Journal of Universal Computer Science, vol. 25, no. 11 (2019), 1396-1416
submitted: 4/3/19, accepted: 26/7/19, appeared: 28/11/19

Detection of Size Modulation Covert Channels Using Countermeasure Variation[1]

Steffen Wendzel
(Fraunhofer FKIE & Worms University of Applied Sciences, Germany
wendzel@hs-worms.de)

Florian Link
(Worms University of Applied Sciences, Germany
inf2283@hs-worms.de)

Daniela Eller
(Worms University of Applied Sciences, Germany
inf2939@hs-worms.de)

Wojciech Mazurczyk
(Warsaw University of Technology, Poland
wmazurcz@ii.pw.edu.pl)

Abstract: Network covert channels enable stealthy communications for malware and data exfiltration. For this reason, developing effective countermeasures for these threats is important for the protection of individuals and organizations. However, due to the large number of available covert channel techniques, it is considered impractical to develop countermeasures for *all* existing covert channels.

In recent years, researchers started to develop countermeasures that (instead of only countering one particular hiding technique) can be applied to a whole family of similar hiding techniques. These families are referred to as *hiding patterns*.

Considering above, the main contribution of this paper is to introduce the concept of *countermeasure variation*. Countermeasure variation is a slight modification of a given countermeasure that was designed to detect covert channels of one specific hiding pattern so that the countermeasure can also detect covert channels that are representing *other* hiding patterns.

We exemplify countermeasure variation using the compressibility score, the ϵ-similarity and the regularity metric originally presented by Cabuk et al. All three methods are used to detect covert channels that utilize the Inter-packet Times pattern and we show that countermeasure variation allows the application of these countermeasures to detect covert channels of the Size Modulation pattern, too.

Key Words: covert channels; network steganography; information hiding; patterns; network security

Category: B.4.1, C.2.2, C.2.5, C.2.6, D.4.6, K.6.5, K.7.m

[1] This article is an extension of the paper [Wendzel et al., 2018]. In comparison to the previous paper, we introduce an improved definition of our core concept (*countermeasure variation*), perform countermeasure variations for two additional metrics (ϵ-similarity and regularity), and compare the results of all three metrics.

1 Introduction

In today's network environments, covert channels represent (usually stealthy) policy-breaking communication channels [Proctor and Neumann, 1992; Millen, 1999; Wendzel et al., 2014; Handel and Sandford, 1996]. They enable several malicious use-cases, e.g., the secret transfer of malware commands or the stealthy exfiltration of confidential data [Mazurczyk et al., 2016; Mazurczyk and Caviglione, 2015].

A few hundred hiding techniques for covert channels are known which can be assigned to different families, called *hiding patterns*. Hiding patterns were introduced in 2015 and are abstract descriptions of hiding methods [Wendzel et al., 2015].[2] For instance, the least significant bit (LSB) pattern specifies that secret data can be hidden in the LSB(s) of a header field, but it does not specify where such a field has to be located in a header, which size or byte order the field can have, or to which protocol the hiding technique can be applied to.

So far, countermeasures for covert channels focus only on a single hiding technique or on a family of similar methods which are assigned to the same hiding pattern. To keep the application of countermeasures feasible in practice, their number should be kept at a minimum. Therefore, it must be studied which countermeasures can be applied to which hiding patterns. However, no work is available that has shown that countermeasures can be applied in a way that works with several *patterns*.

In this paper, we introduce the idea of *countermeasure variation*, i.e., to counter specific covert channels these countermeasures can be potentially applied to multiple patterns instead of only one. As a positive side-effect, countermeasure variation reduces the amount of necessary code per countermeasure as parts of a countermeasure's code can be recycled to counter other patterns. We exemplify the feasibility of countermeasure variation by showing that the compressibility score, the ϵ-similarity, and the regularity metric used to detect covert timing channels of the Inter-packet Times pattern can also be applied to detect covert channels that modulate packet sizes (Size Modulation pattern).

The remainder of this paper is structured as follows. Sect. 2 highlights fundamentals and the linked related work while Sect. 3 introduces countermeasure variation. Sect. 4 first presents the original three countermeasures by Cabuk et al. for detecting Inter-packet Times-based covert channels, followed by our variations of their countermeasures to detect Size Modulation-based covert channels. We evaluate our three countermeasure variations in Sect. 5. A conclusion and an outlook are given in Sect. 6.

[2] For a general introduction into patterns within the security context see [Schumacher et al., 2013]. For the latest taxonomy of covert channel hiding patterns see [Mazurczyk et al., 2018].

2 Fundamentals & Related Work

Several existing works studied how covert channels based on packet length can be realized, e.g., [Ji et al., 2009; Elsadig and Fadlalla, 2017; Mazurczyk and Szczypiorski, 2012; Ling et al., 2013; Girling, 1987; Wolf, 1989; Murdoch and Lewis, 2005]. Such a covert channel is a form of the *Size Modulation* pattern. The basic idea of Size Modulation is that a covert sender selects at least two different packet sizes to encode different secret symbols. For instance, if a packet has a size of 100 bytes it could indicate a binary zero while a packet with a size of 101 bytes could indicate a binary one.[3] Countermeasures for covert channels based on packet length are already available. For instance, Elsadig and Fadlalla developed a traffic normalizer that adds padding bytes to every $n-$th packet of a flow [Elsadig and Fadlalla, 2017]. This is done in a blind manner, i.e., without knowing whether a covert channel is present, or not. Their approach can be categorized as a limiting one (instead of a detecting one). Moreover, Ling et al. propose to simply pad *all* packets so that packet sizes cannot be modified by a covert channel [Ling et al., 2013]. However, these approaches would negatively influence the network performance and a targeted application would require the capability to detect such covert channels before eliminating them.

Wendzel et al. introduced the concept of *pattern variation* in [Wendzel et al., 2015]. The idea of *pattern variation* is that one pattern can change its context, i.e., the network protocol to which it is applied. For instance, the LSB pattern, which hides data in the least significant bit(s) of a protocol header field, can be applied to the TTL field of IPv4 as well as to the Hop Limit field of IPv6. Therefore, the same algorithm can be applied, but the context (network protocol) is changed. Pattern variation is based on the idea of *pattern transformation*. Pattern transformation is used for the dynamic generation of user interfaces so that they fit a given context, e.g., a desktop browser or a mobile browser.

