



Modern Traffic Analysis and its Capabilities

By **Riccardo Bettati**

Department of Computer Science and Engineering and
Center for Information Assurance and Security

Texas A&M University

College Station, TX, 77840-3112, USA

e-mail: bettati@cse.tamu.edu

Abstract

Traffic analysis of encrypted communication in wired and wireless internetworks has developed into a very powerful tool for a variety of users and settings. For example, it has been used to break encryption in ssh sessions, re-construct speech in encrypted VoIP systems, break anonymity systems, remotely identify honeypots and various other bots, and identify and isolate participants in anonymized wireless networks.

In this paper we first survey threats to confidentiality and security in general posed by traffic analysis based attacks. We then proceed to discuss countermeasures and the difficulties to correctly implement them. We will identify situations in which the naive application of traffic-analysis countermeasures is detrimental to security and confidentiality.

Traffic analysis has been traditionally considered to be ineffective when large numbers of traffic flows are aggregated and are indistinguishable. We will show how modern signal processing techniques enable the pre-conditioning collected traffic data in order to separate large aggregates of flows into their individual components.

Finally, we will illustrate a family of attacks that renders anonymity measures in wireless networks largely ineffective.

1. Introduction

Traffic analysis has developed into a very powerful tool for a variety of users and settings: Increasingly, fine-grained analysis of activity and traffic in networks is being used to classify customers, to classify players in multiplayer games [1], to detect many forms of bots [1, 2], to compromise anonymous communication, and for many other objectives.

In many settings, one can make use of only a limited amount of information in the observed traffic streams: First, packet data is typically encrypted and therefore not directly available. Furthermore, header information is often of very limited use as well, because the real traffic is either tunneled, or because the sender appropriately modified the header information. After some initial filtering of traffic (e.g. traffic from trusted sources, or to ports that don't lend themselves to misuse), the observer is left primarily with *timing information* in the traffic only, and she must resort to *timing-based traffic analysis* in order to recover information about the communication or its participants.

Traffic analysis has many applications in system defense: In many large-scale applications we increasingly may want to be concerned about whether the participants in the systems are *bona-fide* users, as opposed to automated proxies, typically *bots*. Bots can occur in a variety of settings, such as game bots [1], fraudulent clickers, web crawlers, honeypots [2], and many others. When one has no access to all of the traffic generated by the bot node, *remote bot detection* schemes are needed to identify such bot processes, so that then appropriate measures can be taken¹. By their very nature, bots are designed to generate traffic that is indistinguishable from that of *bona-fide* users, and bot detectors must challenge suspected bots either at high level (traditional Turing test) or at very low level, to infer whether the timing behavior is triggered by user actions or by other timing control mechanisms [2].

Traffic analysis can clearly strengthen offensive capabilities as well: Operators of networks with *privacy or other confidentiality requirements* must take into account the possibility of observers collecting traffic data and applying similar timing-based traffic analysis techniques with the intent to violate some of the confidentiality. For example, designers of protocols for low-latency applications may need to be aware that user behavior (such as typing patterns in ssh [3]) may be directly detectable in the traffic pattern, with severe effects on confidentiality and system security. Similarly, anonymity protocols are inherently and particularly vulnerable to timing-based traffic analysis attacks (e.g., [4-6] and many others).

The objective of this paper is to illustrate the effectiveness of modern traffic analysis by briefly describing application scenarios across a number of widely disparate domains: We will first set the stage by describing how traffic analysis can be used to violate the security and confidentiality of networked applications. We will then proceed to illustrate the difficulties to counter traffic analysis attacks in internet network settings. Finally, we branch out to describe how fine grained traffic analysis can be married to modern signal processing algorithms to launch formulate powerful attacks in presumably protected wireless settings.

2. Traffic Analysis as Threat to Confidentiality and Security

In this section we use a small number of examples to illustrate the effectiveness of traffic analysis confidentiality mechanisms in a number of settings.

¹ We distinguish *remote bot detection* from traditional bot detection, for example bot detection in enterprise networks. In such systems, the bot detector has access to the entire behavior of the bot node. In our case, we must do with the *projected subset* of the behavior of the bot, for example the interaction of a remote game bot with one player.

