SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:3011–3029
Published online 21 July 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1531

SPECIAL ISSUE PAPER

DAT detectors: uncovering TCP/IP covert channels by descriptive analytics

Felix Iglesias*, Robert Annessi and Tanja Zseby

Institute of Telecommunications, TU Wien, Gusshausstraße 25 / E389, 1040 Wien, Austria

ABSTRACT

Covert channels provide means to conceal information transfer between hosts and bypass security barriers in communication networks. Hidden communication is of paramount concern for governments and companies, because it can conceal data leakage and malware communication, which are crucial building blocks used in cyber crime. We propose detectors based on descriptive analytics of traffic (DAT) to facilitate revealing network and transport layer covert channels originated from a wide spectrum of published data-hiding techniques. DAT detectors transform communication data into flexible feature vectors that represent traffic by a set of extracted calculations and estimations. For the case of covert channels, the core of the detection is performed by the combined application of autocorrelation calculations and multimodality measures built upon kernel density estimations and Pareto charts. DAT detectors are devised to be embedded as extensions of network intrusion detection systems, being able to perform fast, lightweight analysis of numerous flows. The present paper focuses specifically on TCP/IP traffic and provides suitable classifications of TCP/IP fields and related covert channel techniques from the perspective of the statistical detection. The proposed methodology is evaluated with public traffic datasets as well as covert channels generated according to main techniques described in the related literature. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
covert channels; network security; statistical analysis

***Correspondence**
Felix Iglesias, Institute of Telecommunications, TU Wien, Gusshausstraße 25 / E389, 1040 Wien, Austria.
E-mail: felix.iglesias@nt.tuwien.ac.at

1. INTRODUCTION

A *covert channel* is a parasitic communication channel used to clandestinely convey information and bypass security barriers. The information in a covert channel travels in hidden places, which are unexpected but for the sender and the receiver of the data.† Covert channels are used in communication networks to conceal the leakage of confidential information as well as to hide the evidence of communication patterns between hosts. Such non-legitimate communications may be carried out by users to circumvent censorship, for example, in situations where communication itself must remain secret. However, covert channels can also be used by malwares, hackers, criminals, or terrorists to plan, coordinate, and conceal attacks [2,3]. In this way, covert channels are a serious threat for companies and governments.

This work introduces descriptive analytics of traffic (DAT) detectors: a set of methods for detecting covert channels in TCP/IP traffic in a passive warden scenario, that is, not requiring the modification of traffic. It is noteworthy that covert channels have also provoked the proposal of active measures to mitigate the existence of such communications in security networks (e.g., [4]). We specifically target covert channels in TCP/IP because those are the dominant protocols and provide opportunities to hide communication virtually anywhere on the Internet.

Descriptive analytics of traffic schemes are not confined to a specific type of covert channel; instead, they aspire to state an exhaustive methodology that faces the identification of multiple covert channel techniques proposed in the literature. DAT detectors are to be embedded in network intrusion detection system (IDS; especially if located in border gateways; Figure 1). They are intended to perform a fast, lightweight analysis and labeling of flows as addition

† Comprehensive documentation about the design of codes for concealing information in multiple types of signals is provided in [1].

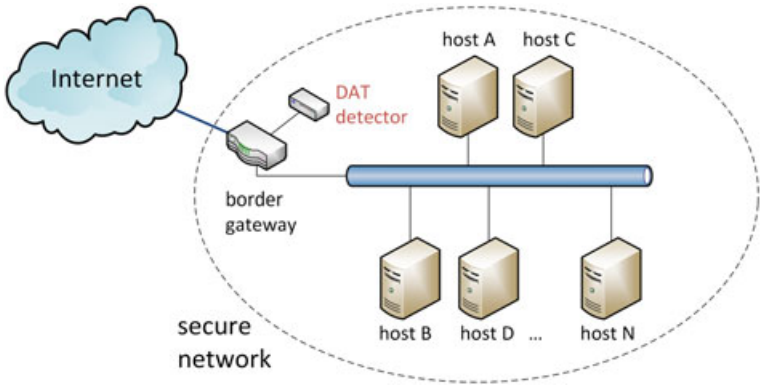

Figure 1. Basic network placement of DAT detectors.

to classical IDS. To this end, simplicity and fast calculation are mandatory requirements, because detectors are set to deal with big amount of data under time constraint conditions. Such demands have determined the overall design of detectors as flexible filters based on traffic data analytics. Hence, the major novelty introduced by DAT schemes is to present a methodology for a first-level security covert channel detection, whose power is maximized by the combined application of proper traffic representation and fast statistical techniques. For the presented approach, multimodality estimations and the sum of autocorrelation coefficients are proposed as core components of the DAT methodology.

Fast covert channel detectors do not attempt to decode presumable covert communications, but rely on communication footprints to establish conjectural assessments about the existence of covert channels. Except for obvious cases, detectors cannot determine the presence or absence of covert channels in an absolute manner. Instead, they provide an evaluation – that is, a *level of suspicion* – that can be understood either as a probability or the pertinence to a fuzzy group [5]. This fact implies a significant and unavoidable degree of uncertainty in covert channel detection outcomes. Due to the lightweight requirements, DAT detectors might be unable to unmask very complex covert channels. Nevertheless, they are designed to mark flows that are susceptible to contain covert channels even if they do not find evidences to trigger a suspicion. Hence, they act as filters that discriminate incoming traffic and reduce the scope for more specific and resource demanding detection approaches. Therefore, DAT detectors are designed to be the first filter instance in a multi-layer detection system (Figure 2). This functionality is in agreement with Dainotti *et al*., who in [6] state that effective IDS must be formed by multi-layer frameworks or combined techniques. Covert channel detectors as flexible plugins or extensions have been also previously introduced in [7].

Data transmitted in a covert channel can be plaintext or encrypted (ciphertext). Without data encryption, communication via covert channels just provides security by obscurity, violating Kerckhoffs's principle [8]. Because

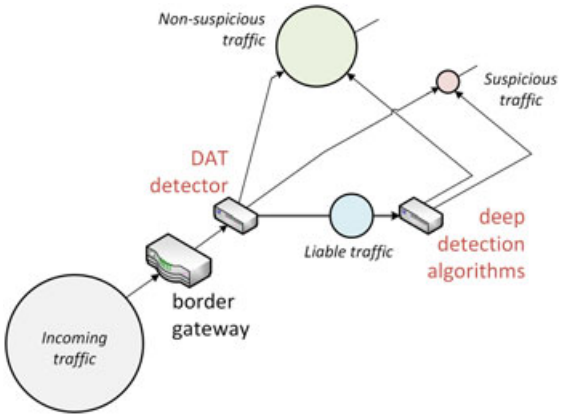

Figure 2. Integration of DAT detectors with deep detection algorithms, which become feasible as their load is significantly alleviated by the incorporation of a first DAT-filtering phase.

encryption changes the statistical properties of the data, the detection of encrypted traffic differs from the detection of plaintext (Section 3.4). Our paper just focuses on the detection of covert communication (both, plaintext and encrypted), decoding or decrypting covert channels is a challenge not addressed here.

In short, this paper introduces the theoretical groundings and central analysis methods of DAT schemes. It also studies covert channels from a statistical perspective and proposes an appropriate classification. We provide analysis examples and a first proof of concept of the DAT methodology with real data. A testbed for the generation of covert channels (grounded by a framework like the one introduced in [9]) and the first beta version of the DAT open-source software are to be released.

1.1. Terminology

To prevent possible ambiguities, we clarify the meaning of some terms used throughout this paper:

- *Flow* (traffic) is defined as 'a set of packets or frames passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties' [10]. We consider unidirectional flows; the common properties that we use to identify packets belonging to the same flow are only source address and destination address. For instance, a TCP connection between client A and server B contains two flows: A to B and B to A.
- *Field* (TCP/IP): every distinct header field defined for the IP and TCP headers according to RFC 791 [11] and RFC 793 [12], respectively. Additionally, we consider the difference in arrival times of subsequent packets – the packet IAT – as a characteristic of a flow, and for simplicity also name it a field.
- *Feature*: every measurement or estimation included in the input vector to be processed by the analysis tools, for example, the number of packets in a flow.
- *Symbol* (covert): the minimal piece of information with an abstract meaning that is sent in a covert channel, for example, '0' and '1' in a binary covert channel, or 'a', 'b', and 'z' in a covert channel that hides ASCII characters.

2. RELATED WORK

Literature about covert channels in communication networks is extensive. A comprehensive approach including covert channels has been recently published in [13]. We can differentiate three types of papers that are also often combined: (a) surveys and classifications, (b) detection schemes and methodologies, and (c) proposals of covert channel techniques. In this section, we focus on surveys and classifications as well as detection methods; techniques for covert channels are introduced later in Section 3. The paper focuses on covert channels in TCP/IP communication, although the introduced methods can be applicable to other protocols.

2.1. Surveys and classifications

In [14], a classification for covert channels is presented. This classification is based on the characteristics of *information flow sequences*‡ and is highly coupled to the proposed identification method. Alternatively to using flow sequences as foundation, covert channels are also classified based on entropy analysis in [15]. A more comprehensive, intuitive, and frequently cited classification is introduced by Zander *et al.* in [16]. Zander, together with other authors, has also published detection approaches [17], novel covert channel techniques [18], and an open software tool for the generation of covert channels [19]. In an attempt to standardize and facilitate the development of countermeasures, a recent survey from 2015 [20] classified covert channels using *patterns* according to the *pattern language markup language* [21]. The authors of this survey found that covert channel techniques published between 1987 and 2013 fit 11 different patterns.