Instead of the patterns, countermeasures can also be 'transformed'; we call this process countermeasure variation (see next sect.). Countermeasure variation modifies a countermeasure to work with another pattern as originally intended. The conference paper that serves as the basis for this article introduced countermeasure variation briefly. In other recent work, we have already shown that countermeasure variation is feasible for covert channels of the so-called *Artificial Re-transmission* pattern [Zillien and Wendzel, 2018]. One additional work evaluated a countermeasure variation for the *(Manipulated) Message Ordering* pattern with good results [Wendzel, under review]. As shown in Fig. 1, countermeasure variation was only studied for three patterns ((Manipulated) Message Ordering, (Artificial) Re-transmission and Size Modulation) and three countermeasures (*compressibility*, *ϵ-similarity* and *regularity*) so far.

[3] See [Wendzel et al., 2015] for a detailed description of the Size Modulation pattern.

	Inter-arrival Times	(Manipulated) Message Ordering	Artificial Re-transmissions	Size Modulation	...
compressibility	[Cabuk et al., 2009] (*)	[Wendzel, under review]	[Zillien and Wendzel, 2018]	[Wendzel, Eller and Mazurczyk, 2018]	
ε-similarity	[Cabuk et al., 2009] (*)	-	[Zillien and Wendzel, 2018]	**this paper**	
regularity	[Cabuk et al., 2009] (*)	-	-	**this paper**	

Figure 1: Summary of existing work on countermeasure variation. (*) indicates the original approaches, i.e., without countermeasure variation.

Please note that although Fig. 1 mentions only three countermeasures, additional countermeasures could be considered, e.g., all countermeasures of [Mazurczyk et al., 2016, Ch. 8]. Also, countermeasures were always transferred from one covert channel technique to another but *i)* not with the focus on hiding patterns and *ii)* in a less systematic manner.

3 Countermeasure Variation

The idea for *countermeasure variation* was already briefly introduced in [Wendzel et al., 2015] but was never experimentally evaluated or detailed.

When a new type of network hiding pattern is found, no countermeasure is instantly available for the new covert channels of the particular pattern. Countermeasure variation allows to transform existing countermeasures so that they can be applied to such a new pattern. Similarly, countermeasure variation can be applied to already known covert channel patterns for which no or only few countermeasures are known. However, there is currently no clear definition of countermeasure variation. For this reason, we provide the following definition:

Definition. Given the two hiding patterns A and B, with $A \neq B$, a *countermeasure variation* is a pattern-based process in which an existing countermeasure that detects, limits, prevents or audits covert channels of pattern A is modified so that it detects, limits, prevents or audits covert channels of pattern B.

The process of countermeasure variation replaces the input attributes (*features*) used for A with features for B and performs a modification of the inner functioning (e.g., the algorithm) used for A in order to work with the new features for B. The alternation of the inner functioning is kept as small as possible, which provides the contrast to developing entirely new countermeasures. In comparison to simply applying the same countermeasure (e.g. a statistical method) to another covert channel technique, countermeasure variation *i) requires* the modification of the inner functioning and *ii)* focuses on hiding patterns, i.e., it needs to consider features that can be used for multiple covert channels belonging to the same pattern. ■

In other words, to perform a variation for a given countermeasure, both, the input and the inner functioning of an existing countermeasure must be adapted to a new hiding pattern's requirements. For instance, instead of packets' inter-arrival time (IAT) values used to detect the *Inter-packet Times* pattern, packet sizes could be used as a feature to detect covert channels of the Size Modulation pattern. Or, as shown in [Zillien and Wendzel, 2018], observations of TCP re-transmissions can be extracted from flows to detect the Artificial Re-transmissions pattern. Indeed, multiple features could be combined.

The inner functioning of a countermeasure must be (slightly) modified since the existing functioning (in almost all cases) will not provide satisfying results with the new inputs. Another reason to modify the inner functioning is given when the new input type is incompatible with the existing function (e.g., because a countermeasure is designed to deal with small floating point values < 1 but now has to deal with 32 bit integers as it was the case in [Zillien and Wendzel, 2018] or because sufficient detection results require a modified string generator [Wendzel, under review]). There is no generalization feasible of how such a countermeasure variation can be performed as countermeasures are highly heterogeneous. However, as we will show in the remainder of this section, performing countermeasure variation is not necessarily a complicated task, which renders the idea a useful and quick method for creating new countermeasures.

As a positive side-effect, recycling the code of one countermeasure to work with a different pattern allows to reduce the overall lines of code: only on a detailed level, the algorithm is slightly altered to fit into the context of the new pattern (i.e., it is *transformed* to the new pattern). However, we do not state that countermeasure variation is *necessarily* less time-consuming than developing new countermeasures from scratch. Instead, its major benefit is to take advantage of existing countermeasures, i.e., it transfers existing countermeasure concepts into new countermeasures.

In this paper, we show the feasibility of countermeasure variation with three countermeasures originally designed for the detection of the Inter-packet Times pattern. After the process of countermeasure variation, the three countermeasures can be applied to the Size Modulation pattern.

While one could argue that the Size Modulation and Inter-packet Times patterns are rather similar in their functioning (both basically modulate integer values) this was not the case for the Artificial Re-transmissions pattern. Thus, we conclude that countermeasure variation can be expected feasible for other patterns than the already evaluated ones. While the detection results for new patterns after countermeasure variation were acceptable in most cases, there were also cases where no acceptable detection results could be achieved.[4] However, no generic conclusion on the quality of detection results after performing a countermeasure variation is feasible due to the diversity of existing hiding patterns.

4 Detecting Size Modulation with Countermeasure Variation

In this section, we first explain the original detection methods as introduced by Cabuk et al., followed by our approaches for countermeasure variations.

4.1 Inter-packet Times Pattern and Its Detection

Cabuk et al. developed a detection approach for covert channels that transfer secret data via delay between network packets (IAT values) in [Cabuk et al., 2009, 2004]. These covert channels fall under the Inter-packet Times pattern. The basic functionality of such covert channels is that before sending new packets they encode secret data using different IATs[5]. For instance, if the time between two packets is 100 ms, this could indicate a binary zero while a time-gap of 200 ms could indicate a binary one. However, detecting such channels is challenging since their coding can vary and because they can easily blend with the legitimate traffic. The three proposed detection metrics for IAT-based channels of Cabuk et al. are the compressibility, the ϵ-similarity, and the regularity.

