On Flow Marking Attacks in Wireless Anonymous Communication Networks

Xinwen Fu, Ye Zhu*, Bryan Graham*, Riccardo Bettati and Wei Zhao
Department of Computer Sciencem, Texas A&M University, College Station, TX 77843 - 3112
E-mail:{xinwenfu, bettati, zhao}@cs.tamu.edu, *{zhuye, bgraham}@tamu.edu

Abstract

This paper studies the degradation of anonymity in a flow-based wireless mix network under flow marking attacks, in which an adversary embeds a recognizable pattern of marks into wireless traffic flows by electromagnetic interference. We find that traditional mix technologies are not effective in defeating flow marking attacks, and it may take an adversary only a few seconds to recognize the communication relationship between hosts by tracking such artificial marks. Flow marking attacks utilize frequency domain analytical techniques and convert time domain marks into invariant feature frequencies. To counter flow marking attacks, we propose a new countermeasure based on digital filtering technology, and show that this filter-based countermeasure can effectively defend a wireless mix network from flow marking attacks.

1 Introduction

This paper studies the degradation of anonymity in a wireless anonymous communication system from flow marking attacks, in which an adversary embeds a recognizable pattern of marks into wireless traffic flows by electromagnetic interference.

Concerns about privacy and security have gained more attention with the rapid growth and public acceptance of the Internet as a means of communication and information dissemination. *Anonymity* has become necessary and legitimate in many scenarios, such as anonymous web browsing, E-Voting, and E-Commerce. In each of these scenarios, encryption alone cannot achieve the anonymity required by participants [40, 41, 13].

Since Chaum pioneered the basic idea of the anonymous communication system, referred to as *mixes*, researchers have developed various anonymity systems for different applications. Although a significant amount of effort has been made in wired networks, not enough attention has been paid to anonymity in wireless environments.

In this paper, we consider a broad range of wireless networks, ranging from networks with all links being wireless to hybrid wired and wireless networks. The wireless links can be either 802.11 (or its extensions) or Bluetooth. A wireless network may use existing mix techniques to provide anonymity for flow-based applications such as anonymous web browsing. We study three mix batching approaches, which are feasible for a flow-based wireless mix network, and find that they are all susceptible to a new flow-level attack, which we call *flow marking attack*.

In a flow marking attack, an adversary uses electromagnetic interference to embed a periodic pattern of marks into traffic flows. By tracking these marks, the adversary can discover the communication relationship between users. To effectively and efficiently detect the pattern of marks, the adversary can use frequency analysis and convert the abstract pattern of marks in the time domain to easily detectable invariant frequency components, denoted as *feature frequencies*, in the frequency domain.

Our major contributions are summarized as follows:

1. We evaluate the performance of a wireless mix network under flow marking attacks in terms of *detection rate*, which is defined as the probability that an adversary correctly recognizes the communication relationship between two hosts. Empirically, we find that existing mixing techniques are all susceptible to flow marking attacks in a wireless mix network with either 802.11 or Bluetooth links. It may take an adversary only a few seconds to achieve a detection rate of 100%.

2. To counter flow marking attacks, we develop a new countermeasure based on digital filter techniques. With appropriate coefficients, a recursive (IIR) filter can effectively and efficiently filter out feature frequencies, thus preserving the effectiveness of a wireless anonymous communication network. Our experiments show that flow marking attacks become ineffective when filters are deployed.

The remainder of this paper is organized as follows: Section 2 reviews existing anonymity systems and related timing attacks in flow-based mix networks. We introduce the wireless mix network model and adversary threat model in Section 3. In Section 4, we give an overview of the flow marking attack technique and its issues. In Section 5, we discuss how to embed marks into wireless traffic and how to intercept wireless traffic. In Section 6, we discuss how to choose an effective pattern of marks and how to recognize the pattern. In Section 7, we empirically evaluate the effectiveness and efficiency of flow marking attacks. In Section 8, we develop a digital filter-based countermeasure to flow marking attacks and empirically prove its feasibility. We summarize the paper in Section 9.

2 Related Work

In his pioneer work [5], Chaum proposed the idea of anonymous computation, and communication. Since then, researchers have applied the idea to different applications such as message-based email and flow-based low-latency communications. Various attack approaches have also been reported in [35, 1, 37, 7, 51] and many others.

For anonymous email applications, Chaum proposed using relay servers, i.e., *mixes*, to reroute messages, which are encrypted by mixes' public keys. Mixes use source routing for message forwarding. An encrypted message is analogous to an onion constructed by a sender, who sends the onion to the first mix. Using its private key, the first mix peels off the first layer, which is encrypted using the public key of the first mix, and retrieves the next mix's address. The rest of the onion is encrypted with the second mix's public key. Consequently, the first mix sends the peeled onion to the second mix. This process proceeds until the core of the onion reaches the receiver. The core is covered by the receiver's address and contains real messages.

Mix techniques can be used for either message-based (high-latency) or flow-based (low-latency) anonymity applications. Message-based email anonymity applications include the first Internet anonymity *remailer* by Helsingius [15], *cypherpunk remailer* by Eric Hughes and Hal Finney [32], *Babel* by Gülcü and Tsudik [14] and *Mixmaster* by Cottrell [27]. Danezis, Dingledine and Mathewson [6] recently developed a so-called Type III Anonymous Remailer Protocol *Mixminion*, whose design considers a relatively complete set of attacks that researchers have discovered.

Low-latency anonymous communication can use either core mix networks or peer-to-peer networks. In a system using a core mix network, users connect to a pool of mixes and select a forwarding path through this core network to the receiver. *Tor* [7], *Onion routing* [42], *Freedom* [3] and many others belong to this category. In a system using a peer-to-peer network, every node is a mix, but it can also be a sender and receiver. A peer-to-peer mix network may scale well and provide better anonymity if a large number of participants use the anonymity service. *Crowds* [36], *Tarzan* [12], *ANODR* [20] and many others belong to this category.