We here propose an alternative classification, more suitable to our purposes, that groups covert channel techniques according to their statistical properties and characteristics for the detection.

‡ The meaning of "flow" in [14] differs from the definition provided here.

2.2. Detection schemes and methodologies

As mentioned in [20], existing detection methodologies in a passive warden scenario can be grouped in different classes:

- *Traffic irregularities*: a significant number of covert channel techniques are based on an irregular use of traffic fields or on the exploitation of protocol flaws. In [22], some of such fixed evidences for TCP/IP traffic are discussed. In [23] and [24], authors additionally propose active packet *normalizations* to eliminate covert channels. Our DAT detector looks for traffic irregularities caused by covert channels in a *packet compliance checking* phase, described in Section 5.2.
- *Statistical analysis*: covert channels change either header fields or the timing of packets, normally altering the distributions of network traffic connections. Therefore, covert channels can be detected by studying statistical characteristics of traffic flows. Such methods are described in [25–29]. We use statistical techniques (multimodality, autocorrelation, and descriptive statistics) as basis to build the input feature vector of our detection scheme (Sections 4 and 5).
- *Machine learning (ML)* has been frequently proposed for the detection of covert channels due to their exploration and classification power, beyond checking for traffic irregularities and mathematical analysis. For instance, suport vector machines [30] [31] [29], neural networks [32], or decision trees [33]. DAT detectors incorporate *inter-field analysis*, which uses a set of patterns obtained from ML classifiers during offline analysis (Section 5.5). Using pre-calculations in offline phases, we can prevent high computational costs during detection, which otherwise would hinder the use of ML.

DAT detectors contribute to the current state of research and application of covert channels by providing a detection methodology with the following qualities:

- *Comprehensive* – covers a wide spectrum of covert channel techniques described in the literature;
- *Lightweight* – devised to be embedded in devices that process large amounts of traffic, such as IDS; and
- *Flexible* – based on traffic represented by feature vectors where new estimations or calculations can be easily added. Also, complementary analyzers and detectors can be integrated downstream.

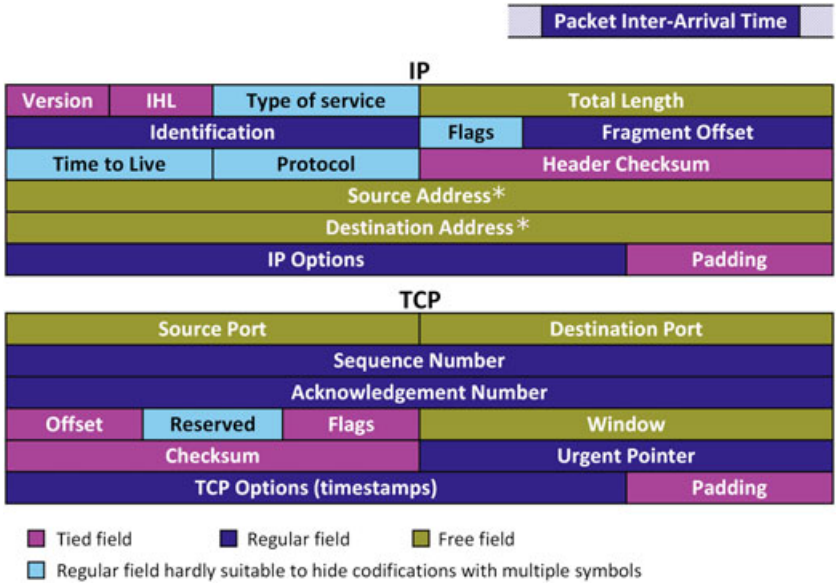

Figure 3. TCP/IP header field classification based on the limitations to exploit covert channels.

3. COVERT CHANNELS

This paper focuses on covert channels hidden in TCP/IP headers, packet sizes, and timing. In this section, we first introduce TCP/IP fields according to their potential to hide covert channels. Later, a representative set of existing, documented covert channel techniques are presented. We underline the used fields in every case and classify them based on their similarities for the detection. Finally, we comment on some field peculiarities related to scale types that are relevant for the statistical analysis as well as discuss about the challenges that message codification and encryption involve.

3.1. TCP/IP field classification

Instead of using the *payload*, covert channels transmit content data inside network traffic header fields. A classification of TCP and IP header fields according to their potential to hide covert data is proposed as follows (Figure 3):

- *Tied fields* have a fixed value, a decisive meaning for the traffic configuration or can be inferred from other fields, for example, Internet header length (IHL), Checksums. The use of covert channels in tied fields is highly improbable and easy to detect.
- *Regular fields* are expected to show common values or follow behavioral patterns in normal traffic situations.
- *Free fields* can take a wide range of different values in normal traffic and show random, unknown or costly traceable distributions. These fields are the most challenging for the covert channel detection.

3.2. Covert channel techniques

From the perspective of the detection, covert channel techniques can be classified as follows:

- *Value to symbol correspondence*. In a value to symbol correspondence either one or multiple field values can correspond to the same symbol of a covert message. For example, in [34], the IP identification conceals covert symbols – previously encrypted by a total automorphism – in the high bits of the field, being the low bits established at random. In this way, multiple IP IDs correspond to the same symbol (see the example in Figure 4, left). Total length [35] [36], protocol [17], and time to live (TTL) [18] [37] or even the destination address [38] have also been used to hide covert symbols by a similar technique. In [39], the author proposes some types of covert channels with value to symbol correspondence: setting the IP identification with a value equal to the covert byte multiplied by 256, or the initial sequence number (ISN) with a value equal to the covert byte multiplied by $65536 \cdot 256$. In [40], the least two significant bits of the type of service – normally unused – mark different covert symbols.

* Depending on the locations of the warden, the covert channel sender and the covert channel receiver, the class assigned to source and destination addresses can be equally *tied*, *regular* or *free*. In any case, because covert channels distributed over multiple flows are not addressed by our implementations of DAT detectors, in this work, source and destination addresses are fixed per flow and not considered to be used for hiding information.

DOI: 10.1002/sec

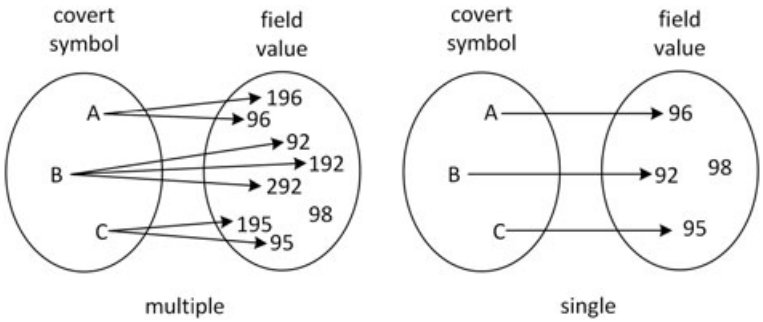

Figure 4. Examples of sets with value to symbol correspondence. Note that not all field values must be necessarily linked to symbols.

In some cases, only one field value maps a covert symbol (Figure 4, right). For instance, in [41], where the Don't Fragment bit of the IP header is used to clandestinely send '0' or '1'. Similarly, in [24], the existence or lack of the checksum – optional for UDP packets – encodes the secret binary data.

- *Value ranges as symbols* refers to a special case of correspondence where covert symbols are represented by ranges of field values. For instance, in [42], authors propose studying the statistical distribution of packet lengths in specific UDP services, Total length ranges are used to represent symbols; therefore, a covert channel is conveyed in a communication where packet length variation matches statistical properties of normal traffic. Also TTL value ranges are studied for this type of covert channel in [43].
- *Container fields*. The fact that a covert channel technique hides one symbol per packet or already a portion of the message instead (some concatenated symbols) involves substantial differences from the perspective of the distribution properties. For such reason, we consider that a covert channel is hidden in container fields when the amount of covert information sent per packet is greater than 1 byte. Container fields are usually (but not always) accompanied by *marker fields*, which inform the receiver about the existence of covert information in the current packet. Markers can also be embedded in the container field itself as *ad hoc* headers.

  In [24], authors present TCP Flags as markers: packets with the TCP RST flag active can contain covert information in the *payload*; also packets where the URG bit is not set can leak 16 bits of information in the urgent pointer field. In [44], up to 36 bytes of covert information are sent in the IP options field by using packets with the record route option activated and the field contents faked. TCP or IP padding fields can also be used to transport covert data [45]. In [32], up to three covert bytes are transmitted in the ISN by using a special encryption that tries to simulate randomness to hamper the detection. Even checksum fields can be used to contain hidden messages (16-bit long for the IP checksum), but requiring manipulation of other fields, usually IP or TCP options [46] to not invalidate the checksum.

  Covert channels in container fields can be difficult to unmask if we only look at field histograms; they have a higher capacity of covert data per packet, which implies more randomness or less effect on distributions. Nevertheless, those channels make use of special or rare packet characteristics that are usually enough to arise suspicion (e.g., packets with options, padding different to zero). The suspicion is highly increased if such packets repeat with variable values in the same flow.
- *Timing channels* conceal information by using timing properties of network traffic – mainly IAT. In most cases, packets are intentionally generated just for such purpose. For instance, the authors of [47] propose sending data bit by bit, where '0' corresponds to a prefixed time delay between consecutive packets and '1' is received when there is no delay or it is negligible. Similarly, in [48], binary symbols are identified by different time delays. In [49], a general methodology for the design of timing and low-bandwidth storage covert channels (at most 1 bit of covert information per packet) is presented.