The *compressibility*-based approach by Cabuk et al. works as follows. For each traffic flow, all n IATs are recorded in a list $\Delta_{t_1}, ..., \Delta_{t_n}$ (we use t to indicate that we focus on timing events). All values > 1 s are filtered out. All remaining values are coded in ASCII characters in the form that the number of leading zeros behind the comma is encoded in upper-case characters starting from A (no zeros) over B (one zero behind the comma) and so forth. All resulting strings are then concatenated to a large string S (e.g. "`A25B2A25B19A24B22`"). Then, S is compressed with a compressor $\Im$, resulting in the compressed string $C = \Im(S)$. As a compressor, Cabuk et al. used *Gzip*. The compressor is a key component and it is integrated to reveal the decrease of the entropy due to the covert channel utilization as its few IATs occur many times. Finally, the authors divide the length of both strings by calculating the value $\kappa = |S|/|C|$. In result, certain ranges of κ values are an indicator for the presence of a covert channel.

In case of the ϵ*-similarity*, all IATs of a flow are first sorted in a list with ascending order. For every packet P_i in the sorted list, the pair-wise timing

[4] The compressibility score did not provide sufficient results for the Artificial Retransmissions pattern while the ϵ-similarity metric did provide acceptable results for the same pattern.

[5] A detailed description of the pattern can be found in [Wendzel et al., 2015; Mazurczyk et al., 2016].

difference λ_i with packet P_{i+1} is calculated, i.e., $\lambda_i = |P_i - P_{i+1}|/P_i$ (if the timestamps are equal, the value 0 is used). Next, a value ϵ is selected as a threshold and all relative increases $\lambda_i < \epsilon$ are counted. The number of λ values below the threshold in comparison to all values is then used as an indicator for the presence of a covert channel.

Finally, the *regularity* can be calculated from a given list of IAT values. This list of values is first divided into sections, called windows, containing 2,000 packets each. Then the standard deviation σ_i is calculated for each window i. Next, the difference values between these standard deviations are determined. The final regularity value is calculated from the standard deviation of these difference values [Cabuk et al., 2004], i.e., regularity $= \mathrm{STDEV}(|\sigma_i - \sigma_j|/\sigma_i, i < j, \forall i, j)$.

4.2 Countermeasure Variation

To perform a countermeasure variation for the *compressibility* metric proposed by Cabuk et al., i.e., transferring the original approach to covert channels which utilize the Size Modulation pattern, we modified the following aspects of the original algorithm. First, we considered the relative differences of packet sizes of a flow instead of its IATs. Thus, for each flow with n packets, we calculated $n-1$ relative size differences Δ_{p_i} (p stands for *packet size*) between the succeeding packets. Second, we concatenated a string S consisting of the relative differences for each flow, separated by commas: $S = \Delta_{p_1}, \Delta_{p_2}, ..., \Delta_{p_n}$. In this string, numbers were represented in ASCII (i.e., the string coding is different to the letter-coded rounded IATs of Cabuk et al.). For instance, if a flow contains five packets with the packet sizes 120, 520, 514, 518, and 520 bytes, then the relative differences $\Delta_{p_1}, ..., \Delta_{p_4}$ would be 400, -6, 4, and 2 bytes. We concatenated the string S using the ASCII representation of the Δ_p values, i.e., "`400,-6,4,2`". We decided to introduce the comma-based separation of values as otherwise, due to the ASCII representation, numbers would not be distinguishable, e.g., the relative differences used above would result in the string "`400-642`", which would influence the compression in an way that does not consider the actual size differences. The remainder of this detection method functions exactly the same as in the case of the original approach. For each flow, we calculated the compressibility of S using a compressor $\Im$ to calculate $C = \Im(S)$, followed by dividing string lengths $\kappa = |S|/|C|$. As already mentioned, the compressor is a key component and it is integrated to reveal the decrease of the entropy due to the covert channel utilization as few utilized covert channel's packet sizes occur many times. Finally, we determined κ values of legitimate traffic and of covert channel traffic to define interval borders in which flows could be considered as covert channel traffic. A similar step is required to categorize κ values in case of the original approach [Cabuk et al., 2009, 2004].

Next, we performed a countermeasure variation for the *ϵ-similarity* as follows. First, we sorted all packet sizes of a flow instead of the IAT values. Then, we calculated the relative differences λ_i based on these values and determined suitable ϵ-thresholds. All other steps of this countermeasure were kept as in the original approach.

Finally, the countermeasure variation for the *regularity* was performed by considering packet sizes instead of IAT values. However, as we will show later, we determined optimal window sizes and also determined how the regularity value differs depending on the number of packets within the flow and the window size. All other steps of the countermeasure were kept as in the original approach.

5 Evaluation

To evaluate how the transformed detection approaches perform with the Size Modulation pattern, we used different data samples as shown in Tab. 2.

The two metrics used to evaluate our detection methods are precision and accuracy. Precision is defined as the number of true positives (TP) divided by the number of all positives (true and false positives) and it is expressed as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

In other words, precision illustrates the percentage of the flows detected as covert channels that were actually covert channels (while other flows may have been detected as "covert channels" but were actually legitimate traffic).

Accuracy, on the other hand, expresses how large the number of correctly classified elements is in comparison to *all* elements. In other words, it represents the percentage of flows that were correctly classified as covert or legitimate in comparison to all classified flows (the total population of true and false positives and negatives). The accuracy is calculated as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

In the remainder of this section, it must be noted that for each detection technique, we first analyze the detectability of covert channels that encode data using two different symbols, i.e., two different packet sizes. Afterwards, we analyze the detectability of traffic that combines all these covert channels. Finally, we analyze the detectability of covert channels with more than two symbols.

5.1 Compressibility

To evaluate compressibility, first we performed a *training* phase to determine κ values of legitimate and covert traffic (100,000 legitimate packets and 100,000 covert channel packets). As a data source for legitimate traffic, we used the NZIX data set [WAND group, 2000] from the University of Waikato's WAND group. In particular, we considered traffic containing at least 200 packets. The κ values of legitimate and covert channel traffic overlap clearly. When the covert channel utilizes more symbols, the median κ value seems to decrease, rendering these channels potentially easier to detect.

The obtained κ values were then used to define interval values that separate covert from legitimate traffic. Covert channel traffic has a κ value of approximately 4 to 6, with an approximate mean of 5.0; the differences are depending on the covert channel's number of symbols and our generation of the string S. Therefore, we decided to use the intervals shown in Tab. 1.