Kong and Hong [20] developed an anonymity protocol for wireless ad-hoc networks. When Alice tries to communicate with Bob, she encrypts the request using secret keys shared with her neighbors and broadcasts the request to them. Her neighbors then broadcast the similarly encrypted request to their own neighbors. This process proceeds until the request reaches Bob, who responds to it through the reverse path. Thus an anonymity path is built from Alice to Bob and each mobile unit on this path acts as a proxy and relays packets from Alice to Bob by replacing the source address of packets with their own ones. The authors and other researchers also mention using broadcast MAC addresses to achieve more protection. But the whole protocol is still susceptible to the flow marking attack shown in this paper.

Above we have reviewed the existing anonymity systems. In this paper, we are interested in attacks degrading flow-based anonymity networks and the corresponding countermeasures for wired networks and wireless networks.

In [41, 16], a quantitative performance analysis is given for an anonymous web server that uses encryption and packet header mangling such as in a NAT proxy. The analysis takes advantage of the fact that a number of HTTP features, such as the number and size of objects, can be used as signatures to identify web pages with some accuracy. Unless the web anonymizer addresses this issue, these signatures are visible to the adversary. Serjantov and Sewell [38] analyzed the possibility of a lone flow along an input link of a mix in peer-to-peer anonymity systems. If the rate of this lone input flow is approximately equal to the rate of a flow out of the mix, this pair of input and outflow flows are correlated.

To find if Bob is communicating with Alice through a flow-based mix network, an adversary may measure the similarity between Bob's outbound traffic and Alice's inbound traffic. The authors of [51] propose using mutual information for the similarity measurement. In the one-mix case, an adversary collects a sample from an input flow and each output flow of the mix. Each sample is divided into multiple equally sized segments based on time. The number of packets in each segment is counted and forms a time series of packet counts. Then the adversary chooses the output link whose flow's packet count time series has the biggest mutual information with the input flow's packet count time series as the input flow's output link. To counter such attacks, we propose the use of adaptive padding, in which the output flows of a mix are synchronized and packets to different output links are sent in a predefined order. If there is no packet to an output link and a deadline is passed, dummy packets are generated for that output link.

Levine *et al.* [25] are also interested in the problem of discovering if Bob is communicating with Alice, but they use cross correlation to measure similarity between flows. If the cross correlation is beyond a threshold, the adversary decides Bob is communicating with Alice; otherwise not. The choice of threshold is the key problem of this attack and it may not be easily derived in practice. The authors propose using defensive dropping to thwart this attack. That is, Bob generates dummy packets to Alice, but intermediate mixes on the flow's path randomly drop those dummy packets.

Andrei Serjantov and Peter Sewell [38] and some other researchers mention very briefly that an adversary may introduce a "spike" into traffic to find the communication relationship between users, but without any in-depth study of how to introduce spikes, what kind of spike should be introduced, or how to recognize the spike. This paper generalizes this kind of attack in wired and wireless networks and builds a complete framework to answer the above questions.

3 Models

In this section, we first present the concept of mix network, and then describe the wireless mix network model used in this paper. Finally, we introduce the threat model.

3.1 Mix Network

A traditional mix is a relay server for anonymous email communication. It has a public key which senders use to encrypt messages. A mix operates as follows: (1) the sender attaches the receiver address to the message and encrypts the entire package by using the mix's public key; (2) the mix collects a batch of messages (from different senders), and decrypts them to obtain the receiver addresses; (3) finally the mix sends decrypted messages out in a rearranged order to corresponding receivers. Batching and reordering are necessary techniques for a mix to prevent the traffic analysis attack, which may correlate input messages and output messages by their timing.

A mix network consisting of multiple mix servers can provide enhanced anonymity. In a mix network, senders route their messages through a series of mixes. Therefore, even if an adversary compromises one mix and discovers the correlation between its input and output messages, other mixes along the path can still provide the necessary anonymity. Figure 1 illustrates the route selection for one message. A sender can choose different routes for each message or use one route for all her messages [6, 49, 50].

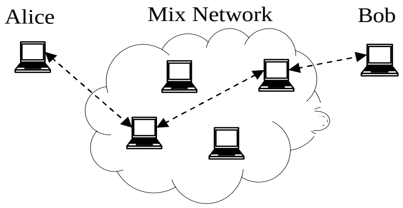


Figure 1. Mix Network

Message-based mix networks have been extended to flow-based networks for applications such as anonymous FTP, Web browsing, video and audio transmission, and many other low-latency applications. In the context of an IP network, the relay servers in Figure 1 form an overlay network and forward packets instead of messages.

In this paper, we will study how an adversary may explore the *dynamics of flows* between users and wireless links' susceptibility to interference and so seriously degrade a flow-based wireless mix network. Researchers have paid attention to attacks exploring *packet-level correlation* in anonymous communication systems. But this is not sufficient and sometimes misleading, since most of today's communications are flow based, with the large majority using TCP. On the Internet, TCP flows constitute 60% $\sim$ 90% of the Internet traffic and UDP flows constitute 10% $\sim$ 40% [10, 45], while all other protocols combined produce less than 5% traffic. On the Sprint IP backbone, new applications such as distributed file sharing and streaming media using TCP and UDP flows constitute 60% of the traffic on some links, while 30% is web traffic [11]. Traffic flows consist of rich features that can be explored to compromise anonymity systems.

Majors difference between flow-based anonymity systems and message-based anonymity systems are as follows:

1. Flow-based anonymity systems usually do not use dummy packets to pad the traffic in order to achieve the anonymity. This is because dummy packets consume additional bandwidth and reduce efficiency [7].

2. Flow-based anonymity systems usually adopt static routing, i.e., one path per flow, in order to avoid the difficulty and overhead caused by using multiple routes for TCP connections, and prevent intersection attacks [50]. This practice coincides with the design of several existing systems, including Crowds [36], Tor [7], and many others.

3. Batching and reordering [38] increase the (worst case) delay and are less preferred methods in flow-based anonymity systems. However, they may be necessary to counter packet-level timing correlation attacks.

In this paper, we will investigate the anonymity of flow-based anonymity systems with several different configurations. In [37], a relatively complete list of batching strategies for a message-based mix has been provided to counter message-level timing attacks. In our opinion, not all of them are appropriate for flow-based systems. For example, in a threshold mix, a mix can transmit the batch of packets only if the number of packets it collects has gone beyond a predefined threshold. This may cause serious problems for traffic of TCP flows, for instance, if the first (SYN) packet cannot be exchanged between a sender and receiver, the TCP flow cannot start, and hence the entire mix network may not be stable. We select three batching strategies which seem to be feasible for a flow-based mix network and summarize them in Table 1.

