Hochschule Worms
University of Applied Sciences

NETWORK INFORMATION HIDING

CH. 6: SOPHISTICATED HIDING METHODS & DISTRIBUTED HIDING PATTERNS

Prof. Dr. Steffen Wendzel
Worms University of Applied Sciences

https://www.wendzel.de (EN) | https://www.hs-worms.de/wendzel/ (DE)

Online Class: https://github.com/cdpxe/Network-Covert-Channels-A-University-level-Course/

Pattern Variation

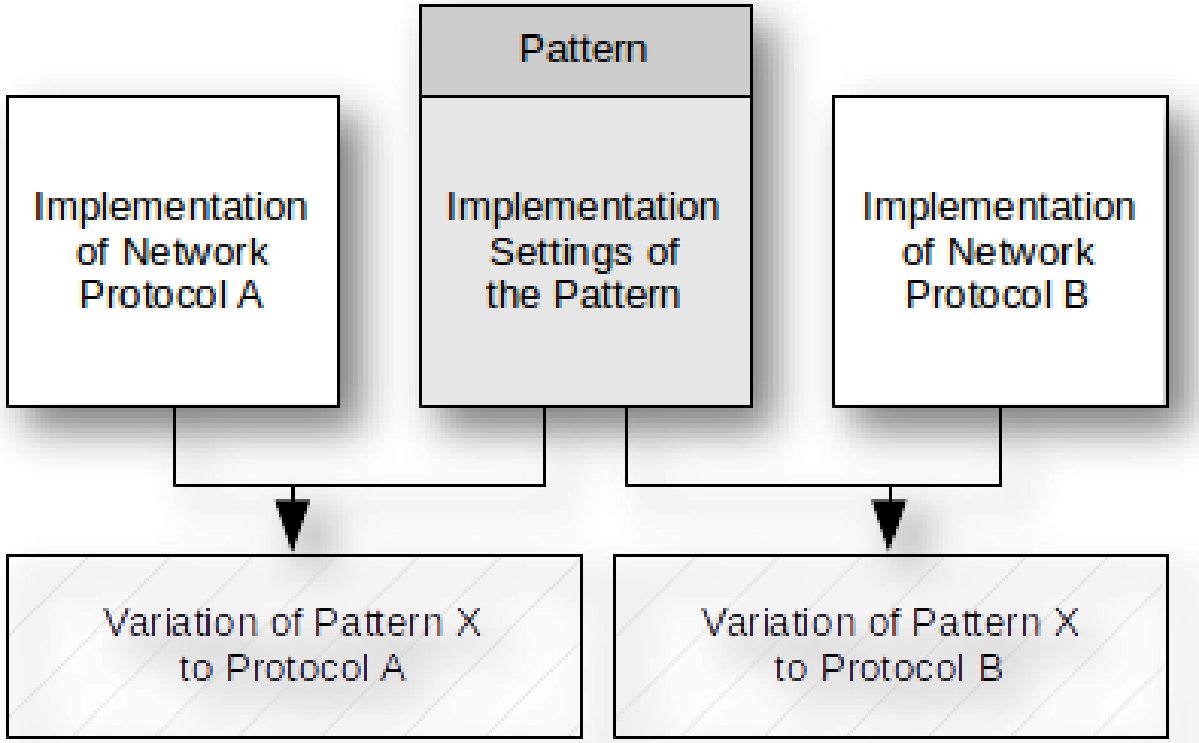
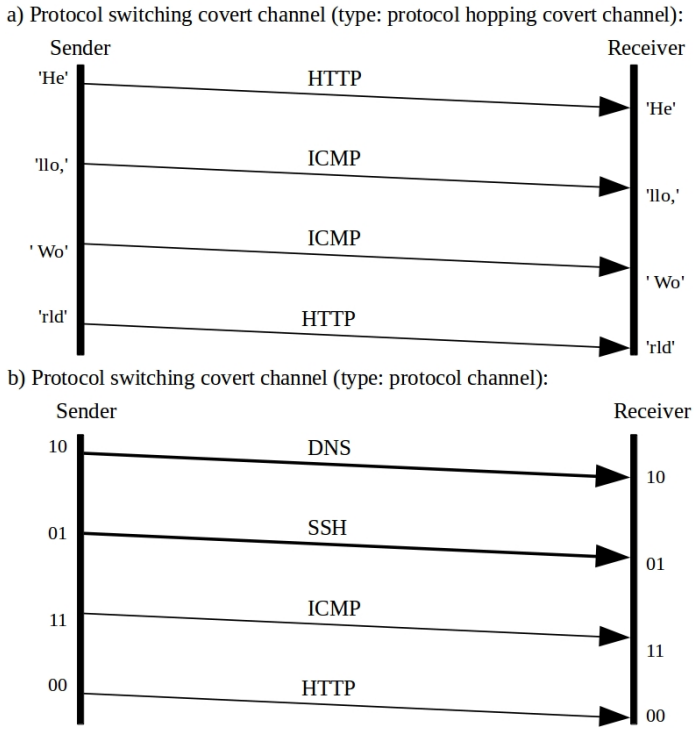


Fig.: [1]

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

Protocol Switching, Protocol Hopping, Pattern Hopping

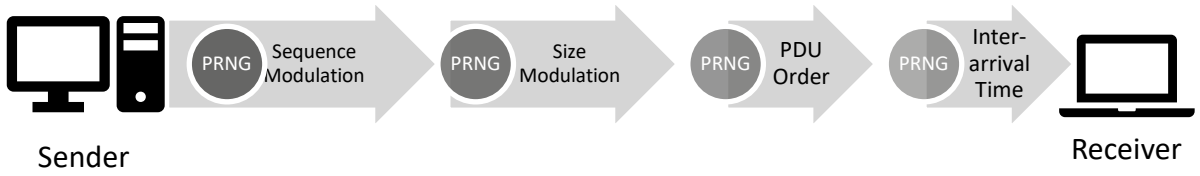
Protocol Hopping Covert Channel (PHCC) [2]:
Secret information is split over multiple network protocols to increase hurdles for a forensic traffic analysis.