No.	**Range**	**No.**	**Range**	**No.**	**Range**	**No.**	**Range**	**No.**	**Range**
0	$\langle 0.0; 0.1\rangle$	1	$\langle 3.7; 4.2\rangle$	2	$\langle 4.1; 4.6\rangle$	3	$\langle 4.2; 4.7\rangle$	4	$\langle 4.3; 4.8\rangle$
5	$\langle 4.6; 5.1\rangle$	6	$\langle 4.7; 5.8\rangle$	7	$\langle 4.9; 6.0\rangle$	8	$\langle 5.3; 7.0\rangle$	9	$\langle 0.0; 99\rangle$

Table 1: Detection intervals

In the following *testing* phase, we applied another 100,000 legitimate packets and 100,000 covert channel packets to test each interval for every particular type of covert channel (see following sub-sections).[6] Since there are no traffic recordings for the Size Modulation-based covert channels available [Elsadig and Fadlalla, 2017], we decided to generate our own covert channel traffic data with a traffic generator. Our covert channels used different packet sizes for their coding to transfer randomized content, i.e., every hidden symbol (packet size) occurred with the same probability. This is a realistic assumption as secret data can be encrypted before being transmitted. Some of the covert channels used a coding with significantly different packet sizes, e.g., sending either a packet of size 100 bytes or of size 1,000 bytes. Other covert channel's coding was only marginally distinguishable, e.g., sending either a packet of size 1,000 or 1,001 bytes. Table 2 provides an overview of the generated covert channels, all following a uniform distribution of covert symbols.

It must be noted that we apply the same detection intervals for the detection of all covert channels, i.e., we do not further optimize the intervals to match a specific channel's characteristics to ease detection. This was decided to reflect realistic conditions.

[6] In case of the combined test of all covert channels using two symbols, we transferred 20,000 packets per covert channel, so that 100,000 packets were processed overall.

Table 2: Size modulation traffic used to evaluate our approach

Data-type	**Payload sizes (transp. layer) [bytes]**	$\lvert\Delta_p\rvert$ **[bytes]**	**# of flows**
legitimate	various	various	100,000
covert channel (2 symbols)	1,000 / 1,001	1	100,000
	100 / 101	1	100,000
	50 / 60	10	100,000
	100 / 200	100	100,000
	100 / 1,000	900	100,000
	all the above	all the above	100,000
covert channel (>2 symb.)	100 / 200 / 300	100, 200	100,000
	100 / 200 / 300 / 400	100, 200, 300	100,000
	100 / 200 / ... / 800	100, 200, ..., 700	100,000

5.1.1 Two-symbol Covert Channels

We first analyzed covert channels with a payload size difference of only 1 byte, i.e., the covert channel that encodes data with 1,000 and 1,001 payload bytes and the one that encodes secrets with 100 and 101 payload bytes. As the differences in packet size were always -1, 0 or +1 bytes for both covert channels, they resulted in highly similar κ values that only depended on the randomly selected covert channel symbols. The mean κ value for the channel with 1,000 and 1,001 bytes was 4.43448 while it was 4.43677 for the channel with 100 and 101 bytes. Thus, the precision and accuracy values for both channels were almost equal for our interval sizes. As shown in Figure 2, the best performing interval was #3, resulting in an accuracy of 97.17% and an F-Score of 95.82%.

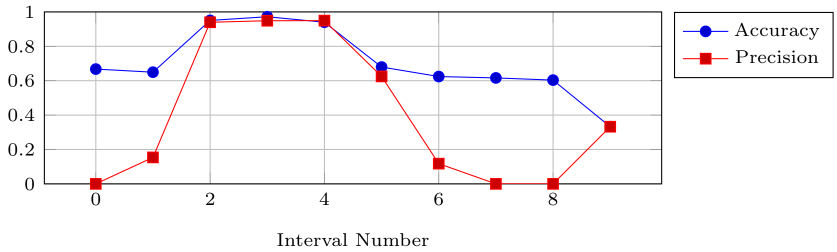

Figure 2: Precision and accuracy for covert channels using the payload sizes 1,000 and 1,001 bytes.

Next, we analyzed the covert channel with a payload size difference of 10 bytes, i.e., payload sizes of 50 and 60 bytes (Figure 3). Here, we could achieve the best accuracy and F-Score values for the interval #7, namely 94.86% and 92.82%. The mean κ value (5.34675).

The covert channel with the payload sizes 100 and 200 bytes provided an accuracy of 93.59% and an F-Score of 91.21% for the best-performing interval (#8), see Figure 4. The mean value for κ was 6.16748. The the mean value for the channel with 100 and 1,000 bytes was almost the same (6.03886), resulting in basically the same detection values for interval #8.

Table 3 summarizes the average κ values that were calculated for the different two-symbol covert channels. As can be seen, the κ value generally increases when the difference of the utilized packet sizes increases.

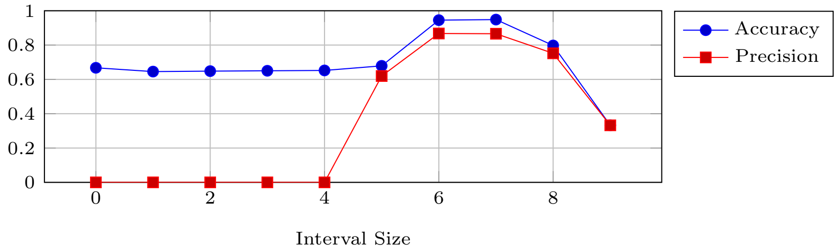

Figure 3: Precision and accuracy for covert channels using the payload sizes 50 and 60 bytes.

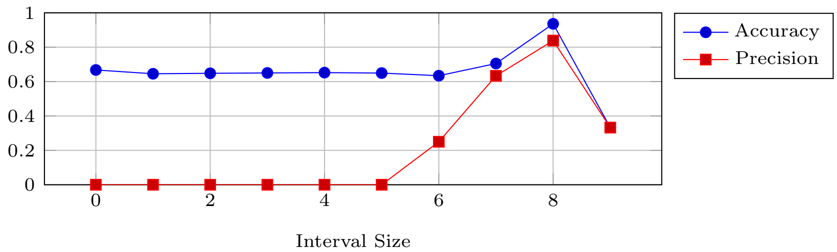

Figure 4: Precision and accuracy for covert channels using the payload sizes 100 and 200 bytes.