2.1. Encrypted VoIP

A question of significant interest is how much information about a VoIP conversation can be re-constructed through traffic analysis of the encrypted packet stream. Such information can be the language spoken, the identity of the speaker, passive call-tracking, and speech re-construction.

In a preliminary study [7] we study an attacker who collects timestamps of packets on a link that carries a VoIP flow. The objective of the attacker is to reconstruct the original sequence of spoken words (symbols) from the collected estimated spurt lengths. For this, she computes the probability distribution, that is, the probability that symbol was spoken, given that a talk spurt of length was observed on the network, over all symbols for each talk spurt in the sequence.

We assume that the attacker has the opportunity to *train* against plaintext data by using a selection of voice signals, each of which represents one of the symbols. We assume that the attacker has sufficient samples of each symbol so as to be able to make statistically significant estimations. The encrypted transmission of the symbols over a VoIP channel can be represented as a *discrete-time Markov-Modulated Process*, where each state (representing the transmission of a symbol) triggers the (observable) emission of a spurt of length l . The *a priori* probability of sequences of symbols to appear in a text can be captured in form of *transition probabilities* between states in the Markov model. Since only the sequence of talk spurt lengths is observable, while the sequence of *transmitted* symbols remains hidden, we call the model a *Hidden Markov Model* (HMM) [8]. During the attack, the observer uses dynamic programming to estimate an optimal path through the states in the Markov model based on the observed spurt lengths. In this way she reconstructs the sequence of words transmitted over the VoIP channel.

We note that the attacker does not rely on packet content, or packet sizes in her attempt to reconstruct the sequence of words transmitted. Instead, she relies on packet timing information only, in order to identify talk spurt boundaries. This attack is therefore effective against encrypted VoIP systems with CBR codecs. Several variations of this attack on encrypted VoIP can be formulated, such as timing attacks on systems with VBR codecs (e.g. [9]), speaker identification, separation and traceback of VoIP flows, and others.

2.2. System Configuration Discovery

Traffic analysis methods (timing analysis in particular) can be used to passively infer information about remote components (hosts, routers, and switches) in the network infrastructure. Such information is opaque in nature: Network traffic for example very rarely explicitly carries information about the configuration of a sending host. In addition, an intruder host or a stepping stone process, as well as an unauthorized overlay-network node, will go to great lengths to hide its presence in the system.

We have argued very early on [10] that timing analysis of network traffic can be used to infer information about applications, system software, and the hardware configuration of a node that sends data. More recently, various forms of *fingerprinting* of physical devices [11, 12] and of device drivers [13] have attracted attention. Similarly, host identification based on thermal signatures reflected in the host's clock skew has been studied in [14].

Such fingerprinting can be used for intrusion detection [10]. In particular, application specific and system specific signatures can be defined using bounds on timing measures, and traffic flows that exceed these bounds can be flagged and appropriately dealt with. Similarly, timing analysis of traffic data can be used by attackers to acquire knowledge of a server configuration (hardware, system software), and to better tailor attacks. Our previous work [10] used deterministic *traffic bounding functions* [15, 16] and *empirical traffic envelopes* [16] as classifiers², but other measures can be used as well.

Preliminary informal measurements in our testbeds showed that different traffic timing features of a traffic stream can be used to determine different parameters of a source machine. For example, cross-correlation of frequency spectra of traffic streams can be used to distinguish Linux-based machines from Windows based ones over a wide variety of underlying hardware platforms (including various laptops and desktop machines). On the other hand, the pairwise mean-square-error between the measured traffic envelopes [16] clearly separates machines with different hardware configurations. Generally, frequency analysis exposes the timing control within the operating system, such as slot allocation by schedulers or timer and clock management by the system, or timing behavior that is dictated by feedback protocols, such as bandwidth availability or RTT in TCP/IP. Similarly, traffic envelope analysis appears to expose allocated resources within the system, for example buffers at various levels.