My open source implementations: https://github.com/cdpxe/NetworkCovertChannels

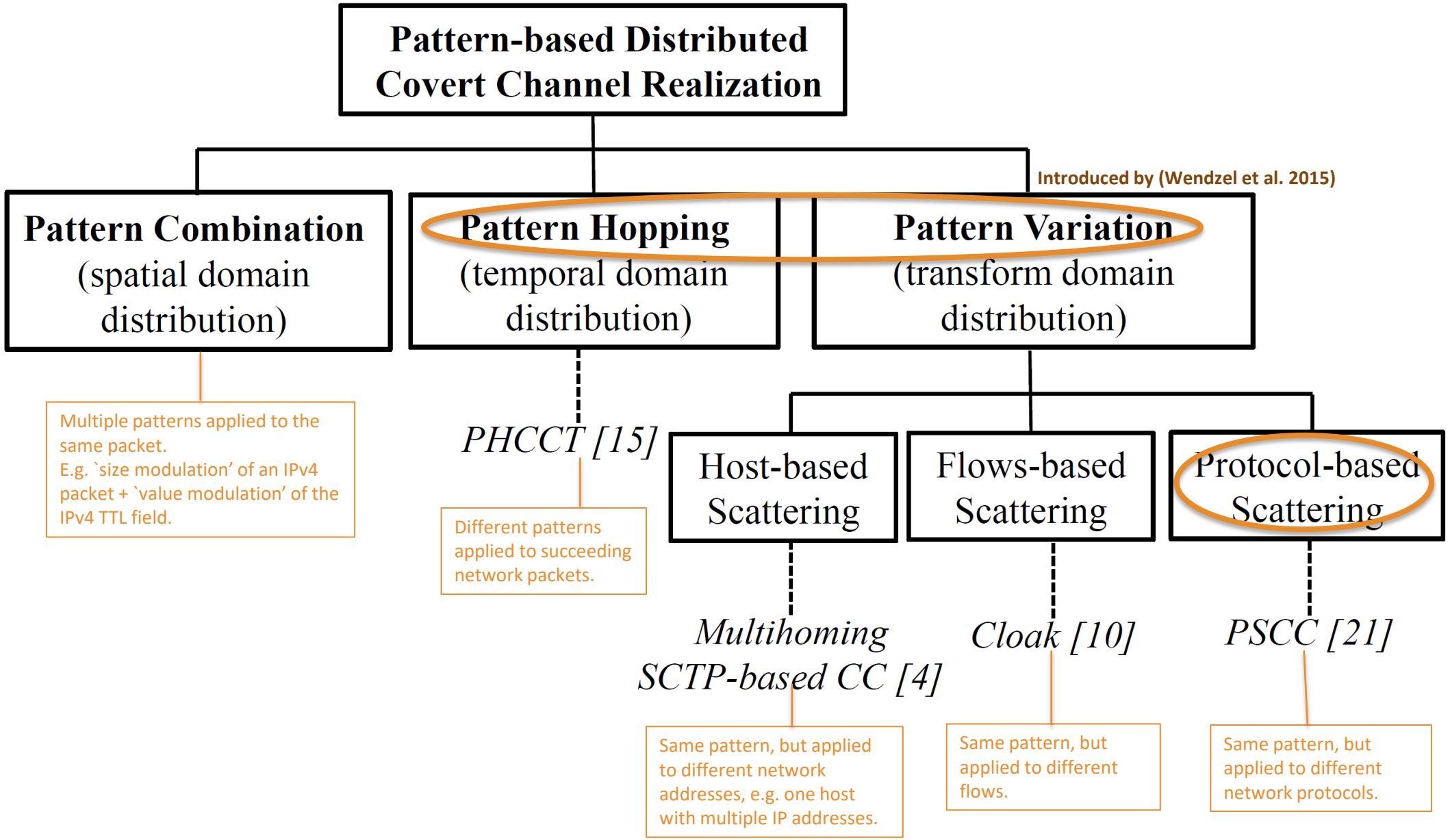
Protocol (Switching Covert) Channel (PSCC) [1]:
Secret information is represented by the protocol itself.

Pattern Hopping [3]:
For every new piece of secret information a PRNG selects one of the patterns (+variation) to transfer the data.



[1] S. Wendzel, S. Zander: Detecting protocol switching covert channels, Proc. *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*. IEEE, 2012.
[2] S. Wendzel, J. Keller: Low-attention forwarding for mobile network covert channels, Proc. Communications and Multimedia Security (CMS), 2011.
[3] S. Wendzel, S. Zander, B. Fechner, C. Herdin: Pattern-based Survey and Taxonomy for Network Covert Channels, ACM CSUR, Vol. 47(3), 2015.

Distributed Hiding Methods [1]