3.2 Wireless Mix Network

Now we introduce the wireless network model used in this paper. There are two popular radio frequency (RF) technologies: IEEE 802.11 [17] (and its extensions such as 802.11a/b/g) and Bluetooth [2].

The IEEE 802.11 standards are widely adopted for wireless LAN (WLAN). Two types of WLAN are supported: one is the infrastructure mode and the other ad-hoc mode. In the infrastructure mode, a station acts as the access point (AP) centrally controlling the WLAN, and other mobile units communicate with the AP. A WLAN in the infrastructure mode is denoted as the basic service set (BSS). In the ad-hoc mode, an AP does not exist. All mobile

Strategy Index	*Name*	*Adjustable Parameters*	*Algorithm*
S_0	Simple Proxy	$none$	No batching or reordering
S_1	Timed Mix	$< t >$	If timer with period t fires, send all the packets queued in the last interval.
S_2	Stop-and-go Mix (Continuous Mix)	$< \mu, \sigma^2 >$	Each packet is assigned a delay (deadline) satisfying a distribution with mean μ and variance σ^2. A packet is sent out when its deadline is reached.

Table 1. Batching Strategies

units (MUs) communicate within each other's transmission range. Ad-hoc routing protocols, such as DSDV [34], DSR [18], AODV [33], and many others, have been developed to extend the range and flexibility of ad-hoc networks. A WLAN in the ad hoc mode is also denoted as an Independent Basic Service Set (IBSS). An Extended Service Set (ESS) consists of multiple BSS/IBSS interconnected by access points and a distribution system, such as ethernet.

Bluetooth[1] is a low cost, low-power, short range radio technology, originally designed as a cable replacement to connect devices such as mobile phone handsets, headsets, and portable computers. In Bluetooth, a group of at least two and up to eight Bluetooth units form a *piconet*, sharing the same wireless channels (hopping sequence). In a piconet, any but only one unit can act as the *master* of the piconet, and the others are *slaves*. The master implements centralized control, and only communication between the master and slaves is possible. The communication between two slaves must be relayed by the master. Piconets can be interconnected and form a *scatternet*. Routing algorithms are proposed in [19] and many others for efficient communication between Bluetooth units (BU) in a scatternet.

Since most anonymity communication systems are built as overlay networks, wireless units (MUs or BUs) can use mixing strategies discussed above and form a wireless mix network. This paper assumes an ESS-like network with combined wireless (Bluetooth or 802.11) and wired links, in which any host can act as a mix. For example, in Figure 1, Alice (sender) and Bob (receiver) can be mobile units, and they may communicate with each other through a wireless or wired mix network.

3.3 Threat Model

In the following, we summarize the adversarial assumptions considered in this paper:

1. The content of wireless communication between legal participants is protected by underlying encryption algorithms and immune to any attack.

2. The adversary is an external one, and therefore is not a legal participant of the wireless network.

3. The adversary can passively eavesdrop on the communication session. We will show that eavesdropping wireless links can be easily realized in Section 5.

4. The adversary can actively interfere with wireless networks by injecting interference traffic. We assume that the adversary uses a reasonably good directional antenna, allowing it to interfere with a selected victim with minimum disturbance to other wireless units [21, 24, 26].

4 Flow Marking Attack

4.1 Overview and Problem Definition

Figure 2 illustrates the basic idea of a flow marking attack. Alice is communicating with Bob through a mix network. To find if Alice is communicating with Bob, an adversary, *interferer*, can embed a series of *marks* into Alice's traffic by interfering with her link. Another adversary, *sniffer*, eavesdrops Bob's inbound traffic. If the sniffer discovers a similar pattern of marks in Bob's traffic, she can be sure that Alice is communicating with Bob.

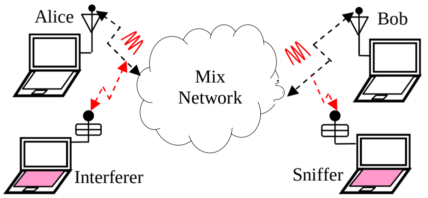


Figure 2. Flow Marking Attack Scenario

Thus, the general problem of the flow marking attack can be defined as follows: given a series of marks embedded into a flow, how can an adversary recognize them at other locations somewhere along this flow's path?

Flow marking is a general technique and can be used in both wired and wireless networks. In wired networks, an adversary may explore TCP's characteristics and use efficient denial of service approaches [22] to introduce marks. Refer to [44] for analysis of flow marking attacks in wired networks. In wireless networks, an interferer can use electromagnetic interference to embed marks into traffic. This is the focus of this paper.

4.2 Issues of Flow Marking Attack

From the viewpoints of both adversaries and defenders, there are four critical issues related to the problem of flow marking attacks:

1. How can an adversary introduce marks into traffic flows and intercept flows?

2. How can an adversary effectively recognize marks?

[1] We focus on the Bluetooth 1.1 wireless standard because of its popularity.

3. How effective and efficient can the flow marking attack be in reality?

4. How can we counter flow marking attack if it is effective?

We intend to address these issues in the following sections.

5 Mark Embedding and Traffic Interception

In this section, we discuss two key issues related to (1) embedding marks into wireless traffic and (2) intercepting wireless traffic. The discussion is not intended to be comprehensive due to the page limitation. Please refer to [44] for details.

5.1 Overview of Radio Frequency Communication

The physical layer of IEEE 802.11 and Bluetooth is where interference may happen. IEEE 802.11 (and its extensions) has two different physical layers: frequency hopping (FHSS) layer and direct-sequence (DSSS) layer[2]. Bluetooth uses FHSS. Both IEEE 802.11 and Bluetooth use license-free ISM (industrial, scientific, and medical) radio frequency (RF) band from 2.4GHz to 2.5GHz. This band is divided into many channels.

In this paper, we assume that an adversary uses a laptop computer equipped with an 802.11b (DSSS) PCMCIA card to apply the interference and embed marks. Below we will focus on how the interference and interception can happen. Related RF specifications are based on the regulation of America's Federal Communications Commission. Please refer to [17] and [2] for RF regulations in other regions.

