Network Flow Watermarking: A Survey

Alfonso Iacovazzi and Yuval Elovici, *Member, IEEE*

***Abstract*—Traffic analysis (TA) is a useful tool aimed at understanding network traffic behavior. Basic network administration often takes advantage of TA for purposes such as security, intrusion detection, traffic shaping and policing, diagnostic monitoring, provisioning, and resource management. Network flow watermarking is a type of TA in which packet features of selected flows are manipulated in order to add a specific pattern easily identifiable when the watermarked flows cross an observation point. While passive TA has been extensively studied with hundreds of papers found in the literature, active TA, and more specifically network flow watermarking, has only recently attracted attention. Enforced robustness against traffic perturbations due to either natural network noise or attacks against passive TA have enhanced the appeal of this technique. The contribution of this paper is a thorough review of the main watermarking algorithms implemented for traffic analysis purposes. We present an overview of the motivations and the objectives that have led to the use of network flow watermarking. We also describe the general architecture of a watermarking system. In addition, we impose clarity and order in this branch of TA by providing a taxonomy of the algorithms proposed in the literature over the years, and categorize and present them based on carrier, visibility, and robustness.**

***Index Terms*—Watermarking, traffic analysis, traffic flow, botnet, traceback.**

I. Introduction

Watermarking is a well-known process of embedding proprietary information in digital content, such as images, audio, or video. According to the definition provided by Cox *et al.* [1], a watermark is an "identification code that is permanently embedded in the data and remains present within the data after any decryption process".

Tirkel *et al.* [2] were the first to introduce the term "digital watermarking" (and "electronic watermarking") in 1993, referring to their algorithms designed to hide data in digital images. Since then, watermarks have been widely used for purposes of security which include data hiding, copyright protection, data authentication, copy prevention, and rightful ownership protection of digital media [3]–[5].

The evolving use of digital watermarking has prompted the research community to adapt this technique to new unexplored contexts with different goals. Examples include: "software watermarking" (embedding a unique identifier within a piece of software) [6]–[8]; "spatial data watermarking" (protecting spatial data ownership in geographic information systems) [9], [10]; "text watermarking" (text document copyright protection) [11], [12]; "design watermarking" (permanently embedding information within a physical design) [13], [14]; and "human electrocardiogram (ECG) watermarking" (signal integrity verification) [15], [16].

In this paper we discuss the watermarking process in another context – Internet "traffic analysis" (TA). As the popularity of the Internet has grown over the last two decades, the role of TA has become more important. Traffic analysis focuses on the development of procedures, algorithms, and strategies in order to monitor, evaluate, control, and manage Internet network traffic (for more information see the survey works [17], [18]). In depth investigation of Internet traffic in computer networks is a critical task which allows analysts to determine and understand the causes and implications of network behavior. Research has shown that it is possible to gather useful information by inspecting and passively analyzing Internet traffic flows (passive TA). For example, by analyzing statistical features of flow packets (e.g., packet lengths, interpacket gaps, packet directions, etc.) and exploiting advanced machine learning algorithms, it is possible to define the type of application protocol in a given set of possible choices [19]–[22], predict the user's contextual location [23], detect abnormal traffic [24], and distinguish malicious traffic flows [25].

Furthermore, passive TA can also be exploited by an attacker in order to infer private or sensitive information about the user's communication. Examples of this type of confidentiality breach include: 1) recognizing the downloaded Web pages despite encrypted and authenticated communications [26]; 2) identification of the conversation language in encrypted VoIP transactions [27]; and 3) obtaining partial transcripts of an encrypted VoIP conversation [28].

Passive TA has three main disadvantages: 1) it requires the use of complex machine learning algorithms which usually do not achieve an optimal balance between scalability and accuracy; 2) a significant amount of sample flows are needed in advance to train the machine learning algorithms; and 3) it is vulnerable to traffic perturbation, either due to normal network behavior or malicious manipulation of the traffic by an adversary.

Algorithms for active TA have been proposed in this field in order to address some of the drawbacks described above. Network flow watermarking is a type of active TA in which a watermark is embedded into selected flows in order to recognise that flows at specific points in the network.

Manuscript received March 11, 2016; revised July 8, 2016; accepted August 12, 2016. Date of publication September 16, 2016; date of current version February 22, 2017.

A. Iacovazzi is with the Temasek Laboratories, Singapore University of Technology and Design, Singapore (e-mail: alfonso_iacovazzi@sutd.edu.sg).

Y. Elovici is with the Cyber Security Research Center, Ben-Gurion University of the Negev, Beer-Sheva 8410501, Israel, and also with the iTrust—Centre for Research in Cyber Security, Singapore University of Technology and Design, Singapore (e-mail: yuval_elovici@sutd.edu.sg).

Digital Object Identifier 10.1109/COMST.2016.2604405

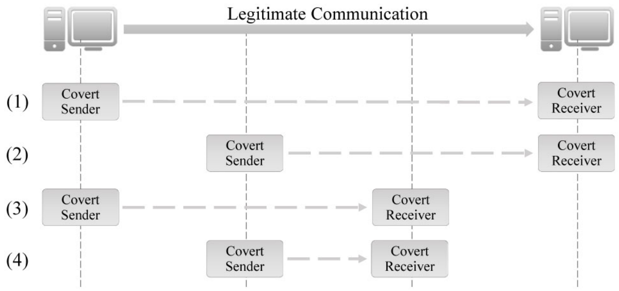

Fig. 1. Four possible covert channel scenarios.

In order to provide a formal definition of network flow watermarking, the concept of the covert channel must be introduced. While covert channels have been discussed for decades, the term was first defined by Lampson [29] in 1973, who described them as "channels not intended for information transfer at all, such as the service program's effect on the system load". A later and more detailed definition was introduced by the U.S. Department of Defense in which a covert channel is defined as "any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy" [30]. In contrast, Zander *et al.* [31] narrow it further and identify the subclass "network covert channel" which is defined as "the hiding of information in network protocols". Zander *et al.* [31] provide an overview of existing covert channels in computer network protocols and distinguish four possible communication scenarios, depending on whether the sender and/or receiver of the communication being delivered by the covert channel are also the sender and/or receiver of the communication being delivered by the legitimate channel. A legitimate system for transferring information is often referred to as an "overt channel." The four communication scenarios described by Zander *et al.* [31] are presented in Figure 1. In the first scenario, both the sender and receiver of the legitimate communication are also the sender and receiver of the covert communication. In the second scenario, the legitimate and covert receiver are the same entity, while the covert sender is different from the legitimate sender, and the latter is unaware that his communication is being exploited as a covert channel by a third party. In contrast, in the third scenario, the legitimate and covert senders are the same, while the two receivers are different. In the last scenario, the two covert channel endpoints are completely distinct from the legitimate endpoints.

Network flow watermarking is a specific case of a covert channel which falls under the last communication scenario presented in Figure 1. The term watermarking was used, rather than covert channel, for the first time in 2001 by Wang *et al.* [32], to address a specific problem in TA, and since that time a watermark is referred to as "a small piece of information that can be used to uniquely identify a connection". Our use of the term watermarking (or network flow watermarking) in this article, is based on Wang's definition.

To adopt a strategy of network flow watermarking, the analyzer should be able to actively and conscientiously alter, near the source, some statistical features of selected flows so that a specific pattern (the watermark) can be embedded in the flows. Thanks to the presence of the watermark, the target flows will be easily identified at any observation point.

Recently, Lu *et al.* [33] and Zhang *et al.* [34] have proposed two different surveys on this topic. The former is a short description of the main watermarking algorithms in the literature which are categorized by class (based on watermark carrier) as payload-based, rate-based, and timing-based algorithms. In addition, the authors describe three possible attacks: timing analysis attacks, multi-flow attacks, and mean-square autocorrelation attacks. The latter survey provides an additional glimpse of this field by classifying the algorithms as interpacket delay-based, interval-based, and interval centroid-based watermarking algorithms. It also provides details regarding three other attacks (timing analysis attacks, multi-flow attacks, and chosen flow attacks). Unfortunately, the latter review is available only in Chinese and thus not easily accessible to all.

In this paper we aim to collect and present all of the literature currently available in this specialized field. Unlike the two surveys cited above, our review provides an extensive and in-depth analysis of the topic. More specifically, this study provides the following contributions: First, we outline the primary objectives of network flow watermarking. Second, we provide a detailed description of the general architecture of a TA watermarking system and present a thorough taxonomy of the existing watermarking algorithms. Lastly, we examine possible attacks targeting the watermarking's key properties.

The rest of the paper is organized as follows. An overview of the main objectives of watermarking in traffic analysis is provided in Section II. Section III outlines the watermarking system architecture and provides details about each component. The generic attack scenario which incorporates the attacks against watermarking algorithms is described in Section IV. The invisibility and robustness vulnerabilities are detailed in Sections IV-A and IV-B, respectively. The performance evaluation process is reported in Section V. Next, in Section VI, we offer future research directions, and our conclusions are presented in Section VII.

II. Objectives of Watermarking in Traffic Analysis

Identifying the same traffic flow in two or more different observation sites of the Internet is the main goal of network flow watermarking. This particular objective is often referred to as "network flow identification," "flow linking," "flow correlation," or "flow trace back," depending on the reference scenario and context. Often such identification results in determining the route (partially or in its entirety) of a flow within the network, from the source to the destination node.

Most of the works in the literature focus on broad scenarios, rather than analyzing specific or limited problems, so that the proposed solutions are more widely applicable. For example, some researchers discuss the generic problem of linking/correlating flows [35], while others refer to breaking the anonymity of the network [36]–[38].

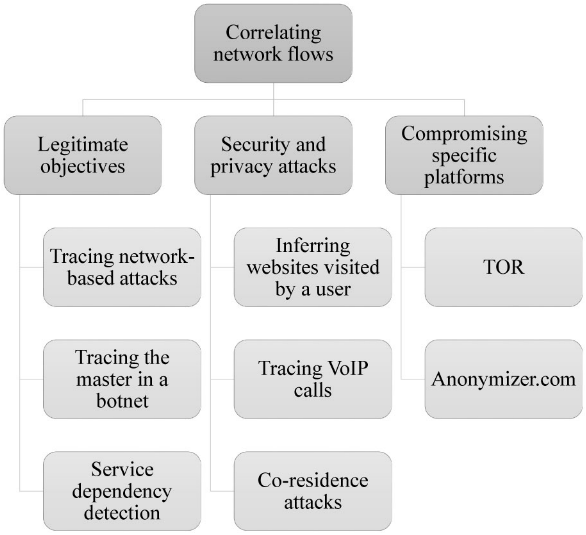
Correlating network flows
Legitimate objectives
Security and privacy attacks
Compromising specific platforms
Tracing network-based attacks
Tracing the master in a botnet
Service dependency detection
Inferring websites visited by a user
Tracing VoIP calls
Co-residence attacks
TOR
Anonymizer.com

Fig. 2. Objectives of network flow watermarking.

On the other hand, other research works address specific and well-defined objectives and scenarios, so as to easily take advantage of distinguishing their features and peculiarities. This type of research can be divided into three categories based on the goals of the attack. The first category contains legitimate goals, that is those aimed at thwarting network cyber attacks (see Section II-A). The second consists of the goals pursued by a malicious entity that wants to infer private information and violate user privacy (see Section II-B). The last category includes the objectives that aim to compromise specific services and platforms, which cannot be considered legitimate or malicious (see Section II-C).

Figure 2 includes all of the specific motivations discussed in the following subsections, however, the boundaries between these categories can become blurred, since objectives often overlap. In addition, the objectives for the main works analyzed are listed in the first column of Table I, (Section III).

A. *Legitimate Objectives*

Legitimate objectives refer to all the pursued goals that are aimed at improving services and performance, or opposing malicious/illegal behavior in the context of computer networks. In this subsection we briefly analyze the use of network flow watermarking to trace network-based attacks, an area which researchers have focused on. We also consider the specific case of tracing botnet communications, and finally we address a more unusual scenario in traffic flow analysis – service dependency detection.

1) Tracing Network-Based Attacks: Network-based attacks refer to a vast group of security and privacy attacks performed by device(s) different from those under attack. Eavesdropping, denial-of-service attacks, man-in-the-middle attacks, data modification, and identity spoofing are just some of the threats in this category. Many of these attacks require an exchange of messages between the victim and the attacker, without the knowledge of the former and under the control of the latter. Because the attacker does not want to reveal its identity or position, it usually creates its own anonymous communication channel to the victim's computers by designing a route that makes use of stepping stones managed by the attacker or it makes use of an anonymous service already available on the Internet (see Section II-C). In this context, researchers have found that by embedding a watermark in some (modifiable) features of traffic flow, the watermark can be resistant and remain observable despite the attacker's use of stepping stones or anonymous networks. For this reason watermarking was found useful to track the traffic flow from the victims to the attacker. Tracking the path of a flow facilitates the discovery of the IP address of the final endpoint of the malicious communication, identification of the attacker, and eventually, the ability to take reactive decisions against it. Wang *et al.* [32] were the first to propose embedding watermarks in traffic flows, in 2001. Since then, many papers have focused on tracing network-based attacks with the aim of seeking the best watermarking strategy in terms of efficiency, robustness, and invisibility (many of these works are analyzed in greater detail in later sections).