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

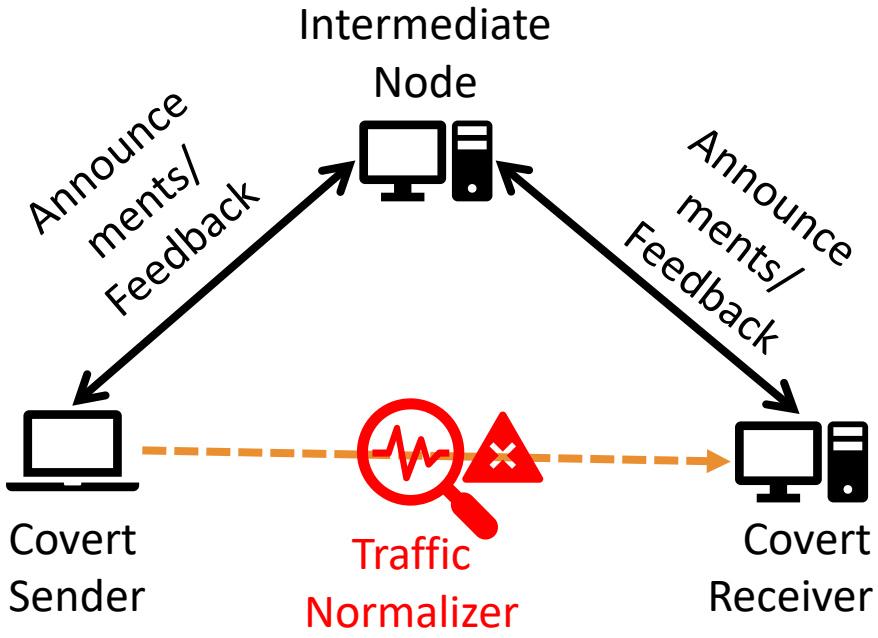
Network Environment Learning

- NEL allows covert channel nodes to determine how filters in their network environment are configured by probing several covert channel techniques.
  - is a constant process.
  - originally introduced by Yarochkin et al. [1]

My NEL tool is available (open source) here:
https://github.com/cdpxe/NELphase

- Circumvention-method improved a few years later by myself in [2] using temporary participant (Intermediate Node):

NEL-driven CC communication cannot be blocked unless *all* covert channels used between CS+CR are blocked.

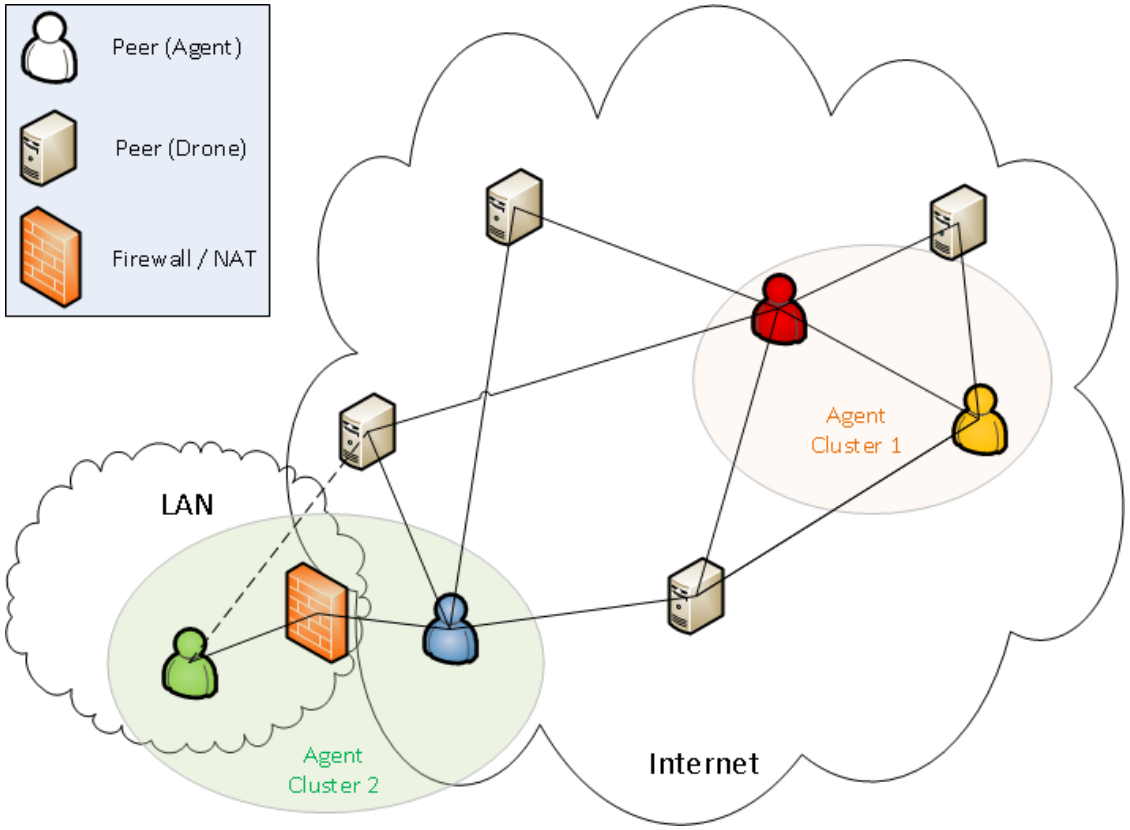


[1] Yarochkin, Fedor V., et al.: Towards adaptive covert communication system, in Proc. *14th IEEE Pacific Rim International Symposium on Dependable Computing*. IEEE, 2008.

[2] Wendzel, Steffen: The Problem of Traffic Normalization Within a Covert Channel's Network Environment Learning Phase, in Proc. *Sicherheit*, GI, 2012.

Dynamic Overlay Routing for Covert Channels

- Introduced already in 2008 by Szczypiorski et al.
- 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



[1] Szczypiorski, K., Margasiński, I., Mazurczyk, W., Cabaj et al.: TrustMAS: Trusted communication platform for multi-agent systems, in Proc. *OTM On the Move to Meaningful Internet Systems,* Springer, 2008.
[2] Backs, P., Wendzel, S., Keller, J.: Dynamic Routing in Covert Channel Overlays Based on Control Protocols, Proc. ISTP, IEEE, 2012.

Reliability & Control (Micro) Protocols

CC-internal control protocols (embedded into a covert channel). Early protocols arose around 2004 with PingTunnel. Later, deGraaf (2005) and Ray and Mishra introduced (2008) protocols [1].

Benefits (copyed from [1]):

- 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

[1] 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.
  - We will cover both later in this chapter.

- Countermeasures for control protocols exist as well, cf. J. Kaur et al.: Covert Channel-internal Control Protocols: Attacks and Defense, *Security and Communication Networks (SCN)*, Vol. 9(15), Wiley, 2016.