Payload sizes [bytes]	**$\varDelta$ of pkt. size**	**Avg. κ value**
1,000 / 1,001	1	4.43448
100 / 101	1	4.43677
50 / 60	10	5.34675
100 / 200	100	6.16748
100 / 1,000	900	6.03886

Table 3: Resulting κ values for two-symbol covert channels.

5.1.2 Combining Two-symbol Covert Channels

Figure 5 illustrates the detectability for a mixture of all above-mentioned covert channels and legitimate data mentioned in Table 2. The fraction of flows and packets per type of covert channel was equal, resulting in 20,000 packets per covert channel (100,000 covert channel packets overall). Again, the same number of legitimate flows and packets were used in this test to provide balanced classes. The best interval in terms of precision, accuracy and recall was #8 (accuracy: 71.54%, F-Score: 64.90%), i.e., without further interval optimization, we cannot detect the mixture of two-symbol covert channels.

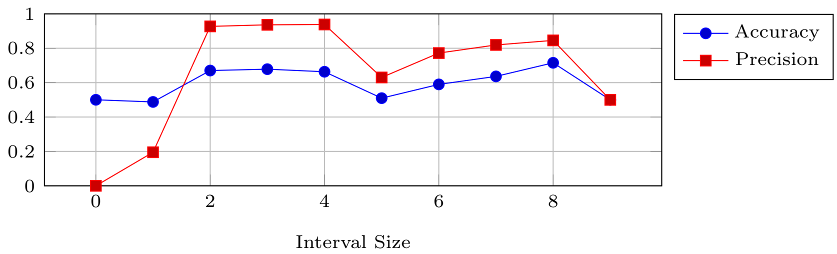

Figure 5: Precision and accuracy for a mixture of *all* two-symbol covert channels.

5.1.3 Covert Channels with > 2 Symbols

Sophisticated covert channels can utilize more than two secret symbols, so that more information can be transferred per packet, i.e., for n symbols, $\log_2(n)$ bits can be transferred per packet. Thus, the required number of packets can be reduced. To analyze such channels, we generated traffic for covert channels using 3, 4 and 8 different symbols which were reflected in the payload sizes between 100 and 800 bytes (Table 2).

First, we analyzed a covert channel with 3 different payload sizes (100, 200 and 300 bytes). As shown in Figure 6, interval #6 performed best (F-Score:

92.90%, accuracy: 94.93%). Results of interval #7 were similarly good. However, all other intervals provided no useful results.

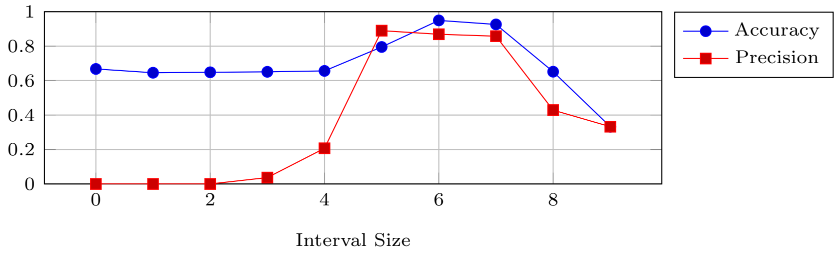

Figure 6: Precision and accuracy for covert channels using the payload sizes 100, 200 and 300 bytes.

In case of a covert channel with four different payload sizes (100 to 400 bytes), the best performing interval was #4 (see Figure 7). The F-Score was 90.66% and the accuracy 94.06%. Other intervals resulted in F-Scores below 80% and where thus considered unsuitable.

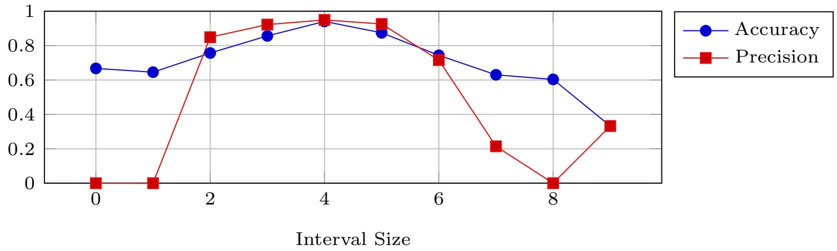

Figure 7: Precision and accuracy for covert channels using the payload sizes 100, 200, 300 and 400 bytes.

Finally, we analyzed a covert channel with 8 different payload sizes (Figure 8). This channel was detectable with an accuracy of 97.66% and an F-Score of 96.59% using interval #1. All other intervals provided no useful results.

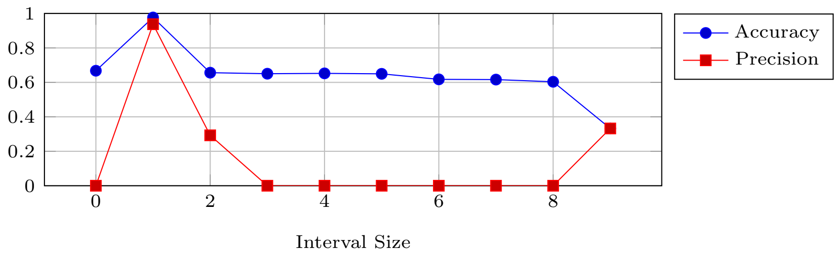

Figure 8: Precision and accuracy for covert channels using the payload sizes 100 to 800 bytes (in steps of 100 bytes).

5.2 ϵ-similarity

Similarly, like a κ value for the compressibility score was initially determined, we first needed to establish suitable thresholds for the ϵ-similarity. It turned out that $\epsilon = 0.1$ (i.e., 10%) provided good results when we expect 0.34% or 0.35% of the λ values below ϵ (see Fig. 9 for a two-symbol channel using 1000 and 1001 bytes).

All results for two-symbol covert channel were highly similar, which is rooted in the fact that the ϵ-similarity counts the number of relative packet size increases while the actual increase is reflected by the configured threshold. For this reason, we could achieve high accuracy and precision for above mentioned limits. However, we could not obtain any useful results for covert channels with three or more symbols when applying the ϵ-similarity, rendering the result of our countermeasure variation unsuitable for such channels.

