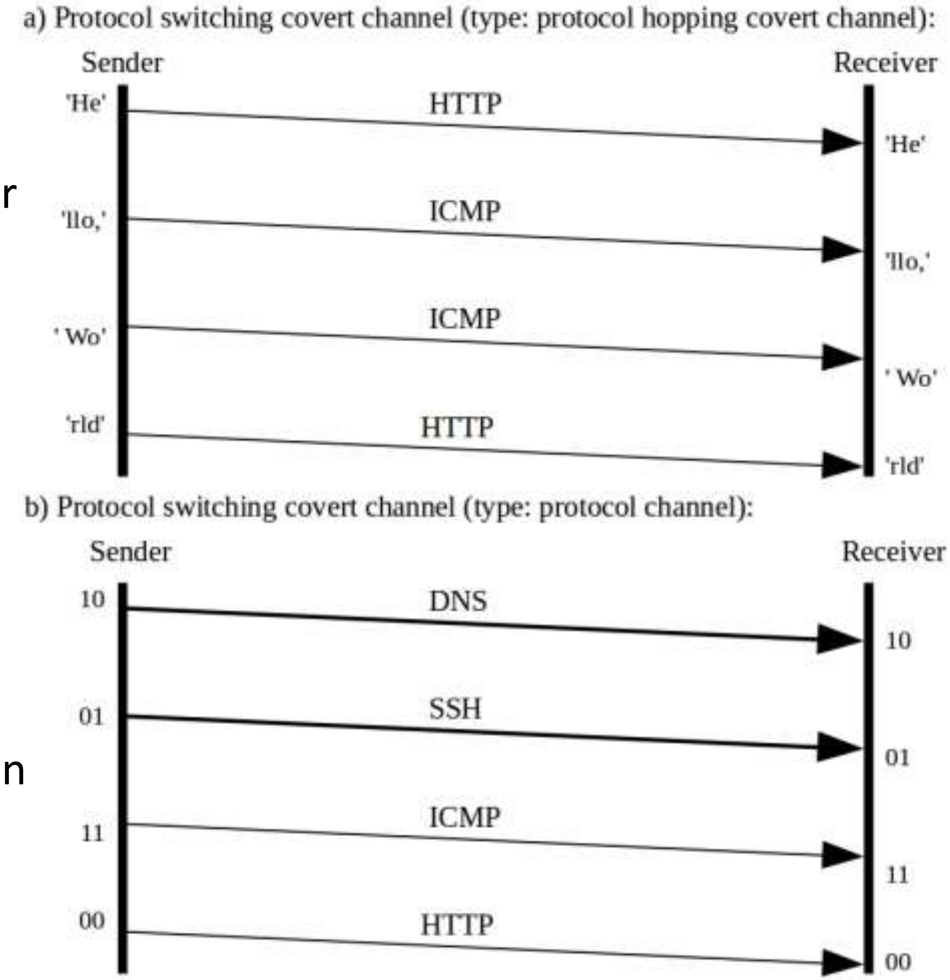
Backs, P., Wendzel, S., Keller, J.: Dynamic Routing in Covert Channel Overlays Based on Control Protocols, Proc. ISTP, IEEE, 2012.

Protocol Switching, Protocol Hopping, Pattern Hopping

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

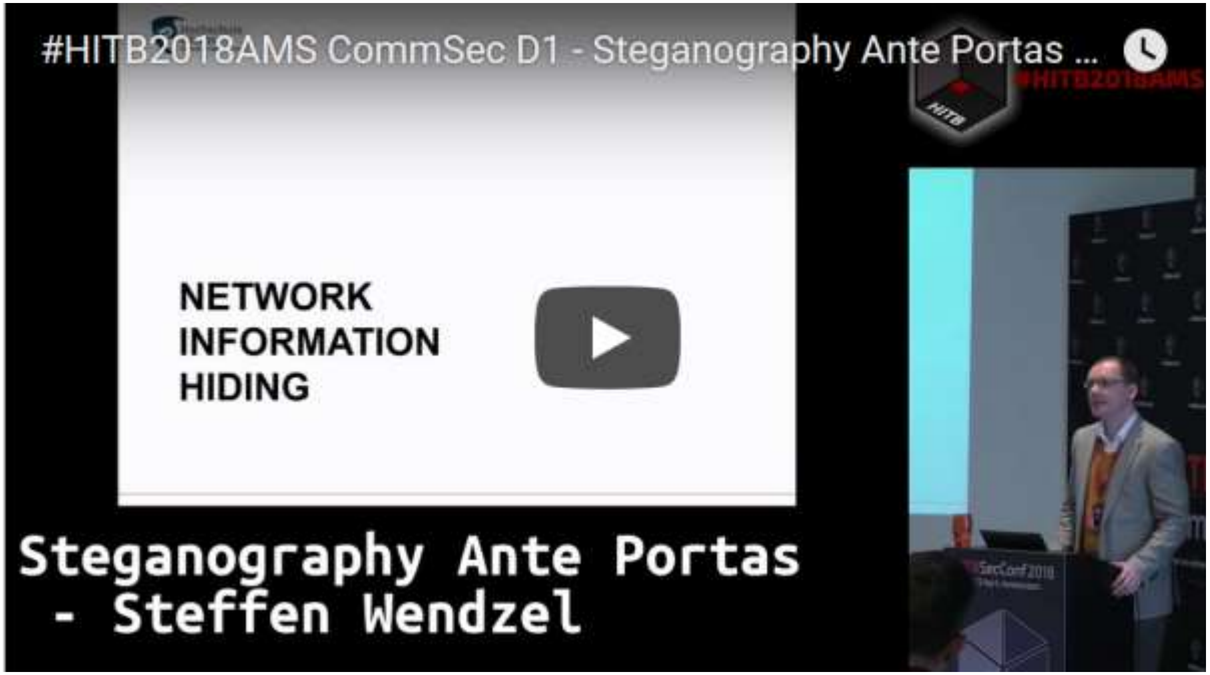
Protocol Switching Covert Channel:
Secret information is represented by the protocol itself.

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


a) Protocol switching covert channel (type: protocol hopping covert channel):
Sender
Receiver
'He'
HTTP
'He'
'llo,'
ICMP
'llo,'
' Wo'
ICMP
' Wo'
'rld'
HTTP
'rld'
b) Protocol switching covert channel (type: protocol channel):
Sender
Receiver
10
DNS
10
01
SSH
01
11
ICMP
11
00
HTTP
00

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.

Video Summary of the Patterns and Sophisticated Hiding Techniques



#HITB2018AMS CommSec D1 - Steganography Ante Portas ...

NETWORK INFORMATION HIDING

Steganography Ante Portas - Steffen Wendzel

SELECTED COUNTERMEASURES

Covert Channel Countermeasures

Prevention/Elimination

Limitation

Detection

Several methods exist …

- cf. (Mazurczyk et al., 2016, Chapter 8) for an overview

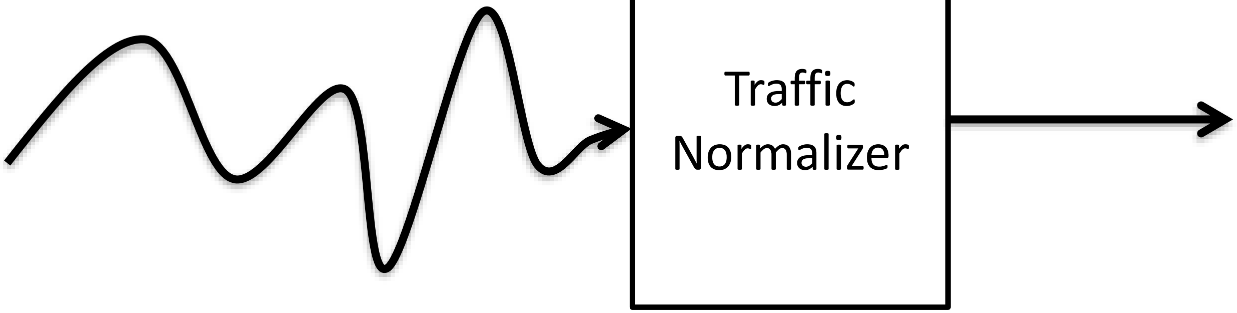
- We will consider only few:

1. **Traffic normalization** to detect various storage channels
2. **Two examples on the detectability** of ‘P8. Inter-arrival Time‘
3. **Protocol Channel-aware Active Warden** (PCAW) to limit protocol switching covert channels

Traffic Normalization

- Also known as packet scrubbing, usually part of firewalls and NIPS today, e.g. in OpenBSD pf and Snort.
- In NIH terminology, it is a form of an active warden

- In a nutshell: traffic is modified so that it becomes “normal”, e.g. reserved bits are cleared, some header fields are set to standard values.
  - usually rule-based



How Normalization Works

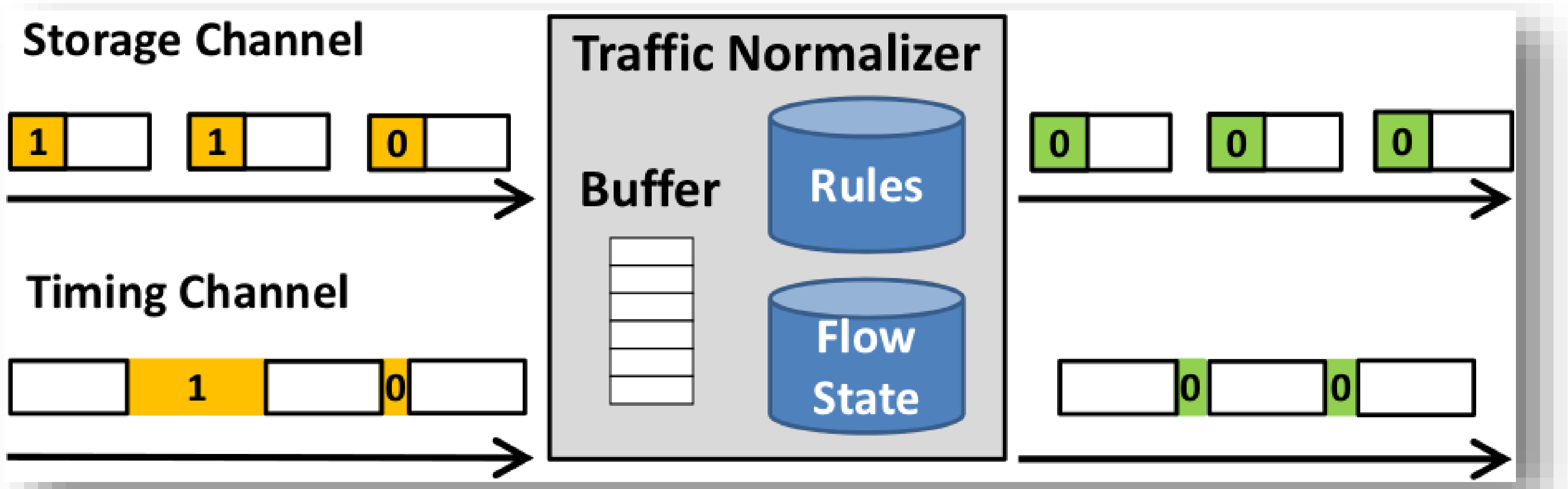
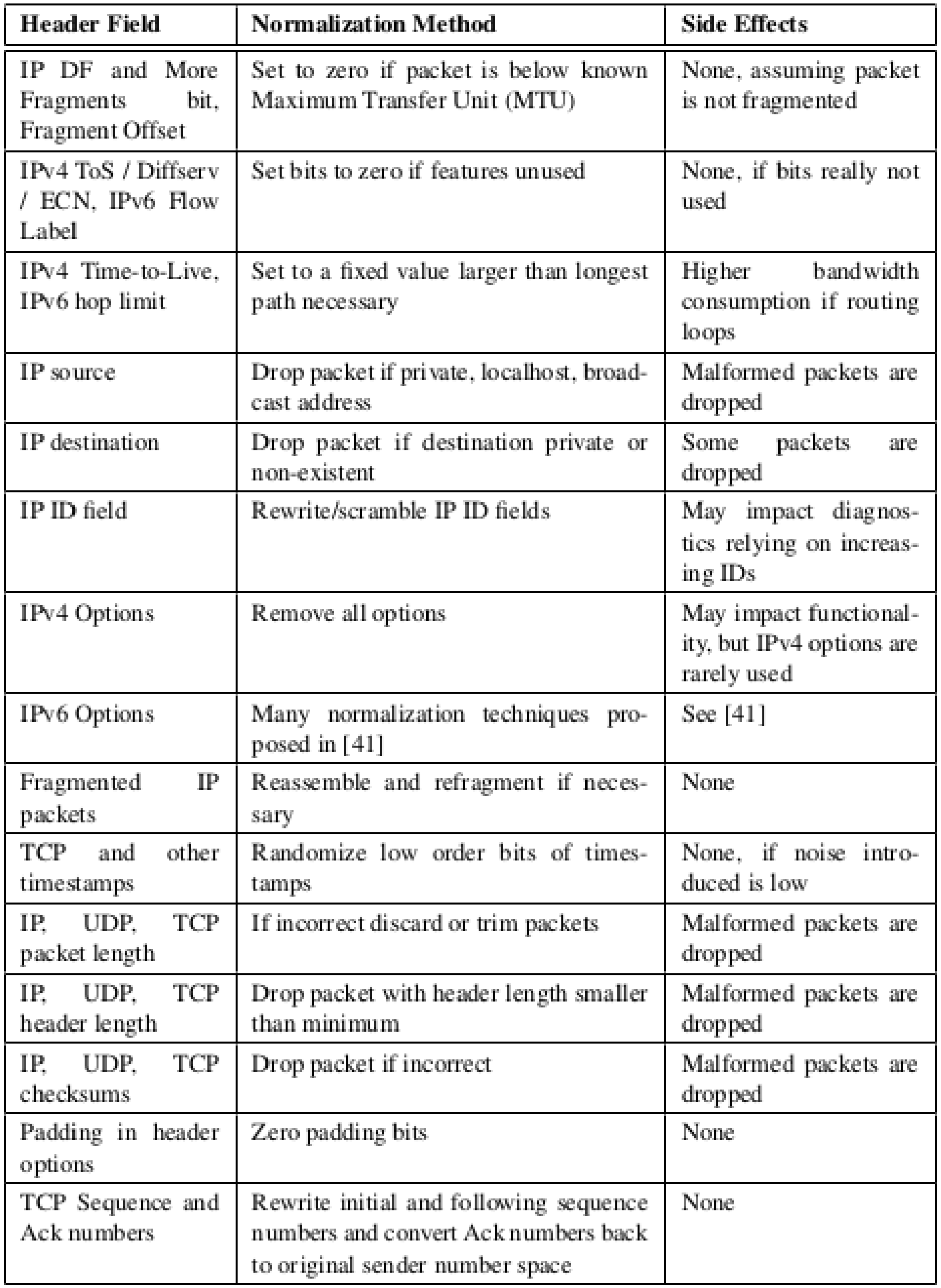


Fig.: (Mazurczyk et al., 2016)

The Problem

- Examples for side effects: table at the side (Mazurczyk et al., 2016).

Table 8.2: Well known techniques to normalize IP, UDP and TCP header fields and their possible side effects



Header Field	Normalization Method	Side Effects
IP DF and More Fragments bit, Fragment Offset	Set to zero if packet is below known Maximum Transfer Unit (MTU)	None, assuming packet is not fragmented
IPv4 ToS / Diffserv / ECN, IPv6 Flow Label	Set bits to zero if features unused	None, if bits really not used
IPv4 Time-to-Live, IPv6 hop limit	Set to a fixed value larger than longest path necessary	Higher bandwidth consumption if routing loops
IP source	Drop packet if private, localhost, broadcast address	Malformed packets are dropped
IP destination	Drop packet if destination private or non-existent	Some packets are dropped
IP ID field	Rewrite/scramble IP ID fields	May impact diagnostics relying on increasing IDs
IPv4 Options	Remove all options	May impact functionality, but IPv4 options are rarely used
IPv6 Options	Many normalization techniques proposed in [41]	See [41]
Fragmented IP packets	Reassemble and refragment if necessary	None
TCP and other timestamps	Randomize low order bits of timestamps	None, if noise introduced is low
IP, UDP, TCP packet length	If incorrect discard or trim packets	Malformed packets are dropped
IP, UDP, TCP header length	Drop packet with header length smaller than minimum	Malformed packets are dropped
IP, UDP, TCP checksums	Drop packet if incorrect	Malformed packets are dropped
Padding in header options	Zero padding bits	None
TCP Sequence and Ack numbers	Rewrite initial and following sequence numbers and convert Ack numbers back to original sender number space	None

Inconsistent TCP-Retransmissions

- Handley et al. [1]:
  - How to handle overlapping TCP segments as such caused by re-transmissions, especially if their payload differs?
  - Example (based on [1]):

```
seq:1, TTL:10, payload=n
seq:1, TTL:12, payload=y
seq:2, TTL:11, payload=o
seq:2, TTL:12, payload=e
seq:3, TTL:10, payload=!
seq:3, TTL:11, payload=s
```

    - We could receive different messages: „`yes`“, „`no!`“, „`yo!`“, …
  - Depending on the TTL: Which segments will reach the receiver?
  - What are the potential consequences of the different scenarios?
  - We need to cache all the data and evaluate all possibilities in the TN.
  - Also cf. previous lecture by **Slobodan Petrović** on July 25th/26th on how the search algorithms can look like.