Reliability & Control (Micro) Protocols

Evaluation of Control Protocols [1]:

1. Header optimized for hiding (we will discuss this later!)
2. Features per header bit
3. Upgradability
4. Handling of non-covert input
5. … (see [1])

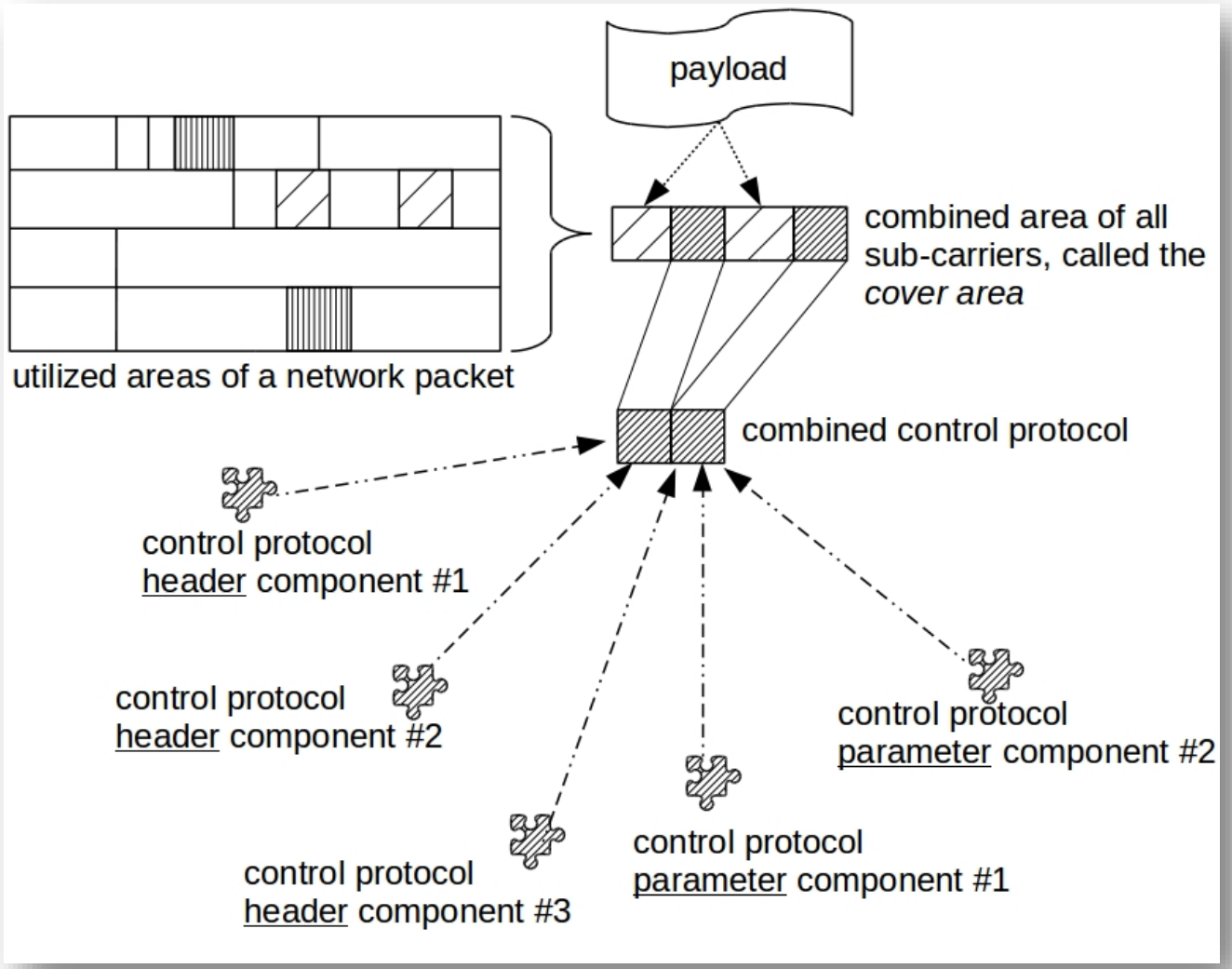
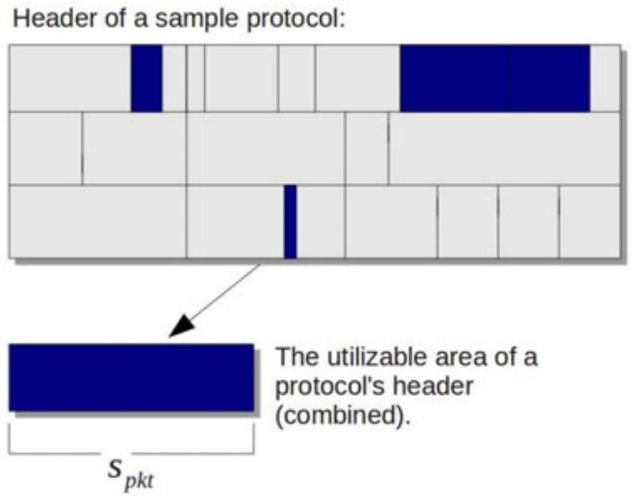


Fig.: W. Mazurczyk, S. Wendzel, S. Zander et al.: Information Hiding in Communication Networks, Ch. 4, Wiley-IEEE, 2016.