2) Tracing the Master in a Botnet: A botnet is a network of devices (bots) connected to the Internet which are infected by a malware developed, managed, and coordinated by a single entity, namely a botmaster [39]. Bots can be exploited to perform different illegal behaviors including, for example, DDoS, spam, and phishing, depending on the attacker's intentions, and can be considered a specific type of network-based attack (previously described). Like any other kind of network-based attacker, the botmaster wants to keep its identity unknown. Although botnets have been around for a long time, with the passage of time, hackers have increasingly created more sophisticated botnets that are able to take advantage of the most recent technologies. For this reason, researchers have invested a lot of time in developing strategies to locate and neutralize both bots and botmasters (currently a number of review articles are available [40]–[43]). Most of the literature is aimed at detecting the bots, while fewer researchers have focused on botmaster detection [44], [45].

The botmaster tracing problem might be considered a sub-category of intrusion tracing (see Section II-A1), but it is often tackled separately due to a number of specific characteristics. For example, the watermarking framework Botmosaic takes advantage of the distributed architecture of botnets [45].

3) Service Dependency Detection: The modular organization of network services has resulted in functionality, security, and reliability, and it has also made isolating the root of a problem difficult when misbehavior occurs. Complex relationships increasingly exist among services available on the Internet (e.g., DNS, Web server, load balancing, etc.) and among the infrastructure supporting these services. For this reason, an instrument able to identify the dependency among network services in a complex platform can provide significant help. Although a somewhat unusual context for the application of TA strategies, Zand *et al.* [46] proposed watermarking as a means to be exploited to detect the dependency that exists among application services that interact with one another.

This issue has been tackled primarily through passive analysis [47]–[52], although recently active strategies have also been proposed [46], [53], [54].

B. Security and Privacy Attacks

Even if network flow watermarking has been designed with a legitimate purpose, it can be used to attempt to breach privacy or security on the Internet. Research groups have explored some of these potential threats which are briefly described in the following subsections.

1) Inferring Websites Visited by a User: An attacker that wants to breach a user's privacy and obtain sensitive information might be interested in knowing which Web page the user visits. Up until a few years ago, only a few Web servers ciphered their communications with clients, thus anyone that had access to the unencrypted traffic was able to easily discover the Web page visited by a user and obtain sensitive information. Nowadays, almost all the Websites use encryption techniques, and therefore it is no longer possible to read the contents of Web communications; nevertheless, as shown by several works, it is still possible to determine the Websites visited by a user with a high degree of accuracy [26], [55], [56]. This can be done through a passive analysis of statistical features of the Internet traffic if an adversary knows the characteristics of the Web pages the user might visit. In addition, one can often find the accessed Website by simply looking at the destination IP addresses of a flow, and even though the complete URLs cannot be read when HTTPS connections are used, URL information can be eavesdropped by taking advantage of the server name indication (SNI) extension of the TLS protocol [57].

Despite the advent of counter-measures taken against passive TA, some research groups have demonstrated how to infer Web pages by exploiting watermarking strategies [37], [58].

2) Tracing VoIP Calls: Currently, VoIP services are widely used, and one of the most studied threats associated with VoIP services is the possibility for an attacker to find out whether a target user A is having a phone conversation with user B at a specific time. Although a significant amount of research has been conducted on how to passively perform some types of privacy attacks [28], [59], the statistical features of VoIP flows are not distinct enough to do so, because the codec used and its packetization methods fully define interpacket delays and packet sizes of VoIP traffic, which are the main features used in TA. For this reason, it is not possible to distinguish among different calls based on passive comparison of the original traffic features of VoIP flows. Nevertheless, some research groups have pointed out that watermarking can be usefully adopted for this privacy threat [60], [61].

3) Co-Residence Attacks: The co-residence threats have been observed with the advent of infrastructure as a service (IaaS), a service which is increasingly being offered by leading IT vendors (Amazon Web Services, Windows Azure, Google Compute Engine, Rackspace Open Cloud, and IBM SmartCloud Enterprise). Offering more efficient and highly flexible services has led vendors to place multiple virtual machines (VMs) on the same physical machine even if leased by different costumers. A malicious party can legitimately run and manage some VM instances in the cloud just like a legitimate service costumer, enabling the attacker to infer information regarding others residing on the same physical machine. This kind of attack is referred to as a co-residence threat, and it was analyzed for the first time by Ristenpart *et al.* [62], in 2009. The possibility of exploiting watermarking strategies for co-residence attacks was considered in two works by Bates *et al.* [63], [64]. In their work, the authors show a method in which several VMs are launched in the cloud so that some artificial traffic leaves the cloud machine, and due to the sharing of the physical interface, the legitimate traffic of co-resident VMs suffers a delay; if the artificial traffic is conveniently created, the delay imposed on the other traffic can be modelled as a watermark which can be identified at a different point in the network.

C. Compromising Specific Platforms

TOR, Crowds, Anonymizer.com, DC-net, GAP, and Hordes are some of the technological platforms developed to provide anonymous and private communication [65]–[70]. Even though they are all potentially vulnerable to both active and passive TA, only TOR and Anonymizer.com have been tested under watermarking attacks.

1) TOR: TOR is the most famous system for anonymous communication based on the second generation of the onion routing paradigm [65]. Its purpose is to protect the privacy and confidentiality of a user's communications by means of a worldwide network of TOR routers managed by and belonging to volunteer users. TOR's security and reliability have been extensively studied due to its popularity, and several methods have been proposed to compromise its anonymity [71]–[74]. In addition, recent works have evaluated the potentiality of watermarking strategies applied to trace flow passing through the TOR network [75]–[79].

2) Anonymizer.com: www.anonymizer.com is one of the leading platforms offering anonymity services on the Internet [67]. It provides an encrypted virtual private network (VPN), flowing through multiple proxies, from the user's device to one of Anonymizer.com's secure servers, allowing the source IP address to be masked so that the user can anonymously surf on the Internet.

Even though many of the previously cited strategies using watermarks could also be used to breach the anonymity of Anonymizer.com, only a couple of studies have tested proposed algorithms on this specific platform to assess its vulnerability [37], [58].

III. Watermarking Framework Architecture

Network flow watermarking techniques are realized by means of two main parts: a watermarker and watermark detector. The placement of these two parts is a design choice made based on the objective pursued and where one would expect to observe the target flows. The watermarker is responsible for converting the information to a watermark code with some specific proprieties and embedding it into the target flow. In contrast, the watermark detector observes the traffic crossing

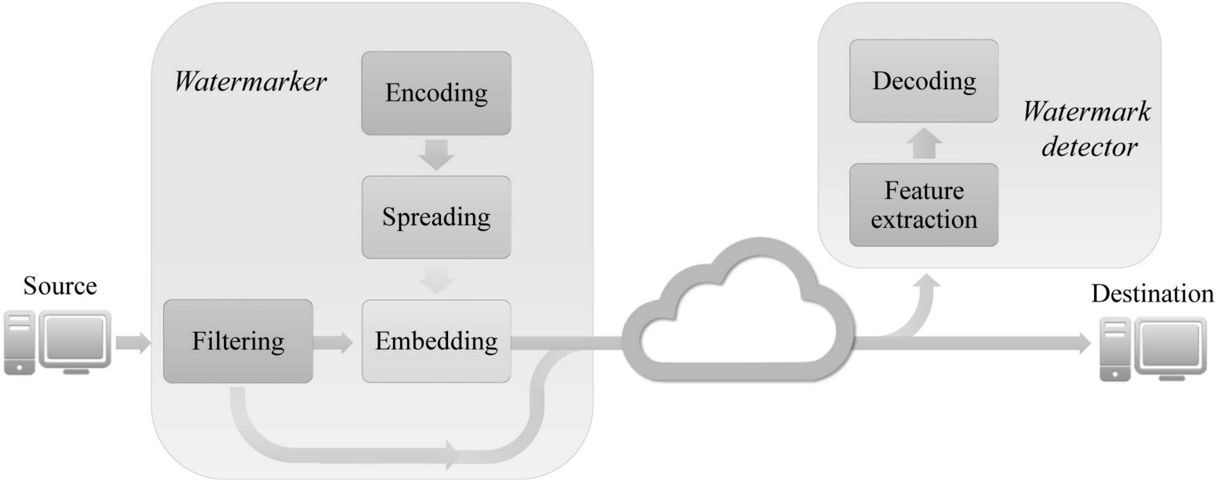

Fig. 3. Watermarking block scheme.

at a specific point in the network and analyzes features of the traffic in order to detect watermarked flows and decode the watermark and thereby obtain any information embedded in it. Figure 3 describes the primary steps of the two parts. Detailed information about the operations conducted by the watermarker and the detector are provided in the following Sections III-A and III-B.

A. Watermarker

We consider a flow as a uni-directional and ordered sequence of messages sent from a source to a destination and flowing through a network. The messages generated by the source at the application layer can be fragmented/aggregated and ciphered before being encapsulated in an IP packet flow according to the Internet protocol suite. Often a passive observer cannot read messages at the application layer because of encryption, and for this reason, most of the literature takes flows at network protocol layer into account. An IP-level flow is a uni-directional and ordered sequence of IP packets identified with the same value of five IP attributes: IP source address, IP destination address, source port, destination port, and IP protocol field. It is important to note that the same application flow can be carried on different IP-level flows at different points in the network due to proxying. Hereafter, two or more IP flows carrying the same application flow are considered to be the same flow.

Following the diagram in Figure 3, we describe the four operations conducted by the watermarker: filtering, encoding, spreading, and embedding. Encoding and spreading can be handled in an off-line mode, while filtering and embedding can be performed only online and in real-time.

1) Filtering: Before the embedding procedure, the watermarker filters the traffic by selecting the target flows that will be watermarked; the non-selected flows will be directly sent towards the network.

2) Encoding: The watermarking system can be viewed as a method to transfer information on a particular channel. For this reason, as in any traditional communication system, the information must be codified and optimally modified to make it suitable for the method. Based on the objectives discussed in Section II, in most cases, information to be conveyed can be mapped into an alphabet with only two symbols ($\mathcal{B} = \{b_0, b_1\}$), as the identifier must be able to distinguish whether an observed flow is watermarked or not.

As described by Houmansadr, in some contexts the system wants to convey one symbol a belonging to a finite alphabet $\mathcal{A} = \{a_0, a_1, \ldots, a_{W-1}\}$ where W is the number of symbols ($W = |\mathcal{A}|$) [80]. Based on the information theory it is known that we can define a code to map every symbol in $\mathcal{A}$ in a sequence on L bits, with $L = \lceil \log_2 W \rceil$. Therefore, this can be considered a generalization of the single watermark bit, and it is sometimes referred to as flow fingerprinting. Hereafter, without loss of generality, we only consider single watermark bits.

We distinguish between two policies to convey a single informational bit $b \in \{b_0, b_1\}$ in a flow:

- (1)-based
- (0/1)-based.

In (1)-based watermarking, the features of the target flow are modified when a bit $b = b_1$ must be embedded, otherwise the flow features remain unchanged. In contrast, in (0/1)-based watermarking, the flow features are always conveniently modified by the watermarker, so that the two symbols are distinguishable (see Section III-A4 for more details).

3) Spreading: Like all communication channels, the channel carrying watermark bits may also be noisy, and the interference added to the carrier signal may destroy the watermark. For this reason, researchers have proposed "diversity schemes" to improve the robustness of watermarking systems and the reliability of the embedded watermarks. Basically, diversity schemes allow for the spread of the original signal in a specific domain. As in traditional telecommunication systems, we consider three different classes of spreading, also called diversity schemes:

- Time diversity
- Frequency diversity
- Space diversity.

TABLE I
OBJECTIVE, DIVERSITY, CARRIER, AND BLINDNESS FOR THE MAIN ALGORITHMS PROPOSED IN THE LITERATURE. OBJECTIVE: CORRELATING NETWORK FLOWS (CNF), INFERRING WEBSITES VISITED BY A USER (WSITE), TRACING VOIP (VOIP), CO-RESIDENCE ATTACKS (CORES), TRACING ATTACK SOURCE (AS), TRACING NETWORK-BASED INTRUSION (NBI), TRACING THE MASTER IN A BOTNET (BOTNET), SERVICE DEPENDENCY DETECTION (SDD), AND ANONYMIZER.COM (ANON). DIVERSITY: TIME, FREQUENCY, AND SPACE. CARRIER: CONTENT-BASED (CONTENT), SIZE-BASED (SIZE), RATE-BASED (RATE), SIMPLE DETERMINISTIC DELAY (DETDELAY), SIMPLE PROBABILISTIC DELAY (PROBDELAY), {IPD, CENTROID, COUNTING}-BASED MEAN BALANCING (MB). "–" IS INSERTED WHEN THE ALGORITHM CANNOT BE CLASSIFIED

	Objective	Diversity	Carrier	Blindness
Wang, 2001 (SLEEPY) [32]	NBI	–	Content	–
Wang, 2003 [81]	NBI	Time	DetDelay	blind
Peng, 2005 [91]	CNF	Time	MB – IPD	blind
Wang, 2005 [60]	VoIP	Time	MB – IPD	blind
Wang, 2007 [37]	Wsite	Time	MB – Centroid	blind
Yu, 2007 [75]	NBI	Freq	Rate	blind
Pyun, 2007 [92]	NBI	Time	MB – Counting	blind
Ramsbrock, 2008 [44]	Botnet	Time	Size	blind
Houmansadr, 2009 (RAINBOW) [83]	NBI	Time	ProbDelay	non-blind
Houmansadr, 2009 [93]	CNF	Time	MB – Centroid	non-blind
Deng, 2009 [89]	AS	Time	MB – Counting	blind
Houmansadr, 2011 (SWIRL) [94]	AS	Time	MB – Centroid	non-blind
Wang, 2011 [95]	AS	Time	DetDelay	blind
Bates, 2012 [63]	CoRes	Time	ProbDelay	non-blind
Houmansadr, 2012 (BOTMOSAIC) [45]	Botnet	Space	MB – Counting	blind
Yu, 2013 [96]	AS	Time	–	blind
Ling, 2013 [58]	Anon	Time	Size	blind

The diversity class depends mainly on the selected carrier (even if the connection is not strictly necessary). Time diversity, in which the single watermark bit b is replicated M times at different times, is the scheme most used by researchers using timing-based carriers (see Section III-A4b). An example of time diversity, proposed in some works, is obtained when the single bit b is replicated multiple times with the same version of the carrier signal [81]–[83]. "Sparsification" is another approach proposed, whereby a single bit b is deterministically mapped to a long sequence of M bits [35], [84].