  Covert timing channels can also be generated by manipulating existing traffic on the go. Considering interactive communication applications (SSH, HTTP, etc.), adding little time variations facilitate the insertion of additional data just by changing the time properties of packets [50]. Another proposal is presented by Gianveccio *et al*. in [51], where the delay variation in covert timing channels is established based on models that match the timing distribution of specific services. In [52], connections are slowed to make low-order bits of the TCP timestamp field match the desired value. Finally, in [53], authors introduce LACK, a mix between timing and container field techniques where the covert information is hidden in the *payload* of intentionally delayed voice over Internet protocol packets.
- *Derivative approaches*. Some covert symbols are not directly hidden in the value of the field but in how this value changes throughout successive packets. For

example, the subtraction of the source port values of consecutive packets are considered ASCII symbols in [54]. The TTL value difference from one packet to its subsequent one implements a binary covert codification in [43] as well as one of the novel techniques proposed in [18].

3.3. Field scale types

In addition to classifying TCP/IP fields according to their potential to hide covert data (Section 3.1), fields can also be classified according to their *assumed* scale type. In TCP/IP headers we basically find three different scale types:

- *Numerical fields*, for which the value is a number, for example, total length or TTL. Arithmetic operations between two different values are meaningful in the normal usage context of the field.
- *Nominal fields*, for which the value is a label, for example, source port or protocol. Arithmetic operations between two different values do not make sense in their normal usage context. We consider flag-type fields (IP flags and TCP flags) as a special form of nominal fields, because every bit of the field has an independent meaning and logic operations between flag-type values are possible.
- *Variable* (complex or combined), for example, IP options or TCP options, where, for instance, timestamps and flags are combined together within the same field. The contents of such fields have a variable meaning; depending on the packet, they can mix different scale types or concatenate diverse sub-fields.

Knowing the scale-type of a field is important to select the right method for detecting suspicious field distributions. Note that covert channels usually make an abnormal usage of field values (different scale type). For instance, a covert channel can utilize a numerical field as nominal (e.g., using a set of TTL values to build a covert channel with *value to symbol correspondence*), as well as it can use a nominal field as numerical (e.g. using the source port to build a *derivative* covert channel).

3.4. Plaintext, encoded, and encrypted data

Data transmitted in a covert channel can be plaintext or encoded (or just non-text data in any possible format). Moreover, covert messages can also be encrypted. Covert channels that transmit plaintext only provide security by obscurity, because the security just lies in the secrecy of the method used to hide the data. Therefore, data are expected to be encrypted before it is transmitted in a covert channel. The whole process then consists of encrypting the plaintext to ciphertext and then translating the ciphertext symbols into field values of the covert channel. The detection of covert channels carrying ciphertext or encoded text is different from the detection of covert channels carrying plaintext. Encryption methods may make ciphertext indistinguishable from a random bit sequence [55] and also make parameters of crypto systems look random [56].

This aspect can be better understood with an example. In a binary covert channel, we can expect a balanced number of symbols representing '0' and '1' values. However, if the covert channel masks symbols that represent ASCII characters and the transferred data are a non-encrypted message, we could expect a very dominant symbol representing the whitespace character followed by a set of around 15–25 symbols with less presence plus another subset with symbols scarcely used. On the other hand, if the covert information is not plaintext, for example, encoded or encrypted text, the expected use of the value distribution changes. Figure 5 illustrates this aspect. Note how both 8-bit codifications (plots on the right side) make a different usage of the available symbols (about 20 vs 256). The conclusion here is that the number of used symbols is more dependent on the codification system than on the transferred message.

Therefore, if ciphertext is hidden in a field value that usually also looks random (e.g., randomly chosen ISN), statistics will not reveal anything; it is hard, if not impossible, to detect such channels. On the other hand, if ciphertext is hidden in fields that have a non-random distribution, the ciphertext will change the distribution in a way that it looks more random. Therefore, ciphertext transmitted in a field with non-random characteristics becomes easy to spot.

4. ANALYSIS METHODS

Network monitoring systems nowadays need to cope with increasing data rates and process big amounts of data in short time. Analysis methods must be fast and lightweight. Consequently, the methods proposed here are based on data aggregation, simple transformations, fast calculations, and inference rules derived from cross comparisons. More complex detection techniques could be suitable in specific cases (e.g., [42], [31]); however, due to their high-computational effort, they cannot be considered as overall solutions able to face multiple fields from large amounts of IP connections, but reduced portions of traffic.

We depict the core methods embedded in DAT detectors in the following subsections.

4.1. Multimodality: distributions (S_k)

Processes for estimating multimodality seek to find the number of statistical modes contained in a population, that is, the number of significant peaks displayed by the probability density function of a random variable. A low computational cost method for exploring multimodality is by using kernel density estimates with normal density as a kernel function. In [57], the kernel density estimate on $Y_1, \ldots, Y_n$ univariate samples is defined by Equation 1:

DOI: 10.1002/sec

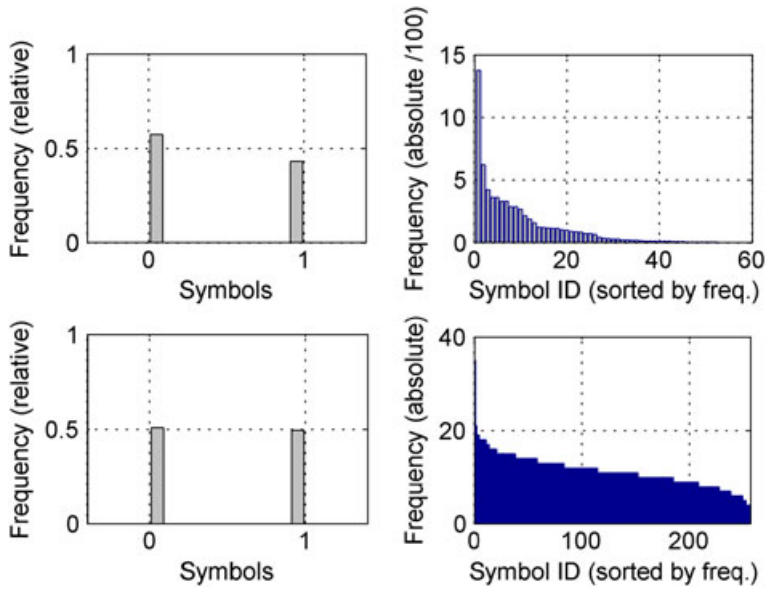

Figure 5. Frequency plots of the symbols used for encoding *The Raven* by E. A. Poe. Upper plot: binary encoding (left) and 8-bit symbols (right) using ASCII encoding. Lower plots: binary encoding (left) and 8-bit symbols (right) of the RAR-compressed text file.

$$\hat{f}(t;h) = \frac{1}{nh}\sum_{i=1}^{n} K\left\{\frac{(t-Y_i)}{h}\right\} \quad (1)$$

where K is a kernel function (e.g., normal density function), n is the number of observations, and h is the window width or smoothing factor. t denotes an event in the (estimated) probability density function, and the resulting probability density function $\hat{f}(t;h)$ is supposed to be the distribution that would generate the observed histograms. The number of modes (S_k) is calculated by a local maxima or peak detector algorithm over $\hat{f}(t;h)$.

Detecting multimodality in a flow can be quite convenient to reveal covert channels that use both *value to symbol correspondences* or *value ranges as symbols*. Figure 6 shows two examples where binary covert channels were concealed using source port values (1300 and 17000) and ranges (from 1200 to 1400 and from 1600 to 1800). Note that we can take the liberty of considering any analyzed variable as a continuous variable because kernel estimations are just used to detect multimodality.

A natural way of selecting the smoothing factor h is proposed in [57], although it involves bootstraping and therefore replications for each analysis, which increases the computational costs. h values can also be adjusted according to the field range and the expected usage – in non-suspicious conditions – of every specific field. Nevertheless, for our proof of concept tests, the direct 'rule of thumb' described in [58] turned out to be sufficient. Moreover, we apply two complementary multimodality estimators (Section 4.2). Both estimations, combined with the rest of descriptive features, provide sufficient information to draw a suitable picture of the flow context (see example in Figure 7).

4.2. Multimodality: symbols (S_s)

The estimated number of main symbols is a complementary approach to assess multimodality. The objective is to find the number of symbols that occur most often. We propose to perform a *Pareto analysis* (descending-sort histogram [59]) of the field under test. Instead of using the 80-20 law, we find the difference between consecutive Pareto bars a good measure to find the breaking point that separates dominant from spurious symbols. Given a Pareto chart $X = \{X_1, X_2, \ldots, X_U\}$ obtained from the histogram of a field, with U being the number of unique symbols, we can select a subset of indices $J = \{j, \ldots, U-1\}$ in a way that j ensures that at least 50% of the cumulative frequency is embraced by the symbols $\{X_1, X_2, \ldots, X_j\}$. Then, the number of main symbols S_s is a scalar defined in Equation 2:

$$S_s = \begin{cases} i \mid \max(X_i - X_{i+1}), & \forall (X_i - X_{i+1}) > \mu_f \\ \quad i \in J & \\ U \quad \text{otherwise} & \end{cases} \quad (2)$$

For cases where all symbols are equally relevant, we take into account the frequency average μ_f of the histogram as a minimum difference to overcome, thus avoiding wrong estimations. Differentiating *main* and *spurious* symbols is not free of subjectivity, as the example of Figure 5 clearly shows.