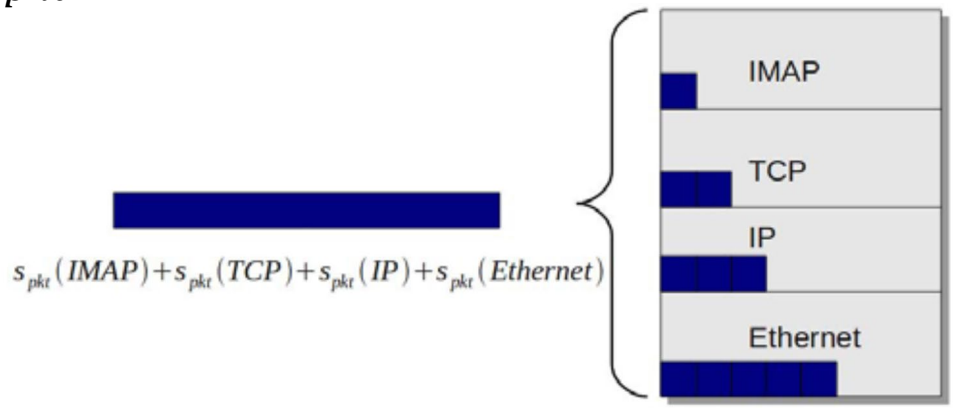
[1] S. Wendzel, J. Keller: Hidden and Under Control, Annals of Telecommunications (ANTE), Springer, 2014.

Optimizing PHCC (Wendzel & Keller, 2011)

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



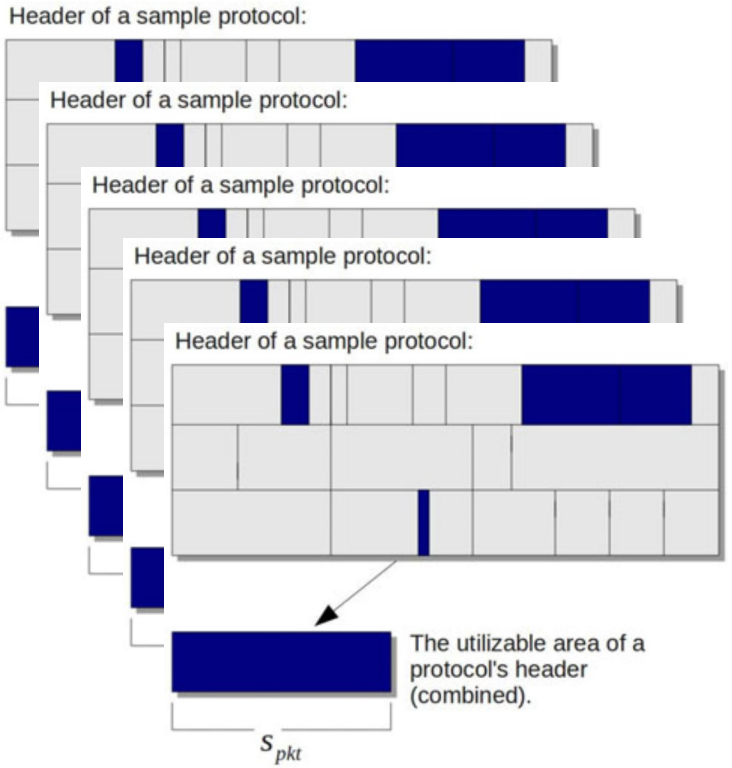
- For a multi-layered protocol, a CC could combine the s_{pkt} values of the protocols in the selected layers, e.g. $s_{pkt} = s_{IMAP} + s_{TCP} + s_{IP}$.



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

Optimizing PHCC (Wendzel & Keller, 2011)

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



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

Optimizing PHCC (Wendzel & Keller, 2011)

- Now, we can optimize a PHCC for different purposes (**QoS**), e.g.
  - A password cracking program needs to transfer a short password string out of a network (e.g. one password/hour).
    => keep a low profile (transfer only few packets: minimize overhead)
  - Urgently (but still covertly) leak videos of harmed protesters in a country with Internet censorship to the press.
    => still keep a low profile, BUT transfer data rather quickly (high throughput).

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

Optimizing PHCC (Wendzel & Keller, 2011)

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

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

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

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

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

Optimizing PHCC (Wendzel & Keller, 2011)

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

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

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

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

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

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

Optimizing PHCC (Wendzel & Keller, 2011)

- One could also optimize **stealthiness** if each protocol (or better: each covert channel technique) is assigned a covertness level $c_i \in \mathbb{N}$.
  - For instance, protocols that might occur regularly in a given network are less of an anomaly than protocols that never occurred in a given network before!

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

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

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

Hochschule Worms
University of Applied Sciences

Optimizing protocol utilization for PHCCs is already nice to have

but can we also optimize the micro protocol so that we raise even fewer attention?