[1] M. Handley et al.: Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, Proc. Usenix Symp. 2001. https://www.usenix.org/legacy/events/sec01/full_papers/handley/handley.pdf

Cold Start

- Handley et al. [1]:

  [The design of a TN] *“can prove vulnerable to incorrect analysis during a **cold start**. That is, when the analyzer first begins to run, it is confronted with traffic from already-established connections for which the analyzer lacks knowledge of the connection characteristics negotiated when the connections were established.*

  For example, the TCP scrubber [8] requires a connection to go through the normal start-up handshake. However, if a valid connection is in progress, and the scrubber restarts or otherwise loses state, then it will terminate any connections in progress at the time of the cold start, since to its analysis their traffic exchanges appear to violate the protocol semantics that require each newly seen connection to begin with a start-up handshake.“

[1] M. Handley et al.: Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, Proc. Usenix Symp. 2001. https://www.usenix.org/legacy/events/sec01/full_papers/handley/handley.pdf

Stateholding Problem

- Handley et al. [1]:

  „*A NIDS system must hold state*" [bspw. TCP-States -> s. Cold Start] "*in order to understand the context of incoming information. One form of attack on a NIDS is a stateholding attack, whereby the attacker creates traffic that will cause the NIDS to instantiate state (see §4.2 below). If this state exceeds the NIDS's ability to cope, the attacker may well be able to launch an attack that passes undetected.*

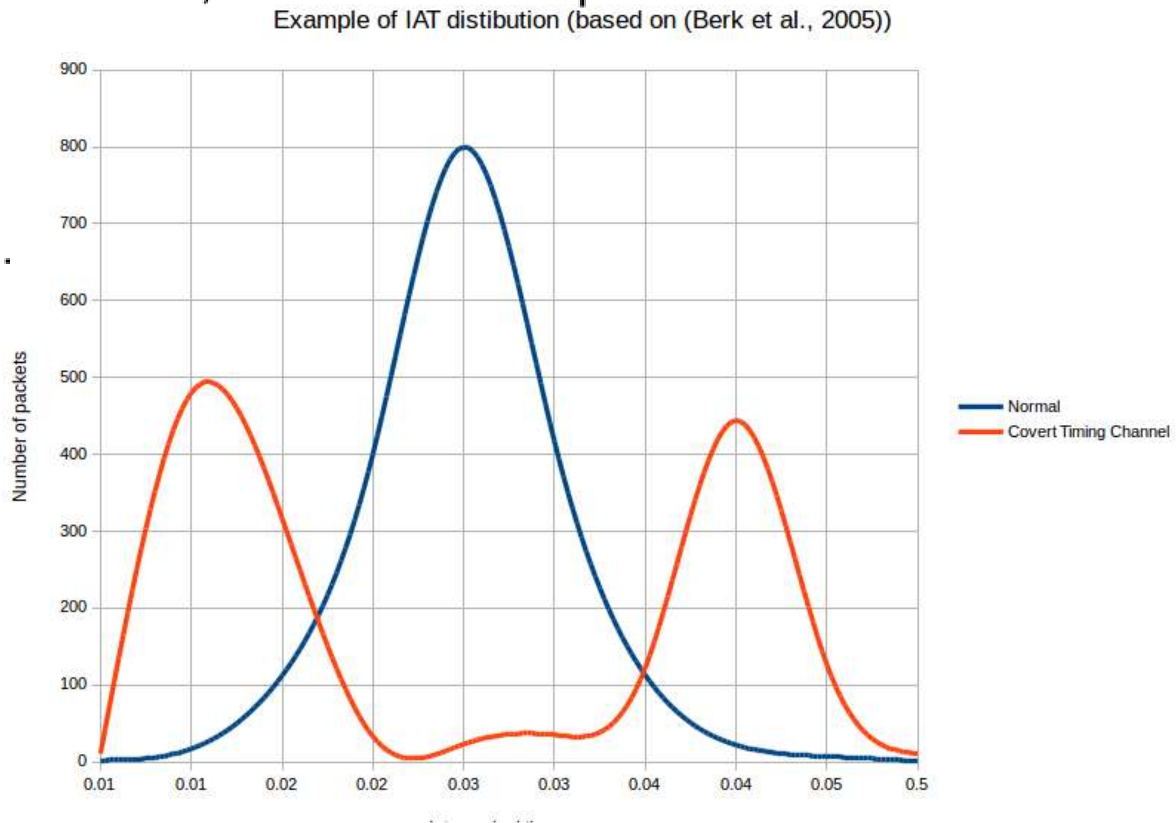
  […]

  An attacker can thus cause the normalizer to use up memory by sending many fragments of packets without ever sending enough to complete a packet."

[1] M. Handley et al.: Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, Proc. Usenix Symp. 2001. https://www.usenix.org/legacy/events/sec01/full_papers/handley/handley.pdf

P8. Inter-arrival Time Pattern Detection: Berk et al.

- (Berk et al., 2005) state that IPGs of non-covert traffic are distributed in a way that most of the measured IPGs are close to an average IPG value X.
- Inter-arrival time-based covert channels, however, signal hidden information using at least two different inter-arrival time encoded symbols, resulting in at least two, instead of one `peak‘ of IPG values:



Example of IAT distibution (based on (Berk et al., 2005))

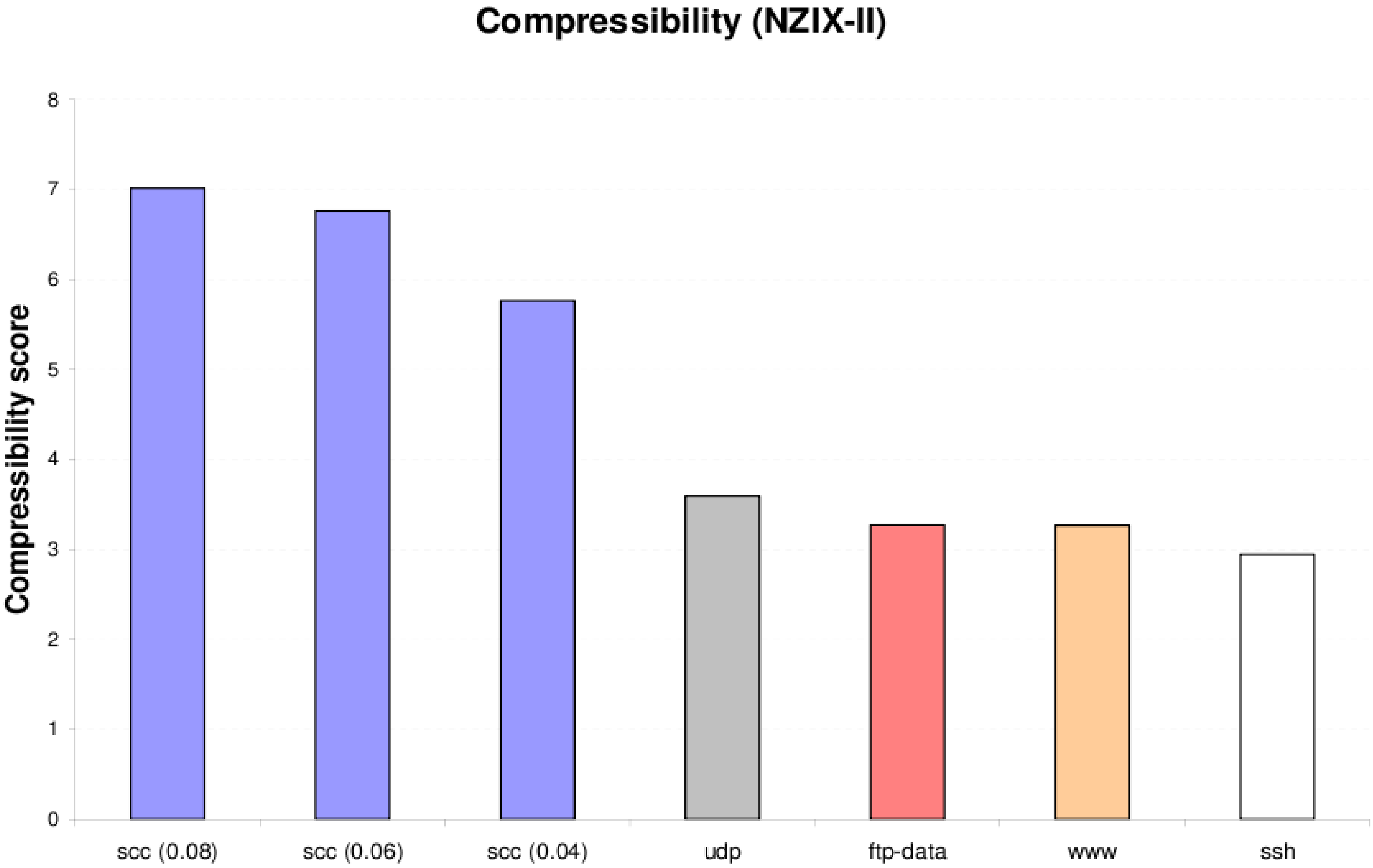
- Procedure:
  - Record all IPGs
  - C_μ: # packets with avg. IAT value
  - C_{max}: highest number of packets with same IAT
  - $P_{CovChan} = 1 - \frac{C_\mu}{C_{max}}$

P8. Inter-arrival Time Pattern Detection: Cabuk et al.

- (Cabuk et al., 2004) presents three techniques for detection of an inter-arrival time-based network covert channel. We consider only technique #3 in this lecture.
- Procedure:
  1. Record a traffic flow (e.g. n=2.000 packets)
  2. Calculate all inter-arrival times (IATs)
     1. To remove noise, remove all IATs that are >1sec
  3. Representation of all IATs as strings
     1. Round all IATs, usually take three+/-1 digits behind comma
     2. First character represents the number of zeroes behind comma
        - e.g. A=1 zeroes, B=2 zeroes, …
     3. Remaining positions of the rounded value are attached to the string
        - Example: 0.00247 becomes B25
  4. Remaining string **S** comprises concatenated IAT string values
  5. Compress S using compressor Θ (e.g. gzip): $C = \Theta(S)$
  6. Calculate Kolmogorov complexity Compressibility $K(S) = \frac{|S|}{|C|}$

P8. Inter-arrival Time Pattern Detection: Cabuk et al., One Selected Method

- $K(S)$ for traffic recordings (from (Cabuk et al., 2009))



Protocol Channel-aware Active Warden (PCAW)

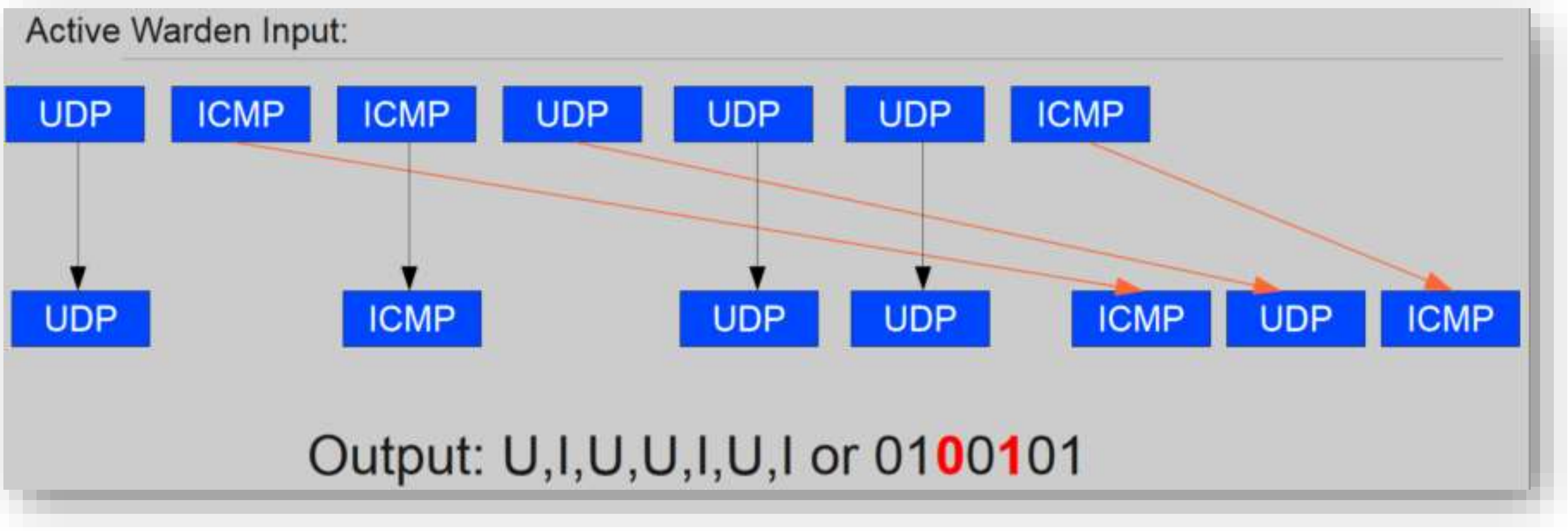
- Protocol (Switching Covert) Channel-aware Active Wardens
  - Limits bitrate of protocol switching covert channels
  - E.g. ICMP=„0“, UDP=„1“, Message „0101“ would then be represented by four packets with the order ICMP-UDP-ICMP-UDP

- How does it work?
  - Having a router that introduces delays for packets if they contain a protocol different from the previous one a particular sender and receiver.

- State-holding: cache the last recently used protocol for each combination of sender and receiver
- Buffer: cache all packets that must be delayed

S. Wendzel, J. Keller: Preventing Protocol Switching Covert Channels, in: Int. Journal Adv. Security, 2012.
https://www.researchgate.net/publication/233765874_Preventing_Protocol_Switching_Covert_Channels

PCAW: Example



S. Wendzel, J. Keller: Design and Implementation of an Active Warden Addressing Protocol Switching Covert Channels, in Proc. ICIMP, Iaria, 2012.

Einfluss des PCAW auf die Bitrate [1]

- Bitrate calculation for a classical network CC (Tsai/Gligor):

$$B = b \cdot (T_R + T_S + 2T_{CS})^{-1}$$

b : number of bits to be transferred per transmission
T_x: time it takes to receive (R), send (S) and process (CS) the covert data.

- For a PSCC that utilizes *n* protocols, $b = \log_2 n$.

S. Wendzel, J. Keller: Design and Implementation of an Active Warden Addressing Protocol Switching Covert Channels, in Proc. ICIMP, Iaria, 2012.

Einfluss des PCAW auf die Bitrate [1]

- Given that we delay packets by *d* and that a switch of a protocol is taking place with probability *p*, while the transfer and covert data processing takes time *T*, we can modify the previous formula:

$$B = \log_2(n) \cdot (pd + T)^{-1}$$

- For a uniform coding with random input using *n* protocols, we know that $p = 1 - 1/n$. So, we can assume that:

$$B = \log_2(n) \cdot \left(\left(1 - \frac{1}{n}\right) d + T \right)^{-1}$$

- Other variants of PSCC exist where p and b may differ (see paper for details).

S. Wendzel, J. Keller: Preventing Protocol Switching Covert Channels, Int. J. Adv. Sec., 2012.

PCAW – theoretische Resultate

For a typical PSCC, we can reduce B to less than *1 bit/s* if we apply *d=2sec* for realistic T.

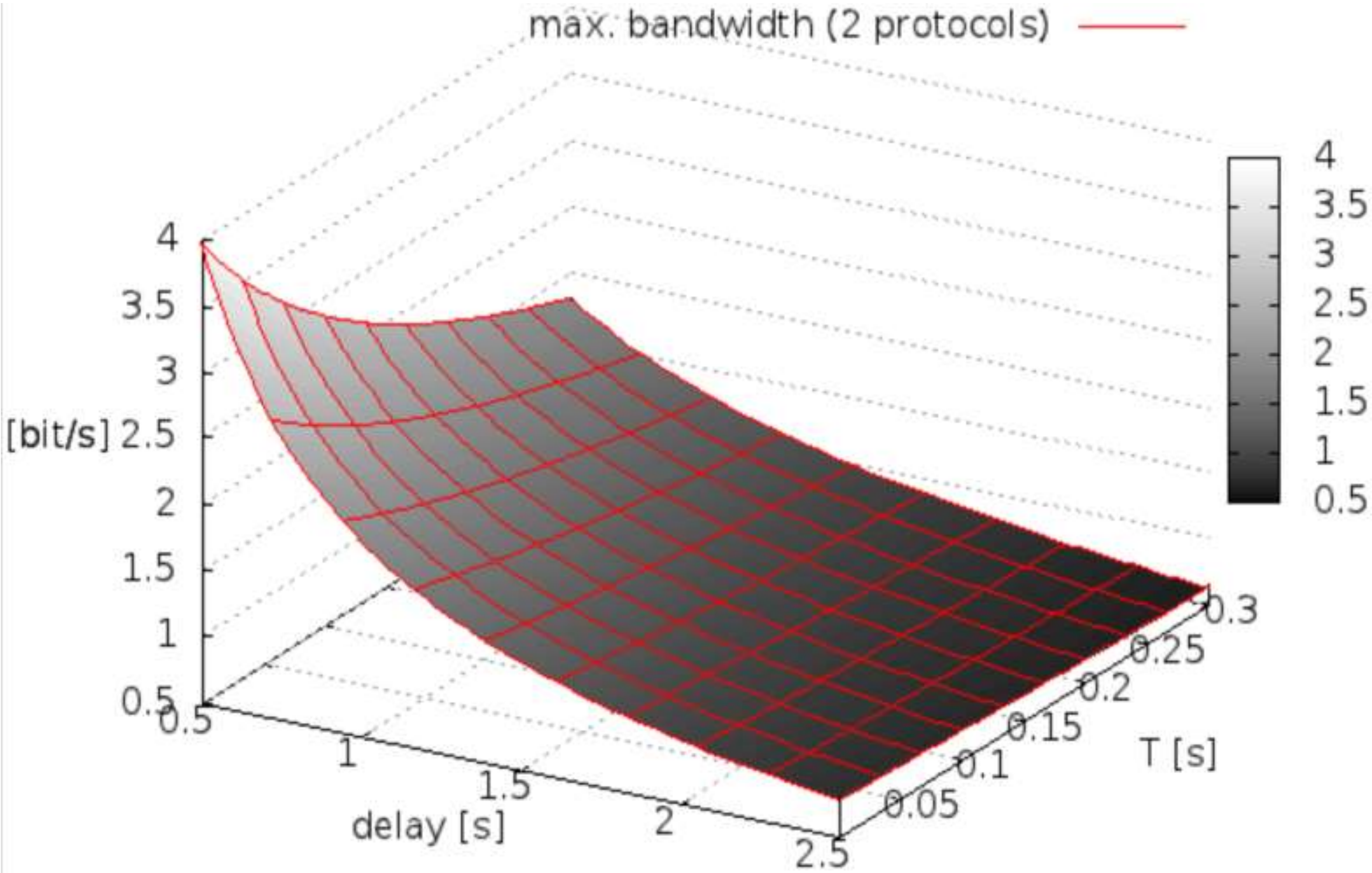


Fig.: S. Wendzel, J. Keller: Design and Implementation of an Active Warden Addressing Protocol Switching Covert Channels, in Proc. ICIMP, Iaria, 2012.