2.3. Bot Detection at a Distance

Most bots typically control the timing behavior of their traffic in order to emulate user behavior (game-bots or crawlers) or system-level latencies (in honeynets). Bot detectors in interactive settings, for example multiplayer games, therefore typically challenge the bot at a “semantically high” level, similar to a Turing test, for example through a separate chat channel. We showed in [2] that the implementation of the timer management mechanism of the operating system easily shows through in form of periodicities in inter-response times in honeynets. In the same work we also showed that simply fiddling with the timer resolution on the bot OS does not make the problem go away. We recently performed a similar analysis on gamebots in the Rangarok Online game and found a similar weakness³. These results lead us to believe that efficient “low-level Turing tests” can be developed that detect at a distance whether implementations of system-level mechanisms are genuine or whether they are approximated using emulation techniques.

3. Countermeasures

Given that the feature space exploited in the methods described above lies exclusively in the time domain, one can be easily misled to believe that simple perturbation of the timing behavior of activities and connection traffic is sufficient to prevent information leakage. This is often not the case, and we will illustrate this with two examples.

2 Traffic bounding functions are used often in the QoS literature to characterize the worst-case amount of traffic carried on a given flow. Leaky buckets are an example of such a function, where a (σ, ρ) bucket bounds the traffic to $\sigma + \rho * I$ units over any interval of length I . The empirical envelope is the experienced worst-case traffic, given in units over any interval of length I . Any valid bounding function bounds the empirical envelope.

3 Several Gamebots in the Rangarok Online game were analyzed in a similar fashion in [1]. In this work the authors measured round-trip time, as opposed to inter-response time of the gamebot, and therefore had a much weaker classifier. The authors developed strong classifiers by using other specific characteristics of the bots.

3.1. Effectiveness of Link Padding

The motivation of link padding is to ensure traffic flow confidentiality, i.e., to prevent the adversary from performing traffic analysis and inferring critical characteristics of the payload traffic exchanged over unprotected networks. We limit the adversary's interest to *payload traffic rate*, that is, the rate at which payload traffic is exchanged between protected subnets.

One way to counter the traffic analysis attacks is to "pad" the payload traffic, that is, to properly insert "dummy" packets in the payload traffic stream so that the real payload status is camouflaged. The most common method to implement padding uses a timer to control packet sending and works as follows: (a) On the padding node, incoming payload packets from the sender Alice are placed in a queue. (b) An interrupt-driven timer is set. When the timer times out, the interrupt processing routine checks if there is a payload packet in the queue: If there are payload packets, one is removed from the queue and sent to the receive. Otherwise, a dummy packet is sent. In order to simplify the network management, this timer often uses a constant interval, and periodically fires with a constant interval between two consecutive timeouts.

Unfortunately, such padding mechanisms are susceptible to traffic analysis. For example, we show in a series of experiments how link padders with constant timers fail to protect against passive traffic analysis [17] and random link padders against active traffic analysis [18].

3.2. Effectiveness of Mix-Based Anonymity Networks

Anonymity preserving technologies have been proposed and used to *mix* traffic in different ways to protect the privacy of anonymity network users, that is, to make senders and/or receivers non-identifiable. Current anonymity networks such as TOR [19] and Onion Routing [20] *mix* network traffic by aggregating a large number of network connections in the spirit of that it is easier to hide in crowds. Other mixing procedures such as batching, pooling, and re-ordering of packets may also be applied to mix traffic as it traverses the network.

Experiments [5] have shown that naive mixing in networks is largely ineffective against flow traceback attacks. More importantly, we have shown that batching of packets to prevent direct correlation in the time domain is actually detrimental for TCP traffic. Batching increases queuing variability, which in turn leads to a more visible timing footprint of TCP flows.

4. Traffic Analysis in Wireless Settings

We show in [21] that traditional schemes for anonymous communication in wireless settings, such as masking of MAC addresses and link padding with dummy traffic, are largely ineffective against statistical timing analysis of network traffic in terms of location privacy. We also found that motion privacy can not be protected as well.