5.2 Interfering With and Intercepting Wireless Communication

802.11 DSSS. It's easy to interfere with and intercept 802.11 DSSS communication. There are 11 channels available, Channels 1 to 11. Hosts in the same channel can interfere with and intercept one another. Furthermore, only Channels 1, 6 and 11 are free of interference with each other, but adjacent channels may interfere with each other.

802.11 FHSS. In FHSS, the transceiver must be synchronized. With both 802.11 and Bluetooth, the ISM band is divided into 79 × 1 MHz channels. The synchronized transmitter and receiver communicate on a series of channels, denoted as *hopping pattern* or *hopping sequence* and only stay on one channel for a predefined amount of time, denoted as *dwell time*.

An 802.11 DSSS device can interfere with an 802.11 FHSS device since 802.11 FHSS's hopping sequence visits the DSSS channel and its adjacent channels regularly, hence potentially causing interference with each other. Intercepting the 802.11 FHSS traffic is not difficult since 802.11 FHSS has only 78 possible hopping sequences divided into 3 sets, and the adversary can know the whole hopping sequence by observing a small fragment of communication using an appropriate spectrum analyzer [8]. Then the adversary can adjust her own 802.11 FHSS device to synchronize with the victim 802.11 device and intercept the traffic. Of course, a full ISM band analyzer can easily intercept 802.11 FHSS traffic.

Bluetooth FHSS. In general, an 802.11 DSSS device can cause more interference to Bluetooth traffic than to 802.11 FHSS traffic since a Bluetooth device visits a fixed DSSS channel more frequently. Bluetooth's hopping sequence has a dwell time of 625 μs, which corresponds to 1600 hops/s. The Bluetooth specification also requires that the hopping sequence distribute the hop frequencies equally over the 79 MHz during a short time interval. An 802.11 FHSS device's hopping rate is often within tens of hops per second.

It is still possible to intercept Bluetooth communication. Although Bluetooth's hopping sequence has a very long period length and does not show repetitive patterns over a short time interval, it has a few defects. First, a piconet uses cleartext frequency hopping sequence (FHS) packets to exchange hopping sequence information between the master and slaves. An adversary can intercept FHS packets, synchronize with the master, and then eavesdrop on the communication. Second, the adversary may have sophisticated Bluetooth listening devices to sniff the communication [23]. Again, a full ISM band analyzer can easily intercept Bluetooth traffic.

6 Mark Pattern Recognition by Feature Frequency

In this section, we address two issues of the flow marking attack: (1) how to choose an effective pattern of marks and (2) how to recognize marks.

6.1 Effective and Efficient Marks

An effective pattern of marks for flow marking attacks must demonstrate uniqueness. That is, the adversary can be certain of recognizing the same series of marks at one location as the one she introduces at another location. Because of the inherent nature of the Internet traffic, an arbitrary pattern of marks may not be effective and efficient for flow marking attacks.

In this paper, we demonstrate that a periodical pattern of marks can be very effective and efficient. That is, an adversary may use *on-off traffic* with a period of T_I, denoted as *interference period*, to interfere with the victim traffic. During an *on period*, the interfering device transmits at a rate as high as possible. This will reduce the available bandwidth for the victim traffic or disrupt packets of the victim traffic. During an *off period*, the interfering device becomes silent and the victim traffic gains the lost bandwidth quickly. In this way, the adversary forces the victim traffic to adapt

[2] Today, most of 802.11 products use DSSS because of its high throughput.

to the pattern of the interfering traffic and the victim traffic develops a similar pattern. The adversary can choose a relatively unique interference period (compared with the background noise traffic) to achieve a series of unique and strong marks within the network. We use an on period (approximately) equal to the off period, with each lasting for $T_I/2$.

Depending on where it is deployed in the path of the flow, the flow marking attack can have different effects on different types of flows. For TCP flows, the attack location can be very flexible. The adversary can apply the interference at any point along a TCP flow's path (i.e., at the sender, intermediate mix, intermediate hop or receiver). Since TCP uses a loop-control mechanism [31], a TCP flow will demonstrate the similar periodicity along its path from the sender to the receiver. For UDP traffic, an adversary may have to deploy the attack as close to the sender as possible. We will focus on a flow marking attack's effect on TCP flows because of their dominant status on the Internet.

6.2 Flow Marking Attack Framework

Now we summarize the framework of flow marking attacks based on pattern recognition [9] in Figure 3. Recall that in a flow marking attack, an adversary tries to discover if Alice is communicating with Bob by checking if the intentionally embedded pattern of marks exist in both Alice's outbound traffic and Bob's inbound traffic. The adversary has to decide what the pattern is and how to evaluate its existence.

Generally speaking, the goal of the pattern recognition process is to use classifiers to *classify* an unknown pattern as belonging to one of several existing pattern *classes* with the help of a feature (or a vector of features). A classifier is trained from training data. In a flow marking attack, there are only two classes of events:

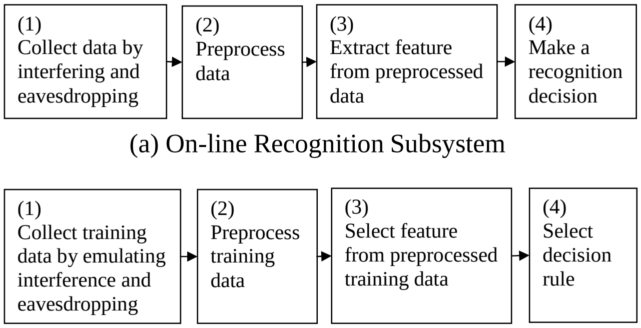
$$\begin{array}{ll} \omega_0: & \text{Alice does not communicate with Bob} \\ \omega_1: & \text{Alice communicates with Bob} \end{array} \quad (1)$$

Following this common practice, a pattern recognition system for flow marking attacks consists of two subsystems: (a) on-line mark recognition and (b) off-line training.

6.2.1 On-line Attack Subsystem

Figure 3 (a) is the framework of on-line attack subsystem. We will discuss the function of each component by using the example in Figure 2.

(1) Collecting packets by interfering and eavesdropping: The interferer interferes with Alice's wireless link and dumps Alice's interfered traffic or records the adversary's own interference traffic. In general, an adversary may not achieve a perfect periodic interference and needs either her own or Alice's traffic to derive the actual interference period. The sniffer intercepts Bob's inbound traffic.