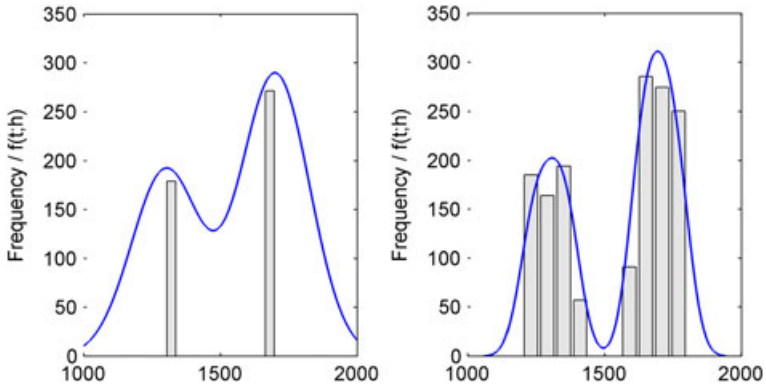

Figure 6. Histograms (bars) and kernel density estimations (curves) of a covert channel with *value to symbol correspondence* (left) and with *value ranges as symbols* (right). Density magnitudes are omitted as only the number of peaks is relevant for the purpose of the estimation.

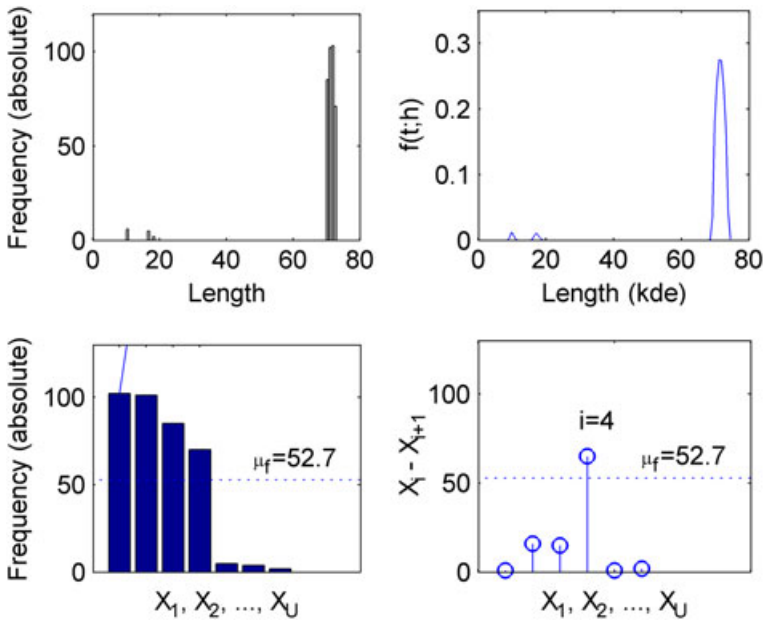

Figure 7. Example of captured packet lengths in a flow. Upper plots: histogram (left) and kernel density estimation (right). Lower plots: Pareto chart (left) and its derivative (right).

The estimation of main symbols is mainly useful to detect covert channels with *value to symbol correspondence* whose characteristics make them challenging for density-based methods. This can be better understood with an example: Imagine a covert channel masked in the Length field, using four symbols that are represented by the values 70, 71, 72, and 73. Other connections, which are not part of the covert channel, use packets with lower lengths. Figure 7 shows some related plots. The wide range of length values and the proximity of the values selected as symbols can disable the capability of the kernel density estimation method to assess the correct number of Modes (S_k). Nevertheless, the right number of main symbols (or modes) can be correctly estimated by S_s in such situations. In any case, as mentioned before, having both indices S_k and S_s allows a better interpretation of the specific context.

4.3. Sum of autocorrelation coefficients (ρ_A)

If we isolate a field of a flow and consider its values as a time series, traffic hiding covert data are different from normal traffic with regard to self-correlation. For instance, some fields, such as the IP *Identification*, are expected to follow self-correlated patterns. Breaking such patterns can suggest the existence of a covert channel. Autocorrelation coefficients have also been previously used for detecting covert channels in [29].

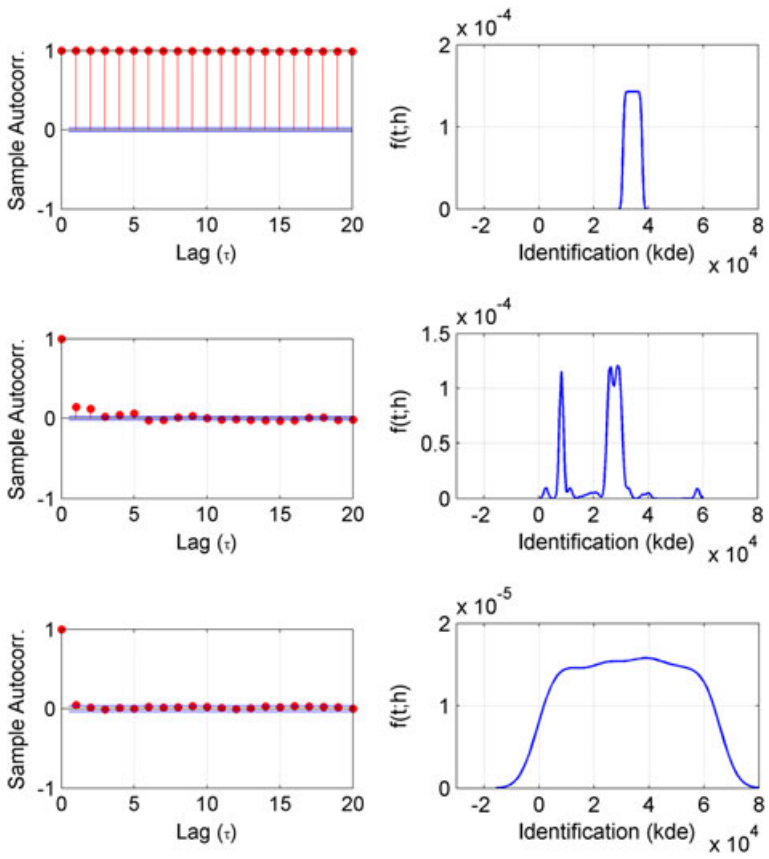

Figure 8. Autocorrelation plots (left) and kernel density estimations (right) of the Internet protocol identification of three different flows. Upper: sequential algorithm. Middle: covert channel technique described in [39] sending ASCII text. Lower: covert channel technique described in [39] sending compressed text.

We define the coefficient ρ_A as the average of a set of selected autocorrelation coefficients with different lags (e.g., $A = \{1, \ldots, 10\}$):

$$\rho_A = \frac{1}{N} \sum_{\tau}^{A} |R_\tau| \tag{3}$$

where N is the number of elements in A. R_τ is the autocorrelation coefficient of the time series $Y_1, \ldots, Y_n$ for the lag τ, defined in Equation 4:

$$R_\tau = \frac{E[(Y_t - \mu)(Y_{t+\tau} - \mu)]}{\sigma^2} \tag{4}$$

with μ and σ^2 being the mean and variance of the time series and E is the expected value.

In Figure 8, diverse flows are compared. The figure shows kernel density estimation curves and autocorrelation coefficients for the time series of the IP identification fields of three different flows. The upper plots correspond to an example of operating systems that establish the IP *identification* sequentially (all the algorithms described in [22] for different operating systems produce high autocorrelation coefficients). The technique presented in [39] has been deployed to covertly send *The Raven* by E. A. Poe as 8-bit ASCII symbols (plots in the middle) and as 8-bit RAR[§] compressed symbols (lower plots). Note that, in spite of the fact that some covert channel techniques can be difficult to detect by multimodality estimations, they can be detected by using autocorrelation coefficients.

5. DAT DETECTOR DESCRIPTION

The building blocks of the DAT detector are displayed in Figure 9 and explained in the following subsections. Previously, the criteria for labeling traffic are introduced.

5.1. Labeling criteria

Every communication checked by a DAT detector will be labeled according to the following criteria:

- *Non-suspicious* traffic is free of covert channels based on the DAT detector capabilities.
- Although improbable, *liable* traffic might contain a covert channel, but the detector does not find any

[§] http://www.rarlab.com/.

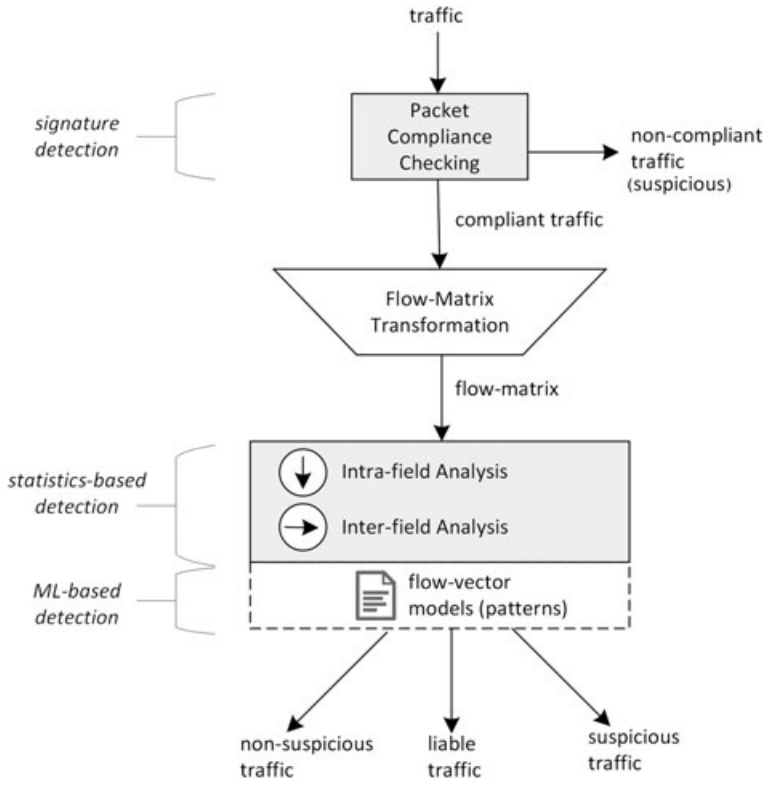

Figure 9. Block diagram of a descriptive analytics of traffic detector.

evidence or footprint that points in that direction. *Liable* traffic is stated as all traffic that is neither *non-suspicious* nor *suspicious*.