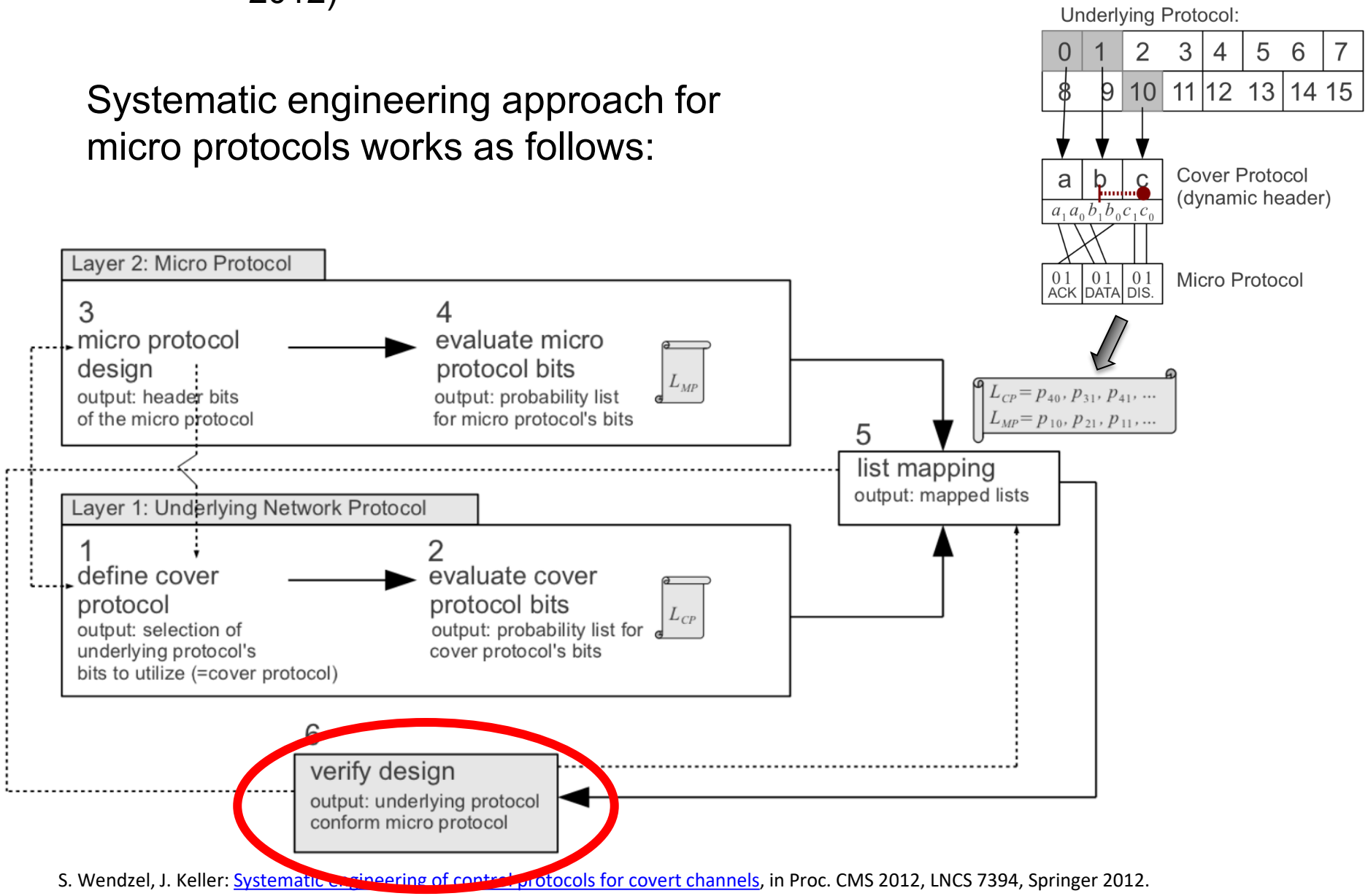
Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

- We skip the part on micro protocol size minimization using protocol engineering, cf. some of my papers on this topic if you are interested.
- **Goal:** Embed the micro protocol in a low-attention raising manner.
- **Answer:** We use a tailored protocol engineering approach (we will cover this at least in a nutshell).

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

Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

Systematic engineering approach for micro protocols works as follows:



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

Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

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

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

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

Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

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

Example:

$\boldsymbol{G_{CP} = (V, \Sigma, P, S)}$, with
$\boldsymbol{V = \{S, A, B, C\}}$,
$\boldsymbol{\Sigma = \{\mathbf{a_0}, \mathbf{a_1}, \mathbf{b_0}, \mathbf{b_1}, \mathbf{c_0}, \mathbf{c_1}\}}$,
$\mathbf{P} = \{\mathbf{S} \rightarrow \mathbf{AB}|\mathbf{AC}$,
$\mathbf{A} \rightarrow \mathbf{a_1}|\mathbf{a_0}$,
$\mathbf{B} \rightarrow \mathbf{b_1}|\mathbf{b_0}$,
$\mathbf{C} \rightarrow \mathbf{b_1c_1}|\mathbf{Bc_0}\}$

$\boldsymbol{G_{MP} = (V, \Sigma, P, S)}$, with
$\boldsymbol{V = \{S, B, C_A, C_B\}}$,
$\boldsymbol{\Sigma = \{\mathbf{a_0}, \mathbf{a_1}, \mathbf{b_0}, \mathbf{b_1}, \mathbf{c_0}, \mathbf{c_1}\}}$,
$\mathbf{P} = \{\mathbf{S} \rightarrow \mathbf{a_0B}|\mathbf{a_1B}$,
$\mathbf{B} \rightarrow \boldsymbol{b_0C_A}|\boldsymbol{b_1C_B}$,
$\mathbf{C_A} \rightarrow \boldsymbol{c_0}$,
$\mathbf{C_B} \rightarrow \mathbf{c_0}|\mathbf{c_1}\}$

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

Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

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

Illustration:

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

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

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

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

Optimizing Micro Protocol Embedding (Wendzel & Keller, 2012)

But what if …?

- **… the language inclusion test fails**? -> modify CP selection or MP design.
- **… we need to model connection-oriented protocols?** See our paper.

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

Dead Drops in the Network [1]

- Did I say, **network** steganography can only be used to **transfer** secret data?
  - Well, that was wrong! You can also use it to **store** secret data!
  - At least for some time ...

- Introduced in [1, text below is a copy from the paper]. Works, e.g. with ARP+SNMP (see Fig.):
  a) CS possesses secret information that it wants to store in the network-accessible ARP cache.
  b) CS exploits the ARP cache of a third-party system by sending a fake ARP request containing a (MAC,IP) tuple. The actual host reflected by the (MAC,IP) tuple does not exist and represents encoded secret information.
  c) The third-party host Dead Drop (D) adds the tuple to its local ARP cache …
  d) … where it can be requested by the CR for some time depending on the lifetime of ARP cache entries on the third-party host.