PCAW [1] – praktische Resultate

- Comparison of formula and actual measurements for *T=0.005s*.

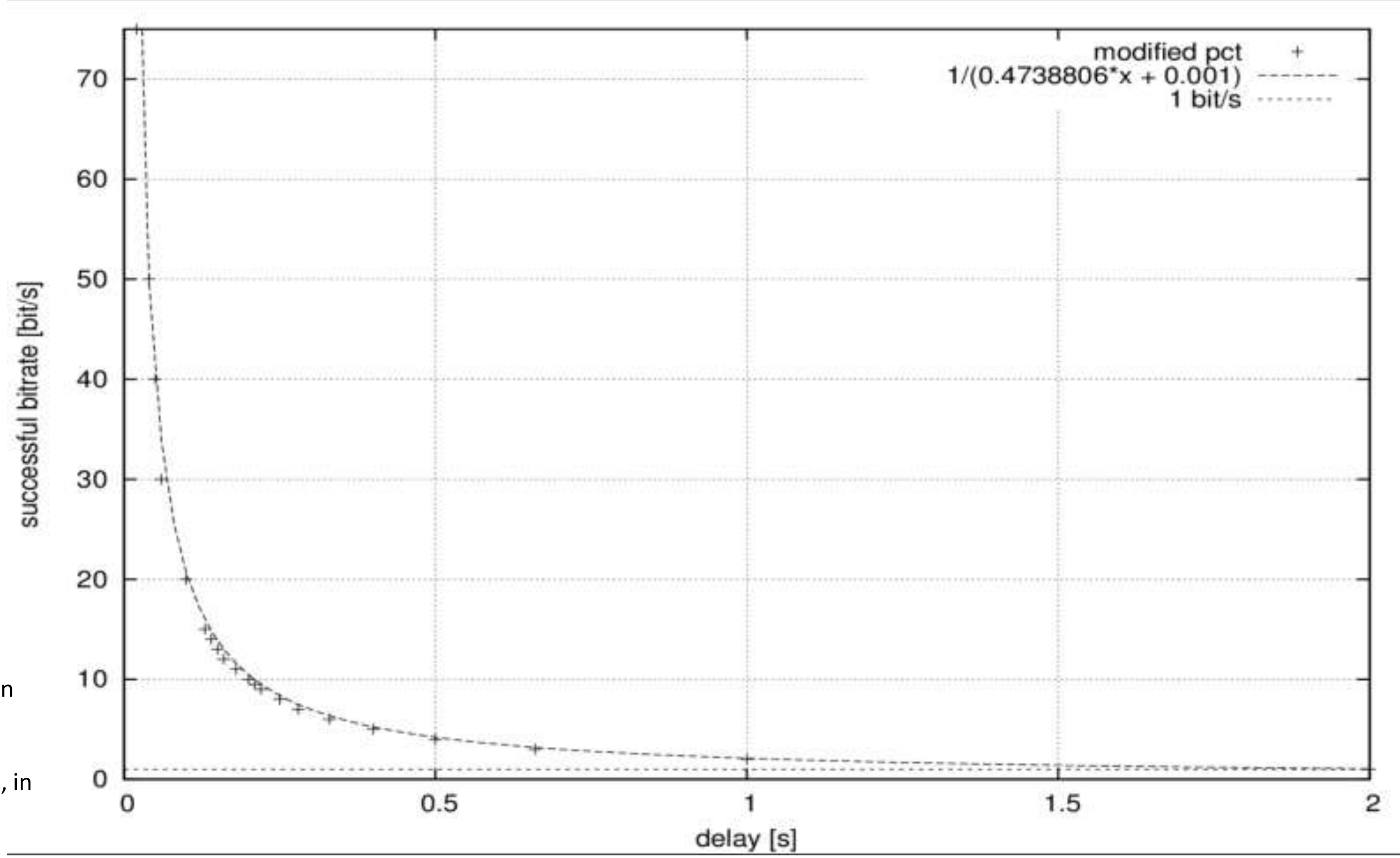


Fig.: S. Wendzel, J. Keller: Design and Implementation of an Active Warden Addressing Protocol Switching Covert Channels, in Proc. ICIMP, Iaria, 2012.

Randomized PCAW

- Problem:
  - Attacker could try to determine *d* and then adjust own sending behavior, so that the warden becomes less effective.

- Solution:
  - Use a randomized delay, so that $d_r \in [0; d[$.
  - Turns out to provide excellent results.

S. Wendzel, J. Keller: Preventing Protocol Switching Covert Channels, in: Int. Journal Adv. Security, 2012.
https://www.researchgate.net/publication/233765874_Preventing_Protocol_Switching_Covert_Channels

Overview

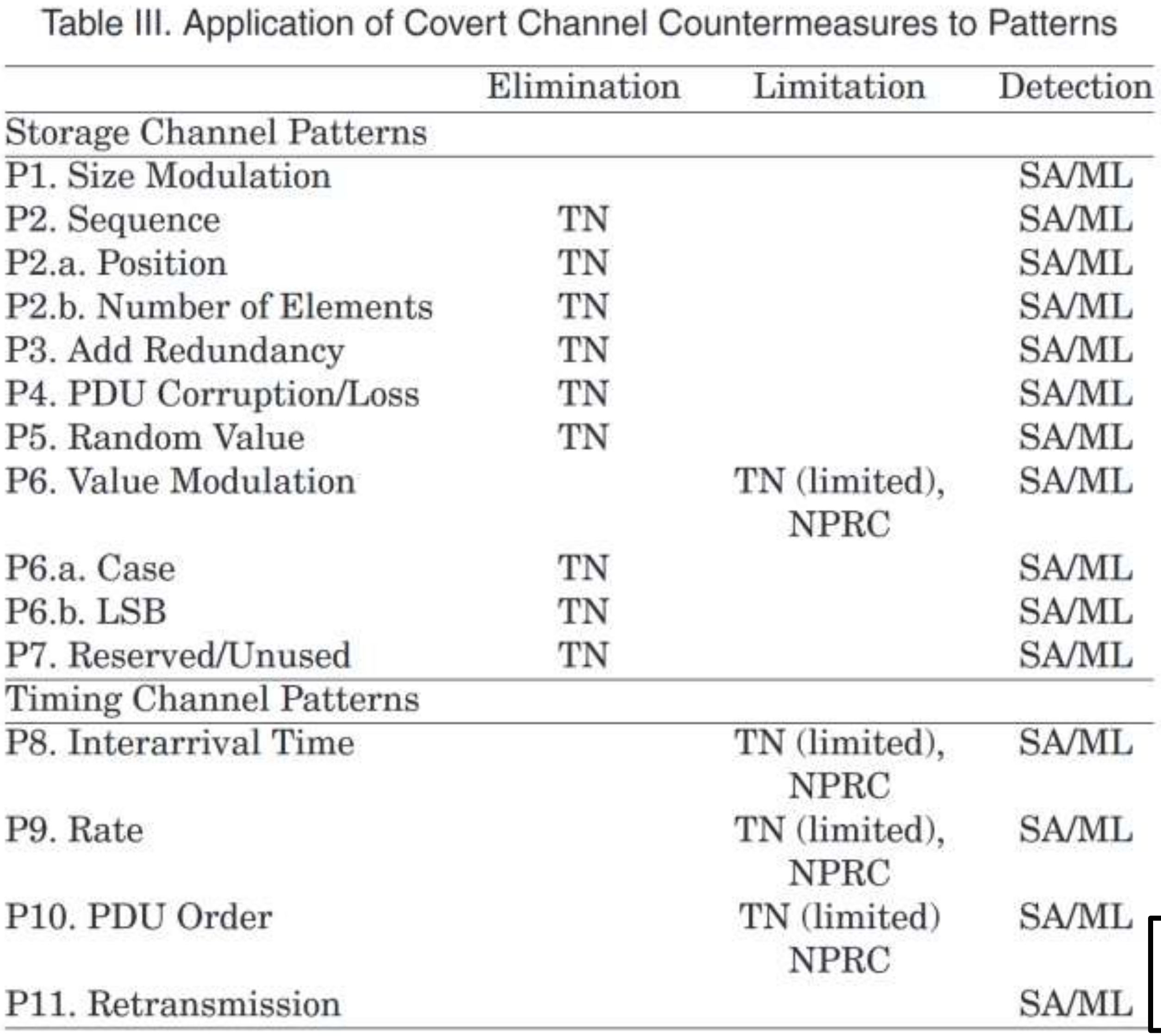


Table III. Application of Covert Channel Countermeasures to Patterns

	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.

Outlook: Countermeasure Variation

Countermeasure variation (transformation)
e.g. applied the compressibility and the epsilon-similarity approaches of Cabuk et al. to new patterns

- *size modulation* pattern
- re-transmission pattern

… and it works fine!

To be published:

S. Wendzel, D. Eller, W. Mazurczyk: *One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels*, in Proc. CECC’18, to appear (November 2018).

REPLICATING EXPERIMENTS

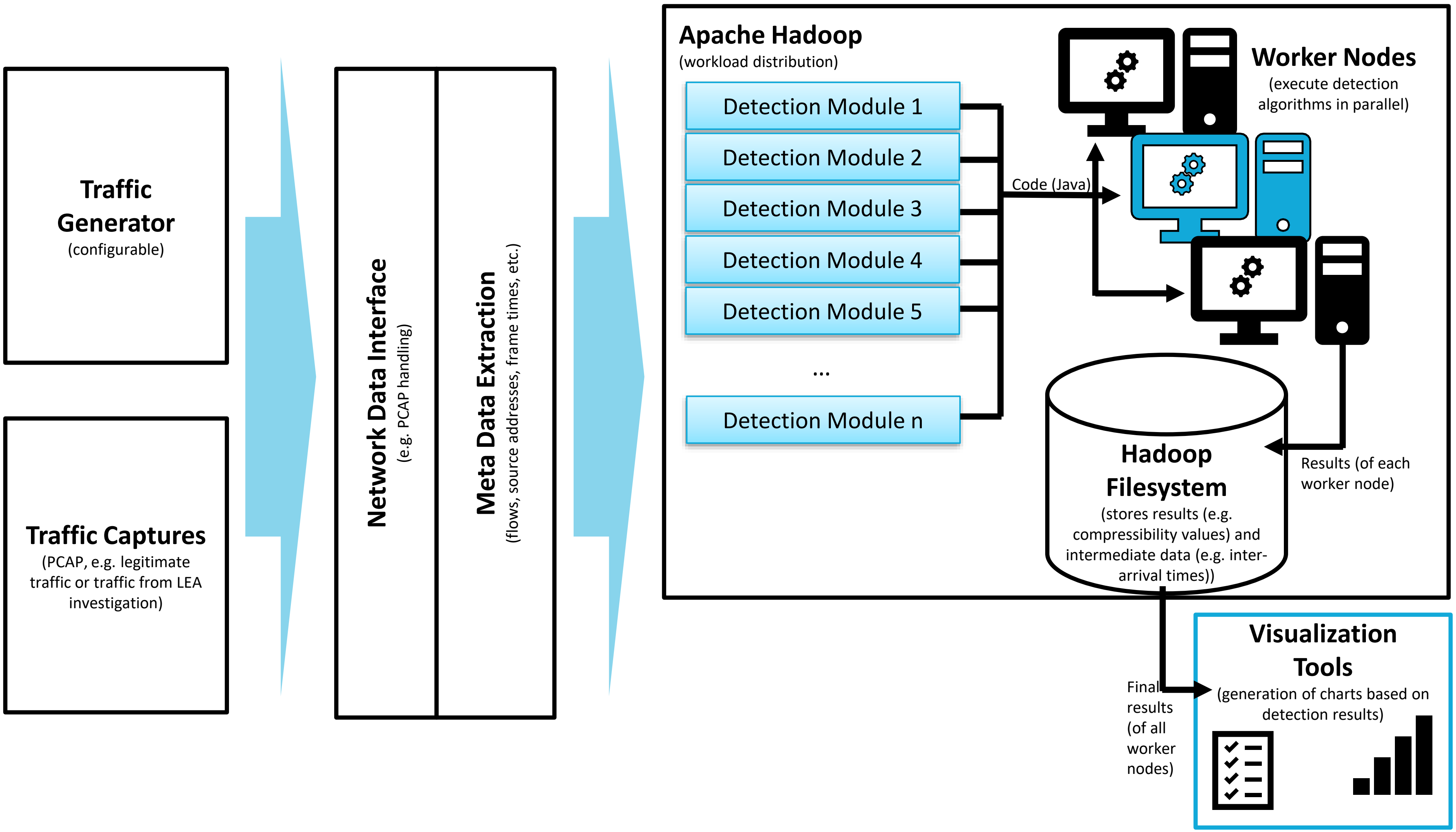
Replicating Experiments

- 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*)



Traffic Generator
(configurable)

Traffic Captures
(PCAP, e.g. legitimate traffic or traffic from LEA investigation)

Network Data Interface
(e.g. PCAP handling)

Meta Data Extraction
(flows, source addresses, frame times, etc.)

Apache Hadoop
(workload distribution)

- Detection Module 1
- Detection Module 2
- Detection Module 3
- Detection Module 4
- Detection Module 5
- ...
- Detection Module n

Code (Java)

Worker Nodes
(execute detection algorithms in parallel)

Results (of each worker node)

Hadoop Filesystem
(stores results (e.g. compressibility values) and intermediate data (e.g. inter-arrival times))

Final results (of all worker nodes)

Visualization Tools
(generation of charts based on detection results)

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.

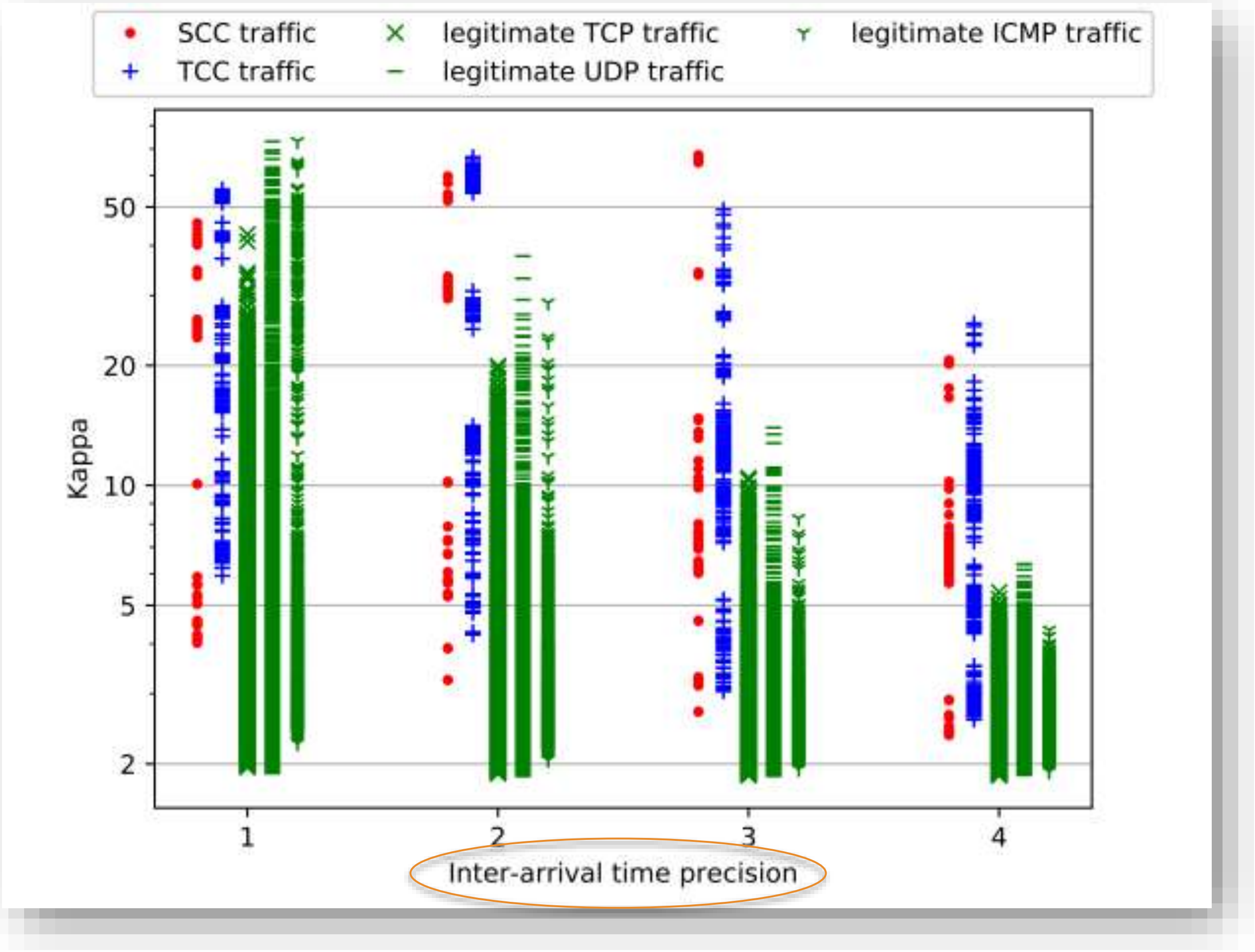
Replication Study: Compressibility of Cabuk et al.