In fact, an attacker can compromise - with the help of a collection of very simple sensors - the location privacy in a densely populated, *perfectly anonymized* wireless network. We call the sensors "simple" because they only need to monitor packets at MAC level or above, do not require directional capabilities, do not need to distinguish packets or relate network packets to senders or receivers, only require coarse time synchronization support, and require only low-bandwidth links for inter-sensor communication. (We don't need support for signal-strength measurement on the sensors either.) Such collections of sensors could be realized by a number of WLAN users that collude and exchange information, or by a separate infrastructure of sensor nodes, such as a sensor network. Given these limited required capabilities, we use the sensors to count packets over intervals of given length,

and to forward the resulting time series of packet counts for analysis to some central location. No information is available about how many nodes are present and sending in the area, and the anonymity measures in the WLAN prevent the sensors from distinguishing packets sent from different nodes.

We can use statistical signal analysis methods to (a) estimate the number of nodes in areas of the network (we call this *node density*) and to (b) separate the overall traffic into estimates of actual traffic sent by nodes in the network to pinpoint the location of sending nodes (*node location*). For the node-density estimation we use *Principal Component Analysis* (PCA). PCA is a classical statistical method used to reduce the dimensionality in a dataset. It can represent a dataset of correlated variables with less uncorrelated variables, which are called principal components. For the traffic separation we use the *Blind Source Separation* (BSS) method [22]. BSS was originally developed to solve the *cocktail party problem*, where the goal is to extract one person's voice signal given a mixtures of voices at a cocktail party. BSS algorithms solve the problem by taking advantage of the independence between voices from different persons. Similarly, in wireless networks, we can use BSS algorithms to separate traffic from different wireless nodes. The separated traffic is not in a form that can be directly associated with any sender node. However, we take advantage of spatial diversity in the collected data to reconstruct the sender location based on the separated traffic.

The poor performance of link-padding based anonymity protocols in wireless settings is due to a large part to the underlying carrier-sensing based MAC protocols, which perturbs the originally padded traffic, and so renders it susceptible to traffic analysis attacks. With privacy of users in mind, we will need to re-evaluate the use of carrier-sensing based versus scheduling based MAC protocols and how to trade-off privacy versus efficiency in such systems. One possible solution is to use TDMA-based MAC or hybrid protocols, such as Z-MAC [23], in order to trade-off between privacy and performance.

5. References

- [1] Kuan-Ta Chen, Jhih-Wei Jiang, Polly Huang, Hao-Hua Chu, Chin-Laung Lei, and Wen-Chin Chen, "Identifying mmorpg bots: a traffic analysis approach," in *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, New York, NY, USA, 2006, p. 4, ACM Press.
- [2] Xinwen Fu, Bryan Graham, Dan Cheng, Riccardo Bettati, and Wei Zhao, "Camouflaging virtual honeypots," Tech. Rep. TR-205-07-03, Department of Computer Science, Texas A&M University, <http://www.homepages.dsu.edu/fux/paper/camouflagingHoneyd.pdf>, July 2005.
- [3] Dawn Xiaodong Song, David Wagner, and Xuqing Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *10th USENIX Security Symposium*, Aug 2001.
- [4] George Danezis and Andrei Serjantov, "Statistical disclosure or intersection attacks on anonymity systems," in *Proceedings of 6th Information Hiding Workshop (IH 2004)*, Toronto, May 2004, LNCS.
- [5] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *4th Privacy Enhancement Technology Workshop (PET 2004)*, 2004.
- [6] Andrei Serjantov, Roger Dingledine, and Paul Syverson, "From a trickle to a flood: Active attacks on several mix types," in *Proceedings of Information Hiding Workshop (IH 2002)*, Fabien Petitcolas, Ed. October 2002, Springer-Verlag, LNCS 2578.
- [7] Tuneesh Lella and Riccardo Bettati, "Privacy on encrypted voice-over-ip," in *Proceedings of the 2007 IEEE International Conference on Systems, Man, and Cybernetics (SMC-2007)*, Montreal, Canada, October 2007.