Frequency diversity is the scheme adopted in a few works [38], [75], [85]–[88]. In this method, often referred to as "direct sequence spread spectrum" (DSSS), a pseudo-noise (PN) code of M bits is used to spread the carrier signal over a spectrum wider than the original signal bandwidth.

Lastly, space diversity, where the watermark is embedded over several different channels, was obtained in Botmosaic developed by Houmansadr and Borisov [45]. In fact, the single bit b is transferred on multiple flows originating from several sources and addressed to a single destination.

The diversity schemes for the main works analyzed are listed in the second column of Table I.

4) Embedding: The watermark embedding process embeds the watermark bit into a selected carrier signal by slightly modifying some of the carrier's features. Typically researchers select a particular carrier and propose methods which consider several factors: the main objective, the ability to modify traffic features, the position of the watermarker, whether the traffic is encrypted or not, etc. Figure 4 provides a taxonomy of several of the carriers adopted by researchers for the purpose of TA, while in the third column of Table I the main watermarking algorithms' carriers are listed. We divide the carriers into four main categories: content, timing, size, and rate-based. Additional information about each carrier type is provided below.

a) Content-based: In content-based watermarking methods, the watermark bit is directly injected into the contents of the exchanged messages. The watermark can be inserted into *payloads* or *headers* of selected protocol data units. To ensure that the watermark can be read by the traffic observer, the messages must be sent on the channel with no encryption. For this reason, content-based watermarking is currently useless in TA. Only a few works fall in this category [32], [89], [90]. An example of a content-based algorithm was designed in a work by Wang *et al.* [32], in which a virtual null string is inserted into the messages so that it appears null to the end of the communication.

b) Timing-based: In the timing-based watermarking approach, the carrier signal is the sequence of arrival (or departure) times of the flow packets observed at a certain point in the network. Watermarks can be embedded by introducing some degree of delay to selected packets of the target flow. Usually, interpacket delays (IPDs) are considered instead of the packet arrival times. Given a flow composed by a sequence of N ordered packets, let t_i (for $i = 0, \ldots, N-1$) be the arrival times at the watermarker site, with the arrival time t_0 of the first packet fixed as time axis origin ($t_0 = 0$). The notation $\tau_{i,j} = t_i - t_j$, with $i > j$ is used throughout the paper for indicating the IPD between two different arrival times. When $i = j+1$ for $i = 1, \ldots, N-1$, the delay is considered between consecutive packets. For simplicity, the IPDs between consecutive packets are indicated by $\tau_i = t_i - t_{i-1}$ for $i = 1, \ldots, N-1$.

We describe two different mechanisms to embed a single watermark bit b: "simple delay" and "mean balancing."

Simple Delay: We call "simple delay" the watermark embedding procedure where the *interpacket delays* (or *packet departure times*) are altered according to the relation:

$$\tau_{i,j}^{b} = \tau_{i,j} + \Delta_{i,j}(b, \mathbf{x}, \mathbf{r}) \quad (1)$$

where $\tau_{i,j}$ and $\tau_{i,j}^{b}$ are the IPDs before and after watermark embedding, respectively, while $\Delta_{i,j}(b, \mathbf{x}, \mathbf{r})$ is an additive delay

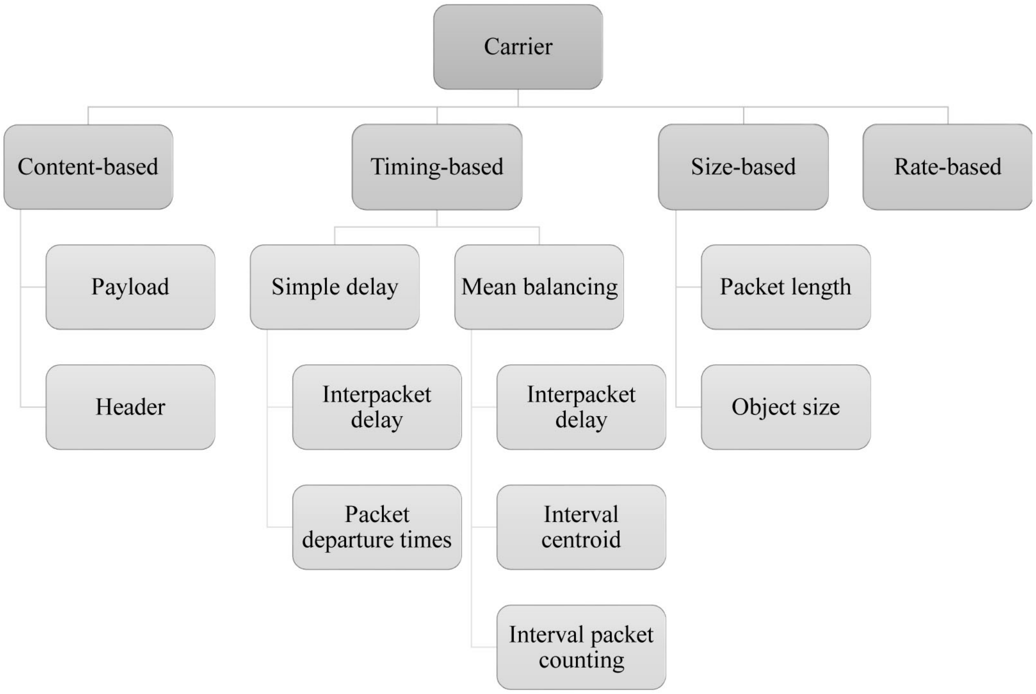

Fig. 4. Watermark carrier used in traffic analysis.

mainly a function of the bit b. Depending on the characteristics of the algorithm, $\Delta_{i,j}$ can also be a function of a vector $\mathbf{x}$ of some deterministic parameters and/or a vector $\mathbf{r}$ of probabilistic parameters appropriately defined in the watermarking algorithm. Some of the parameters in the two vectors $\mathbf{x}$ and $\mathbf{r}$ may have to be secretly shared between the watermarker and the detector. For simplicity, we use Δ_i instead of $\Delta_{i,i-1}$ when referring only to consecutive packets.

Some researchers proposed algorithms which consider the IPDs of sequential packets, with the i-th delay $\Delta_i(b, s, \tau_i)$ a function of only three components: the watermark bit b, the i-th IPD τ_i, and a constant s being the quantization step size [81], [95], [97].

In other works, Δ_i is a function of: the IPD's values $\tau_1, \tau_2 \dots, \tau_i,\ \tau_1^b, \tau_2^b, \dots, \tau_{i-1}^b$, the watermark bit b, and the quantization step size s [35], [84].

Unlike the previously described works, Houmansadr *et al.* propose a watermarking strategy named RAINBOW in which the additive delay Δ_i is a function only of b (with $b \in \{0, 1\}$) and a probabilistic component according to the relation:

$$\Delta_i = b \cdot r_i \tag{2}$$

where r_i is a random variable that can assume one of two real numbers, such that $r_i \in \{r^{(0)}, r^{(1)}\}$, with $r^{(0)} > 0$ and $r^{(1)} \leq 0$, both with the same probability [82], [83].

Mean balancing: Peng *et al.* [91] introduced a new probabilistic timing-based paradigm for watermark embedding. This strategy, hereafter referred to as "mean balancing," is based on the selection of a specific traffic feature x which can be measured or computed at least $2 \cdot d$ times on the flow, where d is a positive integer, the choice of which is based on the number of features to be used to embed the single bit. The selected feature may assume a value in a discrete or continuous, and limited or unlimited domain $\mathcal{D}$. Let x_i be the i-th value of the feature extracted from the target flow, with $i = 1, \dots, 2 \cdot d$, and realization of a random variable X_i with probability distribution $p_{X_i}(x)$. The set of random variables $\{X_i\}_{i=1,\dots,2\cdot d}$ can be pseudo-randomly or deterministically divided into two groups A and B, each of which are composed by d random variables. Let Y_A and Y_B be two new random variables obtained as the average of the random variables in A and B respectively:

$$Y_J = \frac{1}{d} \sum_{X_i \in J} X_i, \quad \text{for } J = A, B. \tag{3}$$

If the random variables Y_A and Y_B have equal expected values, i.e., $E(Y_A) = E(Y_B)$, and the probability distributions $p_{Y_A}(y)$ and $p_{Y_B}(y)$ can be altered slightly in a controlled fashion by delaying some packets of the target flow so that $E(Y_A) \neq E(Y_B)$, then a watermark bit $b \in \{b_0, b_1\}$ can be embedded in the flow following the rule: $E(Y_A) = E(Y_B) - c^{(0)}$ for $b = b_0$ and $E(Y_A) = E(Y_B) + c^{(1)}$ for $b = b_1$ (or vice versa), where $c^{(0)} \geq 0$ and $c^{(1)} > 0$ are two selected constants. When the flow is observed at a different point in the network, the observer can compute the difference between the two realizations $\tilde{y}_A - \tilde{y}_B$ and determine which watermark value has been embedded.

Depending on the traffic feature selected by the developer, we define three mean balancing embedding categories:

- *Interpacket delay*-based
- *Interval centroid*-based
- *Interval packet counting*-based.

In their works Peng, Wang, Park, and Pan adopt the interpacket delay as a feature for the mean balancing algorithm [60], [91], [98], [99]. IPDs between consecutive packets are preferred over those between non-consecutive packets.

The interval centroid-based algorithm was introduced by Wang *et al.* [37]. Given the flow duration T_f, they select an offset $o > 0$ (starting point) and divide a reference duration T_p $(T_p < T_f - o)$ in $2 \cdot d$ intervals $I_1, I_2, \dots, I_{2\cdot d}$, each of them with duration T. The i-th interval I_i will contain n_i packets. Thus, the features selected to be balanced are

$$x_i = \frac{1}{n_i}\sum_{h=1}^{n_i}[(t_h - o) \mod T], \quad \text{for} \quad i = 1, \dots, 2 \cdot d \tag{4}$$

where t_h is the arrival time of the h-th packet in the interval I_i.

Some variations of the interval centroid-based algorithm are considered in a few works [93], [94], [100]–[102].

The algorithm proposed by Pyun *et al.* [92] in 2007 is also interval based, but instead of using the centroid, they considered the number of packets in each interval; we refer to this algorithm as interval packet counting-based. The features selected to be balanced are

$$x_i = n_i \quad \text{for} \quad i = 1, \dots, 2 \cdot d \tag{5}$$

In addition to Pyun, there are several researchers who have used interval packet counting-based algorithms in their works [36], [45], [46], [103], [104].

c) Size-based: The size of the information content exchanged between two communicating parties, which can be inferred by observing the *packet lengths* of a target flow, is one of the main features used in passive TA. In order to exploit this feature in active TA, one needs to alter the packet lengths or the size of the contents encapsulated in the traffic flow; in cases in which the traffic is encrypted, this can be accomplished only if the process is executed before the encryption. Usually the watermarker cannot manipulate the traffic at the endpoints before the encryption, which makes this carrier quite unappealing. In fact, we found only a few works based on this watermark carrier [44], [58], [76].

Packet Length: When the watermarker has access to the packets before the encryption, the packet length modification can be generalized according to the relation:

$$\ell_i^b = \ell_i + \Lambda_i(b, \mathbf{x}, \mathbf{r}) \tag{6}$$

where ℓ_i and ℓ_i^b are the lengths of the i-th packet in the target flow before and after the watermark embedding, respectively, while $\Lambda_i(b, \mathbf{x}, \mathbf{r})$ is an additive positive padding mainly a function of the bit b. Depending on the characteristics of the algorithm, Λ_i can also be a function of a vector $\mathbf{x}$ of some deterministic parameters and/or a vector $\mathbf{r}$ of probabilistic parameters appropriately defined in the watermarking algorithm. Some of the parameters in the two vectors $\mathbf{x}$ and $\mathbf{r}$ may have to be covertly shared between the watermarker and the detector.

The watermarking algorithm developed by Ramsbrock *et al.* [44] randomly selects two packets P_r and P_e in the target flow, called "reference" and "encoding" packets, respectively. Let ℓ_r and ℓ_e be the length of the two packets with $\ell_e > \ell_r$. To embed a watermark b, some padding characters are added to one of the two selected packets so that a specific length difference z_b is achieved. z_b must be associable to the value of the watermark bit b when the watermarked flow is observed. The padding characters added should be difficult to detect by the receiver of the legitimate communication. In this case, Λ_i in Equation (6) can be written as:

$$\Lambda_i = \begin{cases} \max\{(\ell_e - \ell_r - z_b), 0\}, & \text{if } i = r \\ \max\{(z_b - \ell_e + \ell_r), 0\}, & \text{if } i = e \\ 0, & \text{if } i \neq r, e \end{cases} \tag{7}$$

Ling *et al.* [58] created a malicious HTTP Web server capable of generating packets of lengths with statistical distribution recognizable by a client-side sniffer controlled by the attacker. In this case the additive padding $\Lambda_i(r_i)$ is mainly function of a random variable r_i that can assume a real value in [0, 1].