- *Suspicious* traffic in which the DAT detector has discovered anomalies that suggest the use of a covert channel is addressed with a qualitative label (*low*, *medium*, *high*), and a quantitative estimation of the potential amount of covert data that could have been sent: *C* (covert bytes).

5.2. Packet compliance checking

The first phase checks compliance of every analyzed packet, looking mainly at the values of *tied fields*. In short, the compliance checker extends some *firewall* policies and consists of a set of rules that automatically detect traffic, which is corrupted or does not comply to standard practices. Some examples of *non-compliant traffic* are packets with wrong checksums, padding different to 0, or invalid TCP flag combinations.

Non-compliant traffic is labeled and separated from other traffic. If a non-compliant packet is found, additionally all traffic belonging to the same flow (defined by source and destination IP) is filtered out and not processed in the following analysis phases.

This first filtering step already detects several covert channel possibilities like the ones introduced in [24] (payload in TCP RST packets and urgent pointer field without URG bit) and in [45] (padding).

5.3. Flow-matrix transformation

Further analysis requires the representation of traffic in the shape of a vector. Such vector can sketch the traffic from diverse perspectives of analysis. Two possible approaches are as follows:

- *Host characterization* uses just source or destination address to identify a flow and summarizes the unidirectional network activity of either a source or a destination. It requires less computational effort (because less flows are formed), calculates faster, and can detect bouncing covert channels and covert channels generated by using spoofed source IP addresses to hide information. However, it presents an important drawback, because all traffic from one source (or one destination) is considered together as a traffic aggregate. Therefore, covert channel statistical footprints are easily concealed by legitimate traffic.
- *Flow characterization* summarizes unidirectional traffic flows given a source-destination pair. Compared with host characterization, flows are less prone to be disturbed by legitimate traffic. It implies more evaluation effort but provides a highly improved detection rate.

In this work, we focus on flow characterization analysis and do not consider covert channels that use address fields

for information hiding. To this end, flows extracted from the captured traffic become rows of a *flow matrix*. Every flow is represented by the vector in Equation 5 (flow_vec) for a given a sampling time T (e.g., $T = 1h$), which establishes the boundaries of the analysis. So we obtain one vector per flow and one matrix per sample interval T.

$$\text{flow\_vec}_i = \langle fset_1, \ldots, fset_j, \ldots, fset_n, pkts \rangle \quad (5)$$

where $fset_j$ is a set of features defined for the field j and $pkts$ is the total number of packets sent matching the flow i key during time T. These features are not always strictly the same for each field; the convenience of their calculation or estimation depends on the field itself and is based on the classifications proposed in Sections 3.1 and 3.3. Possible features in $fset_j$ are defined as follows:

- U: number of unique values.
- S_k: number of multimodality distribution peaks.
- w_S: relative width of the main distribution.
- S_s: number of symbols.
- Mo: the Mode value[¶].
- $p(Mo)$: number of packets with the Mode value.
- ρ_A: sum of autocorrelation coefficients.
- μ_Δ: mean of differences, where $\Delta_n = field_j[n] - field_j[n-1]$, and $field_j[n]$ represents the value of field j for a given packet n.

Given that we are considering vectors that represent flows, we do not take source address and destination address as fields to analyze; they are fixed for a given vector, which they identify. Additionally, some fields like IP padding, TCP padding, header checksum, or TCP checksum are not considered in the analysis, because they are already checked by the packet compliance checker.

5.4. Intra-field analysis

Intra-field analysis checks fields ($fset_j$) separately. A set of customized rules and thresholds are defined for every specific field according to the expected distribution characteristics,[‖] field type, and value ranges. As a general procedure, given a flow under test, intra-field analysis evaluates the existence of covert channels by performing the following steps for every field:

(1) Check U (number of unique values).
(2) Draw the distribution shape based on S_k, S_s, w_S, Mo, $p(Mo)$ and feature range.
(3) Evaluate *metrics* based on ρ_A, μ_Δ, and $pkts$. Adjust field rules and thresholds according to metrics.
(4) Compare flow indices with profiles for non-suspicious distributions (based on field rules and thresholds).
(5) Calculate C and label the flow.

5.5. Inter-field analysis

During the inter-field analysis, DAT detectors gather intra-field evaluations of different fields and evaluate them together to state a final label for every specific flow: *non-suspicious*, *liable*, or *suspicious*, respectively. The inter-field analysis works with prefixed templates of suspicious field combinations (constructed based on the documentation about covert channel techniques) as well as carries out a deeper, quick analysis by means of trained classification models.

Thus, advanced implementations of the inter-field analysis block are devised to compare complete feature vectors with patterns obtained from supervised classification analysis. To this end, pre-labeled legitimate and covert channel traffic flows undergo a flow-matrix transformation and are processed by supervised classification techniques. The results are models (or patterns) to quickly identify covert channels expressed in the proposed flow-vector format. Such models and patterns are not obtained during the online operation of the detector, but in an offline, controlled development phase (Figure 10).

6. DETECTION OF COVERT CHANNEL CLASSES

Descriptive analytics of traffic detectors make use of the scheme presented in Section 5 to face covert channels introduced in Section 3.2. Each covert channel class is differently handled.

6.1. Value to symbol correspondence

Covert channels included within the *value to symbol correspondence* class are usually detected by the intra-field analysis. Indices U, S_k, S_s, w_S, and $pkts$ are combined to draw the distribution of values and reveal the context of the field usage along the duration of the flow. The context abstraction is compared with references of expected or non-suspicious distributions, which are ultimately shaped by a set of rules and thresholds. ρ_A provides an invaluable support to understand the time evolution of field values and infer the existence of patterns or randomness. The most demanding cases (i.e., free fields) can also require μ_Δ, Mo, and $p(Mo)$ for a deeper definition.

It is important to remark that every traffic field shows a particular behavior in normal conditions and, therefore, different expected distributions. For instance, a common flow usually shows monotonically increasing IP identification values ($U = pkts = S_s, \rho_A \simeq 1$). Instead, TCP flags in

[¶] Not implemented in the proof of concept.
[‖] Specific sets of rules and thresholds are dependent on the final location of the DAT detector and the characteristics of the expected traffic through the respective border gateway (Figure 1). In any case, in order to allow reproducibility of experiments and independent testing, rules and parameters are to be provided with DAT software releases.

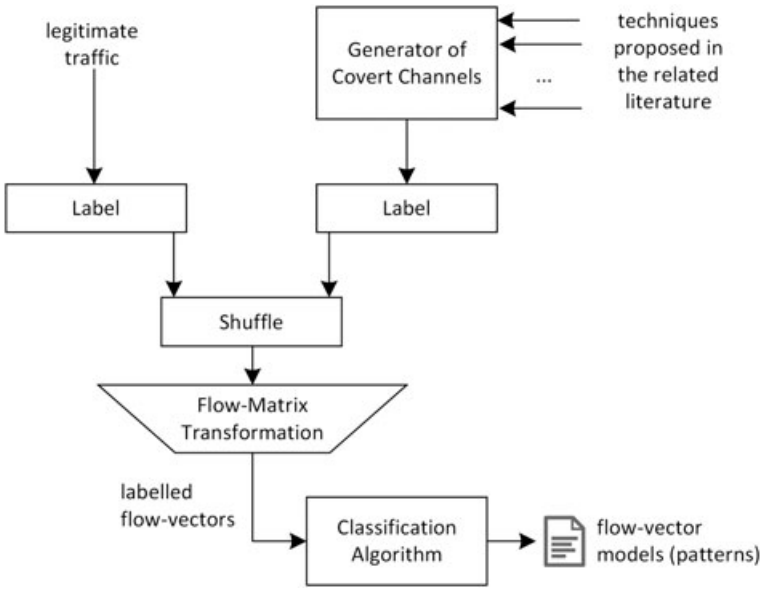

Figure 10. Scheme of the testbed for obtaining classification models (patterns).

regular conditions are expected to have few unique value and only one mode ($1 < U < 10, S_s = 1$), provided the number of packets in the flow is high enough.

6.2. Value ranges as symbols

Covert channels with *value ranges as symbols* are similarly detected to covert channels with *value to symbol correspondence*, but in this case, the number of multimodality distribution peaks S_k prevails over the number of multimodality symbols S_s. This fact does not make a difference for the detector, which simply interprets the distribution context based on the given indices. For example, $U = 2, S_k = 2, S_s = 2, pkts = 100$ would present a clear example of *value to symbol correspondence* distribution; instead, $U = 60, S_k = 2, S_s = 2, pkts = 100$ would perfectly fit a distribution for a *value ranges as symbols* case. A graphical example is shown in Figure 6.

6.3. Container fields

Some covert channels using *container fields* are early detected by the packet compliance checking module, for example, covert channels in TCP or IP padding fields, TCP RST with covert data in the *payload*, or urgent pointer distinct to 0 without the URG bit set. The intra-field analysis block is mainly addressed to check *container fields* in the IP and TCP options fields.