(1)
Collect data by interfering and eavesdropping
(2)
Preprocess data
(3)
Extract feature from preprocessed data
(4)
Make a recognition decision
(1)
Collect training data by emulating interference and eavesdropping
(2)
Preprocess training data
(3)
Select feature from preprocessed training data
(4)
Select decision rule

(a) On-line Recognition Subsystem

(b) Off-line Training Subsystem

Figure 3. Flow Marking Attack Framework

(2) Preprocessing data: The collected data sample will be divided into segments, each of which contains packets within an interval, T_s, denoted as *sampling interval*. Thus, the number of packets in each segment forms a time series. The number of segments in the sample is denoted as *sample size*. This time series of packet counts is denoted as follows:

$$X(T_I, T_s) = \{x_1, \cdots, x_N\} \quad (2)$$

where T_I is the interference period, N is the sample size and x_i the number of packets in the i^{th} segment. We denote *sample length* as the lasting time of the traffic sample and it is equal to NT_s.

(3) Extracting a feature from preprocessed data: This is the key step for flow marking attacks. An appropriate feature extracted from $X(T_I, T_s)$ should represent the pattern of marks.

In flow marking attacks, because the adversary artificially introduces periodicity into the victim traffic, when Fourier transform is applied to $X(T_I, T_s)$, strong amplitudes will be observed around the frequency of $1/T_I$, denoted as *feature frequency*.

(4) Making a recognition decision: If the sniffer can observe the feature frequency in Bob's traffic, she can be sure that Alice is communicating with Bob. Here, we have an implicit assumption: without interference, the amplitude at the feature frequency is not significant. This *a priori* knowledge should be obtained from the off-line training.

6.2.2 Off-line Training Subsystem

Figure 3 (b) is the procedure for off-line training. The procedure is similar to the on-line recognition phase. The difference is that here, all the network traffic from Alice to Bob, denoted as *training traffic*, is generated by the adversary. First, the adversary collects training traffic without applying the interference and derives the *a priori* knowledge of statistics of the amplitude, A_{ω_0}, at the supposed feature frequency. Next, she collects data by emulating the flow marking attack and obtains statistics of the amplitude, A_{ω_1}, at the feature frequency. From statistics of A_{ω_1} and A_{ω_0}, the adversary generates rules used to make recognition decision in the on-line subsystem.

6.2.3 Bayes Classification Rule

In this paper, we assume the adversary uses Bayes classification rule during the on-line pattern recognition.

Bayes decision rule: *The amplitude a at the feature frequency implies ω_1 if*

$$p(\omega_1|a) \geq p(\omega_0|a) \tag{3}$$

That is,

$$p(a|\omega_1)Pr(\omega_1) \geq p(a|\omega_0)Pr(\omega_0) \tag{4}$$

where $Pr(\omega_i)$ $(i = 0, 1)$ is the *a priori* probability that Alice is communicating with Bob or not (set 50% in this paper), and $p(\omega_i|a)$ is the *a posteriori* probability that Alice is communicating with Bob when the collected sample has the amplitude a at the feature frequency.

From (4), the decision boundary d can be derived if we solve the following equation:

$$p(a|\omega_1)Pr(\omega_1) = p(a|\omega_0)Pr(\omega_0) \tag{5}$$

Thus, the rule is, Alice is communicating with Bob if $a > d$.

6.3 Detection Rate as Evaluation Criterion

Detection rate is defined as the probability that an adversary correctly recognizes the fact that Alice is communicating with Bob. To derive the detection rate for the Bayes decision system, the adversary has to estimate *a posteriori* probability distribution of the feature frequency power amplitude in power spectrum for classes ω_0 and ω_1. We assume that the adversary uses a Gaussian kernel function based method to estimate density functions [39].

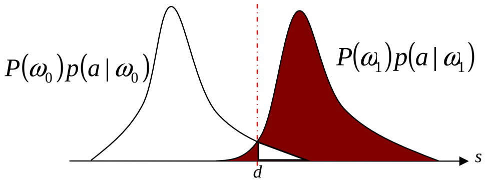


Figure 4. Bayes Decision Rule for Flow Marking Attack

As showed in Figure 4, once $p(a|\omega_1)$ and $p(a|\omega_0)$ are derived, detection rate can be calculated in (6).

$$v = P(\omega_0)\int_{-\infty}^{d} p(a|\omega_0)da + P(\omega_1)\int_{d}^{+\infty} p(a|\omega_1)da \tag{6}$$

6.4 Selection of Interference Interval and Sampling Interval

In flow marking attacks discussed above, there are two parameters: sampling interval T_s and interference period T_I. These parameters are critical to the effectiveness and efficiency of a flow marking attack.

6.4.1 Sampling Interval

We claim that the sampling interval should be smaller than half of the interference period. That is,

$$T_s < T_I/2 \tag{7}$$

This claim can be justified as follows. When we count packets in a sampling interval and derive the packet count time series in Step 3 in Figure 2 (a) and (b), this process is similar to a *zero-order hold* [30] sampling process. We know the feature frequency is $1/T_I$, which has to be preserved for the best effectiveness of flow marking attack. Nyquist's sampling theorem [30] suggests that to preserve this feature frequency, the sampling rate $1/T_s$ should be greater than twice the feature frequency. That is,

$$1/T_s > 2/T_I \tag{8}$$

Thus (7) is verified.

6.4.2 Interference Period

T_I's selection is not arbitrary, either. As discussed above, the interference traffic during the on period of the interference period has to decrease the victim traffic rate, and the off period has to be long enough so that the victim traffic can gain the lost bandwidth. Clearly, for 802.11 DSSS, 802.11 FHSS and Bluetooth, there are different requirements for T_I because of their different physical and protocol characteristics.

The interference period cannot be too long since in practice, a flow may only last for a short time. For example, the duration of a FTP session is determined by the corresponding file size.

Interference period is also related to the requirement of sample length for the effectiveness of flow marking attack. To get a feature frequency, we must sample for at least one complete cycle of interference. Otherwise, we could not resolve the feature frequency [30]. Thus, the sample length of NT_s should be greater than the interference period, i.e.,

$$T_I \leq NT_s \tag{9}$$

7 Evaluation of Flow Marking Attack (FMA)

In this section, we empirically show the failure of a wireless mix network under a flow marking attack in a laboratory environment and discuss its properties.