Object Size: One example of an object size-based algorithm can be observed in an attack on TOR analyzed by Arp *et al.* [79], where Web page requests and replies with ad hoc content sizes are created. Arp *et al.* [79] demonstrate that if an attacker can generate some traffic from a victim by means of JavaScript code, the anonymity of the victim in TOR can be weakened by exploiting driven content sizes of the requests/replies. In this case the driven resources injected in the communication are generated with fixed sizes which can be easily recognized by anyone.

d) Rate-based: The voluntary injection of dummy traffic in a segment of the network may influence the rate of real traffic going through the same segment at that time. The traffic injection (considered interference) can be controlled so that identifiable rate patterns are generated on a target flow. This is the main idea exploited by Yu *et al.* [75] to embed a watermark bit b in a target flow. In fact, they apply weak interference against the flow for a period T when the watermarker wants to embed a bit $b = b_0$, while for $b = b_1$ strong interference is applied. As a result, a flow, characterized by an average traffic rate R, when affected by piloted interference, will have a rate that can be generalized according to the relation:

$$R_{T_i}^b = R_{T_i} + \Gamma_i(b) \tag{8}$$

where R_{T_i} and $R_{T_i}^b$ are the bit rates of the i-th period of duration T in the target flow before and after the watermark embedding, respectively, while $\Gamma_i(b)$ represents the rate variation due to the interference which is mainly a function of the bit b. More specifically Γ_i can be written as:

$$\Gamma_i = \begin{cases} C, & \text{if } b = b_1 \\ D, & \text{if } b = b_0 \end{cases} \tag{9}$$

where C and D are two negative parameters linked to the interference's intensity with $C < D \leq 0$. The same watermarking idea was later used by other researchers [38], [85]–[88], whose approaches differ from one another mainly based on the spreading function adopted.

A different example of watermarking based on the variation of traffic rates can be found in the framework developed by Chan-Tin *et al.* [76] where a malicious burst server is designed with the objective of introducing burst traffic in a TOR connection with a victim. Chan-Tin *et al.* [76] show that the TOR relays crossed by the flow can be identified by observing changing traffic volumes.

B. Watermark Detector

The watermark detector is the component devoted to passively observing the traffic at a specific point in the network, analyzing the traffic features, and determining whether a watermark has been embedded in the observed flows. In cases in which the system contemplates the transmission of a sequence of bits, the correct sequence must be identified. The specific location of one or more identifiers in the network depends mainly on where one expects to observe watermarked flows.

We consider two main phases in the watermark identification procedure: feature extraction and decoding. Additional details are provided in the next subsections.

1) Feature Extraction: Once a flow is sniffed, the detector extracts the features of the flow packets which could potentially transport watermark bits, in accordance with the carrier selected (see Section III-A4). The collection of the selected features will be a descriptor vector of the observed flow.

When the carrier is timing-based, the arrival timestamps of the flow packets will be extracted. Also, rate-based algorithms are required to measure packets' timestamps, while in the case of size-based carriers, the identifier needs to read the sequence of packet lengths. Lastly, when the carrier is content-based, the identifier must read and analyze the content in the packet payloads in order to find the hidden information.

2) Decoding: After extracting the features of the flow to be examined, the identifier computes the value of a function of the extracted features and the parameters previously arranged with the watermarker. This value can indicate whether the watermark bit b is b_0 or b_1 (in the case of (0/1)-based watermarking), or alternatively, whether the flow is watermarked or not (in the case of (1)-based watermarking). The detection problem can be seen as a classification problem, where the detector has to assign a class label to each observed flow. The (0/1)-based policy is based on three classes: unwatermarked, watermarked with bit b_0, and watermarked with bit b_1. In contrast, the classification problem of the (1)-based policy is based on two classes: unwatermarked and watermarked. Algorithms for this classification problem are generally much simpler and faster than those used in passive TA where the traffic features are not actively altered.

We distinguish between two types of watermark decoding algorithms: "blind" and "non-blind" algorithms (see Table I column 4). When the watermark is non-blind, the identifier needs to know the carrier values measured at the watermarker point in order to make a decision, while in case of a blind watermarking system, the detector can base its decision on the features observed at the detection point.

If a diversity scheme has been adopted by the watermarker, the word composed of M bits extracted by the flow must be converted to a watermark b.

In general, the classification choice is made by algorithms that identify the most likely watermark conditionally based on the sequence of M bits. Often the decision is included in the decoding algorithm, as in mean balancing algorithms.

Below, we provide some details about the decision procedure for the four carriers.

a) Content-based: When the watermarks are injected in the payloads, the traffic must not be ciphered, otherwise the detector would be unable to read the watermarks. The detector scans the payload of all of the packets observed in order to find a recognizable label pre-defined with the watermarker.

b) Timing-based: Given a flow observed by the detector of $\tilde{N}$ ordered packets, let $\tilde{t}_i$ (for $i = 0, \ldots, \tilde{N}-1$) be the arrival time of the i-th at the detection point, with the arrival time $\tilde{t}_0$ of the first packet fixed as time axis origin ($\tilde{t}_0 = 0$). As at the watermarker site, the IPD is defined as the difference between two different arrival times $\tilde{\tau}_{i,j} = \tilde{t}_i - \tilde{t}_j$, with $i > j$.

When a simple delay $\Delta_{i,j}(b, \mathbf{x}, \mathbf{r})$ is added by the watermarker (see Equation (1)), the detector computes the estimated bit $\tilde{b}$ as a function of the IPDs between agreed upon pairs of packets:

$$\tilde{b} = f\big(\tilde{\tau}_{i,j}, \mathbf{x}, g(\mathbf{r})\big) \tag{10}$$

where the vector $\mathbf{x}$ of deterministic parameters has been previously shared, while the function $g(\mathbf{r})$ represents the statistical characterization of the probabilistic component $\mathbf{r}$ in Equation (1). In the case of (0/1)-based watermarking strategies, the detector reveals the watermark exactly, if there has been no interference in the information channel, and the watermarking features (i.e., the interpacket delay) were exactly the same as at the watermark embedding site.

In case of mean balancing watermarking the decoding procedure simply computes the function

$$\begin{aligned} D(\tilde{y}_A, \tilde{y}_B) &= \frac{1}{d}\sum_{\tilde{x}_i \in A} \tilde{x}_i - \frac{1}{d}\sum_{\tilde{x}_i \in B} \tilde{x}_i \\ &= \tilde{y}_A - \tilde{y}_B \end{aligned} \tag{11}$$

where $\tilde{x}_i$ is the i-th flow feature value computed or measured at the detection site, while $\tilde{y}_A$ and $\tilde{y}_B$ are the respective values of y_A and y_B of Equation (3) but computed at the detection site. After having appropriately defined two thresholds $c_{th}^{(0)}, c_{th}^{(1)} \geq 0$ such that $c_{th}^{(0)} < c^{(0)}$ and $c_{th}^{(1)} < c^{(1)}$, if $D(\tilde{y}_A, \tilde{y}_B) > c_{th}^{(1)}$, the flow is classified as watermarked with the bit b_1. If $D(\tilde{y}_A, \tilde{y}_B) < -c_{th}^{(0)}$, the flow is classified as watermarked with the bit b_0. Otherwise the flow is considered unwatermarked.

Although in some works the function $D(\tilde{Y}_A, \tilde{Y}_B)$ is computed after all of the values $x_i \in A \cup B$ are available, some researchers have proposed algorithms to make the detection faster without necessitating waiting until the entire flow is analyzed [105], [106].

c) Size-based: Size-based watermark detection requires the analysis of flow packet lengths. By applying a simple decision rule on the observed preprocessed lengths, it is possible to estimate whether a bit b_0 or b_1 has been embedded.

d) Rate-based: Rate-based watermark detection requires the analysis of flow rate variations; with a simple decision rule applied on the observed rate, it is possible to estimate whether a bit b_0 or b_1 has been embedded.

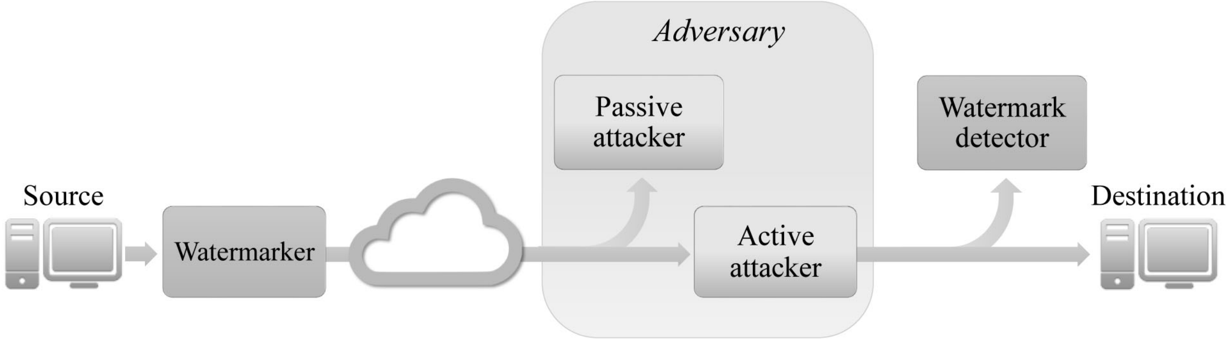

Fig. 5. Attack scenario.

C. Watermarking Algorithm Comparison

For comparison purposes, the objectives, diversity schemes, carriers, and blindness features of the main algorithms proposed in the literature are listed in Table I. It is worth noting the following points. Although watermarking techniques have been adopted in several scenarios with a number of different goals, tracing network-based attacks is certainly the most pursued objective. This is due to the fact that identifying the source of an attack and location of the attacker remain challenging problems and are of particular interest to the scientific community working on network security. Other scenarios have generated less curiosity, but it is interesting to note that network flow watermarking may be used in a versatile manner, in different contexts, and with different objectives which often are in contrast with one another.

Timing-based algorithms are usually preferred to size, content, and rate-based solutions, mainly due to the fact that packet times are more easy to manipulate and timing alterations are harder to identify by a third party. In addition, mean balancing algorithms are preferred among timing-based algorithms; the main reason for this preference is that the algorithms in this category add short delays to each packet of the target flow.

Time diversity is the primary diversity scheme adopted in the literature, mainly because the choice of this feature is closely related to the choice of the carrier.

Undoubtedly, blind algorithms are preferred to non-blind ones. Although blind methods can be more accurate in recognizing watermarks, they require collaboration and synchronization between the watermarker and detector which need to share a database of the watermarked and unwatermarked features. In contrast, non-blind solutions offer a more flexible watermark detector which is completely independent from the watermarker.

IV. Attacks Against Watermarking

Figure 5 shows a generic scenario of an attack against watermarking. We identify two kinds of attacks: passive and active. In a passive attack, the adversary can discover the presence of a watermark in the flow only by observing the traffic, but unlike the watermark detector, the adversary has no knowledge about secret parameters used to embed the code. This threat is also referred to as an attack against the invisibility. In an active attack, the adversary can alter the statistical features of the traffic in order to remove the information embedded in the flow. This attack aims at damaging the watermarking algorithm's robustness and the ability to identify the watermark by the detector.

In Figure 6 the main vulnerabilities of network flow watermarking are schematized, while additional details about attacks against the invisibility and robustness are provided in Sections IV-A and IV-B.

A. Invisibility

Digital watermarks can be visible or invisible. In the first case, the watermark is embedded so that anyone observing the marked object may be able to read it. In contrast, the invisible watermark is used when the watermarker wants to protect the existence and content of the embedded information. In network flow watermarking, legitimate and malicious objectives require invisibility mainly due to the vulnerability of the carriers (see Section IV-B). For this reason the main algorithms proposed in the literature claim to be invisible [35], [75], [83], [84], [87], [94]. Unfortunately, in recent years several works have shown that some types of watermarks can be easily identified by third parties.

Formally, a watermarking system is said to be invisible if, for each set of selected flow features, any watermarked flow is indistinguishable from any unwatermarked flow. The concept of "indistinguishability" is used with several purposes in cryptography. Recall the definitions of "statistical" and "computational" indistinguishability provided by Goldreich [107].

Definition 1 (Statistical Indistinguishability): Let $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ be two ensembles of random variables, each ranging over a finite domain Ω. X and Y are statistically indistinguishable if for some negligible function $\mu : \mathbb{N} \to \Omega$ and for every $n \in \mathbb{N}$,

$$\sum_{\alpha\in\Omega} |Pr(X_n = \alpha) - Pr(Y_n = \alpha)| < \mu(n). \tag{12}$$

That means the statistical difference between the two distributions is negligible.