Packets with IP options are very rare in Internet traffic. By default, DAT detectors consider the repetition of packets with different IP options within the same flow as a preliminary suspicion. Such verification requires evaluating the size of the packets (the IHL field) and the IP options contents together. For the case of TCP options, DAT detectors differentiate between TCP packets with timestamps and those with other TCP options; the latter are dealt alike packets with IP options. Timestamp, however, is faced as an independent field, because timestamps are expected to be monotonically increasing and fake values are easy to detect. Finally, low bandwidth channels like the one introduced in [52] are well characterized by using μ_Δ values.

In addition to the TCP options field, the ISN can be used as a covert channel container. Legitimate TCP ISN algorithms generate ISN time series that show identifiable footprints when considering the sum of autocorrelation coefficients (ρ_A), the number of multimodality distribution peaks (S_k), and the relative width of the main distribution (w_S) indices together. ISN generated by algorithms that conceal covert channels (we have tested [32], [39], and variations) are easily detectable based on ρ_A, S_k, and w_S values.

6.4. Timing channels

Most timing channels can be considered as special covert channels with *value ranges as symbols* for the packet IAT field and, therefore, approached as exposed in Section 6.2.

6.5. Derivative approaches

Derivative channels are fundamentally detected by using the sum of autocorrelation coefficients ρ_A and the mean of differences μ_Δ. *A priori*, signs of a *derivative* covert channel are absence of dominant symbols (high S_s and/or $S_k = 1$ with wide width of the mean distribution w_S), high ρ_A, and specific μ_Δ value ranges. Such clues identify

triangular and saw tooth shapes in the time series of the field under test, a characteristic that *derivative* covert channels usually manifest.

7. INTRA-FIELD ANALYSIS EXAMPLES

In this section, we introduce some examples to show how the intra-field analysis of the DAT detector works. Cases include communications of legitimate traffic as well as covert channels in tied, regular, and free fields. They have been selected to show a variated combination of covert channel techniques in different types of TCP/IP fields. Vector values are displayed in Table I.

Case 1 shows the summary of data related to TCP flags for a real TCP connection. Four unique values ($U = 4$) indicate diverse packet types; most of them correspond to ACK packets ($S_s = 1$), but there are also SYN, PSH/ACK, and FIN/PSH/ACK packets. With only a single multimodality symbol ($S_s = 1$), the possibilities of a covert channel are minimum. In normal traffic, a low autocorrelation coefficient ρ_A is expected for cases with only a single mode ($S_s = 1$) and more than one unique value ($U > 1$). Nevertheless, autocorrelations are usually much higher for fields that show multimodality ($S_s > 1$ or $S_k > 1$) and do not hide covert channels (different frequent symbols are prone to be arranged in patterns).

In Case 2, a binary covert channel has been added with '0' corresponding to ACK and PSH/ACK packets and '1' for URG/ACK and URG/PSH/ACK, similar to the scenario proposed in [41] but for TCP flags. The intra-field analysis detects two clear modes and a very low autocorrelation coefficient ($\rho_A = 0.01$) and, therefore, a *medium/high suspicious* level is stated. In this case, the repeated use of the URG flag would be enough indication of an abnormal usage of a TCP connection. A legal flow showing multimodality in the TCP Flags is completely possible, but in such cases, the autocorrelation would be higher as the use of TCP flags is not chaotic but follows repetitive patterns.

A legitimate use of a regular field – Case 3 – directly shows one unique value ($U = 1$), therefore this field is considered *non-suspicious* for the given flow$_i$. For Case 4, a flow showing two different values ($U = 2$) is enough evidence to trigger a suspicion of a covert channel. Nevertheless, there are legitimate scenarios for flows with two distinct TTL values. For instance, we can imagine a Windows machine and a Linux machine on the same network, both connecting to the same external server through a network address translation (NAT) gateway and using different default TTL values. The NAT changes the source addresses of both hosts to the same source IP, therefore packets of both machines appear as if they were generated by the same source. In any case, Case 4 is generated according to the techniques proposed in [18] and shows two multimodality symbols ($S_s = 2$), a considerable number of bytes ($C = 287$) and a low autocorrelation $\rho_A = 0.06$, resulting in a *medium/high suspicious* evaluation. For a legitimate NAT case with $S_s = 2$, we would expect a considerably higher autocorrelation coefficient as it is quite unlikely that both machines connect to the same host simultaneously and just for the same time interval.

The legitimate Case 5 shows a real TCP connection to a Web server using packets with different lengths, of which most have a size of either 162 or 1438 bytes. Even in spite of $S_s = 2$, the high autocorrelation ($\rho_A = 0.7$) discards the suspicion. In Case 6, however, using exactly the same packet lengths to conceal a message, autocorrelation is much lower ($\rho_A = 0.09$) and confirms the *medium/high suspicious* assessment.

Case 7 checks the IAT values of a legitimate communication with a microsecond resolution. Case 8 uses a covert timing channel as defined in [47]. In both cases, due to the microsecond resolution, the number of multimodality main symbols is very close or the same as the number of unique values: $S_s = U = 179$ and $S_s = U = 790$, respectively; meaning that there are no prevailing symbols. Nevertheless, multimodality distribution peaks disclose monomodal (case 7, $S_k = 1$) and bimodal (case 8, $S_k = 2$) distributions. A low autocorrelation coefficient is normal for monomodal distributions, but in bimodal cases, it suggests the presence of a covert channel. The DAT detector, thus, labels Case 7 as *non-suspicious* and Case 8 as *highly suspicious*.

Cases 9, 10, and 11 show three cases with free fields; Case 9 is a legitimate communication, but Cases 10 and 11 hide covert channels (*value to symbol correspondence* and *derivative*, respectively). Case 9 shows three multimodality distribution peaks ($S_k = 3$); the high autocorrelation $\rho_A = 0.99$ reveals that the occurrence of main symbols is not overlapped. In addition, the combined assessment of U, S_k, w_S, S_s, and $p(Mo)$ show a low-sparse frequency spectrum that *a priori* does not suggest the existence of a covert channel, although it cannot be completely discarded (i.e., it is labeled *liable*).

In Case 10, the number of unique values $U = 64$ together with the amount of multimodality symbols $S_s = 13$ are strong indicators of the possible existence of a covert channel masking ASCII characters. Similarly, in Case 11 – proposed in [54], the large number of unique values and packets ($U = 4143$ and $pkts = 4349$), the high autocorrelation ($\rho_A = 0.89$), and the value of the mean of differences ($\mu_\Delta = 180.9$) denote the possibility of a *derivative* channel concealing ASCII characters. The DAT detector labels Cases 10 and 11 as *highly suspicious*.

Case 12 is a flow where ISNs are established by means of an algorithm that performs random positive increments (specifically from a RedHat Linux machine); in Case 13 a covert channel is embedded using the Nushu protocol [32]. Even in spite of being encrypted, the Nushu protocol includes some control bits that determine the distribution shape and can be detected in the flow vector. The DAT detector finds case 12 *non-suspicious* and Case 13 *medium suspicious*.

Cases 14 and 15 involve the Window field. Whereas Case 14 is a legitimate flow (*non-suspicious*), Case 15

Table I. Case examples.

Case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	TCP Flags	TCP Flags	TTL	TTL	Length	Length	IAT	IAT	Source Port	Source Port	Source Port	Seq Number	Seq Number	Wind.	Wind.
Type	Tied	Tied	Reg.	Reg.	Free	Free	Reg.	Reg.	Free	Free	Free	Reg.	Reg.	Free	Free
Cov. chan.	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	Yes	No	Yes	No	Yes
pkts	2548	2548	2012	2012	1522	1522	1552	1552	4349	4349	4349	4349	501	1633	1633
U	4	6	1	2	23	11	179	790	635	64	4143	3972	498	43	1477
S_k	1	2	1	2	2	2	1	2	3	4	1	4	1	2	7
w_S	0.6	0.2	0.7	0.3	< 0.1	0.3	< 0.1	< 0.1	0.9	0.2	0.9	0.9	0.9	0.4	0.2
S_s	1	2	1	2	2	2	179	790	139	13	4143	3972	498	4	1477
$p(Mo)$	2391	1283	2012	1113	694	771	270	169	69	730	4	6	2	487	4
ρ_A	0.03	0.01	—	0.06	0.70	0.09	< 0.01	< 0.01	0.99	0.04	0.89	0.99	0.44	0.43	0.05
μ_Δ	—	—	0	33.8	1034	1033	0.02	0.07	1.5	30.1	180.9	4.1M	29M	1.1K	4.1K
C	—	364	—	287	—	217	—	221	—	4349	4349	—	1503	—	3266

DOI: 10.1002/sec

shows some suspicious feature values. For instance, the large number of unique values and multimodality symbols ($S_s = U = 1477$) are completely unusual for normal traffic. Moreover, seven multimodality distribution peaks ($S_k = 7$) with a narrow main distribution width of $w_S = 0.2$, as well as such low autocorrelation coefficient $\rho_A = 0.05$, introduce a 3- or 4-bit covert channel with *symbols as value ranges*. Hence, the DAT detector labels Case 15 as *highly suspicious*.