7.1 Experiment Environment

Figure 5 illustrates the experiment setup in the lab. It is a typical one-mix anonymous communication network with wireless links, i.e., an ESS-like wireless network. Alice uses FTP to download a file from Bob through a mix. To simplify our discussion, we assume that only Alice's link is wireless, and she communicates with other parts of the network through a machine performing access-point-like functions. We also install NISTNet [4] on this access-point-like computer to simulate delay and other network dynamics when necessary. One computer acts as a noise maker to generate noise traffic to Bob. In this way, we can evaluate noise's impact on the performance of flow marking attacks.

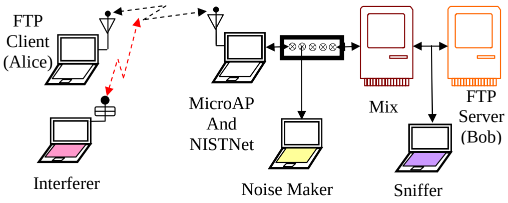


Figure 5. Experiment Setup

Mixing strategies are implemented on the TimeSys/Real Time Linux operating system for its timer accuracy [46]. We integrate the mix control module performing batching and reordering functions into Linux's firewall sub-system *Netfilter* [28], and firewall rules are used to specify what traffic should be protected.

We use WaveLAN silver PC card as 802.11 DSSS devices, Spectrum24 LA 3021 PC card as 802.11 FHSS devices, and Belkin Bluetooth PC card as Bluetooth devices. Wireless traffic and wired traffic is dumped by tcpdump [43]. Wireless channels can be changed by iwconfig [47].

In our experiments, a timed mix's timer has a period of 100ms. The stop-and-go mix assigns a exponentially distributed delay to packets with average delay of 25ms. This delay cannot be too long, otherwise it may cause a large number of packet reordering and hence seriously disrupt TCP's normal behavior.

7.2 Failure of Mix Networks under FMA

Figure 6 shows the power spectrum by 64-point FFT for a stop-and-go mix network with an 802.11 DSSS wireless link. We can see that the feature frequency, 2Hz ($1/T_I$), has a very strong amplitude compared to the case without flow marking attacks, in which every frequency component has roughly equal amplitudes.

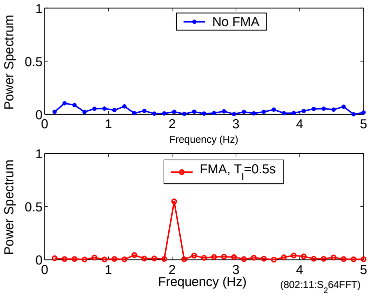


Figure 6. Power Spectrum of 802.11 DSSS Traffic for stop-and-go mix

Figure 7 shows the relationship between detection rate and sample length for all the three mixing techniques in Table 1 and three types of wireless links. In all the experiments, interference period $T_I = 0.5s$, and sampling interval $T_s = 0.1s$. 802.11 DSSS and 802.11 FHSS links have a bandwidth capacity of 2Mbps while the Bluetooth link has a bandwidth capacity of 1Mbps. We have the following observations from Figure 7:

1. A wireless anonymous communication system may completely fail under flow marking attacks. As sample length increases, a flow marking attack can achieve a detection rate of 100% in all cases in Figure 7.

2. An adversary only needs a few seconds of sampling to get a detection rate of 100%. This shows that flow marking attacks can be effective and efficient for on-line piracy tracing even if an anonymous file exchange service is used on the Internet since most of the file downloading times are greater than a few seconds [44].

7.3 Detection Rate v.s. Different Wireless Links

Figure 8 compares detection rate for the three different wireless links. Stop-and-go mixes are used in experiments.

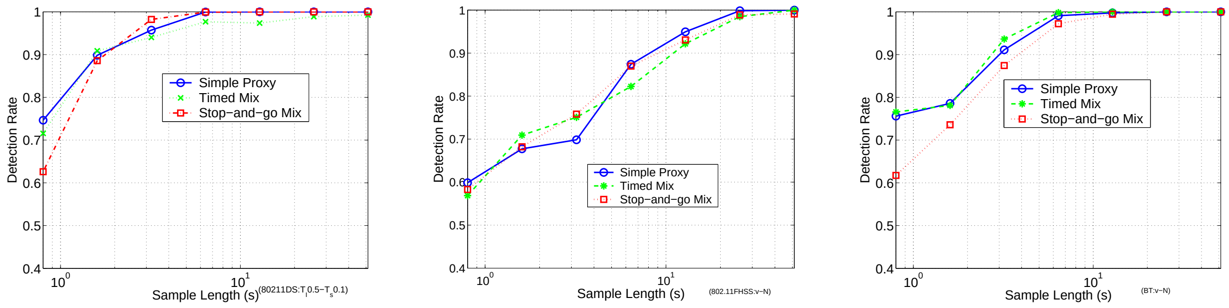
As we analyze in Section 5, since an adversary can use the same 802.11 DSSS channel to interfere with the victim 802.11 DSSS wireless link, she achieves the highest detection rate in this case. Because of a higher hopping rate, a Bluetooth FHSS link is more susceptible to the 802.11 DSSS interference than an 802.11 FHSS link, where the Spectrum24 PC card has a hopping rate of 10 hops/s. The adversary achieves higher detection rate in the case of interfering with a Bluetooth link.

Because of the space limitation, in the following, we concentrate on properties of flow marking attacks of 802.11 DSSS wireless links. Please refer to [44] for other cases.

7.4 Time to Achieve Detection Rate of 95%

Figure 9 shows the minimum amount of time an adversary takes to achieve a detection rate of 95% for each interference period. From Figure 9, we can see that at the interference period of 0.5s, it takes the adversary about 1.6 seconds to achieve a detection rate of 95%. This indicates that there is an *optimal* interference period by which the sample length is minimized. That is, flow marking attacks can be very effective and efficient.