Definition 2 (Computational Indistinguishability): Let $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ be two ensembles of random variables, each ranging over a finite domain Ω. X and Y are computationally indistinguishable if for every probabilistic polynomial-time algorithm $\mathcal{A}$, for every sufficiently large

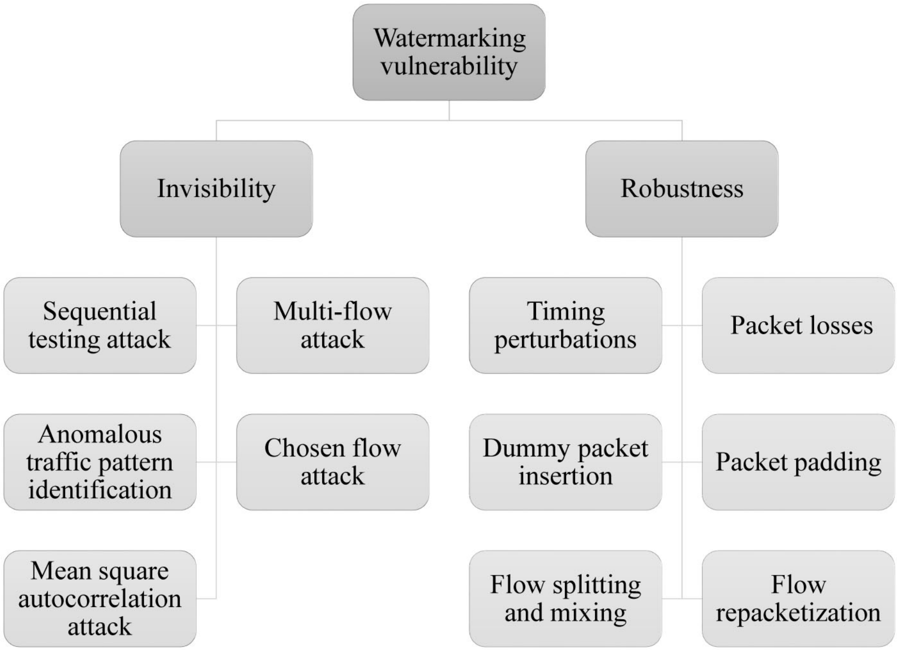
Watermarking vulnerability
Invisibility
Robustness
Sequential testing attack
Multi-flow attack
Anomalous traffic pattern identification
Chosen flow attack
Mean square autocorrelation attack
Timing perturbations
Packet losses
Dummy packet insertion
Packet padding
Flow splitting and mixing
Flow repacketization

Fig. 6. Attacks against network flow watermarking.

n, there exists a negligible function $\mu : \mathbb{N} \rightarrow \Omega$ so that

$$|Pr(\mathcal{A}(X_n) = 1) - Pr(\mathcal{A}(Y_n) = 1)| < \mu(n). \tag{13}$$

Based on definitions 1 and 2, we can accordingly define the two concepts of "invisibility" in a watermarking system.

Definition 3 (Statistical Invisibility): Let $X = \{X_n\}_{n\in\mathbb{N}}$ be an ensemble of random variables, and w be a symbol (watermark) selected in a finite set Ψ. Instead, let $Y = \mathcal{G}_w(X)$ be a transformation algorithm on the ensemble X. w is a statistically invisible watermark if X and Y are statistically indistinguishable.

Definition 4 (Computational Invisibility): Let $X = \{X_n\}_{n\in\mathbb{N}}$ be an ensemble of random variables, and w be a symbol (watermark) selected in a finite set Ψ. Instead, let $Y = \mathcal{G}_w(X)$ be a transformation algorithm on the ensemble X. w is a computationally invisible watermark if X and Y are computationally indistinguishable.

Although the watermarking algorithms developed and illustrated in Section III claim to be invisible, this property has never been formally proven, and in fact, a number of attacks against the presumed invisibility have come out in recent years. Confirmation of this can be observed in the first column of Table II. In the following subsections more details about the attacks against invisibility are provided.

1) Sequential Testing Attack: The first attack mentioned in the literature against the invisibility in network flow watermarking is the "sequential testing attack" which was proposed by Peng *et al.* [108] in 2006. In this attack a strong assumption is hypothesized: the sequence of packet delays $\delta_1, \delta_2, \ldots, \delta_N$ between two nodes in the network, one placed before and the other placed after the watermarker, are known by the adversary, considering that K of N delays had been expanded by the watermarker. Let θ be the ratio K/N. In order to detect whether a watermark exists on a flow, the authors apply the sequence probability ratio test algorithm on the value θ [109]. Their results show that a watermark can be detected with greater than 90% accuracy.

2) Multi-Flow Attack: Kiyavash *et al.* [110] show another statistical attack – the "multi-flow attack". They demonstrate that an adversary is able to reveal the presence of either an interval packet counting-based or interval centroid-based watermark proposed by Wang *et al.* [37] and Pyun *et al.* [92] in 2007. In addition, they show how to recover the secret watermarking parameters. The multi-flow attack is based on two assumptions: 1) the attacker can collect a small number of watermarked flows, all containing the same watermark code, and 2) the normal traffic can be modelled as a Markov-modulated Poisson process (MMPP). Basically, with this attack the adversary can plot all of the packet timestamps of all the selected flows on the time axis; if the flows are watermarked, well separated clusters can be viewed on the plot, and otherwise the timestamps are uniformly distributed.

3) Anomalous Traffic Pattern Identification: Luo *et al.* [111] developed a watermark detection system with the main idea of identifying any anomalous sequence of low-throughput in the network flows. The system was designed against rate-based watermarks spread out with a DSSS scheme, and it is based on two main steps: identification of low throughput periods in the flows and detection of abnormal patterns in the selected periods. In their experiments, a detection rate close to 100% is obtained through this attack.

Backlit is an example of an attack carried out against timing-based watermarks [112]. Unlike the work of Luo *et al.* [112], the abnormal traffic is identified by means of a one-class classifier trained with only non-watermarked traffic patterns [113]. In the training phase, the algorithm defines the boundary of the known pattern class. The classification process consists

TABLE II
VISIBILITY AND ROBUSTNESS OF THE MAIN ALGORITHMS IN THE LITERATURE. WHEN NOT ENOUGH INFORMATION WAS FOUND, "–" IS INSERTED

	Visibility	Robustness against active attacks	Robustness against specific perturbations
Wang, 2001 (SLEEPY) [32]	visible [32]	weak	–
Wang, 2003 [81]	visible[108]	weak [110]	random timing perturbation
Peng, 2005 [91]	–	week [91]	chaff packet injection
Wang, 2005 [60]	visible [110]	weak [110]	–
Wang, 2007 [37]	visible [110], [112]	weak [110], [116]	packet dropping, repacketization, flow splitting
Yu, 2007 [75]	visible [110], [111], [115]	weak [110]	–
Pyun, 2007 [92]	visible [110], [112]	weak [110]	repacketization, random timing perturbation
Ramsbrock, 2008 [44]	–	–	–
Houmansadr, 2009 (RAINBOW) [83]	visible [112]	weak [114]	–
Houmansadr, 2009 [93]	visible [112]	weak [114]	multi-flow attack
Deng, 2009 [89]	–	–	–
Houmansadr, 2011 (SWIRL) [94]	visible [112]	weak [114]	multi-flow attack
Wang, 2011 [95]	–	robust (not proven)	timing perturbations
Bates, 2012 [63]	visible [112]	weak [114]	multi-flow attack
Houmansadr, 2012 (BOTMOSAIC) [45]	visible [110], [112]	week [45]	–
Yu, 2013 [96]	–	–	–
Ling, 2013 [58]	–	–	–

of determining whether an observed flow is a member of the known class or not. This classifier was tested against four watermarking algorithms proposed by Wang *et al.* [37], Houmansadr *et al.* [83], [94], and Pyun *et al.* [92]. The Backlit framework shows anomalous statistical behavior of the mean, and variance of the times elapsed between sending a request message and receiving a response from the server when the flows are watermarked with the RAINBOW [83] and SWIRL algorithms [94]. Backlit also identifies anomalies in the statistics of interpacket delays of flows watermarked by mean balancing algorithms [37], [92]. These anomalies allow Backlit to easily recognize watermarked flows with up to 100% accuracy for all tested algorithms.

4) Chosen Flow Attack: A set of timing-based attacks was illustrated in a work by Lin and Hopper [114] in 2012. In their model, called "chosen flow attack," the authors consider the histogram of the IPDs measured for a flow under observation and apply the cosine similarity measure on the bins of the histogram. Thus they compute a metric based on the similarity score which disclosures the presence of a watermark. Their experiments show that flows watermarked with the RAINBOW [83] and SWIRL algorithms [94] can be correctly identified within the chosen flow attack, and even in this case 100% accuracy is obtained.

5) Mean Square Autocorrelation Attack: The "mean square autocorrelation attack" (MSAC) was recently proposed by Jia *et al.* [115]. This attack is designed against the DSSS diversity scheme, both in time and frequency domains. They analytically demonstrate that when the same PN code is used to spread each bit of the watermark sequence, it is possible to identify timing sequence correlations in the flow. In particular, they exploit the square autocorrelation function applied to time-shifted periods of the spread watermarked signal. When a watermark is embedded in a flow, periodic peaks can be observed in the mean square autocorrelation function. The MSAC enables the attacker to detect the presence of a watermark and obtain the PN code. Experimental results show detection rates greater than 60%.

B. Robustness

Invisibility is not the only property a watermark must have; a watermark should also be robust. Poor robustness is one of the main reasons why researchers investigate the use of watermarking as an alternative to passive TA. In fact, it is more robust than passive analysis against network noise. That notwithstanding, robustness aspects of watermarking require particular attention, because watermarks are vulnerable to a number of network behaviors and attacks making watermark undetectable.

For instance, encryption algorithms can be considered as an attack to content-based watermarking, as they do not allow violating data integrity and confidentiality by third parties.

Several threats to timing, size, and rate-based watermarking are known in the literature. We identify six types of threats:

- Timing perturbations
- Packet losses
- Dummy packet insertion
- Packet padding
- Flow splitting and mixing
- Flow repacketization.

The second column of Table II indicates the robustness of the main algorithms in the literature against active attacks, and the third column contains the threats tested against the same algorithms.

1) Timing Perturbation: Packets flowing through the network will suffer from jitter, however if the delay is the same for all packets in the same flow, it will not affect passive and active TA. Unfortunately, there are a number of legitimate protocol behaviors in the network leading to different packet delays. Variations in link and router congestion, packet queuing, and multipath routing are just some examples of the causes of jitter in a network.

The impact of natural timing perturbations on the robustness of timing-based watermarks was studied in several works, where the authors show that the correlations of packets' timestamps before and after perturbations remain unaltered [81], [86], [95], [99].

Much more challenging is the delay voluntarily added to packets by an adversary in order to noisily perturb the packet timestamps and destroy any information carried by statistical features of the traffic. Active timing perturbations have been widely studied [116]–[119]. Some researchers studied timing perturbation with the aim of obtaining flows at constant packet rate [116], [119]. A constant packet rate would completely remove timing information. They also consider the usability of this kind of attack and conclude that it involves a significant degradation of quality of service intolerable by many adversaries.

The minimum timing distortion required for an adversary to eliminate watermarking information was analyzed by Wang and Reeves [95]. They show that it is not feasible for an adversary to cancel all of the information contained in the packet timestamps if real-time constraints are considered.

The model of active random timing perturbation created by Donoho *et al.* [118] in 2002, and its impact on mean balancing watermarking are analyzed by several works [92], [100], [102], [104], whereas real active perturbation in low latency anonymous networks is considered by Park and Reeves [98].

Kiyavash *et al.* [110] introduced a different timing perturbation attack directed at completely removing the watermark from the flow in cases in which the watermark parameters are known or acquired by means of the multi-flow attack.

An opposite approach to timing perturbation is the "copy attack" developed by Lin and Hopper [114] in 2012. This attack was inspired by the "protocol attack" applied to multimedia watermarks where a watermark is copied from a digital product and embedded into another [120]. The main idea is that if the attacker is able to discover the watermark and extract it, the same watermark can be embedded in other different flows so as to degrade performance by increasing the false positives of the watermark detector. The authors show the feasibility of this attack on RAINBOW [83] and SWIRL [94].

2) Packet Losses: IP protocol does not guarantee every sent packet will reach its destination. Therefore, some packets of the IP flow can be lost. The main causes of packet loss are node or link failure, detected errors and discarded frame, congestion problems and buffer overflow, and deliberate loss due to safety or efficiency reasons.

The loss of a packet in a watermarked flow implies not only the loss of its features, but also the misalignment of packet sequence from the point of view of the passive watermark detector.

Some packets can also be actively discarded by an adversary in order to impede the identifier from detecting the watermark.

Gong *et al.* [35] evaluated the robustness of their watermarking algorithms in the presence of random packet losses with loss probability less than 10%. In contrast, the robustness of cases with traffic up to 40% of loss is demonstrated with rate-based algorithms by Liu *et al.* [86].

3) Dummy Packet Insertion: Dummy packets can be actively injected in the traffic by an adversary in order to weaken watermarks. From the adversary perspective, dummy packets should not be recognized by a watermark detector and should be identifiable by the flow destination so they can be discarded. Some existing standard protocols already offer the possibility of inserting dummy packets in a flow; for instance, IPSec can be configured to obtain connections with packets sent at random intervals or shaped as any actual traffic distribution [121].

As for packet losses, dummying implies a misalignment of packets observed by the watermark detector. In addition, new piloted features are added which introduce interference to the carrier, and this can cause real packet timestamps to be perturbed.

Robustness against dummy packet insertion in mean balancing watermarking algorithms was studied by Peng *et al.* [91]. The watermarking algorithms were tested under uniformly distributed timing perturbation and dummy packet insertion with 11 different types of Poisson distribution.