8. CHALLENGES

The high variety of covert channel techniques involves some challenging situations that can be hard to diagnose by DAT detectors. We emphasize some observed cases:

- *Noisy channels*. When covert traffic coexists with normal traffic in the same flow, the detection of the covert channel becomes more difficult. Depending on the case, a DAT detector can label such cases as *low suspicious* or just *liable*. The solution implies more in depth analysis after additional filtering steps.
- *Covert channels in address fields and bouncing covert channels*. Covert channels in source and destination address fields can be detected by performing host characterization (Section 5.3). Bouncing covert channels consist of covert communications that use intermediate servers to prevent detection and tracking of the sources (e.g. [39]). Such cases also involve headers with faked addresses; in the case of TCP communications, it means obvious violations of the TCP three-way handshake. Due to our flow definition, such channels are not considered in this paper and might require additional analysis techniques.
- *Covert channels modeled according to field distribution properties*. Some covert channel techniques use the statistical characteristics of IP services to mask covert information by imitating such characteristics (e.g., [42]). In spite of the fact that these covert channel techniques cannot be applied to most of the fields, their detection requires cost-demanding approaches. DAT detectors might not be able to discover such cases, but at least can identify which flows could be exploited that way and label them as *liable*.
- *Covert channels with encrypted messages*. Covert channels that transmit ciphertext instead of plaintext are either easier or harder to detect, depending on the underlying distribution of the field values. If field values in an overt communication would look random, ciphertext can easily be hidden in such fields. Instead, if the typical value distribution follows other (deterministic or biased) patterns, the hiding of ciphertext may be revealed because it would shift the distribution to a more random shape.

In general, novel covert channels can be faced by adding new descriptive features to DAT schemes. For instance, entropy calculations [26] or detecting pseudo-random generation [25] do not entail high computational costs; hence, they are two potential enhancements to incorporate in future implementations of the DAT detector.

9. PROOF OF CONCEPT

Unfortunately there is no open, public network traffic dataset available for testing covert channel detection schemes. Therefore, we generated our own dataset to evaluate the proposed methods (and for the examples in Section 7). To this end, as baseline, we used public datasets, which to the best of our knowledge only contain overt communication. For the generation of covert channels, we implemented several approaches from the literature. The datasets are described as follows:

- A *normal traffic* dataset, which consists of real TCP traffic taken from the LBNL/ICSI Enterprise Tracing Project [60][**]. The PCAP files contain 2.313,433 and 8,340,355 packets from 473 and 825 flows (10 min and 1 h of captures, respectively). We assume that this traffic is free of covert channels, and we label its traces as 'normal'. We do not expect any covert channel to be detected – the less (false) positives the better.

- A dataset with *covert channels*, combining binary, 4-bit and 8-bit symbol channels (also two cases of *container* fields). We implemented 26 flows (319,660 packets) with covert channels based on the techniques proposed in [18,24,32,34,37–41,43,47,48,54] with both encrypted and non-encrypted data configurations. All this traffic is labeled as "covert". Unlike the previous case, we expect here to detect all covert channels — the less (false) negatives the better.

Results with a low rate of false positives and false negatives indicate the capability of the analysis methods to abstract the covert channel structures and patterns and differentiate them from normal traffic.

9.1. Prototype implementation

The prototype implementation of the DAT detector applied for evaluation does not include all proposed functionality, but a reduced set. The main characteristics compared with the presented proposal are

- *Blocks:* Only generic, coarse-grained versions of the packet compliance checking block and the intra-field analysis block have been implemented. Filtering rules are not field specific for the *intra-field analysis*; instead, in this respect, fields are split into two groups: *tied and short-regular fields* and *free and long-regular fields*.

[**] *ftp://ftp.bro-ids.org/enterprise-traces/hdr-traces05/*, files: *lbl-internal.20041004-1305.port002.dump*, and *lbl-internal.20041215-0410.port006.dump*.

Table II. Traffic datasets – results.

	Normal traffic		Covert channels	
Non-suspicious	837	(64.5%)	0	(0.0%)
Liable	429	(33.0%)	0	(0.0%)
Low-suspicious	26	(2.0%)	2	(7.7%)
Medium suspicious	6	(0.5%)	11	(42.3%)
Highly suspicious	0	(0.0%)	23	(50%)
	1298	(100.0%)	26	(100%)

- *Features:* U, S_k, S_s, $p(Mo)$, Mo, ρ_A ($A = \{1, \ldots, 20\}$) and μ_Δ (not included: w_S). The smoothing factor h for the multimodality estimation has been set according to the Silverman's 'rule of thumb' [58].
- *Fields: packet IAT, version, IHL, type of service, total length, identification, IP flags, fragment offset, TTL, protocol, source port, destination port, ISN, offset, TCP flags, Window, urgent pointer*, *TCP options*.

9.2. Results and discussion

Table II shows the results after analyzing traffic with the prototype version of the DAT detector. It is worth noting two important facts: (i) DAT detector outcomes are just estimations and (ii) no proof exists that the *normal traffic* dataset is 100% free of covert channels.

A careful checking of the cases that can be identified as *false positives* in Table II (i.e., mainly the *medium-suspicious* group of the *normal traffic* column) reveals that they are mostly triggered by fields defined as *free* in Section 3. The *covert channels* column discloses that all covert channels were detected and not a single *false negative* was obtained. Taking into account that the implemented prototype of the DAT detector is a simple version intended just for a proof of concept, results displayed in Table II are promising. The incorporation of all described functionalities and a careful field-grained definition of filtering rules and thresholds for the *intra-field analysis* and the *inter-field analysis* is expected to further improve detection rates significantly.

9.3. DAT detector performance

The DAT detector prototype was tested on a machine with the following characteristics: 8× Intel(R) Core(TM) i7-4770T CPU 2.50 GHz, 16 GB RAM, Ubuntu 12.04 LTS, kernel Ubuntu 3.13. A total of 10,973,448 packets corresponding to 1324 flows were processed and analyzed. The times required to process the PCAP files are as follows:

- The TCP/IP field extraction from PCAP files was carried out by `tshark`[††]. The field extraction process of all traffic together took 1 h, 16 min, and 36 s. Because ready-to-use (commercial) solutions exist for extracting information from TCP/IP traffic at high-speed, we did not tackle this problem specifically.
- The flow-matrix generation and the inter-field analysis processing was undertaken by scripts written in `Python`[‡‡] and supported by `R`[§§]. The processing took 19 min and 40 s. Performance optimization to the flow-matrix generation and inter-field analysis in the proof-of-concept implementation could potentially reduce processing time by an order of magnitude.
- The intra-field analysis was also implemented in `Python`; processing time was about 220 ms.

10. CONCLUSIONS

Under the name of 'DAT detectors', this paper proposes the application of descriptive analytics for the detection of covert channels in TCP/IP communication networks. The paper provides comprehensive, general-purpose, statistic-based schemes feasible for real implementations and thus fills a gap in the existing literature related to covert channel detection techniques. Analysis methodologies are combined in schemes that jointly explore field features based on descriptive statistics, aggregations, autocorrelation indices, and multimodality calculations by means of kernel density estimations and Pareto analysis.

Given the introduced flow vector representation, covert channels are expected to show characteristic behaviors, meaning that their shapes can be captured in the form of patterns. This work is built upon such assumption, which is checked by examples as well as by a proof of concept of the DAT detector prototype. The prototype was tested with a significant set of different covert channels techniques belonging to the state of the art. The perspective of the conducted detection methodology suggests effective classification schemes for TCP/IP fields as well as for existing covert channels approaches, both depicted in the current paper.

Moreover, because DAT detectors are grounded by descriptive analytics, they exhibit a high flexibility and allow the easy incorporation of new traffic features for future detection methodologies. Such enhancements can be embedded inside the analysis structures as well as applied downstream by considering the DAT detector as a first, fast filtering stage.

We are currently developing a first open-source version of the DAT software as well as a testbed for the generation of covert channels based on techniques described in the related literature. Moreover, we have already incorporated some of the introduced detection techniques in our 'Network Security Advanced Laboratory' at TU Wien [61], included in the academic program of electrical engineering students.

[††] https://www.wireshark.org/docs/man-pages/tshark.html.

[‡‡] https://www.python.org/.

[§§] http://www.r-project.org/, http://rpy.sourceforge.net/rpy2.html.