- Published in ACM Transactions on Information and System Security (TISSEC), as an extended version of an ACM CCS paper.
- 130/450 citations
- However, compressibility was only covered in the journal version.

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.

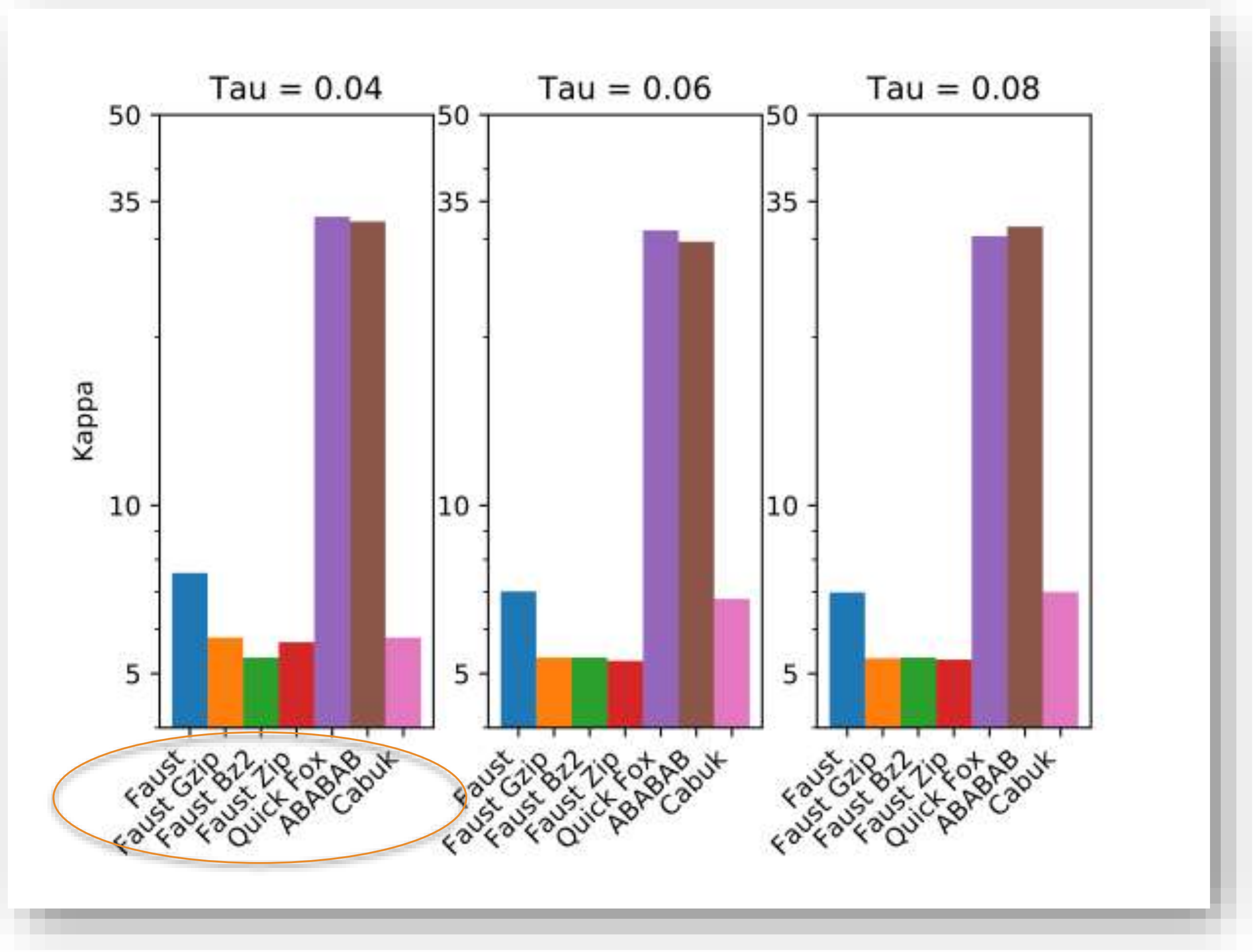
Let’s see how the precision of the measured IAT values influences Kappa…



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.

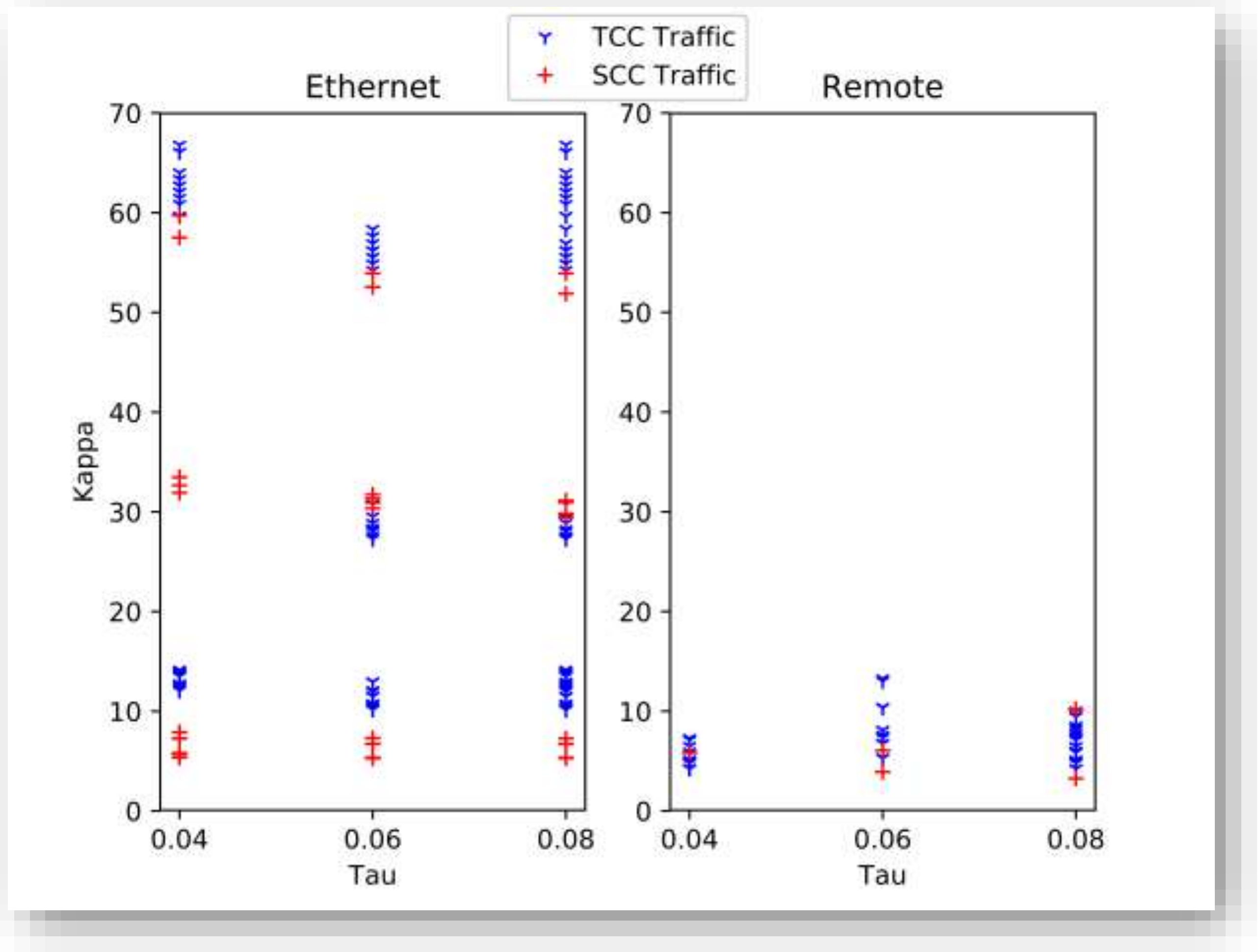
Let's see what happens if we transfer slightly different data over the covert channel …



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.

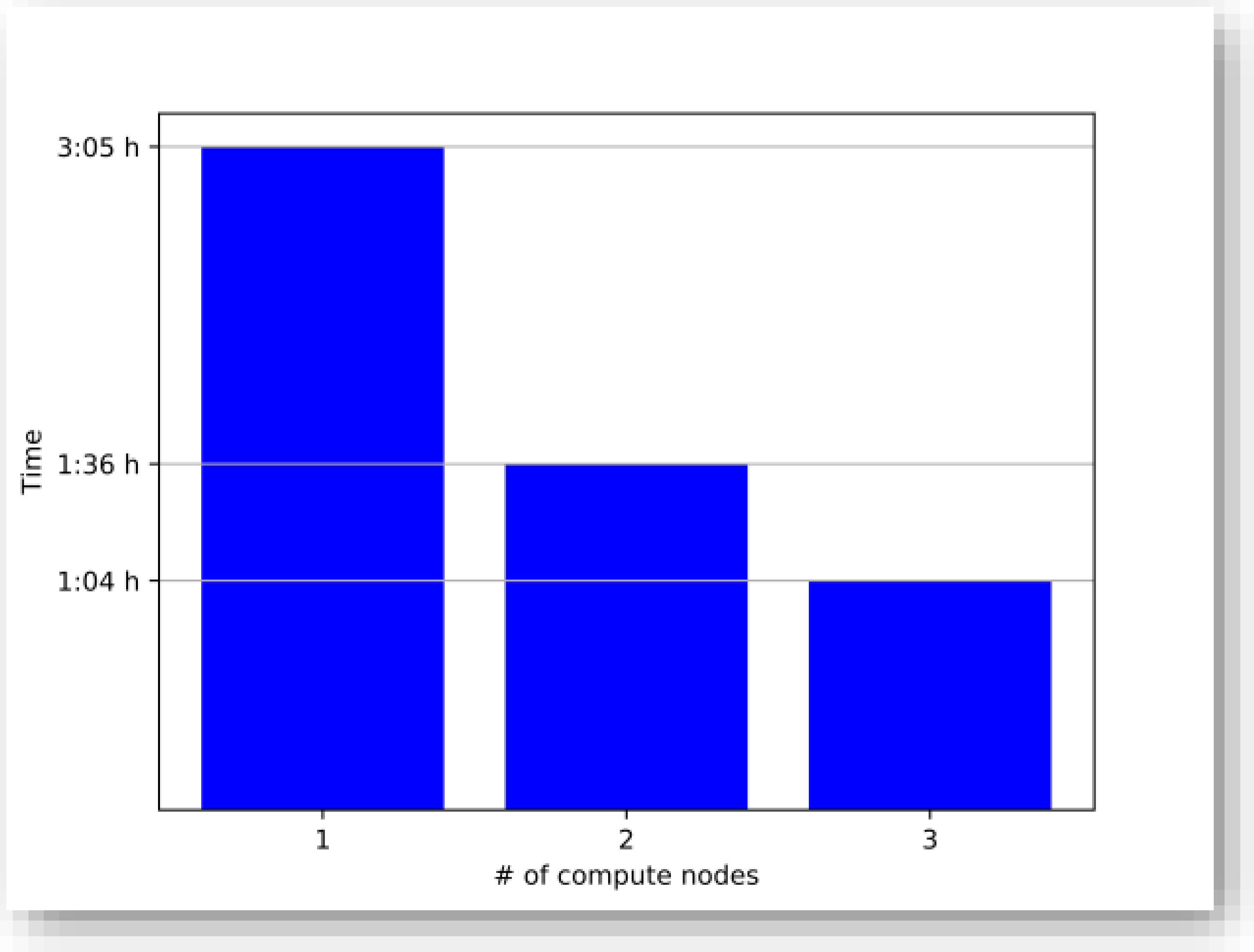
Let’s see how Kappa differs when we utilize a different network connection …



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.



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

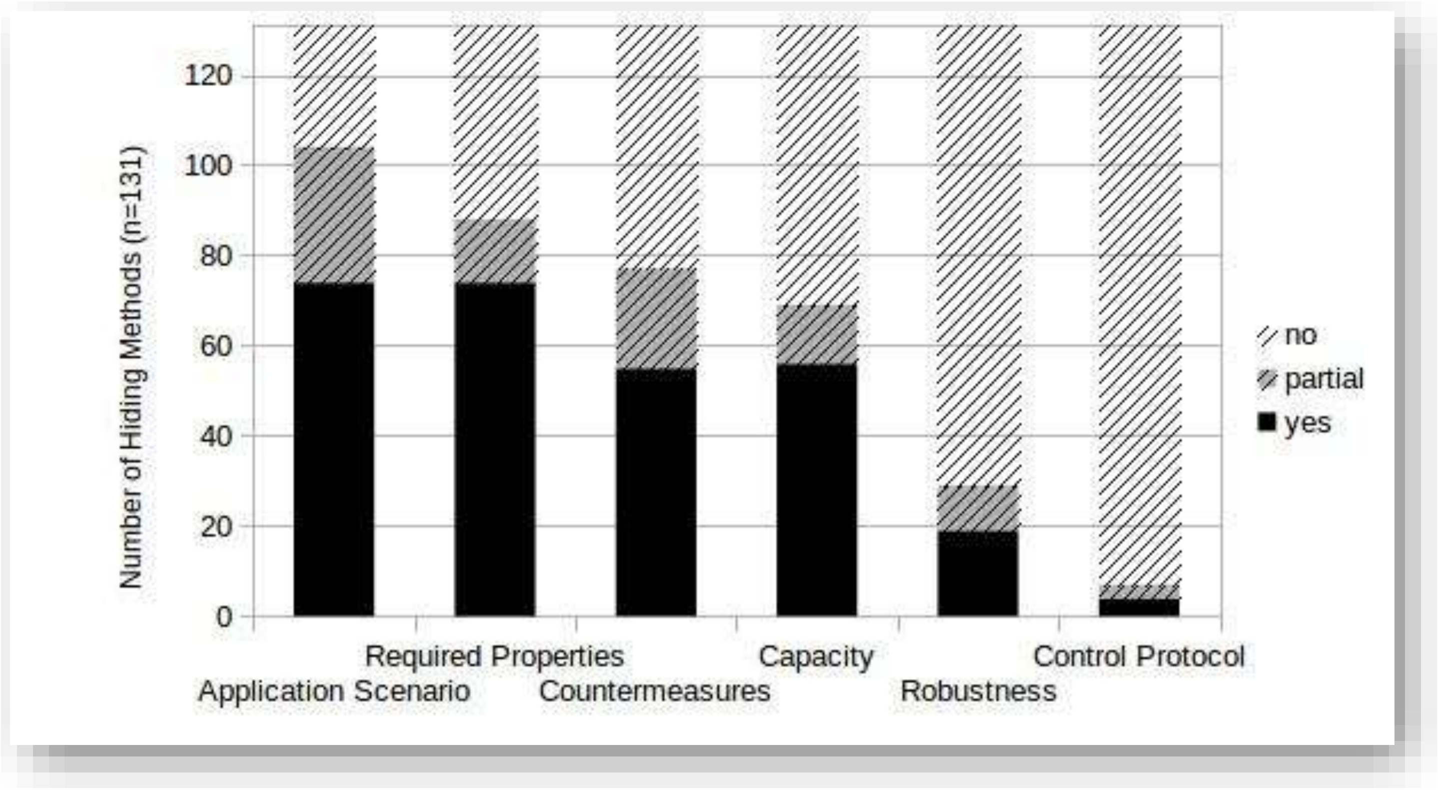
- Replication can be done and it can lead to new insights.

- 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 slightly changing circumstances.

HOW TO DESCRIBE A NEW HIDING METHOD?

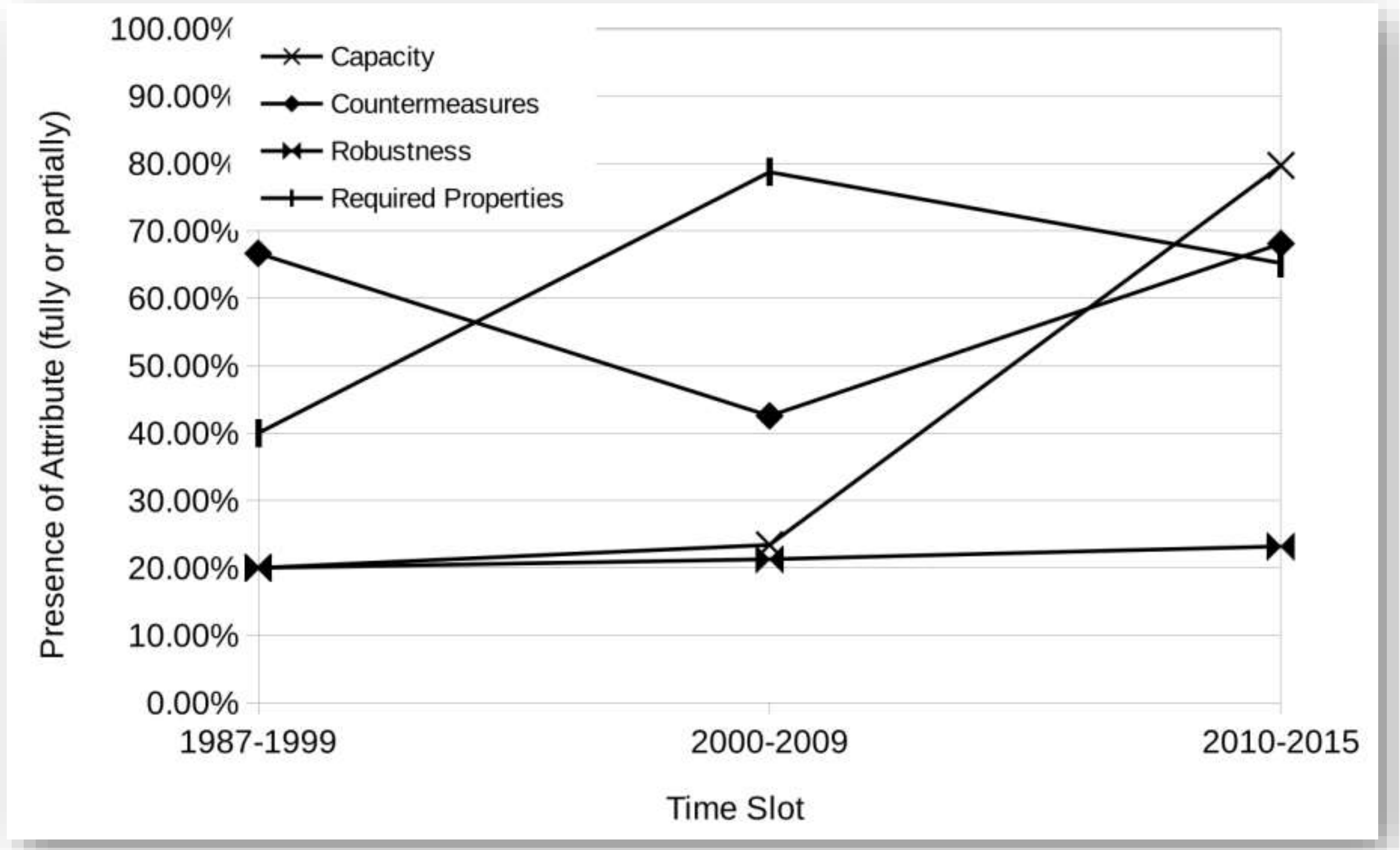
Analysis of 131 Hiding Techniques

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, 2016.