4) Packet Padding: When an attacker wants to remove the side-channel information contained in packet lengths, padding can be added to the packets to modify the lengths. With padding, only the lengths can be increased. Even if efficient padding insertion strategies were developed such that the correlation between the original and the padded packets of a flow was completely eliminated, padding is not typically applied in real contexts [122], [123]. For this reason, there are no papers addressing the impact of padding on watermarked flows in the case of size-based watermarks. Liu *et al.* [86] briefly analyzed the impact of padding insertion (applied to the flows simultaneously with timing perturbation) in the case of rate-based watermarks.

5) Flow Splitting and Mixing: Another strategy to alter the features of a packet flow is flow splitting whereby a single IP flow is divided into multiple distinct (ciphered) subflows. Each subflow will have a subset of packets of the original flow. When an observer looks at the split flows, he does not know that the split flows are part of the same flow, unless he can recreate the original flow through cross-analysis.

Flow mixing is the opposite of flow splitting. In this attack, two or more unrelated flows are mixed together in a single (ciphered) flow, so when an observer looks at the mixed flow, even if he might be able to understand whether a flow is an aggregate, he cannot separate the original flows to extract a potential watermark. From the viewpoint of the watermark detector, split flows can be thought of as flows with a high percentage of lost packets, while a mixed flow can be viewed as a form of dummying applied to the target flow.

Flow splitting and mixing and their impact on network flow watermarking are analyzed by Gong *et al.* [35], Wang *et al.* [37], and Jin and Wang [124].

6) Flow Repacketization: The process of transforming a flow f, composed of N packets $P_1, P_2, \ldots, P_N$, in a new flow f' of N' packets $P'_1, P'_2, \ldots, P'_{N'}$ containing the same contents of f is called "repacketization." In this conversion, the new

TABLE III
METRICS ANALYZED FOR THE MAIN ALGORITHMS PROPOSED IN THE LITERATURE: TRUE POSITIVE (TP), FALSE POSITIVE (FP), FALSE NEGATIVE (FN), NUMBER OF PACKETS (# PKTS), LATENCY (LAT), PERCENTAGE OF BIT CORRECTLY (OR INCORRECTLY) IDENTIFIED (% C/I), MEMORY USAGE (MMU), CROSS-OVER ERROR RATE (COER), KOLMOGOROV-SMIRNOV TEST (KS), AND ENERGY CONSUMPTION (EC)

	TP	FP	FN	# Pkts	Lat	% c/i	MU	COER	KS	EC
Wang, 2001 (SLEEPY) [32]	–	–	–	–	✓	–	–	–	–	–
Wang, 2003 [81]	✓	✓	–	–	–	–	–	–	–	–
Peng, 2005 [91]	✓	✓	–	✓	–	–	–	–	–	–
Wang, 2005 [60]	✓	✓	–	–	–	✓	–	–	–	–
Wang, 2007 [37]	✓	✓	–	–	–	–	–	–	–	–
Yu, 2007 [75]	✓	✓	–	–	–	–	–	–	–	–
Pyun, 2007 [92]	✓	–	–	–	–	–	–	–	–	–
Ramsbrock, 2008 [44]	–	–	–	–	–	✓	–	–	–	–
Houmansadr, 2009 (RAINBOW) [83]	✓	✓	✓	–	–	–	✓	✓	✓	–
Houmansadr, 2009 [93]	–	–	–	–	–	–	–	–	–	–
Deng, 2009 [89]	✓	–	–	–	–	–	–	–	–	✓
Houmansadr, 2011 (SWIRL) [94]	✓	✓	–	✓	–	–	–	✓	–	–
Wang, 2011 [95]	✓	✓	–	–	–	–	–	–	–	–
Bates, 2012 [63]	–	–	–	–	–	–	–	–	✓	–
Houmansadr, 2012 (BOTMOSAIC) [45]	–	–	–	–	–	✓	–	✓	–	–
Yu, 2013 [96]	✓	✓	–	–	–	–	–	–	–	–
Ling, 2013 [58]	✓	–	–	–	–	–	–	–	–	–

packets have different lengths and timestamps than the original ones. The transformation can be obtained by fragmenting one packet in two or more shorter packets, and/or aggregating more packets in a longer one. Repacketization can be a side effect of the legitimate behaviors of many applications (e.g., SSH [125]), but it can also occur as the result of an anonymous system with the precise aim of introducing interference (e.g., Anonymizer.com [67]).

The robustness of watermark detection in the presence of flow transformation with a repacketized packet ratio of 10-11.8% combined with timing perturbations is discussed in two works by Pyun *et al.* [92], [104].

V. PERFORMANCE EVALUATION

The watermarking evaluation processes may be classified into two main classes, depending on the trace collecting procedure. In the first typology of tests, the watermarking system is tested through real-time experiments on real or emulated/simulated scenarios. The watermarks can be directly added to real flows, collected in real networks for a different purpose, or they can be synthetically generated.

There are three main steps in the evaluation process of a watermarking system which utilizes real-time experiments:

- Embedding the watermarker and the traffic sniffer in the real network; embedding any potential component of an active attack against watermarking.
- Conducting the experiment a number of times based on different settings in the real system; collecting both watermarked and unwatermarked flows.
- Applying watermark detection algorithms on the collected flows to evaluate the detection accuracy; calculating other performance metrics.

Some examples of real-time experiments can be found in a few works proposed by Wang *et al.* [37], Huang *et al.* [38], Ramsbrock *et al.* [44], and Zand *et al.* [46]. In some cases the experiments are embedded in the real world [37], [38], [44], [46], while in other works, the tests are conducted in a controlled and closed environment [83], [92]. We also distinguish experiments where the traffic is synthetically generated [83], from experiments with real traffic [37], [38], [87].

The bulk of the research found in the literature is based on off-line experiments. To conduct these experiments, traces from Bell Labs [81], [91], [95], [98] and CAIDA [35], [83], [94], [126] are used in some works. In other cases traces are synthetically generated [35], [81], [95].

A. Metrics

The quality of a watermarking system can be evaluated according to the capability of detecting watermarked flow, the watermark invisibility, and the robustness against any of the attacks described in Section IV-B.

In the following subsections the more common evaluation metrics are listed, and Table III shows a comparison among the metrics measured by the main watermarking methods.

1) Accuracy: Metrics used to evaluate the accuracy of a network flow watermarking system are the same as in most common cases of classification problems. In particular, "true positive" (TP) and "false positive" (FP) rates are the two metrics most suitable for the problem under review, and this is confirmed by Table III which shows that TP and FP are the most commonly used metrics in the literature. In a few cases "false negative" (FN) rates are also analyzed.

In order to evaluate the trade-off between FP and FN, some authors have measured the cross-over error rate (COER) which is defined as the value of a system parameter calculated when the FP rate equals the FN rate.

When a watermark is implemented as a codeword composed of *M* bits, the accuracy may be measured per bit and evaluated as the percentage of bits correctly (or incorrectly) identified.

2) Invisibility: The Kolmogorov-Smirnov (KS) test is a common instrument used to evaluate the invisibility of a watermarking algorithm [35], [45], [63], [64], [83]. In fact, roughly speaking, the KS test provides a measure of the distance

TABLE IV
PERFORMANCE COMPARISON OF THE MAIN ALGORITHMS IN THE LITERATURE. WHEN INSUFFICIENT INFORMATION WAS FOUND, "–" IS INSERTED

		Ideal conditions		Perturbed traffic		
	#pkts	TP (%)	FP (%)	TP (%)	FP (%)	type of attack
Wang, 2001 (SLEEPY) [32]	–	100	–	–	–	–
Wang, 2003 [81]	300	100	0.4	84	–	timing perturbation (up to 1s)
Peng, 2005 [91]	1500	–	–	100	0.5	chaff packet injection & timing perturbation (up to 7s)
				100	0	chaff packet injection
				44	0	timing perturbation (up to 7s)
Wang, 2005 [60]	1200	100	0.1	–	–	–
Wang, 2007 [37]	512	100	0.3	100	0.5	chaff packet injection
				79	–	flow splitting (3 subflows)
				20	–	timing perturbation (up to 0.5s)
Yu, 2007 [75]	3900	100	0.01	–	–	–
Pyun, 2007 [92]	400	100	0.5	68	1	timing perturbation (up to 2s)
				94	1	repacketization
Ramsbrock, 2008 [44]	–	100	–	–	–	–
Houmansadr, 2009 (RAINBOW) [83]	500	75	0	50	0.8	chaff packets & packet losses (5%)
Houmansadr, 2009 [93]	–	–	–	–	–	–
Deng, 2009 [89]	–	100	–	40	–	packet losses (30%)
Houmansadr, 2011 (SWIRL) [94]	1280	100	0.001	–	–	–
Wang, 2011 [95]	600	100	0.5	40	–	timing perturbation (up to 1.4s)
Bates, 2012 [63]	–	–	–	–	–	–
Houmansadr, 2012 (BOTMOSAIC) [45]	–	–	–	–	–	–
Yu, 2013 [96]	–	100	4	45	25	MSAC attack
Ling, 2013 [58]	–	100	0	–	–	–

between two distribution functions. If the set of statistical features of watermarked and unwatermarked flows are drawn from two distribution functions that look similar, the watermarking system can be considered invisible.

Checking the vulnerability of the watermarking algorithm against the multi-flow attack is another way to evaluate the invisibility [35], [110].

3) Robustness: In order to evaluate the robustness of the watermarking system, some research groups compared the performance obtained in cases of ideal conditions and under some selected types of attack (see Section IV-B).

In a few cases, the robustness analysis is made in an analytical form [81], [95].

4) Other Metrics: Computational cost and memory usage are two main aspects to be evaluated for the effective realization of a watermarking system. Latency, i.e., the time taken to process and forward IP packets, is a metric used for the computational cost evaluation at the watermarker, while the number of packets required to make a decision is used to evaluate the computational cost at the watermark detector device. Energy consumption is a critical aspect in a wireless sensor network which is analyzed in only a few cases [89].

B. Results Comparison

Some of the performance results obtained by the watermarking algorithms are compared in Table IV. We have extracted the numerical values that describe each algorithm provided in the papers.[1] Unfortunately, the comparison is difficult due to the different evaluation processes involved. Nevertheless, we believe that even this comparison may provide interesting insights.

The first column of the table lists the number of packets that must be observed by the watermark detector before making a decision. This measure gives an estimation of the computational cost of the detector device. In addition, when the required number of packets is small enough, watermarked flows can be detected online, and some kind of proactive action can be taken. The solutions listed do not require many packets in order to make a decision (usually less than 1500 packets, and in some cases only few hundred packets are needed). This represents one of the strength of network flow watermarking.

The TP and FP rates for ideal conditions are shown in the second and third columns. Almost all of the algorithms obtain TP rates equal to 100% which means that the detector recognizes all of the watermarked flows, and this is an important result, especially when compared with passive analysis algorithms. Excellent performance is also obtained for FP rates which are almost always below 1%.

The fourth and fifth columns show the TP and FP rates in cases in which the traffic flows are perturbed with one of the attacks described in Section IV-B. The specific attack tested by each work is listed in the last column. Although the comparison is weak, because each solution has been tested against different types of attacks, we can observe that in about half of the cases, the TP rates drastically drop to a level below 50%. This aspect highlights the lack of robustness of most algorithms. In a few cases the accuracy remains high, but the fact remains that only a subset of attacks have been tested. FP rates remain very low for all of the algorithms.

VI. DISCUSSION AND OPEN PROBLEMS

Watermarking applied to Internet flows is a new branch of TA which offers many directions to pursue. In the last few years watermarking techniques have been shown to be

[1] The numerical values shown in Table IV are estimates extracted from the results plotted in graphs by authors in their papers. For this reason they might be subject to imprecision because of human error.

an interesting and promising instrument to use to address a variety of problems in the TA field. Many papers in the literature have demonstrated its potential to assist Internet service providers (ISPs) with malicious behavior detection, tracing malicious flows, identifying the sources of attacks, etc. This area has been also investigated in order to better understand how it can be exploited for malicious purposes such as eavesdropping sensitive information from the data transported on observed flows; awareness of these issues can help researchers develop solutions to mitigate this type of threat.

The algorithms developed so far have, for the most part, shown excellent performance results with high accuracy in correctly identifying embedded watermarks and very low values of false positives. Unfortunately, the other side of the coin shows a set of vulnerabilities which require further investigation. Several attempts have been made to overcome the invisibility and robustness issues. Our analysis has shown that some watermarking systems are indeed robust against the natural noisy behavior of the network, but these same systems are not robust enough when traffic flow perturbations are deliberately injected into the network by an adversary in order to weaken watermark recognition. In addition, attempts to devise algorithms to generate invisible watermarks have been unsuccessful; in fact although some works claimed to have accomplished this, subsequent works demonstrated their lack of invisibility.

Experience tells us that it is not sufficient to test network flow watermarking only on current attacks and threat landscapes. We believe that in order to improve upon present solutions, the next step is to create watermarks that can be formally demonstrated to be robust and invisible. Robustness and invisibility are two properties that contrast with each other, and for this reason they should be studied and analyzed together, as opposed to separately as has been so far.