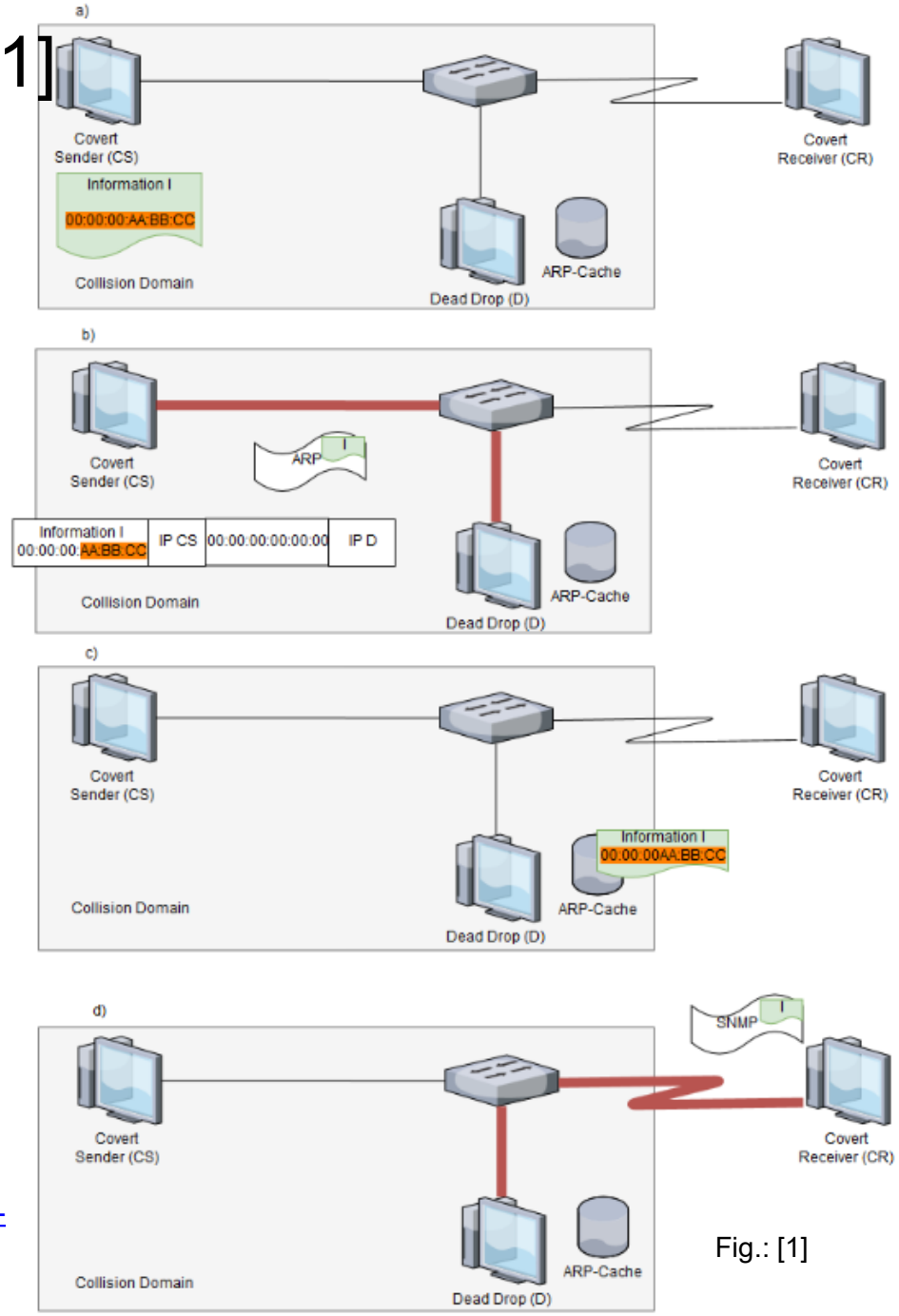


Fig.: [1]

[1] T. Schmidbauer, S. Wendzel, A. Mileva, W. Mazurczyk: Introducing Dead Drops to Network Steganography using ARP-Caches and SNMP-Walks, in Proc. ARES 2019, ACM.

Dead Drops in the Network [1]

- Lifetime of entries depends on the system and caching limits (all Figures taken from [1]):