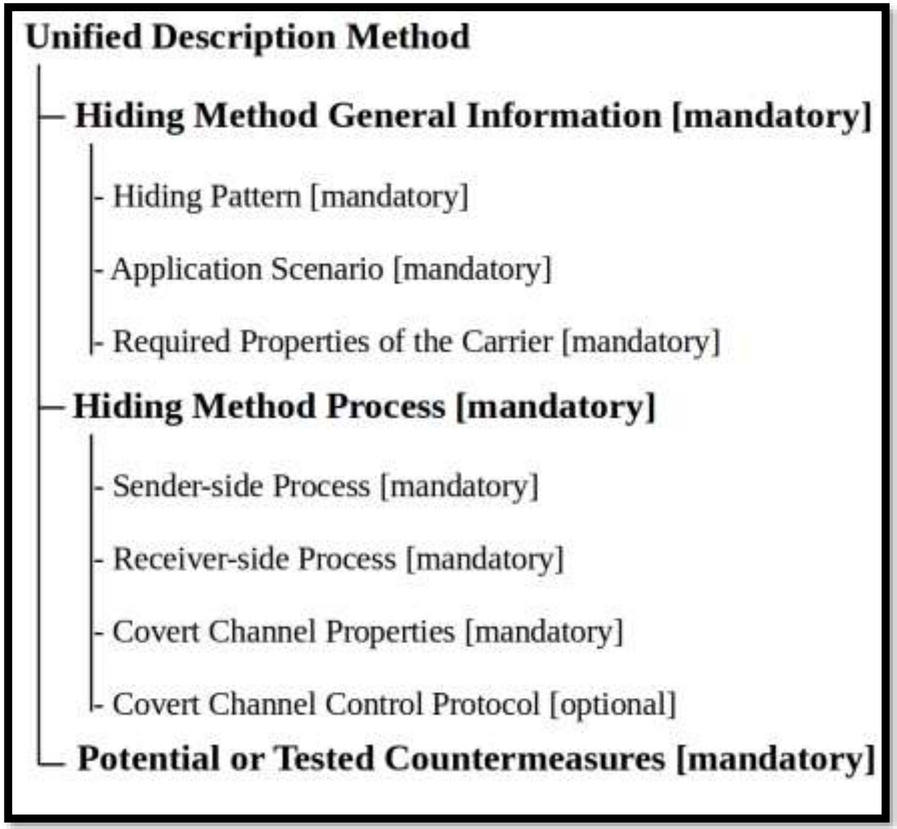
Analysis of 131 Hiding Techniques



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

How to describe a new hiding method?

- For this reason, 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.



Unified Description Method

- **Hiding Method General Information [mandatory]**
  - Hiding Pattern [mandatory]
  - Application Scenario [mandatory]
  - Required Properties of the Carrier [mandatory]
- **Hiding Method Process [mandatory]**
  - Sender-side Process [mandatory]
  - Receiver-side Process [mandatory]
  - Covert Channel Properties [mandatory]
  - Covert Channel Control Protocol [optional]
- **Potential or Tested Countermeasures [mandatory]**

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

Two Examples for Applying the Unified Description Method …

… can be found here:

http://www.jucs.org/jucs_22_11/unified_description_for_network.

Section based on S. Wendzel et al.: Don’t you touch my nuts – information hiding in cyber-physical systems, in Proc. IEEE S&P workshops, 2017, and on a follow-up journal article in JCSM [download].

STEGANOGRAPHY IN THE IOT

Information Hiding & Cyber-physical Systems

Information Hiding:
Steganography, copyright marking, anonymity, obfuscation [1]

+

Cyber Physical Systems (CPS):
integrations of computation with physical processes [2]

=

Information Hiding in Cyber-physical Systems
(specially Steganography for CPS (**CPSSteg**))

Related Work (Chronological Order)

- *Wendzel/Kahler/Rist* (2012) [3]:

  Scenario identification and description of secret data transmission in networked buildings; MLS-based protection approach

- *Tuptuk/Hailes* (2015) [4]:

  Two covert channels (relying on modulation of transmission power and of sensor data) in persuasive computing.

- *Howser* (2015) [5]:

  Data leakage in CPS and MLS-based protection (DLP)

- *Tonejc/Güttes/Kobekova/ Kaur* (2016) [6]:

  Detection of selected covert channels in building automation networks using unsupervised machine learning methods.

Network covert and side channels in CPS

Covert Channels in CPS
Exemplified using Smart Buildings

Network Covert Channel:

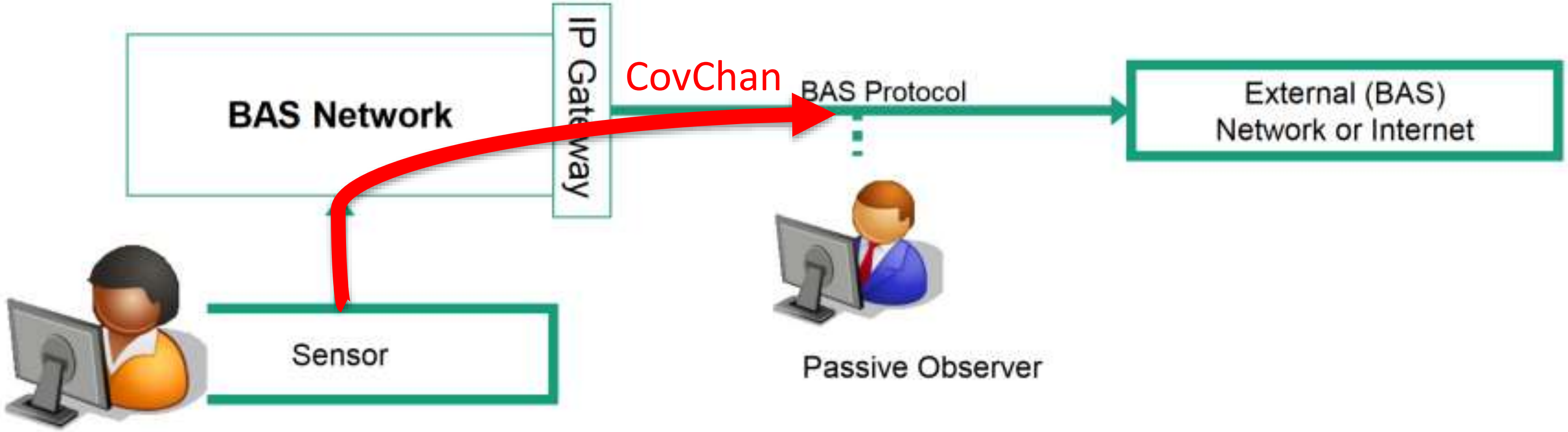
Data exfiltration to bypass common filter technologies such as DLP

Network Side Channel:

Policy-breaking observation of physical events, e.g. whether someone’s boss in his office or planning a theft (reading presence sensor, temperature sensor etc.).

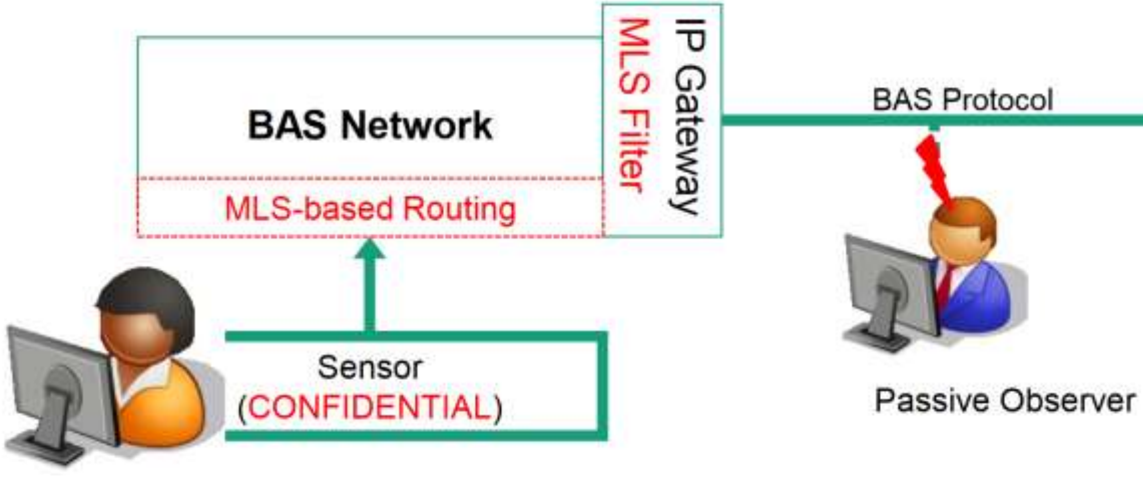
- e.g. breaking MLS-policy

Data Exfiltration through a Building Automation System



Wendzel, S., Kahler, B., & Rist, T. (2012). Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet. In Proc. *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on* (pp. 731-736). IEEE.

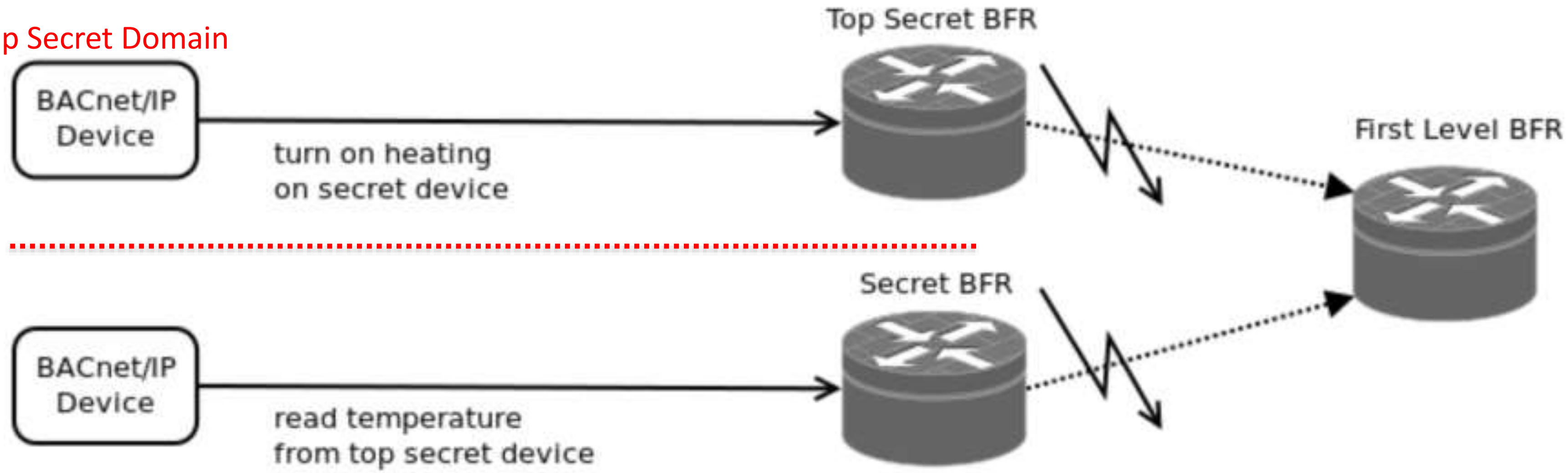
Data Exfiltration through a Building Automation System: MLS-Gateway



Wendzel, S., Kahler, B., & Rist, T. (2012). Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet. In Proc. *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on* (pp. 731-736). IEEE.

Data Exfiltration through a Building Automation System: MLS-Gateway

Top Secret Domain


Top Secret BFR
BACnet/IP Device
turn on heating on secret device
First Level BFR
Secret BFR
BACnet/IP Device
read temperature from top secret device

Secret Domain

Wendzel, S., Kahler, B., & Rist, T. (2012). Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet. In Proc. *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on* (pp. 731-736). IEEE.

Data Exfiltration through a Building Automation System: MLS-Gateway

Alternative:

Middleware solution for smart building apps.

CPS as a data storage

Goals & Strategies

- **Goals:**
  - Determining **how much data can be hidden** in a CPS **and for how long**.

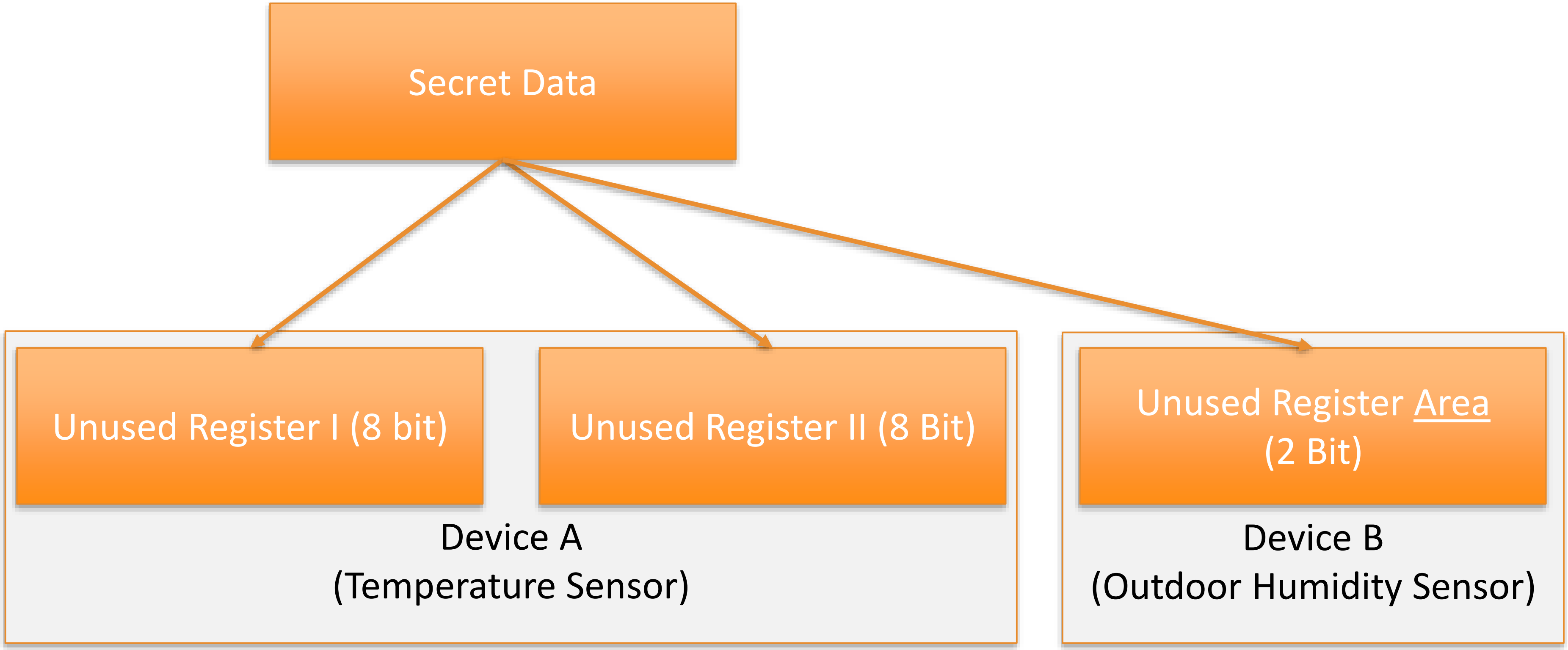
- **Possible Benefits:**
  - Storing secret data in a location where currently nobody will search for it, e.g. embedding a cryptographic key in a smart home.
  - *Fighting product piracy* [in progress]

- Analyzed **two different strategies**:
  - Register strategy: utilization of unused memory registers
  - Actuator strategy: storing data in actuator states (e.g. heating level of a heater) in a way that it will not be recognized

Option 1: Register Strategy

Register Strategy: Concept

- We store data in unused registers of CPS components.


Secret Data
Unused Register I (8 bit)
Unused Register II (8 Bit)
Unused Register Area (2 Bit)
Device A (Temperature Sensor)
Device B (Outdoor Humidity Sensor)

Register Strategy: Concept

- Drawbacks:
  - Writing registers may require direct (local bus) access to a CPS device
  - Register size (and thus steganographic storage) limited
  - Each different device model must be analyzed separately (e.g. datasheets)

- Advantages:
  - Several CPS components and CPS types contain unused registers
    - We used a temperature sensor that contains two unused registers; sensor could be embedded in several types of CPS.
  - Good reading and writing performance
  - Valuable to compare performance of **actuator strategy** (later) – is the more sophisticated approach actually *better*?

Register Strategy: Experiments

- Used Maxim Integrated Products, Inc., 1-Wire DS18B20 temperature sensor
  - Communication via 1-Wire protocol

- Approach:
  - Store data in the alarm registers (2x8 bits) of up to 4 sensors.
  - Sort data by sensor-internal unique serial number (can be read via bus connection)

- In experiment, measured time consumption of 100 reading operations from 1, 2 and 4 sensors simultaneously and of 100 writing operations (0x0000 followed by 0xffff in a loop) to one sensor.