An important issue that has been raised in the literature is the absence of common procedures and metrics for evaluating the algorithms. Historically, each research group has tested its algorithms in a individualized fashion, and this makes comparison of proposed solutions challenging. In addition, some significant metrics have received limited attention including, for example, complexity, memory consumption, overhead, etc. Our opinion is that much more effort needs to be put into the evaluation of these metrics in order to investigate the impact of watermarking on the systems in which it is embedded.

One additional future research direction includes the deployment of watermark detectors in the network. The ideal solution would be to implement a detector in each link of the network under observation, but it is clear that this solution is not scalable. Finding the optimal deployment of monitoring nodes is still an open problem not adequately investigated.

VII. Conclusion

Traffic analysis has become an important tool, widely used in the network traffic engineering field during the last two decades. Internet service providers increasingly take advantage of TA techniques to support tasks such as network administration, traffic shaping/policing, diagnostic monitoring, provisioning, resource management, security issue detection, and improving the reliability of the NIDS (network intrusion detection system).

In this paper we have investigated the use of watermarking in traffic analysis. TA tools can be made easier and more robust by embedding a specific and recognizable pattern (watermark) in flow features. The literature contains a significant amount of work regarding this branch of TA, and we have collected, analyzed, and categorized these works. In particular, we have pointed out the objectives and problems addressed by network flow watermarking, highlighting the differences between malicious and legitimate objectives. We have also provided a detailed description of the general architecture to create a network flow watermarking system. The analysis shows that one of the most important aspects in the design of such systems is the choice of the watermark carrier.

Although a watermarking system can be characterized by low complexity in terms of both architectural and algorithmic aspects, watermarks should possess two important proprieties, i.e., robustness and invisibility. These two properties make the creation of watermarking algorithms more challenging. To the best of our knowledge, no watermarking algorithms are really invisible, and only one algorithm can be considered robust against some active attacks. There is still much work to do in order to achieve satisfactory results.

References

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, “A secure, robust watermark for multimedia,” in *Proc. 1st Int. Workshop Inf. Hiding*, Cambridge, U.K., 1996, pp. 185–206.

[2] A. Z. Tirkel *et al.*, *Electronic Watermark*, Aust. Pattern Recognit. Soc., Canberra, ACT, Australia, 1993, pp. 666–673.

[3] V. M. Potdar, S. Han, and E. Chang, “A survey of digital image watermarking techniques,” in *Proc. 3rd Int. Conf. Ind. Informat. (INDIN)*, Perth, WA, Australia, 2005, pp. 709–716.

[4] S.-J. Lee and S.-H. Jung, “A survey of watermarking techniques applied to multimedia,” in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, Busan, South Korea, 2001, pp. 272–277.

[5] P. Singh and R. S. Chadha, “A survey of digital watermarking techniques, applications and attacks,” *Int. J. Eng. Innov. Technol.*, vol. 2, no. 9, pp. 165–175, Mar. 2013.

[6] C. Collberg and C. Thomborson, “Software watermarking: Models and dynamic embeddings,” in *Proc. 26th ACM SIGPLAN-SIGACT Symp. Principles Program. Lang. (POPL)*, San Antonio, TX, USA, 1999, pp. 311–324.

[7] J. Palsberg *et al.*, “Experience with software watermarking,” in *Proc. 16th Annu. Conf. Comput. Security Appl. (ACSAC)*, New Orleans, LA, USA, 2000, pp. 308–316.

[8] W. Zhu, C. Thomborson, and F.-Y. Wang, “A survey of software watermarking,” in *Proc. Intell. Security Informat. Conf. (ISI)*, 2005, pp. 454–458.

[9] M. Voigt and C. Busch, “Watermarking 2D-vector data for geographical information systems,” in *Proc. Int. Symp. Electron. Imag.*, 2002, pp. 621–628.

[10] R. Ohbuchi, H. Ueda, and S. Endoh, “Robust watermarking of vector digital maps,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Lausanne, Switzerland, 2002, pp. 577–580.

[11] Z. Jalil and A. M. Mirza, “A review of digital watermarking techniques for text documents,” in *Proc. Int. Conf. Inf. Multimedia Technol. (ICIMT)*, 2009, pp. 230–234.

[12] M. A. Qadir and I. Ahmad, “Digital text watermarking: Secure content delivery and data hiding in digital documents,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 21, no. 11, pp. 18–21, Nov. 2006.

[13] A. B. Kahng *et al.*, “Robust IP watermarking methodologies for physical design,” in *Proc. 35th Annu. Design Autom. Conf.*, San Francisco, CA, USA, 1998, pp. 782–787.

[14] A. B. Kahng *et al.*, "Constraint-based watermarking techniques for design IP protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1236–1252, Oct. 2001.

[15] M. Engin, O. Çıdam, and E. Z. Engin, "Wavelet transformation based watermarking technique for human electrocardiogram (ECG)," *J. Med. Syst.*, vol. 29, no. 6, pp. 589–594, 2005.

[16] A. Ibaida, I. Khalil, and D. Al-Shammary, "Embedding patients confidential data in ECG signal for healthcare information systems," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Buenos Aires, Argentina, 2010, pp. 3891–3894.

[17] W. Wei, X. Zhang, W. Shi, S. Lian, and D. Feng, "Network traffic monitoring, analysis and anomaly detection [Guest Editorial]," *IEEE Network*, vol. 25, no. 3, pp. 6–7, Aug. 2011.

[18] E. Biersack, C. Callegari, and M. Matijasevic, *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*. Heidelberg, Germany: Springer, 2013.

[19] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[20] H. Kim *et al.*, "Internet traffic classification demystified: Myths, caveats, and the best practices," in *Proc. ACM Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Madrid, Spain, 2008, pp. 1–12.

[21] A. Callado *et al.*, "A survey on Internet traffic identification," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 37–52, 3rd Quart., 2009.

[22] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Netw.*, vol. 26, no. 1, pp. 35–40, Jan./Feb. 2012.

[23] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Contextual localization through network traffic analysis," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, 2014, pp. 925–933.

[24] C. Callegari *et al.*, *A Methodological Overview on Anomaly Detection*. Heidelberg, Germany: Springer, 2013.

[25] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[26] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proc. 2nd Int. Workshop Privacy Enhancing Technol. (PET)*, San Francisco, CA, USA, 2002, pp. 171–178.

[27] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob?" in *Proc. 16th USENIX Security Symp.*, Boston, MA, USA, 2007, pp. 43–54.

[28] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose, "Phonotactic reconstruction of encrypted VoIP conversations: Hookt on Fon-iks," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, 2011, pp. 3–18.

[29] B. W. Lampson, "A note on the confinement problem," *Commun. ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[30] D. C. Latham, *Trusted Computer System Evaluation Criteria*. Dept. Defense, Arlington County, VA, USA, 1986.

[31] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 3, pp. 44–57, 3rd Quart., 2007.

[32] X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework," in *Proc. Trusted Inf. New Decade Challenge IFIP TC11 16th Annu. Working Conf. Inf. Security (IFIP/SEC)*, Paris, France, 2001, pp. 369–384.

[33] T. Lu, R. Guo, L. Zhao, and Y. Li, "A systematic review of network flow watermarking in anonymity systems," *Int. J. Security Appl.*, vol. 10, no. 3, pp. 129–138, 2016.

[34] L.-C. Zhang, Z.-X. Wang, and H.-S. Liu, "Survey on network flow watermarking technologies," *Comput. Sci.*, vol. 38, no. 11, Nov. 2011.

[35] X. Gong, M. Rodrigues, and N. Kiyavash, "Invisible flow watermarks for channels with dependent substitution, deletion, and bursty insertion errors," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1850–1859, Nov. 2013.

[36] L. Zhang, Z. Wang, Y. Wang, and H. Liu, "Interval-based spread spectrum watermarks for tracing multiple network flows," in *Proc. 12th IEEE Int. Conf. Commun. Technol. (ICCT)*, Nanjing, China, 2010, pp. 393–396.

[37] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proc. IEEE Symp. Security Privacy (SP)*, Oakland, CA, USA, 2007, pp. 116–130.

[38] J. Huang, X. Pan, X. Fu, and J. Wang, "Long PN code based DSSS watermarking," in *Proc. 30th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Shanghai, China, 2011, pp. 2426–2434.

[39] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, Feb. 2013.

[40] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Comput. Surveys*, vol. 45, no. 4, 2013, Art. no. 45.

[41] M. Mahmoud, M. Nir, and A. Matrawy, "A survey on botnet architectures, detection and defences," *Int. J. Netw. Security*, vol. 17, no. 3, pp. 272–289, 2015.

[42] L. Zhang, S. Yu, D. Wu, and P. Watters, "A survey on latest botnet attack and defense," in *Proc. IEEE 10th Int. Conf. Trust Security Privacy Comput. Commun. (TrustCom)*, Changsha, China, 2011, pp. 53–60.

[43] A. Karim *et al.*, "Botnet detection techniques: Review, future trends, and issues," *J. Zhejiang Univ. SCIENCE C*, vol. 15, no. 11, pp. 943–983, 2014.

[44] D. Ramsbrock, X. Wang, and X. Jiang, "A first step towards live botmaster traceback," in *Proc. 11th Int. Symp. Recent Adv. Intrusion Detection (RAID)*, Cambridge, MA, USA, 2008, pp. 59–77.

[45] A. Houmansadr and N. Borisov, "BotMosaic: Collaborative network watermark for the detection of IRC-based botnets," *J. Syst. Softw.*, vol. 86, no. 3, pp. 707–715, 2013.

[46] A. Zand, G. Vigna, R. A. Kemmerer, and C. Kruegel, "Rippler: Delay injection for service dependency detection," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, 2014, pp. 2157–2165.

[47] P. Bahl *et al.*, "Discovering dependencies for network management," in *Proc. 5th Workshop Hot Topics Netw. (Hotnets-V)*, 2006, pp. 97–102.

[48] P. Bahl *et al.*, "Towards highly reliable enterprise network services via inference of multi-level dependencies," in *Proc. Conf. Appl. Technol. Architect. Protocols Comput. Commun. (SIGCOMM)*, Kyoto, Japan, 2007, pp. 13–24.

[49] X. Chen, M. Zhang, Z. M. Mao, and P. Bahl, "Automating network application dependency discovery: Experiences, limitations, and new solutions," in *Proc. 8th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, San Diego, CA, USA, 2008, pp. 117–130.

[50] S. Kandula, R. Chandra, and D. Katabi, "What's going on?: Learning communication rules in edge networks," in *Proc. Conf. Appl. Technol. Architect. Protocols Comput. Commun. (SIGCOMM)*, Seattle, WA, USA, 2008, pp. 87–98.

[51] A. Natarajan, P. Ning, Y. Liu, S. Jajodia, and S. E. Hutchinson, "NSDMiner: Automated discovery of network service dependencies," in *Proc. Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, 2012, pp. 2507–2515.

[52] L. Popa, B.-G. Chun, I. Stoica, J. Chandrashekar, and N. Taft, "Macroscope: End-point approach to networked application dependency discovery," in *Proc. Conf. Emerg. Netw. Experiments Technol. (CoNEXT)*, Rome, Italy, 2009, pp. 229–240.

[53] R. Fonseca, G. Porter, R. H. Katz, S. Shenker, and I. Stoica, "X-trace: A pervasive network tracing framework," in *Proc. 4th Symp. Netw. Syst. Design Implement. (NSDI)*, Cambridge, MA, USA, 2007, pp. 271–284.

[54] A. Brown, G. Kar, and A. Keller, "An active approach to characterizing dynamic dependencies for problem determination in a distributed environment," in *Proc. IEEE/IFIP Int. Symp. Integr. Netw. Manag. (IM)*, Seattle, WA, USA, 2001, pp. 377–390.

[55] Q. Sun *et al.*, "Statistical identification of encrypted Web browsing traffic," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, 2002, pp. 19–30.

[56] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy vulnerabilities in encrypted HTTP streams," in *Proc. 5th Int. Workshop Privacy Enhancing Technol. (PET)*, Cavtat, Croatia, 2005, pp. 1–11.

[57] K.-W. Lim, S. Secci, L. Tabourier, and B. Tebbani, "Characterizing and predicting mobile application usage," *Comput. Commun.*, to be published.

[58] Z. Ling *et al.*, "Novel packet size-based covert channel attacks against anonymizer," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2411–2426, Dec. 2013.

[59] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations," in *Proc. IEEE Symp. Security Privacy (SP)*, Oakland, CA, USA, 2008, pp. 35–49.

[60] X. Wang, S. Chen, and S. Jajodia, "Tracking anonymous peer-to-peer VoIP calls on the Internet," in *Proc. 12th ACM Conf. Comput. Commun. Security (CCS)*, Alexandria, VA, USA, 2005, pp. 81–91.

[61] H. Sengar, Z. Ren, H. Wang, D. Wijesekera, and S. Jajodia, "Tracking Skype VoIP calls over the Internet," in *Proc. 29th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, USA, 2010, pp. 96–100.

[62] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proc. 16th ACM Conf. Comput. Commun. Security (CCS)*, Chicago, IL, USA, 2009, pp. 199–212.

[63] A. Bates *et al.*, "Detecting co-residency with active traffic analysis techniques," in *Proc. ACM Workshop Cloud Comput. Security (CCSW)*, Hangzhou, China, 2012, pp. 1–12.

[64] A. Bates *et al.*, "On detecting co-resident cloud instances using network flow watermarking techniques," *Int. J. Inf. Security*, vol. 13, no. 2, pp. 171–189, 2014.

[65] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proc. 13th USENIX Security Symp.*, San Diego, CA, USA, 2004, pp. 303–320.