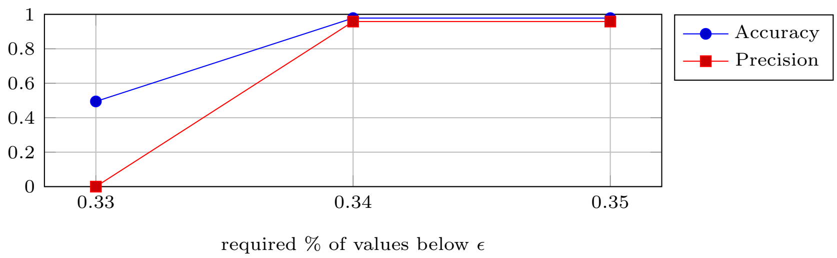

Figure 9: ϵ-similarity: precision and accuracy for covert channels using the payload sizes 1,000 and 1,001 bytes; depending on interval size.

5.3 Regularity

For this detection approach, we first studied the impact of the number of packets in a flow and of the window size on the regularity (see Fig. 10 for results obtained for ten different flow sizes, and see Fig. 11 for results obtained from a single flow with ten different window sizes). The results reveal that a regularity-based covert channel detection is only feasible under two conditions. First, the number of packets of the flow to be tested for covert channels must be approximately the same as in the flow used for the comparison. Secondly, the same window size must be applied.

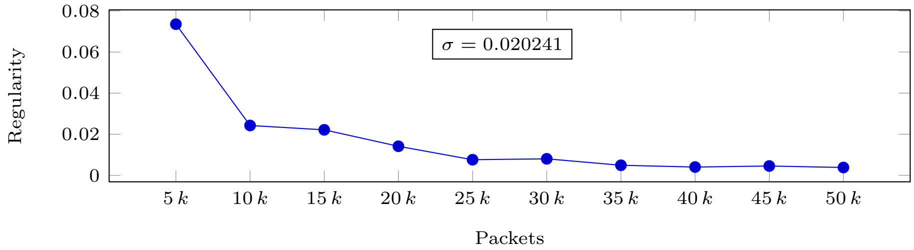

Figure 10: Regularity values depending on flow sizes (constant window size)

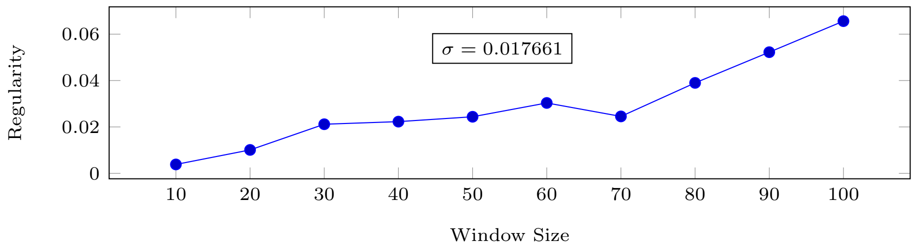

Figure 11: Regularity values depending on window size (constant flow size)

To determine typical regularity values for covert channels, we determined these values by creating covert channels with several different packet sizes. Therefore, we generated 91 flows, ranging from 1,000 to 10,000 packets (incremented in steps of 100 packets) for each symbol combination. From each of these flows the regularity value was calculated with every window size from 10 to 100 (incremented in steps of 10). This results in a list of 910 values for each symbol combination that we used to determine the presence of a covert channel. Note, that these values are similar to the κ values of the compressibility, i.e, they will later be used together with an interval to detect a covert channel.

Next, we generated one set of traffic data for each symbol combination. Each of these datasets contained 800 flows. 400 of them were legitimate flows extracted from the NZIX data set. The other 400 are flows containing a covert channel and appropriately parameterized symbols. In order to have homogeneous test data, the flows with covert channels were mapped to the same flow sizes as the legitimate flows, i.e., resulting in the same number of packets per flow to be compared.

Analogous to the previous countermeasure variations, we calculated precision and accuracy for different interval sizes. All experiments were repeated five times to have the proper statistical relevance.

First, we analyzed a two-symbol covert channel with a one-byte difference in packet sizes (1,000 and 1,001 bytes). Fig. 12 shows the accuracy and precision based on the interval size (average values over all window sizes) and based on the window size (average values over all interval sizes). As visible in the figure, acceptable combinations of accuracy and precision were not determined. We obtained similar results for the packet sizes 100 and 101.

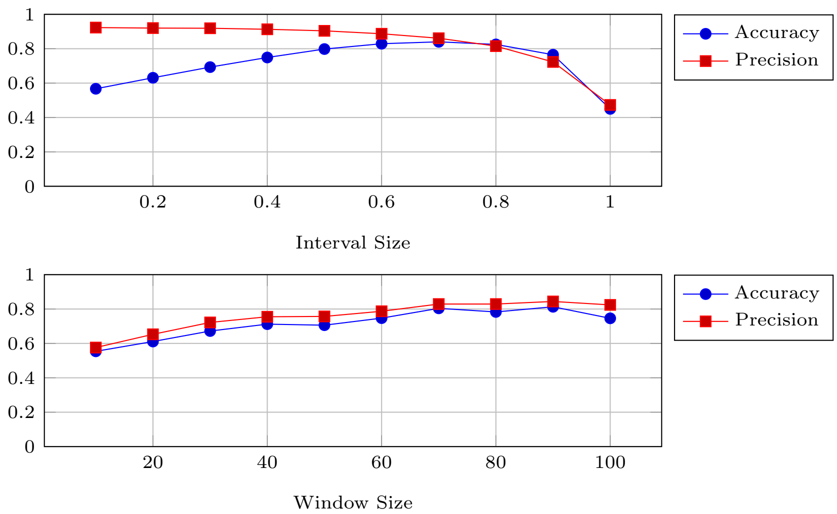

Figure 12: Regularity: precision and accuracy for covert channels using the payload sizes 1,000 and 1,001 bytes; depending on interval size (calculated over all window sizes) and on the window size (calculated over all interval sizes).

Next, we analyzed a covert channel with a 10 byte difference (50 and 60 bytes, two-symbol channel). As shown in Fig. 13, we could achieve a high accuracy (99.3% and 99.2%) combined with a precision of 92.4% and 93.4% for the interval sizes 0.8 and 0.9. All other interval sizes were lacking a high precision. A window size of 90 provided the best results. For this reason, a window size of 90 combined with an interval size of 0.8 to 0.9 is preferable.