Register Strategy: Results

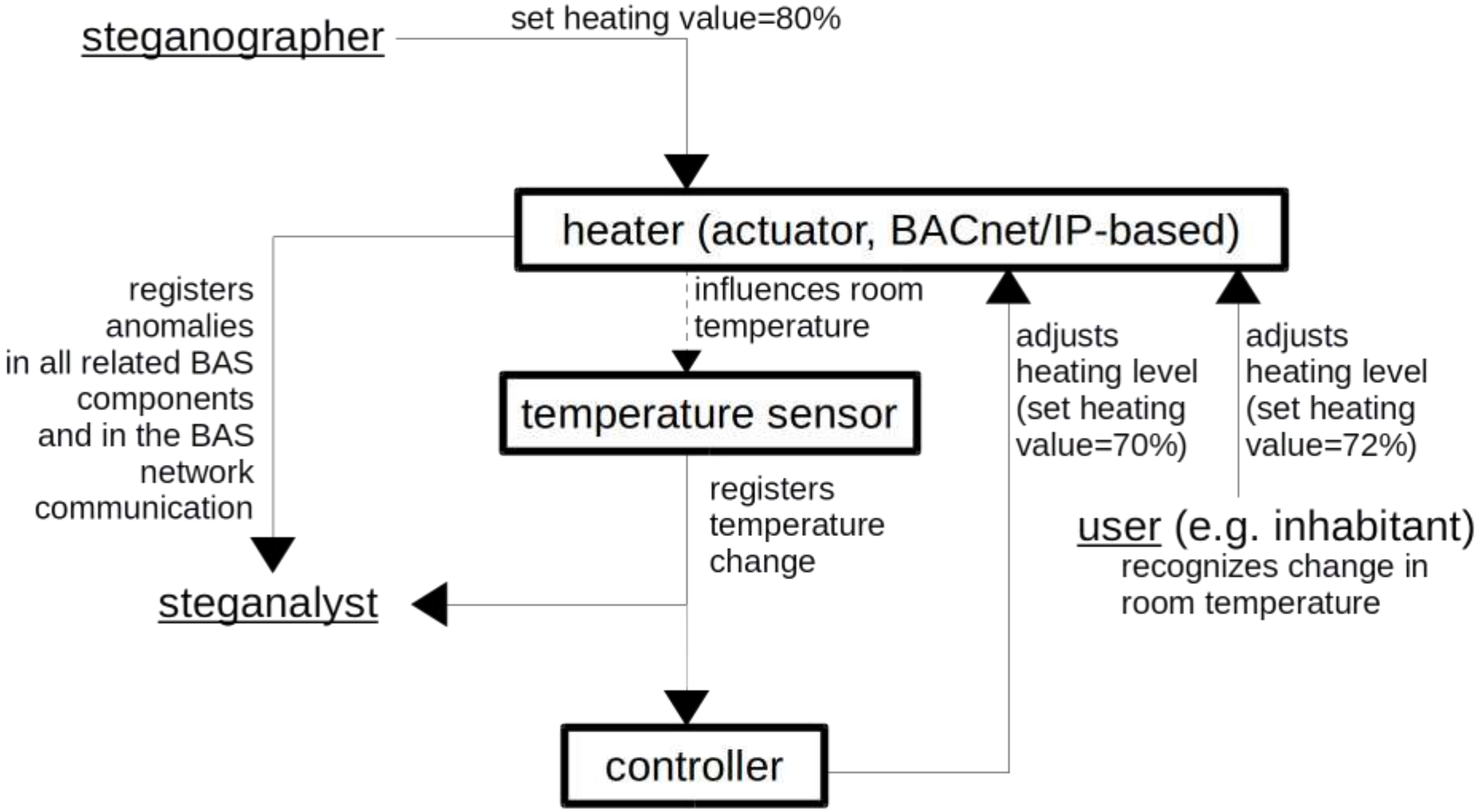
- Reading performance (avg.) per sensor increased with the number of sensors as addresses were only required to be fetched once.
- Values remained robust (0% reading errors within 180.000 operations)
- Thus, performance for steganographic operations not an issue.

Scenario	Avg. Time [µs]	Min. Time [µs]	Max. Time [µs]
Reading 1 Sensor	12.841	12.800	12.844
Reading 2 Sensors	12.804	12.784	12.806
Reading 4 Sensors	12.802	12.788	12.804
Writing 1 Sensor	71.827	71.800	71.834

No general conclusion on storage space possible, *probably* around *#SelectedDevices * 4-8 bits* (available register bits on average).
A single 128 bit crypto key would then require 16-32 devices.

Option 2: Actuator Strategy

Actuator Strategy: Concept


steganographer
set heating value=80%
heater (actuator, BACnet/IP-based)
registers anomalies in all related BAS components and in the BAS network communication
influences room temperature
temperature sensor
adjusts heating level (set heating value=70%)
adjusts heating level (set heating value=72%)
registers temperature change
user (e.g. inhabitant) recognizes change in room temperature
steganalyst
controller

Animal Scatter Hoarding

- For storing collected food, determine locations (caches) which remain mostly untouched by competing animals.
- Split food storage over many storage locations

Image source: Wikipedia, © SajjadF

Adaptive Information Hiding

- How to **determine suitable actuators** for secret data storage?
  - Scan for devices in a CPS environment, e.g. BACnet: “Who-Is” broadcast to determine present devices
  - Afterwards scan these devices to determine their objects and present values
  - Monitor changes of all actuator values over time and sort out unsuitable devices (e.g. door openers or devices with frequently changed states)

- Not a perfect solution:
  - Steganographer operates on the assumption that the CPS will behave as it behaved in the past (based on recordings of its historic behavior)
  - But future CPS behavior cannot be predicted with 100% accuracy based on the historic behavior
    - imagine an open house presentation: building automation system’s actuators will most likely be used in different way, e.g. a previously unused room will be heated
  - Still requires use of error detecting/correcting codes, e.g. parity bits or spreading of redundant data over several devices

Actuator Strategy: Experiments

- Simulated scatter hoarding using the BACnet protocol
  - ISO standard for communication in automated buildings
  - **How well can we store steganographic data under different conditions?**

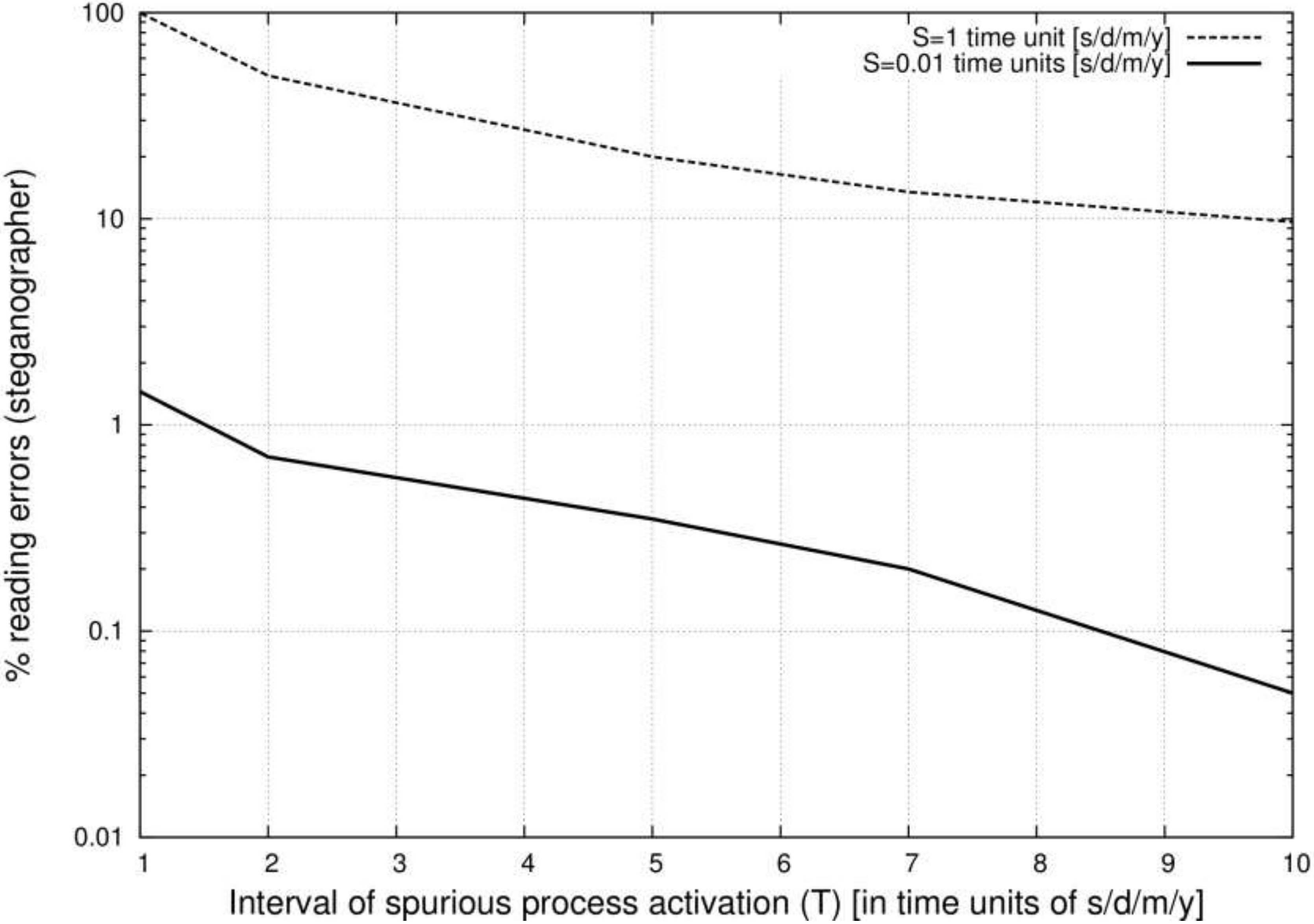
- **In general:** Wrote 100,000 values to an actuator (iterating through values 0°C…100°C). After each value written, the current value was read from the device.

- **Experiment 1**: Introduction of a Spurious Process [7] (**Read-only**)
  - Spurious read-only process (resulted in slow-down of steganographer's process but no data loss as BACnet protocol was able to re-send non-acknowledged packets).

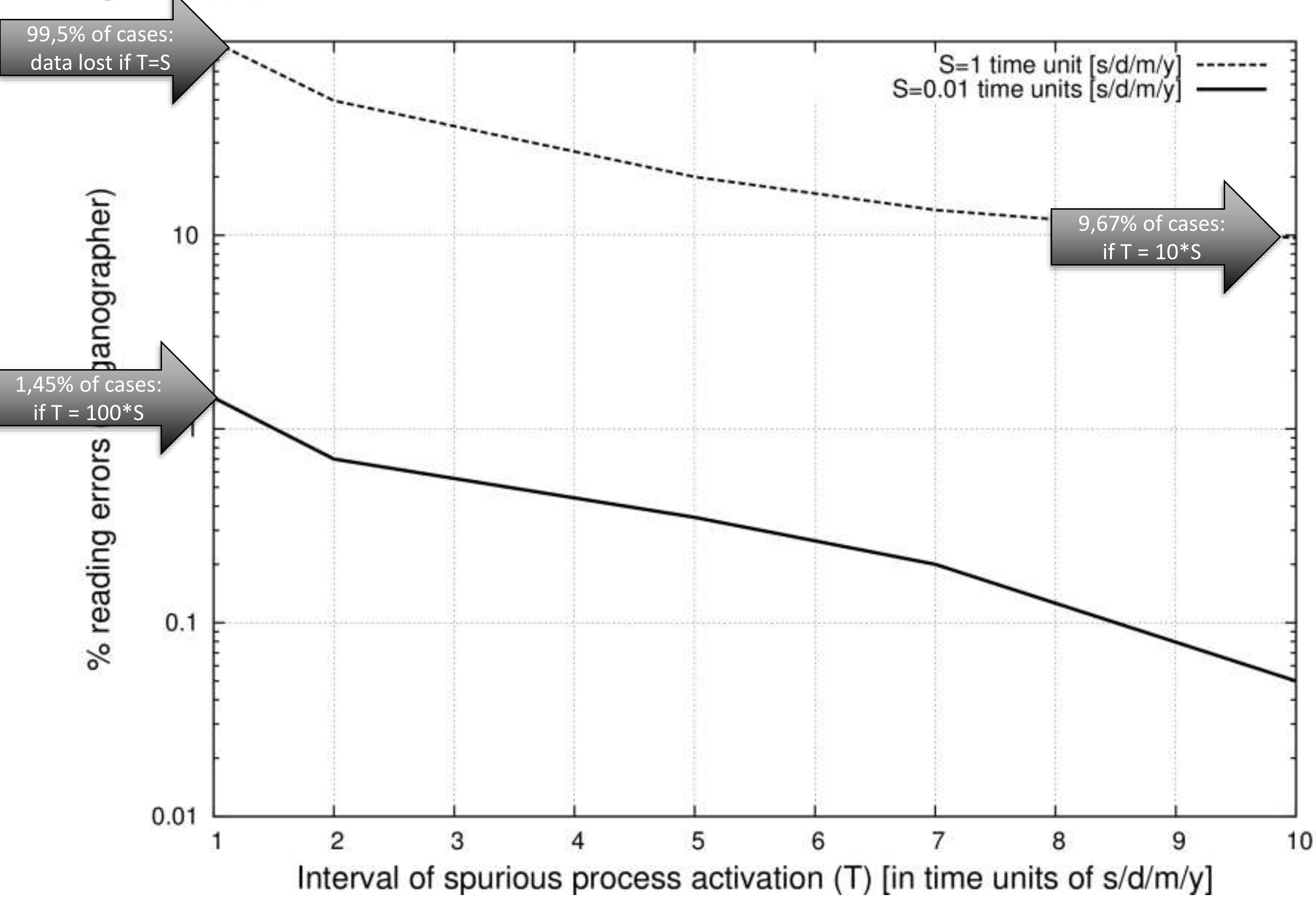
Actuator Strategy: Experiments

- **Experiment 2**: Spurious Process (**Read-Write**)
  - SP represents inhabitant or control logic that changes actuator states
  - Competing animal detects hoarding location (read) and steals food (replacing stored value with a random value)
  - Spurious process writes data every T seconds while the desired storage time was S seconds.
  - Simulated situations reaching from highly spurious ($T = S$) to few spurious intrusions ($S \ll T$).

Actuator Strategy: Results



Actuator Strategy: Results

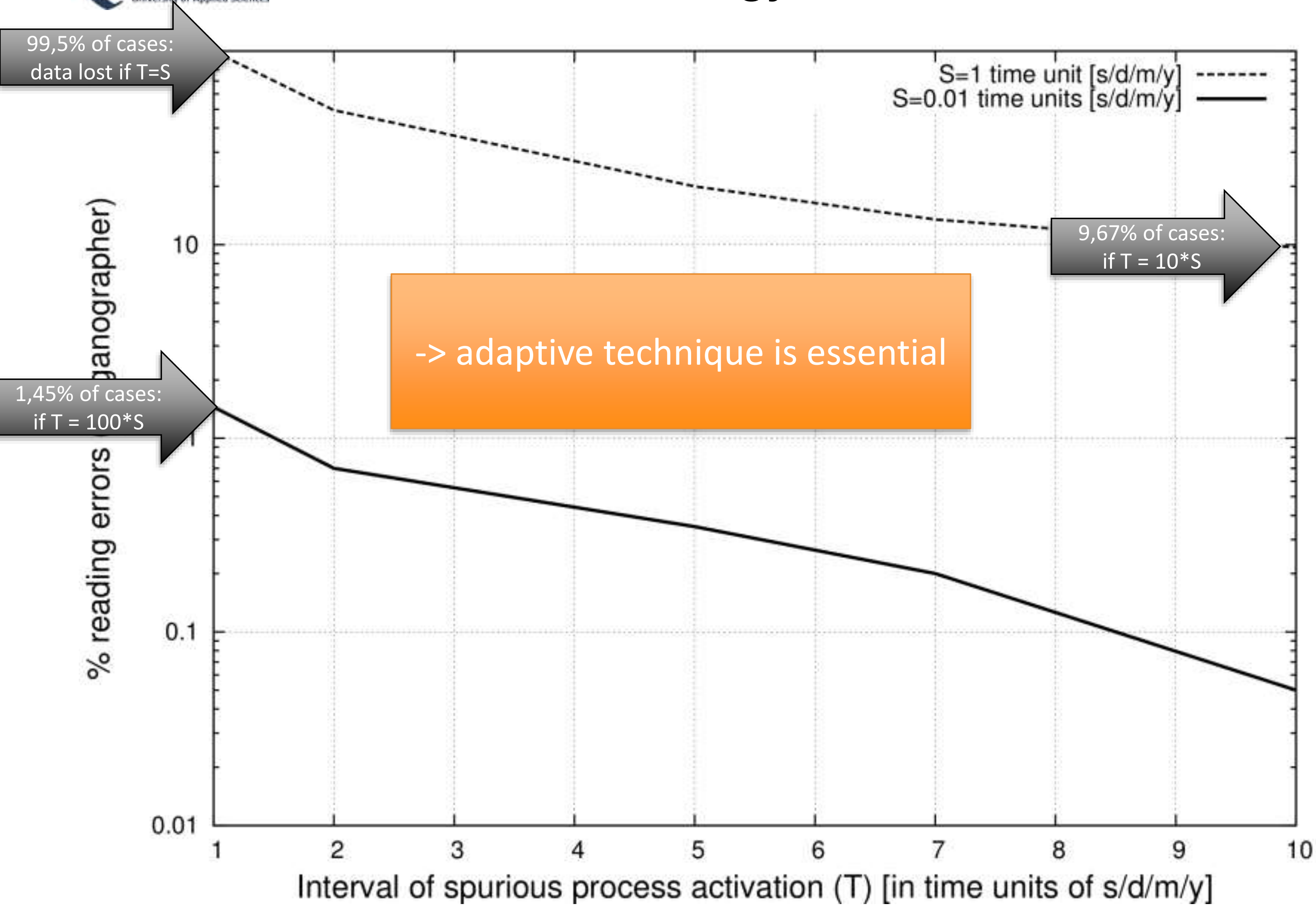


99,5% of cases: data lost if T=S

9,67% of cases: if T = 10*S

1,45% of cases: if T = 100*S

Actuator Strategy: Results



99,5% of cases: data lost if T=S

9,67% of cases: if T = 10*S

1,45% of cases: if T = 100*S

-> adaptive technique is essential

Actuator Strategy: Results

- Storage capacity of actuators highly depends on actuator type (e.g. boolean on-off switches or heaters that provide a fine distinction between heating levels).
- We can assume storage capacity of *2-7* bits per *useful* actuator
  - *18-64* actuators for a 128 bit AES key (**more than in case of register approach!**)
- If we further assume *5-10%* of actuators could be utilized in medium-sized BACnet environments (e.g. 1,000-20,000 actuators), we could store approx. *350* bits - *1.7* Kbytes if *7* secret bits/device can be stored.
- **Advantage: unified accessibility** of actuator approach using common protocol (BACnet) over register approach (individual register access needed!)
  - Especially in larger installations
- Performance: ~0.0055 sec per value that must be written/read
  - some actuators much slower
  - some bus systems much slower
  - 0% reading errors without RW spurious process

Limitations and Future Work

- Structure, environments and capabilities of CPS can vary strongly between different CPS types (e.g. smart building vs. wearable).
  - Further studies needed for other CPS types.
  - Caused influence of steganographer on CPS (and its physical environment) not necessarily clear -> CPSSteg considered risky.
    - Probably not suitable for ICS.