[66] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web transactions," *ACM Trans. Inf. Syst. Security*, vol. 1, no. 1, pp. 66–92, 1998.

[67] *Anonymizer Inc*. Accessed on Feb. 28, 2016. [Online]. Available: www.anonymizer.com

[68] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *J. Cryptol.*, vol. 1, no. 1, pp. 65–75, 1988.

[69] K. Bennett and C. Grothoff, "Gap–practical anonymous networking," in *Proc. 3rd Int. Workshop Privacy Enhancing Technol. (PET)*, Dresden, Germany, 2003, pp. 141–160.

[70] C. Shields and B. N. Levine, "A protocol for anonymous communication over the Internet," in *Proc. 7th ACM Conf. Comput. Commun. Security (CCS)*, Athens, Greece, 2000, pp. 33–42.

[71] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against Tor," in *Proc. ACM Workshop Privacy Electron. Soc. (WPES)*, Alexandria, VA, USA, 2007, pp. 11–20.

[72] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *Proc. IEEE Symp. Security Privacy (SP)*, Oakland, CA, USA, 2005, pp. 183–195.

[73] R. Pries, W. Yu, X. Fu, and W. Zhao, "A new replay attack against anonymous communication networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Beijing, China, 2008, pp. 1578–1582.

[74] J. A. Elices and F. Pérez-González, "Fingerprinting a flow of messages to an anonymous server," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2012, pp. 97–102.

[75] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "DSSS-based flow marking technique for invisible traceback," in *Proc. IEEE Symp. Security Privacy (SP)*, Oakland, CA, USA, 2007, pp. 18–32.

[76] E. Chan-Tin, J. Shin, and J. Yu, "Revisiting circuit clogging attacks on Tor," in *Proc. 8th Int. Conf. Availabil. Rel. Security (ARES)*, Regensburg, Germany, 2013, pp. 131–140.

[77] Z. Ling, J. Luo, K. Wu, W. Yu, and X. Fu, "TorWard: Discovery, blocking, and traceback of malicious traffic over Tor," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2515–2530, Dec. 2015.

[78] Z. Ling *et al.*, "Protocol-level attacks against Tor," *Comput. Netw.*, vol. 57, no. 4, pp. 869–886, 2013.

[79] D. Arp, F. Yamaguchi, and K. Rieck, "Torben: A practical side-channel attack for deanonymizing Tor communication," in *Proc. 10th ACM Symp. Inf. Comput. Commun. Security (ASIA CCS)*, Singapore, 2015, pp. 597–602.

[80] A. Houmansadr and N. Borisov, "The need for flow fingerprints to link correlated network flows," in *Proc. 13th Int. Symp. Privacy Enhancing Technol. (PETS)*, Bloomington, MN, USA, 2013, pp. 205–224.

[81] X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *Proc. 10th ACM Conf. Comput. Commun. Security (CCS)*, Washington, DC, USA, 2003, pp. 20–29.

[82] A. Houmansadr, N. Kiyavash, and N. Borisov, "Non-blind watermarking of network flows," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1232–1244, Aug. 2014.

[83] A. Houmansadr, N. Kiyavash, and N. Borisov, "Rainbow: A robust and invisible non-blind watermark for network flows," in *Proc. 16th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2009, pp. 1–13.

[84] B. Assanovich, W. Puech, and I. Tkachenko, "Use of linear error-correcting subcodes in flow watermarking for channels with substitution and deletion errors," in *Proc. 14th IFIP TC 6/TC 11 Int. Conf. Commun. Multimedia Security (CMS)*, Magdeburg, Germany, 2013, pp. 105–112.

[85] L. Zhang, Z. Wang, Q. Wang, and F. Miao, "MSAC and multi-flow attacks resistant spread spectrum watermarks for network flows," in *Proc. 2nd IEEE Int. Conf. Inf. Financ. Eng. (ICIFE)*, 2010, pp. 438–441.

[86] Z. Liu, J. Jing, and P. Liu, "Rate-based watermark traceback: A new approach," in *Proc. 6th Int. Conf. Inf. Security Pract. Exp. (ISPEC)*, Seoul, South Korea, 2010, pp. 172–186.

[87] X. Pan, J. Huang, Z. Ling, B. Lu, and X. Fu, "Long PN code based traceback in wireless networks," *Int. J. Performability Eng.*, vol. 8, no. 2, pp. 173–182, 2012.

[88] X. Li, C. Yu, M. Hizlan, W.-T. Kim, and S. Park, "Physical layer watermarking of direct sequence spread spectrum signals," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, San Diego, CA, USA, 2013, pp. 476–481.

[89] H. Deng, X. Sun, B. Wang, and Y. Cao, "Selective forwarding attack detection using watermark in WSNs," in *Proc. ISECS Int. Colloq. Comput. Commun. Control Manag. (CCCM)*, Sanya, China, 2009, pp. 109–113.

[90] Y.-S. Choi, D.-I. Seo, S.-W. Sohn, and S.-H. Lee, "Network-based real-time connection traceback system (NRCTS) with packet marking technology," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, Montreal, QC, Canada, 2003, pp. 31–40.

[91] P. Peng, P. Ning, D. S. Reeves, and X. Wang, "Active timing-based correlation of perturbed traffic flows with chaff packets," in *Proc. 25th IEEE Int. Conf. Distrib. Comput. Syst. Workshops (ICDCS)*, Columbus, OH, USA, 2005, pp. 107–113.

[92] Y. J. Pyun, Y. H. Park, X. Wang, D. S. Reeves, and P. Ning, "Tracing traffic through intermediate hosts that repacketize flows," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2007, pp. 634–642.

[93] A. Houmansadr, N. Kiyavash, and N. Borisov, "Multi-flow attack resistant watermarks for network flows," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2009, pp. 1497–1500.

[94] A. Houmansadr and N. Borisov, "SWIRL: A scalable watermark to detect correlated network flows," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2011, pp. 1–15.

[95] X. Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by flow watermarking," *IEEE Trans. Depend. Secure Comput.*, vol. 8, no. 3, pp. 434–449, May/Jun. 2011.

[96] W. Yu *et al.*, "On effectiveness of hopping-based spread spectrum techniques for network forensic traceback," in *Proc. 14th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distrib. Comput. SNPD*, Honolulu, HI, USA, 2013, pp. 101–106.

[97] V. Patil and R. Gawande, "Effective correlation in stepping stones by flow watermarking in encrypted attack traffic," *Int. J. Elect. Electron. Comput. Eng.*, vol. 3, no. 1, pp. 173–176, 2014.

[98] Y. H. Park and D. S. Reeves, "Adaptive watermarking against deliberate random delay for attack attribution through stepping stones," in *Proc. 9th Int. Conf. Inf. Commun. Security (ICICS)*, Zhengzhou, China, 2007, pp. 1–15.

[99] Z. Pan, H. Peng, X. Long, C. Zhang, and Y. Wu, "A watermarking-based host correlation detection scheme," in *Proc. Int. Conf. Manag. e-Commer. e-Government (ICMECG)*, 2009, pp. 493–497.

[100] X. Wang, J. Luo, and M. Yang, "An interval centroid based spread spectrum watermark for tracing multiple network flows," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, San Antonio, TX, USA, 2009, pp. 4000–4006.

[101] X. Wang, J. Luo, and M. Yang, "A double interval centroid-based watermark for network flow traceback," in *Proc. 14th Int. Conf. Comput. Supported Cooperative Work Design (CSCWD)*, Shanghai, China, 2010, pp. 146–151.

[102] J. Luo, X. Wang, and M. Yang, "An interval centroid based spread spectrum watermarking scheme for multi-flow traceback," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 60–71, 2012.

[103] L. Zhang, Z. Wang, J. Xu, and Q. Wang, "Multi-flow attack resistant interval-based watermarks for tracing multiple network flows," in *Proc. Int. Conf. Comput. Intell. Syst. (ICCIC)*, Wuhan, China, 2011, pp. 166–173.

[104] Y. J. Pyun, Y. Park, D. S. Reeves, X. Wang, and P. Ning, "Interval-based flow watermarking for tracing interactive traffic," *Comput. Netw.*, vol. 56, no. 5, pp. 1646–1665, 2012.

[105] X. Wang, J. Luo, and M. Yang, "An efficient sequential watermark detection model for tracing network attack flows," in *Proc. IEEE 16th Int. Conf. Comput. Supported Cooperative Work Design (CSCWD)*, Wuhan, China, 2012, pp. 236–243.

[106] X. Wang, M. Yang, and J. Luo, "A novel sequential watermark detection model for efficient traceback of secret network attack flows," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1660–1670, 2013.

[107] O. Goldreich, *The Foundations of Cryptography*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[108] P. Peng, P. Ning, and D. S. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, 2006, pp. 334–349.

[109] A. Wald, *Sequential Analysis*. North Chelmsford, MA, USA: Courier Corp., 1973.

[110] N. Kiyavash, A. Houmansadr, and N. Borisov, "Multi-flow attacks against network flow watermarking schemes," in *Proc. 17th USENIX Security Symp.*, Washington, DC, USA, 2008, pp. 307–320.

[111] X. Luo, J. Zhang, R. Perdisci, and W. Lee, "On the secrecy of spread-spectrum flow watermarks," in *Proc. 15th Eur. Symp. Res. Comput. Security (ESORICS)*, Athens, Greece, 2010, pp. 232–248.

[112] X. Luo *et al.*, "Exposing invisible timing-based traffic watermarks with backlit," in *Proc. 27th Annu. Comput. Security Appl. Conf. (ACSAC)*, Orlando, FL, USA, 2011, pp. 197–206.

[113] D. M. J. Tax, *One-Class Classification*. TU Delft, Delft Univ. Technol., Delft, The Netherlands, 2001.

[114] Z. Lin and N. Hopper, "New attacks on timing-based network flow watermarks," in *Proc. 21st USENIX Security Symp.*, Bellevue, WA, USA, 2012, pp. 381–396.

[115] W. Jia *et al.*, "Blind detection of spread spectrum flow watermarks," *Security Commun. Netw.*, vol. 6, no. 3, pp. 257–274, 2013.

[116] J. D. Padhye, K. Kothari, M. Venkateshaiah, and M. Wright, "Evading stepping-stone detection under the cloak of streaming media with sneak," *Comput. Netw.*, vol. 54, no. 13, pp. 2310–2325, 2010.

[117] A. Iacovazzi and A. Baiocchi, "Investigating the trade-off between overhead and delay for full packet traffic privacy," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, 2013, pp. 1345–1350.

[118] D. L. Donoho *et al.*, "Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," in *Proc. 5th Int. Symp. Recent Adv. Intrusion Detection (RAID)*, Zürich, Switzerland, 2002, pp. 17–35.

[119] A. Iacovazzi and A. Baiocchi, "Protecting traffic privacy for massive aggregated traffic," *Comput. Netw.*, vol. 77, pp. 1–17, Feb. 2015.

[120] A. Adelsbach, S. Katzenbeisser, and H. Veith, "Watermarking schemes provably secure against copy and ambiguity attacks," in *Proc. 3rd ACM Workshop Digit. Rights Manag.*, Washington, DC, USA, 2003, pp. 111–119.

[121] S. Kent, "IP encapsulating security payload (ESP)," RFC 4303, The Internet Society, Reston, VA, USA, 2005.

[122] A. Iacovazzi and A. Baiocchi, "Optimum packet length masking," in *Proc. 22nd Int. Teletraffic Congr. (ITC)*, Amsterdam, The Netherlands, 2010, pp. 1–8.

[123] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2009, pp. 1–14.

[124] J. Jin and X. Wang, "On the effectiveness of low latency anonymous network in the presence of timing attack," in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, Lisbon, Portugal, 2009, pp. 429–438.

[125] T. Ylonen and C. Lonvick, "The secure shell (SSH) protocol architecture," RFC 4251, The Internet Society, Reston, VA, USA, 2006.

[126] J. I. Gilbert, D. J. Robinson, J. W. Butts, and T. H. Lacey, "Scalable wavelet-based active network detection of stepping stones," in *Proc. Cyber Sens.*, vol. 8408. Baltimore, MD, USA, 2012, pp. 1–13.

Alfonso Iacovazzi received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in information and communication engineering from the Sapienza University of Rome, Italy, in 2008 and 2013, respectively. He is currently a Post-Doctoral Research Fellow with Temasek Laboratories, Singapore University of Technology and Design. His main research interests include communication security and privacy, traffic analysis and monitoring, anomaly detection, and traffic anonymization.

Yuval Elovici received the B.Sc. and M.Sc. degrees in computer and electrical engineering from the Ben-Gurion University of the Negev (BGU) and the Ph.D. degree in information systems from Tel-Aviv University. He is the Director of the Telekom Innovation Laboratories, BGU, the Head of BGU Cyber Security Research Center, the Research Director of iTrust, SUTD, the SUTD Director of ST Electronics-SUTD Cyber Security Laboratory, and a Professor with the Department of Information Systems Engineering, BGU. He has published articles in leading peer-reviewed journals and in various peer-reviewed conferences. In addition, he has co-authored a book on social network security and a book on information leakage detection and prevention. His primary research interests are computer and network security, cyber security, web intelligence, information warfare, social network analysis, and machine learning. He also consults professionally in the area of cyber security and is the Co-Founder of Morphisec, a startup company that develop innovative cyber-security mechanisms that relate to moving target defense.