- [8] Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," pp. 267–296, 1990.
- [9] C.V. Wright, L. Ballard, S.E. Coull, F. Monroe, and G.M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted voip conversations," *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pp. 35–49, May 2008.
- [10] R. Bettati, W. Zhao, and D. Teodor, "Real-time intrusion detection in ATM networks.," in *1st USENIX Workshop on Intrusion Detection and Network Monitoring*, San Jose, CA, April 1999.
- [11] Tadayoshi Kohno, Andre Broido, and K. C. Claffy, "Remote physical device fingerprinting," in *2005 IEEE Symposium on Security and Privacy (SP 2005)*, Washington, DC, USA, 2005.
- [12] Ryan Gerdes, Thomas Daniels, Mani Mina, , and Steve Russell, "Device identification via analog signal fingerprinting: A matched filter approach," in *Network and Distributed System Security Symposium Conference (NDSS 2006)*, 2006.
- [13] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoie, Jamie Van Randwyk, and Douglas Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *USENIX Security '06*, Vancouver, BC, CANADA, August 2006.
- [14] Steven J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *(CCS'06)*, Alexandria, Virginia, USA, October 2006.
- [15] R. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan 1991.
- [16] H. Zhang D. E. Wrege, E. W. Knightly and J. Liebeherr, "Deterministic delay bounds for vbr video in packetswitching networks: fundamental limits and practical tradeoffs," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 352–362, 1996.
- [17] X. Fu, B. Graham, D. Xuan, R. Bettati, and W. Zhao, "Analytical and empirical analysis of countermeasures to traffic analysis attacks," in *32nd International Conference on Parallel Processing (ICPP-2003)*, 2003.
- [18] X. Fu, B. Graham, D. Xuan, R. Bettati, and Wei Zhao, "Empirical and theoretical evaluation of active probing attacks and their countermeasures," in *Proceedings of the 6th Information Hiding Workshop*, Toronto, CANADA, May 2004.
- [19] Roger Dingledine, Nick Mathewson, and Paul Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [20] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM (USA)*, vol. 42, no. 2, pp. 39–41, 1999.
- [21] Ye Zhu and Riccardo Bettati, "Compromising privacy in wireless networks using sensors with limited information," Tech. Rep., Texas A&M University, July 2006, Department of Computer Science, 2006-7-1.
- [22] J. Cardoso, "Blind signal separation: statistical principles," *Proc. of the IEEE*, vol. 9, no. 10, pp. 2009–2025, 1998, Special issue on blind identification and estimation.
- [23] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min, "Z-mac: a hybrid mac for wireless sensor networks," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, New York, NY, USA, 2005, pp. 90–101, ACM Press.

About the Author: Riccardo Bettati



Riccardo Bettati is Professor in the Department of Computer Science at Texas A&M University, where he leads Center of Information Assurance and Security and the Real-Time Systems Research Group. He received his Diploma in Informatics from the Swiss Federal Institute of Technology (ETH), Zuerich, Switzerland, in 1988 and his Ph.D. from the University of Illinois at Urbana-Champaign in 1994. From 1993 to 1995, he held a postdoctoral position at the International Computer Science Institute in Berkeley and at the University of California at Berkeley. His research interests are in traffic analysis and privacy, real-time distributed systems, real-time communication, and network support for resilient distributed applications. He was the Program and General Chairs of The IEEE Real-Time and Embedded Technology and Applications Symposia in 2002 and 2003, respectively. He shares Best Paper awards with collaborators and students in the IEEE National Aerospace and Electronics Conference and in the Euromicro Conference on Real-Time Systems.

UNITED STATES ARMY COMBINED ARMS CENTER FORT LEAVENWORTH, KANSAS

The NEXUS is published by the USACEWP, a directorate of the Combined Arms Center (CAC) at Fort Leavenworth, Kansas. Permission is granted to print single copies of this article for personal, non-commercial use. Requests for reprint of articles in other publications, lesson plans, or similar multiple copy venues must be submitted via e-mail to: USACEWPStratcomm@conus.army.mil; or in writing to the following address: USACEWP, 950 Bluntville Ave, Fort Leavenworth, KS 66027. Also, it should be noted that The NEXUS does not fact check statements or reputed sources in documents published. The responsibility for the accuracy of information asserted, sources, and (where appropriate) security clearance for release is deemed the personal and professional responsibility of the author.