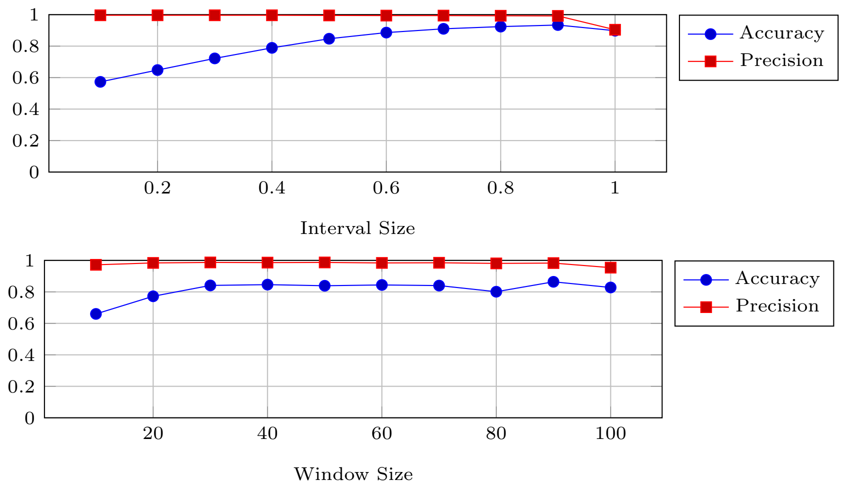

Figure 13: Regularity: precision and accuracy for covert channels using the payload sizes 50 and 60 bytes.

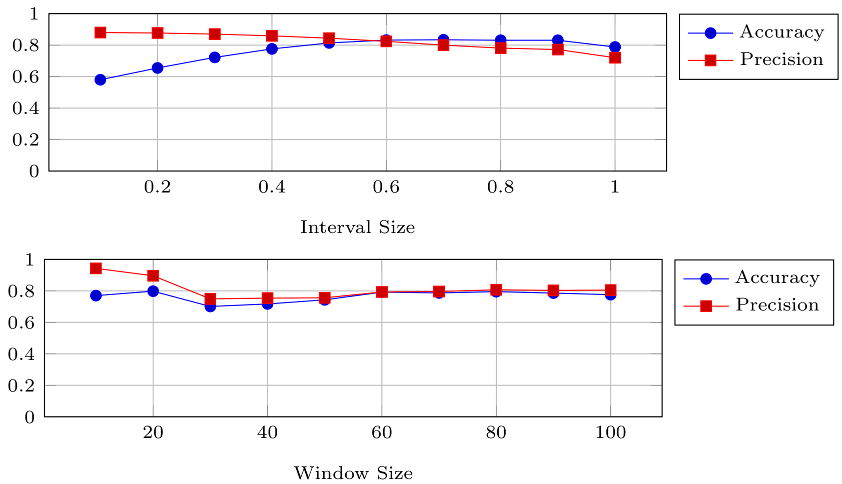

Figure 14: Regularity: precision and accuracy for covert channels using the payload sizes 100 and 200 bytes.

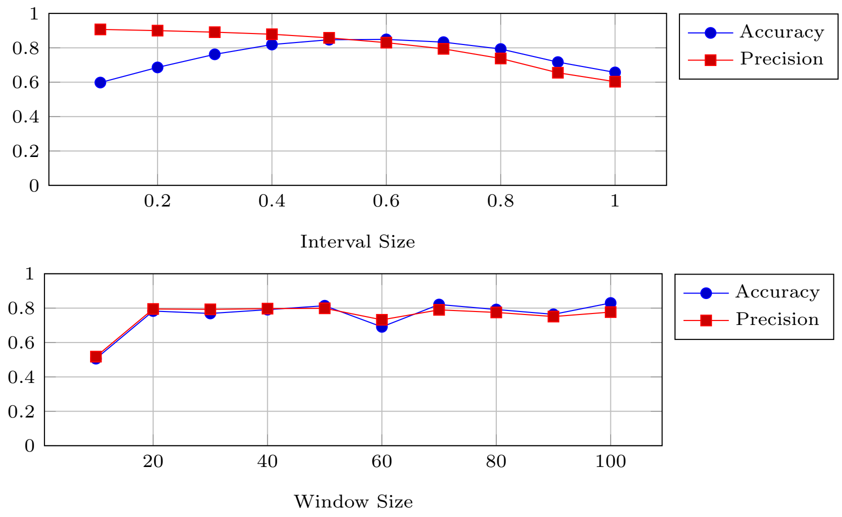

Figure 15: Regularity: precision and accuracy for covert channels using the payload sizes 100, 200 and 300 bytes.

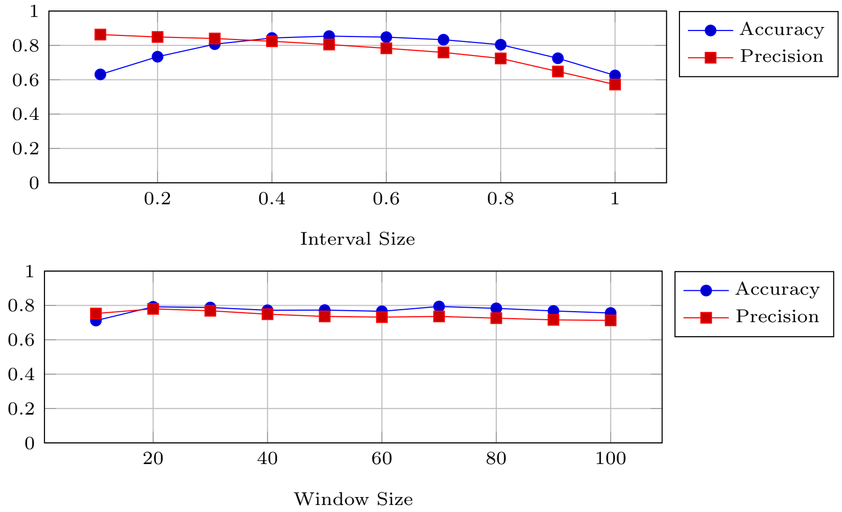

Figure 16: Regularity: precision and accuracy for covert channels using the payload sizes 100, 200, 300 and 400 bytes.

We further increased the difference between the symbols to 100 bytes (two-symbol channel with 100 and 200 bytes). Interestingly, no practically useful results could be obtained (Fig. 14) for this setup. It is possible that our tested ranges were either not suitable for this channel or that regularity is not the appropriate tool to detect larger packet size differences, which will be subject to further analysis.

We also considered a three-symbol (100, 200, and 300 bytes) and a four-symbol covert channel (100 to 400 bytes). As visualized in Figs. 15 and 16 we could not obtain acceptable results for these channels. After further investigating this issue we could determine that the degradation of result quality is not primarily linked to the number of symbols but to the increase in packet size difference. We could obtain high accuracy and precision results for the three-symbol and four-symbol channels (100, 101, 102 bytes) and (100, 101, 102, 103 bytes).