We note that the curve is concave up. The reason is: if the interference period is too small, a TCP flow does not have enough time to reduce the rate during interference and to increase the rate during the silent time of the flow marking attack. Thus, the introduced pattern is very weak in the TCP flow. It may take more time to effectively detect the pattern of marks. On the other hand, if the interference period is very long, from (9), we know the flow marking attack needs at lease one interference period of sample to be effective. Thus, the longer the interference period, the larger the sample length. Clearly, the large sample length is caused by the unnecessarily large interference period.



(a) 802.11 DSSS (b) 802.11 FHSS (c) Bluetooth

Figure 7. Detection Rate by Flow Marking Attack

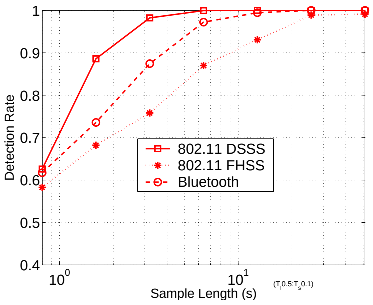


Figure 8. Detection Rate for Different Wireless Links

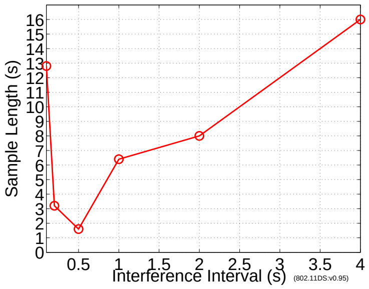


Figure 9. Sample Length Required to Achieve Detection Rate of 95%

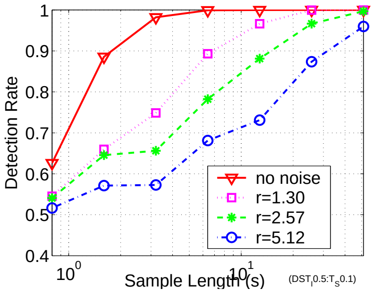


Figure 10. Detection Rate v.s. Noise Traffic

7.5 Impact of Noise Traffic

Figure 10 shows noise's impact on the effectiveness of a flow marking attack. We use r to represent the ratio of the number of noise traffic's packets to the number of TCP traffic's payload packets. The noise traffic is generated with an inter-arrival time satisfying a Pareto distribution with the shape parameter of 1.5 [48].

We have the following observations:

1. Noise traffic has a clear impact on the performance of a flow marking attack. We can see that as r increases, detection rate decreases. The reason is that noise traffic adds randomness into the aggregated traffic, and the power spectrum at the feature frequency has more randomly distributed energy with more noise traffic. This decreases detection rate.

2. Noise traffic's impact on the flow marking attack is limited. We can see that an adversary may still achieve a detection rate of 100% even if $r \approx 5$, which corresponds to a 60% utilization rate for Bob's 10Mbps link.

8 Countermeasures by Filtering

In this section, we develop possible countermeasures to flow marking attacks. Our idea comes from signal processing theory. That is, we use digital filters to filter out possible feature frequencies introduced by adversaries.

The filter-based countermeasure works as follows:

(1) We deploy filters at locations where traffic shaping and filtering is needed.
(2) The filter utilizes a periodic timer of period T_f to sample the traffic rate. It buffers packets arriving in its current timer interval, say the n^{th} interval, and counts the number of packets, $x(n)$, in this interval[3].
(3) Then we can calculate the required number, $y(n)$, of packets we should send out in order to filter out feature frequencies by using the following formula

$$y(n) = \sum_{k=0}^{M} a(k)x(n-k) - \sum_{l=1}^{M} b(l)y(n-l) \qquad (10)$$

where M is the filter order, $x(n-k)$ and $y(n-k)$ are the number of input packets and output packets of the filter respectively during the past k^{th} interval, and $a(k)$ and $b(k)$ are filter coefficients, which are discussed in Section 8.1. Please refer to [29] for general knowledge of the design of a recursive (IIR) filter specified in (10).
(4) For different $x(n)$ and $y(n)$:

(a) if $x(n) > y(n)$, the filter sends out $x(n)$ payload (user) packets when the timer fires and holds the remaining $y(n) - x(n)$ payload packets, which will be counted into

[3] In fact, $x(n)$ is the sum of incoming packets in the current interval and packets left over from the previous interval. Refer to Step 4.

the next round of incoming packets, i.e., $x(n+1) = x(n+1) + y(n) - x(n)$.

(b) if $x(n) < y(n)$, the filter generates $y(n) - x(n)$ dummy packets and send them out with $x(n)$ payload packets;

(c) if $x(n) = y(n)$, the filter just sends out all the $x(n)$ payload packets.

8.1 Selection of Filter Coefficients

Filter coefficients have to be carefully chosen for the best performance in countering flow marking attacks. To do so, we first determine the possible feature frequency band (F_l, F_u), which should be filtered out. In reality, the interference frequency of an adversary are bounded because: (1) it takes time for the victim traffic to respond to the interference and reduce its rate. Time is also needed for the victim traffic to gain the bandwidth when the interference stops. This gives feature frequency an upper bound, F_u; (2) a traffic flow only lasts for a limited interval, for example, the duration of a FTP session is determined by the file size. This gives feature frequency a lower bound, F_l.

Then we set a sufficiently large filter order M and use the yulewalk function from Matlab to derive the filter coefficients $a(k)$ $(k = 0, \cdots, M)$ and $b(l)$ $(l = 1, \cdots, M)$. The filter is of band-stop as we just filter out the band of possible feature frequencies. The benefit is that details of traffic are kept and the number of dummy packets can be reduced.

8.2 Evaluation of Filter-based Countermeasure

Figure 11 gives the detection rate when we put a filter of $T_f = 0.1s$ on MicroAP in Figure 5, where a stop-and-go mix is used. The interference period is 0.5s and the filter has an order of 20.

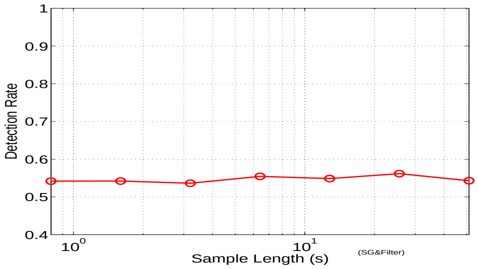


Figure 11. Detection rate with filter-based countermeasure

We can see that detection rate approaches 50%, which is the minimum value in a two-class pattern recognition. So traffic filtering can be used as an effective countermeasure for flow marking attacks in combination with mixes in a wireless mix network.