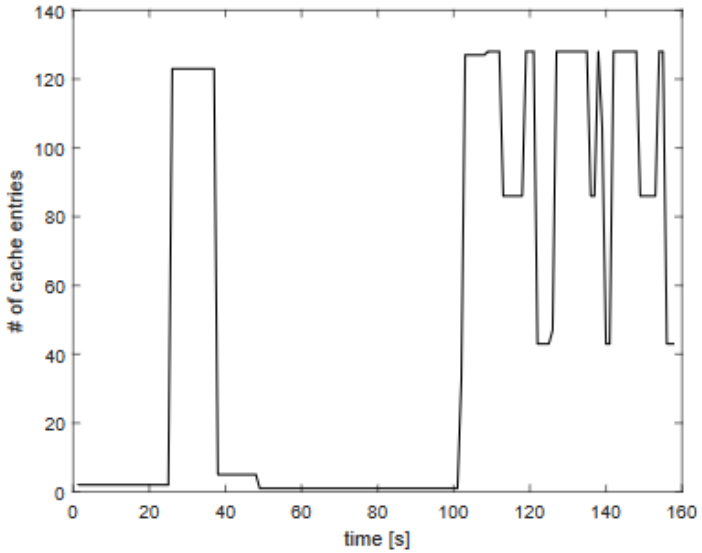


Figure 5: OpenSuSE caching behavior

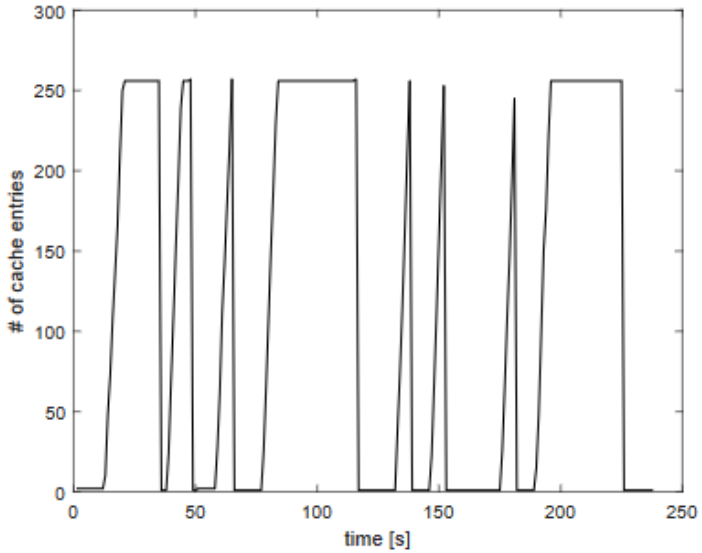


Figure 7: Windows 7 caching behavior

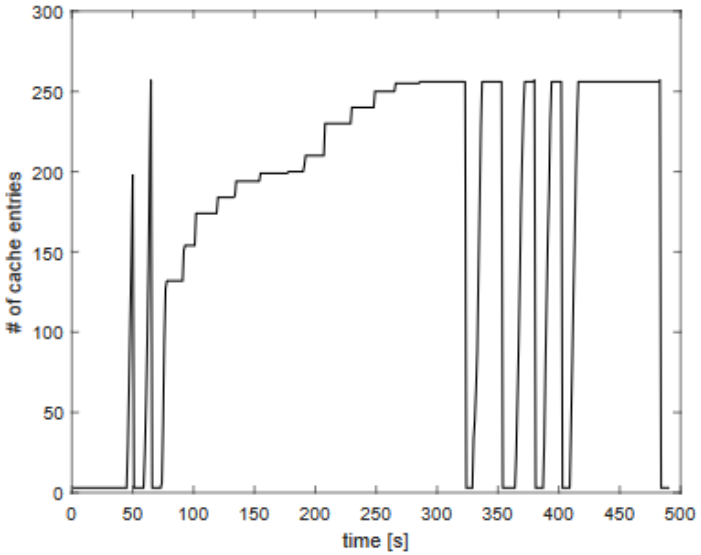
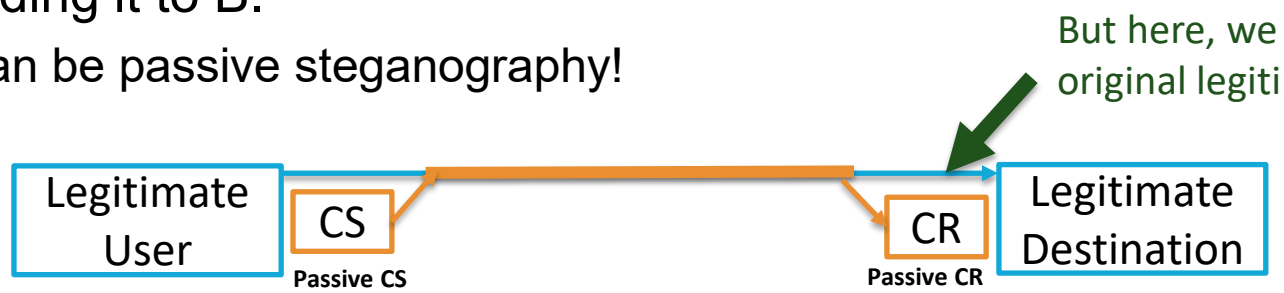


Figure 6: Windows 10 caching behavior

[1] T. Schmidbauer, S. Wendzel, A. Mileva, W. Mazurczyk: Introducing Dead Drops to Network Steganography using ARP-Caches and SNMP-Walks, in Proc. ARES 2019, ACM.

Reversible Data Hiding (RDI) in Networks [1]

- **Core Idea:** Modify legitimate traffic from A to B for steganographic purpose as a MitM. CS embeds the secret data while CR receives the secret data and restores its original form, before forwarding it to B.
  - Yes, this can be passive steganography!



Reversibility Levels (copied from [1]):

- *fully-reversible*: the secret receiver can completely revert the altered fields to the original state.
- *quasi-reversible*: it is not possible to precisely determine the original "state" of the carrier, hence the covertly communicating endpoints can only restore it in a statistical manner.
- *non-reversible*: the covert receiver has no access to the proper knowledge required to restore the protocol data units or network statistics to the original state.

[1] W. Mazurczyk, P. Szary, S. Wendzel and L. Caviglione: Towards Reversible Storage Network Covert Channels, in Proc. Third International Workshop on Criminal Use of Information Hiding (CUING 2019), pp. 69:1-69:8, ACM, 2019.

Reversible Data Hiding (RDI) in Networks [1]

- **RDI Methods in the Network (also copied from [1]):**
  - *intrinsically*: the data hiding technique is constructed in such a way that the covert receiver can completely restore the overt traffic to its original form.
  - *explicitly*: the covert sender also transmits the "state" of the overt traffic that is utilized as a hidden data carrier to allow the covert receiver to restore the carrier to its original state. This may require to sacrifice a part of the available steganographic bandwidth, thus, slowing down the covert communication exchange.
  - *implicitly*: the covert receiver restores the overt traffic by removing the secret message and by "guessing" the potential original form of the overt traffic before the steganographic modifications were applied, for instance, via estimations or past observations.

[1] W. Mazurczyk, P. Szary, S. Wendzel and L. Caviglione: Towards Reversible Storage Network Covert Channels, in Proc. Third International Workshop on Criminal Use of Information Hiding (CUING 2019), pp. 69:1-69:8, ACM, 2019.

Summary of this Chapter

- **Distributed hiding methods**
  - Make limitation/prevention of the covert channels harder.
  - Circumvent filters (e.g. traffic normalizers) using NEL and enhanced NEL

- **Covert channel-internal control protocols (a.k.a. micro protocols)**
  - Necessary for reliable and enhanced/sophisticated features, such as dynamic overlay routing.
  - Can be optimized in several ways.

- **Other mentionable sophisticated techniques:**
  - Dead drops for (short-/mid-term data storage in network caches)
  - Reversible data hiding (re-constructs original form of cover data)