6 Conclusion

We have shown that countermeasure variation for covert channels is feasible. Therefore, it is necessary to *transform* a detection method, i.e., to adapt it so that it works with another hiding pattern. We exemplified the feasibility of countermeasure variation by transforming the compressibility score κ, the ϵ-similarity and the regularity that were originally introduced to detect Inter-packet Times-based covert channels so that they can be applied to Size Modulation-based covert channels.

After applying countermeasure variation to the *compressibility* metric, we were able to detect several two-symbol covert channels with accuracy values over 97% and F-Score values over 95% when suitable intervals were selected. The detectability of mixed two-symbol covert channels was not satisfying (71.5% accuracy and 64.9% F-Score). However, all channels with 3, 4 or 8 symbols were detectable with accuracy and F-Score values between 90% and 97%.

The countermeasure variation of the *ϵ-similarity* provided high-quality results for two-symbol covert channels under narrow limits (0.34%/0.35% of all values below $\epsilon = 0.1$). Out of these limits, no detection was possible. The approach did not provide useful results when the channels had more than two symbols.

Finally, the performance of the *regularity* metric depended on suitably selected interval and window sizes. Promising results were achieved for two-symbol channels with 50 and 60 bytes as well as for channels with a difference of only 1 byte in their size, even if they contained more than two symbols.

In future work, we will extend our parameters to determine whether modifications of the proposed approaches can improve the quality of the detection results for all three countermeasure variations.

References

[Cabuk et al., 2004] Cabuk, S., Brodley, C. E., Shields, C.: "IP covert timing channels: design and detection"; Proc. 11th ACM conference on Computer and Communications Security (CCS'04); 178–187; ACM, 2004.

[Cabuk et al., 2009] Cabuk, S., Brodley, C. E., Shields, C.: "IP covert channel detection"; ACM Trans. Inf. Syst. Secur.; 12 (2009), 4, 22:1–22:29.

[Deutsch and Gailly, 1996] Deutsch, P., Gailly, J.-L.: "Zlib compressed data format specification version 3.3 – RFC 1950"; (1996); `https://www.ietf.org/rfc/rfc1950.txt`.

[Elsadig and Fadlalla, 2017] Elsadig, M. A., Fadlalla, Y. A.: "A balanced approach to eliminate packet length-based covert channels"; Proc. 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS); 1–7; IEEE, 2017.

[Girling, 1987] Girling, C. G.: "Covert channels in LAN's"; IEEE Transactions on Software Engineering; 13 (1987), 292–296.

[Handel and Sandford, 1996] Handel, T. G., Sandford, M. T.: "Hiding data in the OSI network model"; Information Hiding Workshop; 23–38; Springer, 1996.

[Ji et al., 2009] Ji, L., Jiang, W., Dai, B., Niu, X.: "A novel covert channel based on length of messages"; Proc. 2009 International Symposium on Information Engineering and Electronic Commerce; 556–559; IEEE Computer Society, 2009.

[Ling et al., 2013] Ling, Z., Fu, X., Jia, W., Yu, W., Xuan, D., Luo, J.: "Novel packet size-based covert channel attacks against anonymizer"; IEEE Transactions on Computers; 62 (2013), 12.

[Mazurczyk and Caviglione, 2015] Mazurczyk, W., Caviglione, L.: "Steganography in modern smartphones and mitigation techniques"; IEEE Communications Surveys & Tutorials; 17 (2015), 1, 334–357.

[Mazurczyk and Szczypiorski, 2012] Mazurczyk, W., Szczypiorski, K.: "Evaluation of steganographic methods for oversized ip packets"; Telecommunication Systems; 49 (2012), 207–217.

[Mazurczyk et al., 2016] Mazurczyk, W., Wendzel, S., Zander, S., Houmansadr, A., Szczypiorski, K.: Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures; Wiley-IEEE, 2016.

[Mazurczyk et al., 2018] Mazurczyk, W., Wendzel, S., Cabaj, K.: "Towards deriving insights into data hiding methods using pattern-based approach"; Proc. ARES'18 (CUING Workshop); 10:1–10:10; ACM, 2018.

[Millen, 1999] Millen, J.: "20 years of covert channel modeling and analysis"; Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on; 113–114; IEEE, 1999.

Murdoch, S. J., Lewis, S.: "Embedding covert channels into TCP/IP"; Proc. Information Hiding Conference 2005; volume 3727 of LNCS; 247–261; Springer, 2005.

[Proctor and Neumann, 1992] Proctor, N. E., Neumann, P. G.: “Architectural implications of covert channels”; Proceedings of the Fifteenth National Computer Security Conference; volume 13; 28–43; 1992.

[Schumacher et al., 2013] Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., Sommerlad, P.: Security Patterns: Integrating security and systems engineering; John Wiley & Sons, 2013.

[WAND group, 2000] WAND group: “NZIX data set”; https://wand.net.nz/wits/nzix/2/ (2000).

[Wendzel, under review] Wendzel, S.: “Protocol-independent detection of ‘messaging ordering’ network covert channels”; (under review); preliminary title.

[Wendzel et al., 2014] Wendzel, S., Mazurczyk, W., Caviglione, L., Meier, M.: Hidden and uncontrolled – On the emergence of network steganographic threats; Proc. ISSE 2014; 123–133; Springer-Vieweg, 2014.

[Wendzel et al., 2015] Wendzel, S., Zander, S., Fechner, B., Herdin, C.: “Pattern-based survey and categorization of network covert channel techniques”; Computing Surveys (CSUR); 47 (2015), 3.

[Wendzel et al., 2018] Wendzel, S., Eller, D., Mazurczyk, W.: “One Countermeasure, Multiple Patterns: Countermeasure Variation for Covert Channels”; Proc. Central European Cybersecurity Conference; ACM, 2018.

[Wolf, 1989] Wolf, M.: “Covert channels in LAN protocols”; Proc. Local Area Network Security; volume 396 of LNCS; 89–101; Springer, 1989.

[Zillien and Wendzel, 2018] Zillien, S., Wendzel, S.: “Detection of covert channels in TCP retransmissions”; Proc. 23rd Nordic Conference on Secure IT Systems (NordSec); volume 11252 of LNCS; 203–218; Springer, 2018.