9 Final Remarks

This paper studies the degradation of an anonymous wireless communication system under flow marking attacks. Detection rate is defined as the probability that the adversary finds the communication relationship of "Alice" and "Bob" if they are communicating with each other. We show that it takes only a few seconds for an adversary to achieve a detection rate of 100%. This is, in a wireless environment, flow marking attacks can be very effective and efficient even if traditional mix technologies are used.

To counter flow marking attacks, we introduce digital filters to filter out the suspect band of feature frequencies. Our filter is an IIR recursive one. We empirically demonstrate the success of this digital-filter based countermeasure. With a filter deployed in a wireless mix network, the detection rate can be maintained near the minimum value of 50%.

We leave the theoretical analysis of detection rate as the future work. It is possible since there exist a variety of studies on interference between different wireless protocols.

References

[1] A. Back, U. Möller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Proceedings of Information Hiding Workshop*, 2001.

[2] Bluetooth-SIG. *Specification of the Bluetooth System, Version 1.1*. Bluetooth Special Interest Group, 2001.

[3] P. Boucher, A. Shostack, and I. Goldberg. Freedom systems 2.0 architecture, Dec. 2000.

[4] M. Carson and D. Santay. Nist net - a linux-based network emulation tool. *Computer Communication Review*, 33, July 2003.

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[6] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[7] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[8] R. Dixon. *Spread Spectrum Systems, 2nd Edition*. John Wiley & Sons, 1984.

[9] R. O. Duda and P. E. Hart. *Pattern Classification*. John Wiley & Sons, 2001.

[10] M. Fomenkov, K. Keys, D. Moore, and K. Claffy. Longitudinal study of internet traffic in 1998-2003. In *Proceedings of the winter international synposium on Information and communication technologies*, 2004.

[11] C. Fraleigh, S. Moon, C. Diot, B. Lyles, and F. Tobagi. Packet-level traffic measurements from a tier-1 ip backbone. Technical report, Sprint, 2001.

[12] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, 2002.

[13] X. Fu, B. Graham, D. Xuan, R. Bettati, and W. Zhao. Empirical and theoretical evaluation of active probing attacks and their countermeasures. In *Proceedings of 6th Information Hiding Workshop (IHW2004)*, 2004.

[14] C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Proceedings of the Network and Distributed Security Symposium - NDSS '96*, 1996.

[15] J. Helsingius. Press release: Johan helsingius closes his internet remailer. `http://www.penet.fi/press-english.html`, 1996.

[16] A. Hintz. Fingerprinting websites using traffic analysis. `http://guh.nu/projects/ta/safeweb/safeweb.html`, 2002.

[17] IEEE Computer Society. *Part 11: Wireless LAN Media Access Control (MAC) and Physical Control Specifications (802.11)*. IEEE, 1999.

[18] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 1996.

[19] R. Kapoor and M. Gerla. A zone routing protocol for bluetooth scatternets. *IEEE Wireless Communications and Networking Conference*, 2003.

[20] J. Kong and X. Hong. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003.

[21] K. Krizman, T. Biedka, and T. S. Rappaport. Position location: fundamentals, implementation strategies, and sources of error. *IEEE Communications Magazine*, 1997.

[22] A. Kuzmanovic and E. W. Knightly. Low-rate tcptargeted denial of service attacks. In *Proceedings of ACM SIGCOMM*, Oakland, California, 2003.

[23] A. Laurie. Serious flaws in bluetooth security lead to disclosure of personal data. `http://www.thebunker.net/release-bluestumbler.htm`, 2003.

[24] S.-J. Lee, W. Su, and M. Gerla. Wireless ad hoc routing with mobility prediction. *Mobile Network sand Aplications*, 2000.

[25] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright. Timing attacks in low-latency mix-based systems. In *Proceedings of Financial Cryptography*, 2004.

[26] G. Liu and G. J. Maguire. A class of mobile motion prediction algorithms for wireless mobile computing and communication. *Mobile Networks and Applications-Special issue: routing in mobile communications networks*, 1(2), October 1996.

[27] U. Möller and L. Cottrell. Mixmaster Protocol — Version 2. `http://www.eskimo.com/˜rowdenw/crypt/Mix/draft-moeller-mixmaster2-proto%col-00.txt`, January 2000.

[28] netfilter.org. Netfilter. `http://netfilter.samba.org/`, 2003.

[29] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. PrenticeHall, Inc., 1975.

[30] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab. *Signals and systems*. Prentice-Hall, Upper Saddle River, NJ 07458, USA, second edition, 1997.

[31] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM*, 1998.

[32] S. Parekh. Prospects for remailers - where is anonymity heading on the internet. `http://www.firstmonday.dk/issues/issue2/remailers/`, 1996.

[33] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. *Milcom*, 1997.

[34] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of SIGCOMM*, 1994.

[35] J. Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In *Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, 2001.

[36] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 1998.

[37] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: active attacks on several mix types. In *Proceedings of Information Hiding Workshop*, 2002.

[38] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. In *Proceedings of European Symposium on Research in Computer Security (ESORICS)*, 2003.

[39] B. Silverman. *Density Estimation for Statistics and Data Analysis, Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1986.

[40] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of 10th USENIX Security Symposium*, 2001.

[41] Q. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of IEEE Symposium on Security and Privacy*, 2002.

[42] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, California, 1997.

[43] tcpdump.org. *tcpdump*, 2004.

[44] The Center for Information Assurance and Security at Texas A&M University. Flow marking attacks. Technical report, Computer Science Department, Texas A&M University, 2004.

[45] K. Thompson, G. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network magazine*, 11(6), November/December 1997.

[46] TimeSys. Timesys linux docs. `http://www.timesys.com/`, 2003.

[47] J. Tourrilhes. Wireless tools for linux. `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`, 2004.

[48] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[49] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium - NDSS '02*. IEEE, February 2002.

[50] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[51] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao. On flow correlation attacks and countermeasures in mix networks. In *Proceedings of Workshop on Privacy Enhancing Technologies (PET2004)*, 2004.