- Novel approaches for information hiding in CPS can be expected.
  - One could use **BACnet COV Subscription** relationships to encode steganographic data.
  - Embedding data for copyright marking, e.g. DRM for smart buildings to fight piracy of products (e.g. using CPS traffic obfuscation or covert channels).

Conclusion

- Covert data exfiltration over CPS is a promising practical approach.

- The amount of data we can store in a CPS highly depends on
  - How we embed data (hiding method)
  - How many devices are available (e.g. #actuators)

- Similarly, these factors influence the robustness of the embedded data.

References

1. W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, K. Szczypiorski: **Information hiding in communication networks**, Wiley-IEEE, 2016.

2. E. A. Lee: **Cyber physical systems: design challenges**, Proc. 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), IEEE, 2008.

3. S. Wendzel, B. Kahler, T. Rist: **Covert channels and their prevention in building automation protocols: a prototype exemplified using BACnet**, Proc. IEEE CPSCom Workshop on Security of Systems and Software Resiliency, IEEE, 2012.

4. N. Tuptuk, S. Hailes: **Covert channel attacks in pervasive computing**, Int. Conf. on Pervasive Computing and Communications (PerCom), IEEE; 2015.

5. G. Howser: **Using information flow methods to secure cyber-physical systems**, in: Critical Infrastructure Protection IX, Springer, 2015.

6. J. Tonejc, S. Güttes, A. Kobekova, J. Kaur: **Machine learning methods for anomaly detection in BACnet networks**, Journal of Universal Computer Science (J.UCS), Vol. 22(9), 2016.

7. Y. Fadlalla: **Approaches to Resolving Covert Storage Channels in Multilevel Secure Systems**, Ph.D. Thesis, University of New Brunswick, 1996.

A brief outlook on the

ANALYSIS OF CITATIONS

IN COVERT CHANNEL/STEGO RESEARCH. BASED ON THIS PAPER.

Background

In Scientometrics, quite some research was conducted to analyze impacting factors on citations.

Q: Are the observations made in other domains also applicable in information security?

We performed a first study on citations in the covert channel/ steganography domain (to be published next month; based on meta-data extracted from IEEEXplore).

Here comes some preview that also features other domains such as NetSec and Crypto.

S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

Limitations

- Solely based on data from IEEE Xplore (and partially of Google Scholar – not shown in this lecture), i.e. cannot draw conclusions on data of Springer, ACM, Elsevier etc.

- No distinction between self-citations and citations of other authors.

- No analysis of the quality of a paper's content.

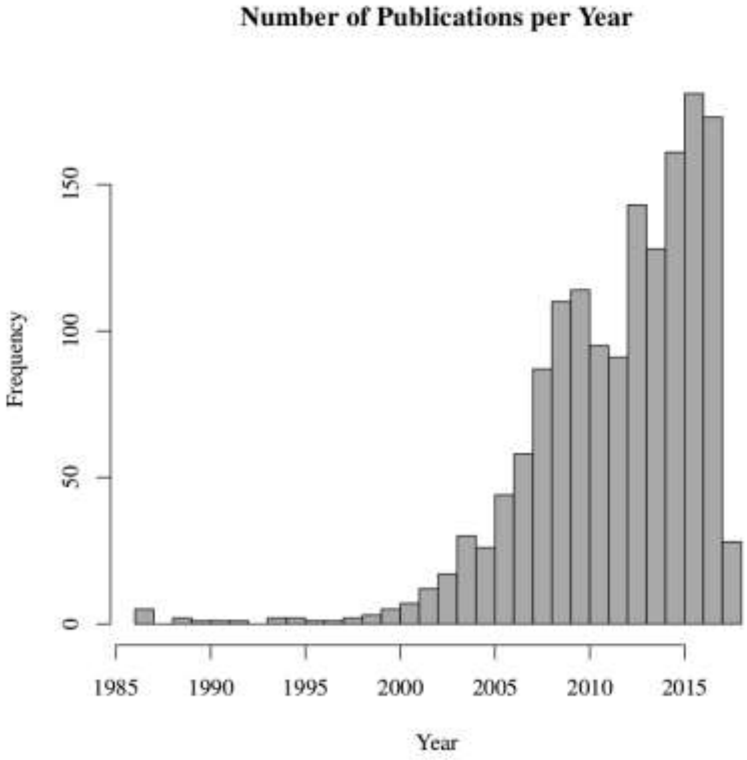
- Goal is not to draw any conclusions in the sense of: If you do X, your paper will receive more citations.

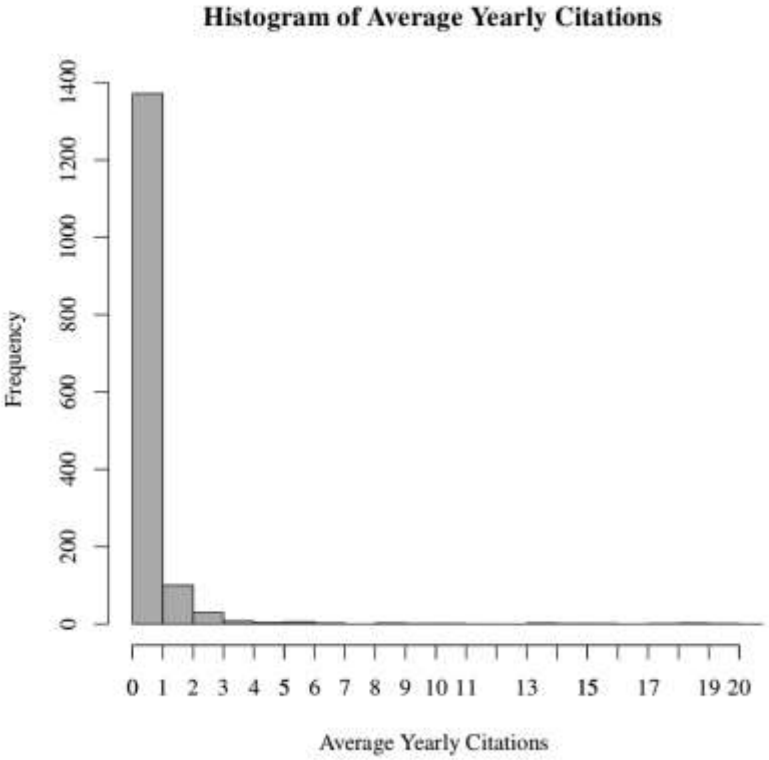
  Example:

  We can say that papers with attribute X (e.g. more pages) receive more citations.
  but maybe this is because of conferences that demand a higher number of pages and the papers of the particular conference receive more citations than other conferences). → Performed no analysis of such aspects!

S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

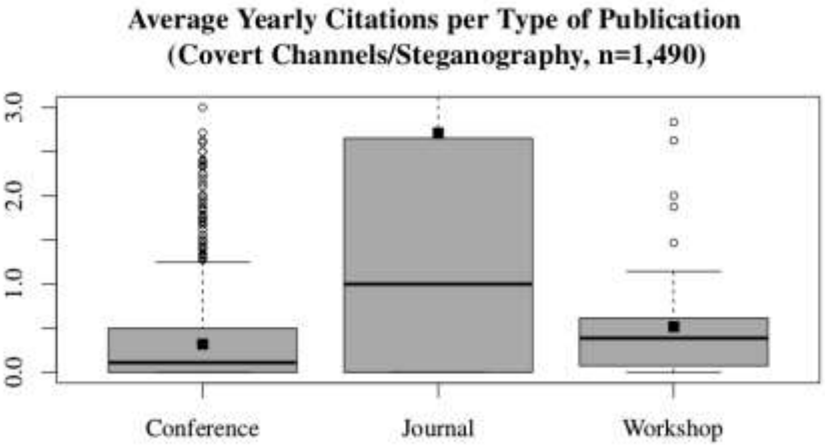
Results in a Nutshell





S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

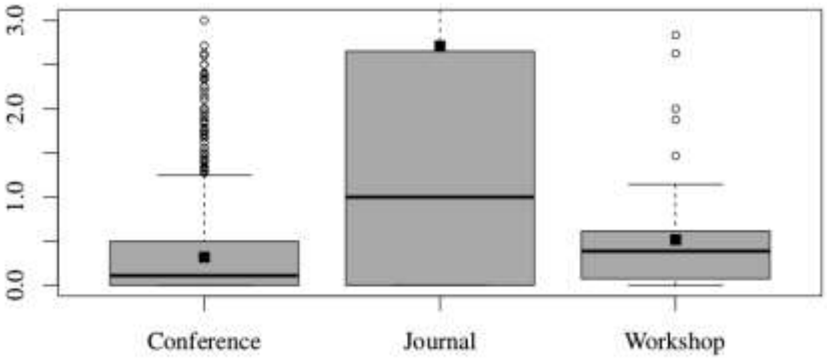
Results in a Nutshell



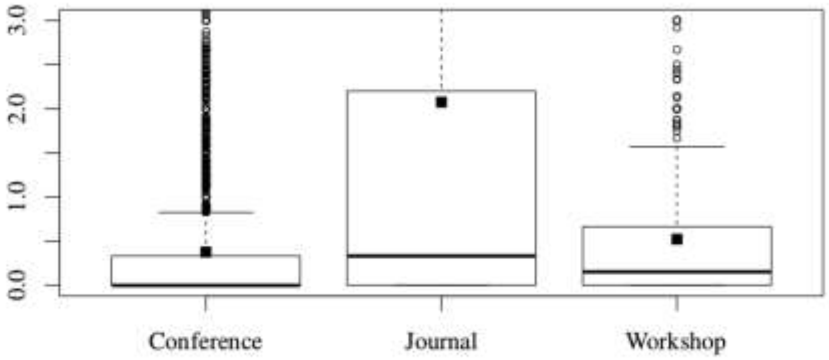
S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

Results in a Nutshell

Average Yearly Citations per Type of Publication
(Covert Channels/Steganography, n=1,490)

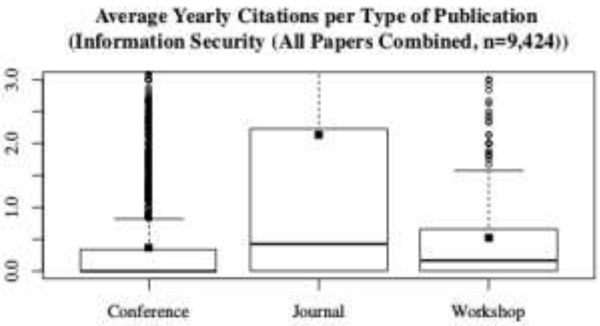


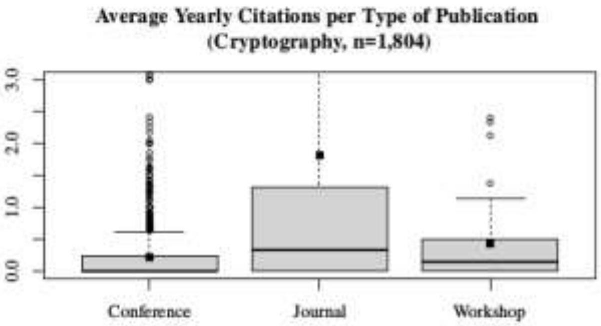
Average Yearly Citations per Type of Publication
(Non−Covert Channel/Steganography Domains, n=7,934)

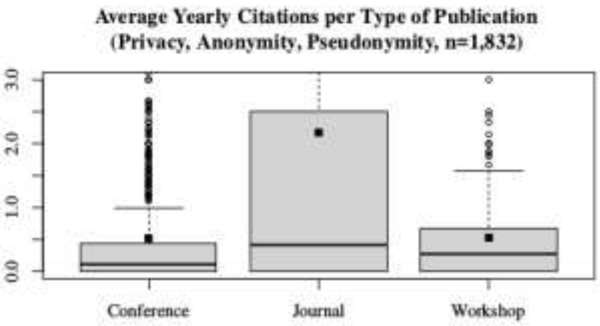


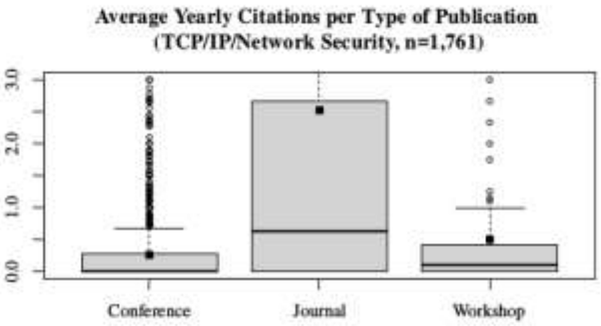
S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES’18.

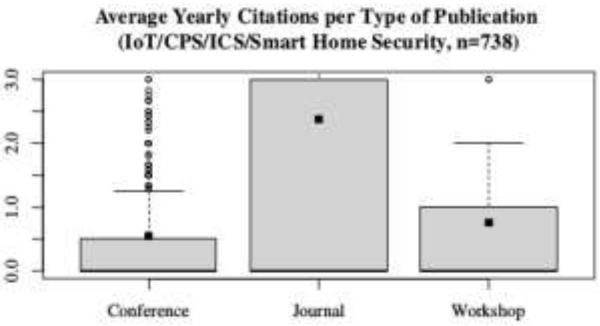
Results in a Nutshell

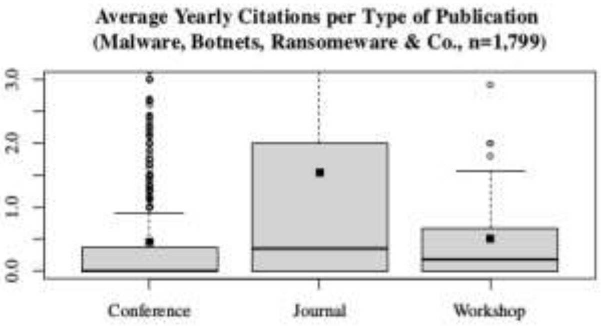

Average Yearly Citations per Type of Publication
(Information Security (All Papers Combined, n=9,424))


Average Yearly Citations per Type of Publication
(Cryptography, n=1,804)


Average Yearly Citations per Type of Publication
(Privacy, Anonymity, Pseudonymity, n=1,832)

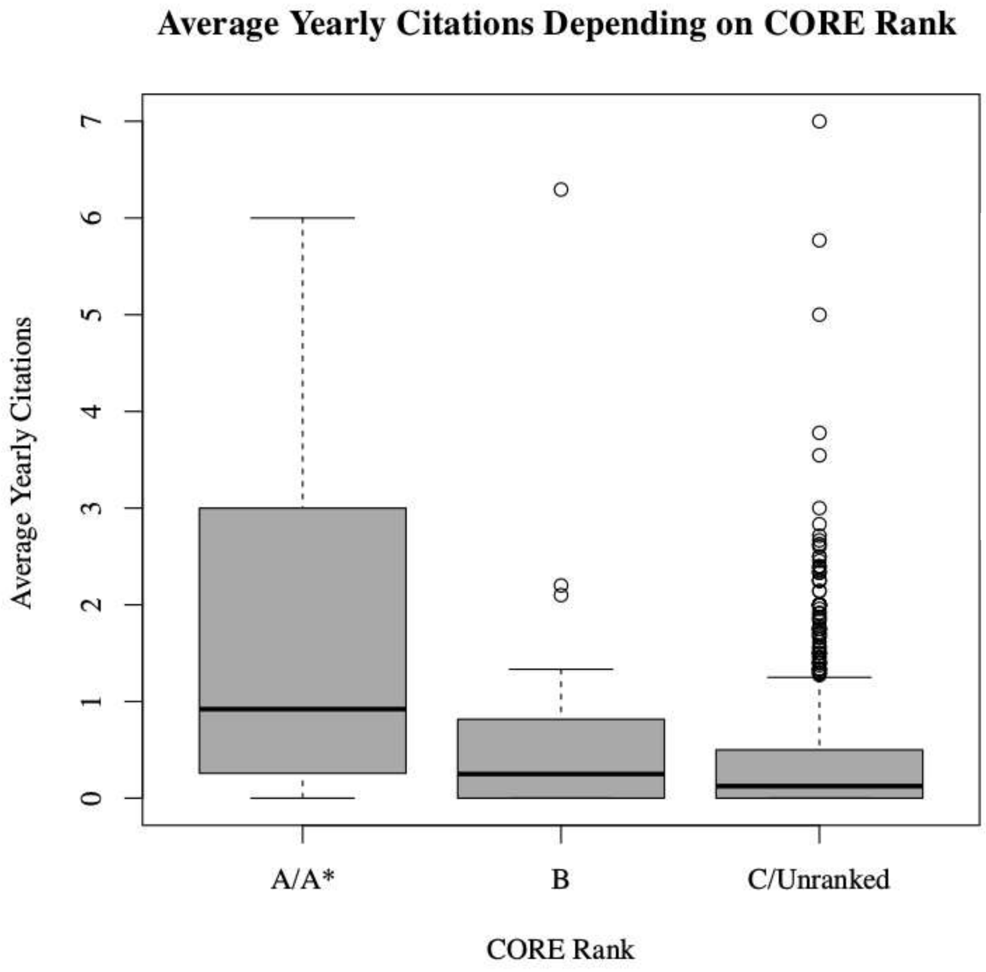

Average Yearly Citations per Type of Publication
(TCP/IP/Network Security, n=1,761)


Average Yearly Citations per Type of Publication
(IoT/CPS/ICS/Smart Home Security, n=738)


Average Yearly Citations per Type of Publication
(Malware, Botnets, Ransomeware & Co., n=1,799)

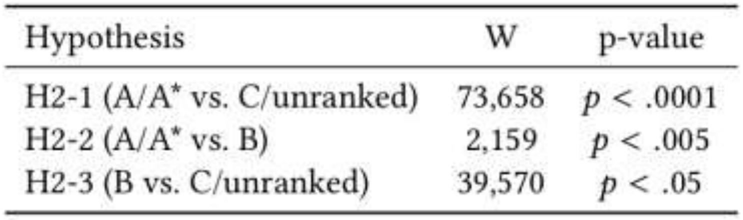
S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES’18.