REFERENCES

1. Moulin P, Koetter R. Data-hiding codes. *Proceedings of the IEEE* 2005; **93**(12): 2083–2126.
2. Zielińska E, Mazurczyk W, Szczypiorski K. Trends in steganography. *Communication ACM* 2014; **57**(3): 86–95.
3. Mazurczyk W, Caviglione L. Information hiding as a challenge for malware detection. *Security Privacy, IEEE* 2015; **13**(2): 89–93.
4. Schulz S, Varadharajan V, Sadeghi AR. The silence of the LANs: efficient leakage resilience for IPsec VPNs. *Information Forensics and Security, IEEE Transactions on* 2014; **9**(2): 221–232.
5. Zadeh L A. Fuzzy logic, neural networks, and soft computing. *Communication ACM* 1994; **37**(3): 77–84.
6. Dainotti A, Pescape A, Claffy K. Issues and future directions in traffic classification. *Network, IEEE* 2012; **26**(1): 35–40.
7. Wu DLWYMANKSU J, Wang Y. C2detector: a covert channel detection framework in cloud computing. *Security Communication Networks* 2014; **7**: 544–557.
8. Kerckhoffs A. La cryptographie militaire. *Journal des Sciences Militaires.* 1883; **9**: 5–38.
9. Rezaei FHM, H S. A novel automated framework for modeling and evaluating covert channel algorithms. *Mathematical Methods in the Applied Sciences* 2015; **38**: 649–660.
10. RFC 7011 – Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. *Technical Report*, Internet Engineering Task Force (IETF), 2013. https://www.ietf.org/rfc/rfc7011.txt.
11. RFC 791 - Internet protocol, *Technical Report* Information Sciences Institute, University of Southern California, 1981. http://tools.ietf.org/html/rfc791.
12. RFC 793 - Transmission control protocol, *Technical Report* Information Sciences Institute, University of Southern California, 1981. http://tools.ietf.org/html/rfc791.
13. Mazurczyk W, Wendzel S, Zander S, Houmansadr A, Szczypiorski K. *Information Hiding in Communication Networks: Fundamentals, Mechanisms, and Applications*. Wiley-IEEE Press, 2015.
14. Shen J, Qing S, Shen Q, Li L. Optimization of covert channel identification. *Security in Storage Workshop, 2005. SISW '05. Third IEEE International*, 2005; 13–95.
15. Zhiyong C, Yong Z. Entropy based taxonomy of network convert channels, Power Electronics and Intelligent Transportation System (PEITS), 2009 2nd International Conference on, 2009Dec; 451–455.
16. Zander S, Armitage G, Branch P. A survey of covert channels and countermeasures in computer network protocols. *Commun. Surveys Tuts.* 2007; **9**(3): 44–57.
17. Wendzel S, Zander S. Detecting protocol switching covert channels, Local Computer Networks (LCN), 2012 IEEE 37th Conference on, 2012; 280–283.
18. Zander S, Armitage G, Branch P. An empirical evaluation of IP time to live covert channels. *Networks, 2007. ICON 2007. 15th IEEE International Conference on*, 2007; 42–47.
19. Zander S, Armitage G. CCHEF – covert channels evaluation framework design and implementation. *080530A*, CAIA, Centre for Advanced Internet Architectures, Swinburne University of Technology, 2008.
20. Wendzel S, Zander S, Fechner B, Herdin C. Pattern-Based Survey and Categorization of Network Covert Channel Techniques. *ACM Computing Surveys* 2015; **47**(3): 26 DOI: 10.1145/2684195.
21. Fincher S, Finlay J, Greene S, Jones L, Matchen P, Thomas J, Molina P J. Perspectives on HCI patterns: Concepts and tools. In *Chi '03 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '03, ACM: New York, NY, USA, 2003; 1044–1045.
22. Murdoch S J, Lewis S. Embedding covert channels into TCP/IP. In *Proceedings of the 7th International Conference on Information Hiding*, IH'05. Springer-Verlag, 2005; 247–261.
23. Handley M, Paxson V, Kreibich C. Network intrusion detection: evasion, traffic normalization, and end-to-end protocol semantics. In *Proceedings of the 10th Conference on USENIX Security Symposium – Volume 10*, SSYM'01. USENIX Association: Berkeley, CA, USA, 2001; 1–18.
24. Fisk G, Fisk M, Papadopoulos C, Neil J. Eliminating steganography in Internet traffic with active wardens. *Revised Papers from the 5th International Workshop on Information Hiding*, IH '02, 2003.
25. Zhao H, Shi YQ. Detecting covert channels in computer networks based on chaos theory. *Information Forensics and Security, IEEE Transactions on* 2013; **8**(2): 273–282.
26. Gianvecchio S, Wang H. An entropy-based approach to detecting covert timing channels. *Dependable and Secure Computing, IEEE Transactions on* 2011; **8**(6): 785–797.
27. Cabuk S, Brodley C E., Shields C. IP covert channel detection. *ACM Transactions on Information and System Security* 2009; **12**(4): 22:1–22:29.
28. Crespi V, Cybenko G, Giani A. Engineering statistical behaviors for attacking and defending covert channels. *Selected Topics in Signal Processing, IEEE Journal of* 2013; **7**(1): 124–136.

29. Shrestha P, Hempel M, Rezaei F, Sharif H. Leveraging statistical feature points for generalized detection of covert timing channels. *Military Communications Conference (milcom), 2014 IEEE*, 2014; 7–11.
30. Shen HLL XY, Yang W. A novel comprehensive steganalysis of transmission control protocol/Internet protocol covert channels based on protocol behaviors and support vector machine. *Security Communication Networks* 2015; **8**: 1279–1290.
31. Sohn T, Seo J, Moon J. A study on the covert channel detection of TCP/IP header using support vector machine. In *Information and Communications Security, Lecture Notes in Computer Science*, Vol. 2836, Qing S, Gollmann D, Zhou J (eds). Springer Berlin: Heidelberg, 2003; 313–324.
32. Rutkowska J. The implementation of passive covert channels in the Linux kernel, 2004.
33. Zander S, Armitage G, Branch P. Stealthier inter-packet timing covert channels. In *Networking 2011, Lecture Notes in Computer Science*, Vol. 6640, Domingo-Pascual J, Manzoni P, Palazzo S, Pont A, Scoglio C (eds). Springer Berlin: Heidelberg, 2011; 458–470.
34. Ahsan K, Kundur D. Practical data hiding in TCP/IP. In *Proc. Workshop on Multimedia Security at ACM Multimedia*, (Vol. 2, No. 7). 2002.
35. Nair A, Kumar A, Sur A, Nandi S. Length based network steganography using UDP protocol. *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011; 726–730.
36. Ji L, Jiang W, Dai B, Niu X. A novel covert channel based on length of messages. *Information Engineering and Electronic Commerce, 2009. IEEC '09. International Symposium on*, 2009; 551–554.
37. Qu H, Su P, Feng D. A typical noisy covert channel in the IP protocol. *Security technology, 2004. 38th Annual 2004 International Carnahan Conference on*, 2004; 189–192.
38. Girling C. Covert channels in LAN's. *IEEE Transactions on Software Engineering* 1987; **13** (2): 292–296.
39. Rowland C H. Covert channels in the TCP/IP protocol suite. *First Monday* 1997; **2**(5). http://firstmonday.org/ojs/index.php/fm/article/view/528/449.
40. Handel T G, Sandford M T II. Hiding data in the OSI network model. In *Proceedings of The First International Workshop on Information Hiding*. Springer-Verlag, 1996; 23–38.
41. Xu B, Wang Jz, Peng Dy. Practical protocol steganography: hiding data in IP header. *Modelling Simulation, 2007. AMS '07. First Asia International Conference on*, 2007; 584–588.
42. Zhang L, Liu G, Dai Y. Network packet length covert channel based on empirical distribution function. *Journal of Networks* 2014; **9**(6): 1440–1446.
43. Lucena N, Lewandowski G, Chapin S. Covert channels in IPv6. In *Privacy Enhancing Technologies, Lecture Notes in Computer Science*, Vol. 3856, Danezis G, Martin D (eds). Springer Berlin: Heidelberg, 2006; 147–166.
44. Trabelsi Z, El-Sayed H, Frikha L, Rabie T. Traceroute based IP channel for sending hidden short messages. *Proceedings of the 1st International Conference on Security*, Springer-Verlag, 2006; 421–436.
45. Arkin O, Anderson J. EtherLeak: Ethernet frame padding information leakage. *Technical Report*, @stake Inc., 2003.
46. Abad C. IP checksum covert channels and selected hash collision. *Technical Report*, UCLA, 2001.
47. Cabuk S, Brodley C E, Shields C. IP covert timing channels: design and detection. *ACM Conference on Computer and Communications Security*, ACM, New York, NY, USA, Washington, DC, USA, 2004; 178–187.
48. Berk V, Giani A, Cybenko G. Detection of covert channel encoding in network packet delays. *TR2005-536*, Dartmouth College, 2005.
49. Smith R, Knight S. Predictable three-parameter design of network covert communication systems. *Information Forensics and Security, IEEE Transactions on* 2011; **6**(1): 1–13.
50. Shah G, Molina A, Blaze M. Keyboards and covert channels. *Proceedings of the 15th Conference on Usenix Security Symposium – Volume 15*, 2006; 59–75.
51. Gianvecchio S, Wang H, Wijesekera D, Jajodia S. Model-based covert timing channels: automated modeling and evasion. *Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection*, 2008; 211–230.
52. Giffin J, Greenstadt R, Litwack P, Tibbetts R. Covert messaging through TCP timestamps. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, PET'02. Springer-Verlag: San Francisco, CA, USA, 2003; 194–208.
53. Mazurczyk W, Szczypiorski K. Steganography of VoIP streams. In *On the Move to Meaningful Internet Systems: OTM 2008, Lecture Notes in Computer Science*, Vol. 5332, Meersman R, Tari Z (eds). Springer Berlin, 2008; 1001–1018.
54. Gimbi J, Johnson D, Lutz P, Yuan B. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). In *Proceedings of the International Conference on Security and Management (SAM)*, Las Vegas, USA, 2012; 1–5.

55. Mller B. A public-key encryption scheme with pseudo-random ciphertexts. In *Computer Security ESORICS 2004*, Samarati P, Ryan P, Gollmann D, Molva R (eds), Lecture Notes in Compute Science. Springer Berlin: Heidelberg, 2004; 335–351.
56. Bernstein D J, Hamburg M, Krasnova A, Lange T. Elligator: elliptic-curve points indistinguishable from uniform random strings. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13. ACM: New York, NY, USA, 2013; 967–980.
57. Silverman B W. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society: Series B* 1981; **43**(1): 97–99.
58. Silverman B W. *Density Estimation*. Chapman and Hall: London, 1986. eqn (3.31).
59. Newman M. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 2005; **46** (5): 323–351.
60. Lawrence Berkeley National Laboratory and ICSI. *The LBNL/ICSI Enterprise Tracing Project*, 2013 (last published). http://www.icir.org/enterprise-tracing/.
61. Zseby T, Iglesias F, Bernhardt V, Frkat D, Annessi R. A network steganography lab on detecting TCP/IP covert channels. *IEEE Transactions on Education* 2016; **59**(4): 1–9.