Results in a Nutshell



Average Yearly Citations Depending on CORE Rank

Table 2: Test results for hypotheses H2-1 and H2-2.



Hypothesis	W	p-value
H2-1 (A/A* vs. C/unranked)	73,658	$p < .0001$
H2-2 (A/A* vs. B)	2,159	$p < .005$
H2-3 (B vs. C/unranked)	39,570	$p < .05$

S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

Results in a Nutshell

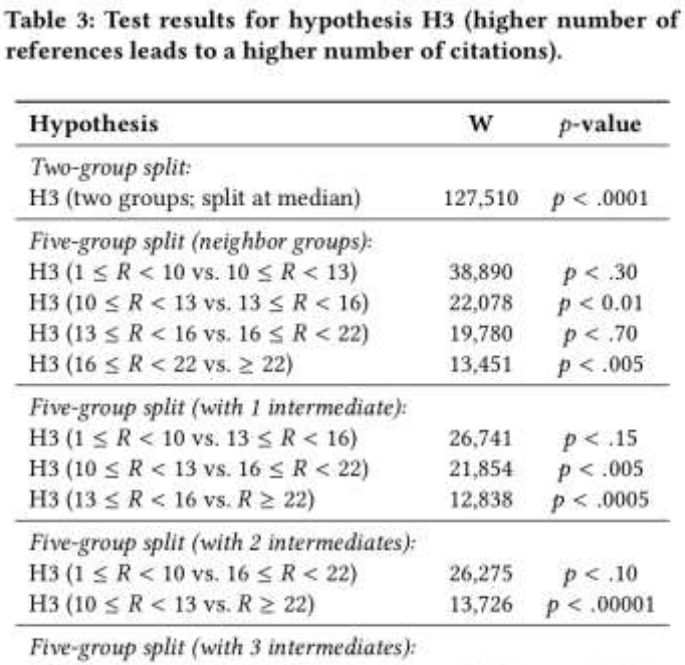
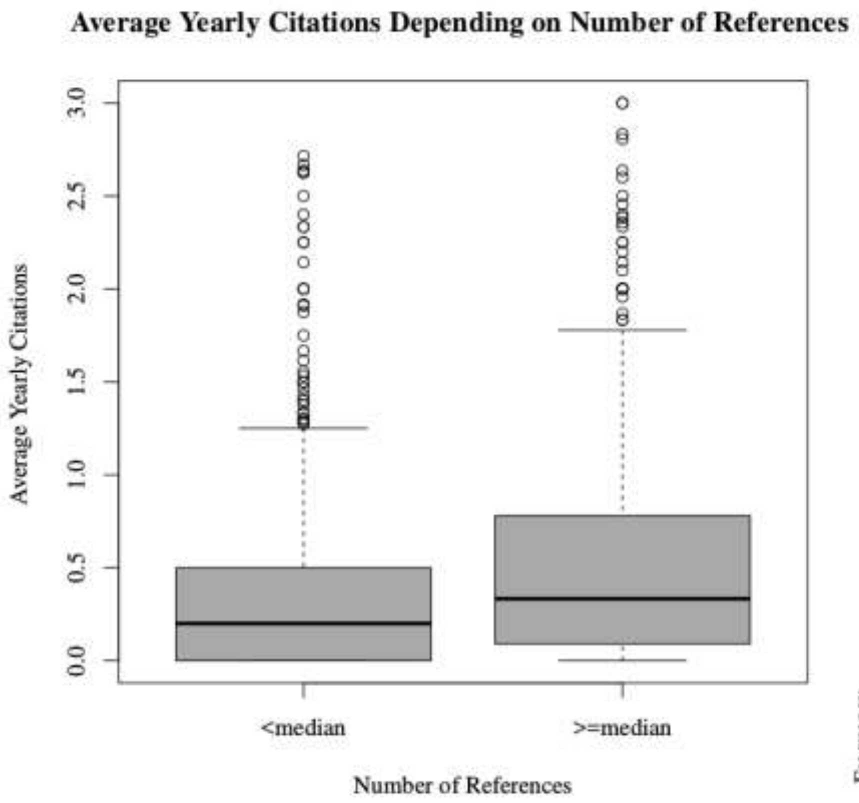
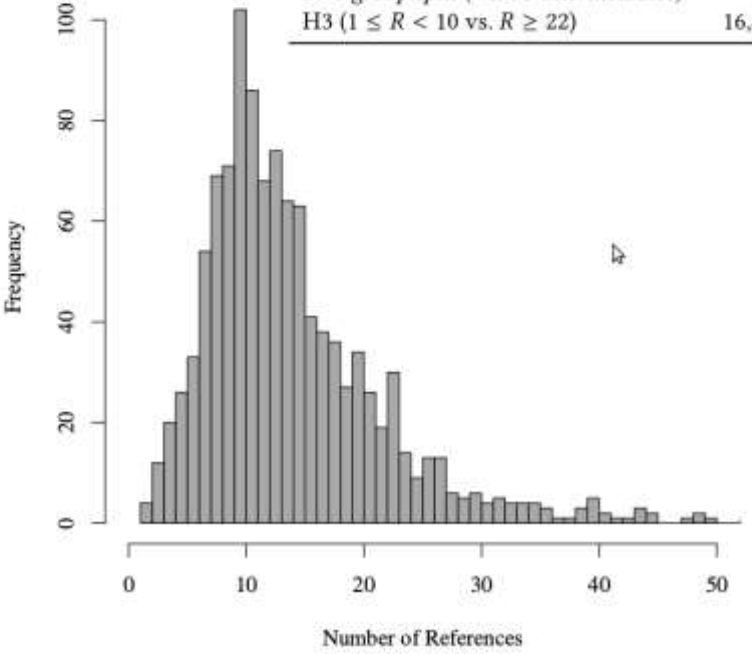


Table 3: Test results for hypothesis H3 (higher number of references leads to a higher number of citations).

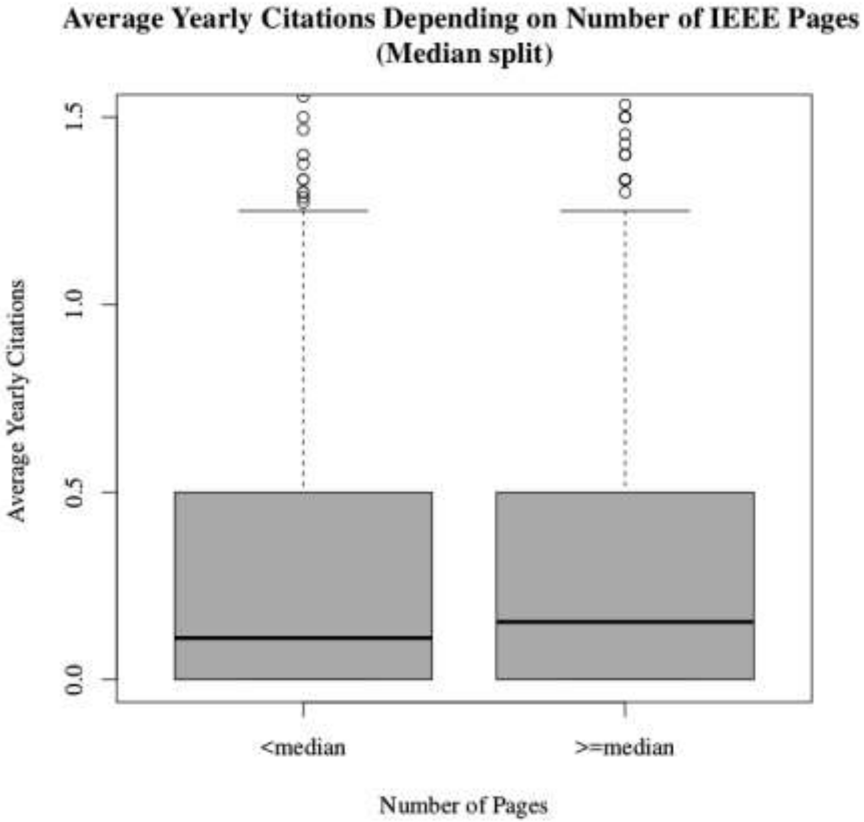
Hypothesis	W	p-value
Two-group split:		
H3 (two groups; split at median)	127,510	$p < .0001$
Five-group split (neighbor groups):		
H3 ($1 \leq R < 10$ vs. $10 \leq R < 13$)	38,890	$p < .30$
H3 ($10 \leq R < 13$ vs. $13 \leq R < 16$)	22,078	$p < 0.01$
H3 ($13 \leq R < 16$ vs. $16 \leq R < 22$)	19,780	$p < .70$
H3 ($16 \leq R < 22$ vs. ≥ 22)	13,451	$p < .005$
Five-group split (with 1 intermediate):		
H3 ($1 \leq R < 10$ vs. $13 \leq R < 16$)	26,741	$p < .15$
H3 ($10 \leq R < 13$ vs. $16 \leq R < 22$)	21,854	$p < .005$
H3 ($13 \leq R < 16$ vs. $R \geq 22$)	12,838	$p < .0005$
Five-group split (with 2 intermediates):		
H3 ($1 \leq R < 10$ vs. $16 \leq R < 22$)	26,275	$p < .10$
H3 ($10 \leq R < 13$ vs. $R \geq 22$)	13,726	$p < .00001$
Five-group split (with 3 intermediates):		
H3 ($1 \leq R < 10$ vs. $R \geq 22$)	16,837	$p < .00001$

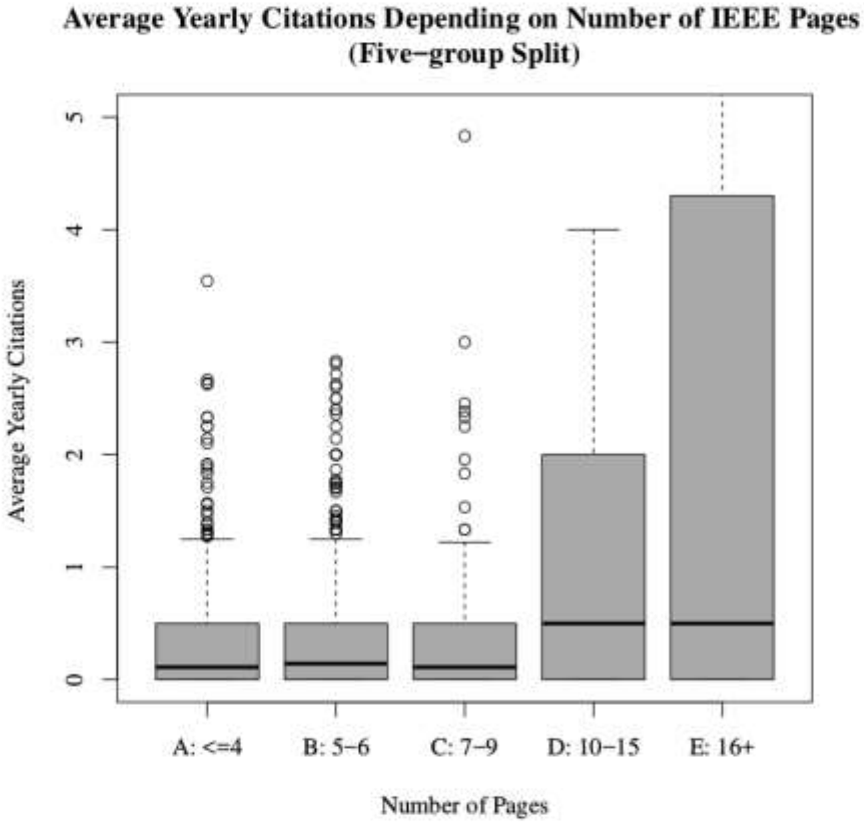




S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

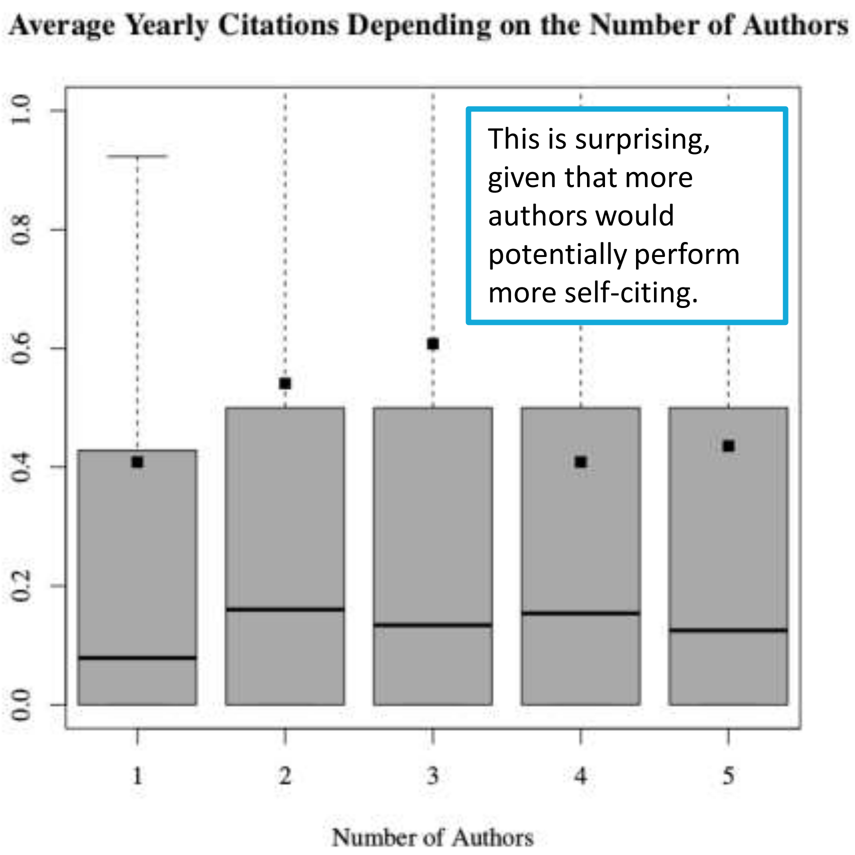
Results in a Nutshell


Average Yearly Citations Depending on Number of IEEE Pages (Median split)


Average Yearly Citations Depending on Number of IEEE Pages (Five−group Split)

S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES’18.

Results in a Nutshell


Average Yearly Citations Depending on the Number of Authors

This is surprising, given that more authors would potentially perform more self-citing.

No statistically significant difference was found between any of these groups of papers (also not for a two-group split at the median).
This is surprising since other research on other sciences has shown that when the number of authors is twice the mean, the observed citations of publications increased by between 24% (physics) and 52% (mathematics). (cf. *E. S. Vieira and J. A. Gomes. 2009. Citations to scientific articles: Its distribution and dependence on the article features. Journal of Informetrics 4 (2009), 1–13.*)

S. Wendzel: Get me cited, Scotty! Analysis of Citations in Covert Channel Research, Proc. ARES'18.

Future Work

- As mentioned, our work is in a pretty early stage. Several aspects are subject to future work.
  - Studying impact of self-references
    - We are working on it but names are not unique …
  - Impact of the length of the abstract
  - Impact of the number of keywords
  - …

Most importantly: studying non-covert channel domains.

OVERALL CONCLUSION

Conclusion

- There are 150+ hiding techniques for network data (+ countless techniques for payload, i.e. digital media steganography).
- Developing countermeasures to limit/detect/prevent all of these hiding techniques separately is extremely challenging.
- However, we can categorize all these hiding techniques in few **hiding patterns** (at least 11, a finer-granular distinction with 14 patterns is also available).

Open Research Problems

- Pattern-based countermeasures can be considered a promising concept, especially **countermeasure variation**.

- In general: development of sophisticated countermeasures is more challenging and more interesting than development of new hiding methods.

- We already know many hiding methods for several protocols. However, for **upcoming** network protocols, a covert channel analysis is a good idea (if described with e.g. the unified description method, so that results can be compared later).

- CPS steganography still in its infancies. Impact unclear.

Are there any questions?

THANK YOU FOR YOUR KIND ATTENTION.

My publications are available here.