SELECTIVE CROSS CORRELATION IN PASSIVE TIMING ANALYSIS ATTACKS AGAINST LOW-LATENCY MIXES

by

TITUS ABRAHAM

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2010

To my parents for their prayers and blessings and to my brother Arun for his consistent support

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr Matthew Wright, for supervising me on this thesis and helping me in every step of my work. My thanks and appreciation to Dr. Yonghe Liu and Dr. Hao Che for taking time to serve on my thesis committee. I would also like to thank my student colleagues, Kush Kothari and Pranav Krishnamoorthy for creating positive work environment in iSec lab and being available any time to discuss problems. Finally, I thank my parents for instilling in me confidence and a drive for pursuing my thesis.

April 16, 2010

ABSTRACT

SELECTIVE CROSS CORRELATION IN PASSIVE TIMING ANALYSIS ATTACKS AGAINST LOW-LATENCY MIXES

Titus Abraham, M.S.

The University of Texas at Arlington, 2010

Supervising Professor: Matthew Wright

A mix is a communication proxy that hides the relationship between incoming and outgoing messages. Routing traffic through a path of mixes is a powerful tool for providing privacy. When mixes are used for interactive communication, such as VoIP and web browsing, attackers can undermine user privacy by observing timing information along the path. Mixes can prevent these attacks by inserting dummy packets (cover traffic) to obfuscate timing information in each stream. Two recently proposed defenses, defensive dropping and adaptive padding, enhance cover traffic by ensuring that timing information seen at the sender is very different from that seen at the receiver.

In this work, we propose Selective Cross Correlation (SCC), an attack that an eavesdropper could employ to de-anonymize users despite the use of defensive dropping or adaptive padding. The main insight of our approach is that, with either defense, the timings at one end of the stream are a subset of the timings at the other end of the stream. By considering the network conditions and the defensive mechanism used, SCC can be used to effectively remove the cover traffic, thereby

enabling the attacker to correlate both ends of the stream. We conducted real network experiments and found that SCC greatly improves attacker effectiveness over prior techniques against both the defenses. With SCC, the attacker is nearly as successful as when neither defense is applied. This attack demonstrates the need for more robust defenses against statistical timing attacks.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Anonymity systems, such as Tor [1] and AN.ON (first described in [2]), can provide personal privacy and protection for journalists, activists, whistleblowers, law enforcement officers, and more (see `http://www.torproject.org/torusers.html.en` for a discussion of who uses Tor). These systems allow users to connect to the Internet anonymously via a series of proxies to hide their IP address from would-be eavesdroppers and the servers, such as Web sites, to which they connect. Connecting normally, without such a service, allows a user to be tracked, located, and possibly even identified based on her IP address.

Since these anonymity systems are generally based on the idea of mixes [3], and they are designed to facilitate real-time communication, such as Web browsing and SSH, we refer to them as *low-latency mixes*. We note that users send their traffic through multiple proxies, chained together in a *circuit* or *path*, which prevents any single proxy from being able to observe everything that the user does. A simple cryptographic technique called *layered encryption* allows the user's IP address to be hidden from all but the first proxy and the user's connections to be hidden from all but the last proxy [3].

In *high-latency mixes*, which are mixes designed for non-interactive communication like email, the mixes employ various defenses that make it difficult for an attacker to attempt to trace the path of a given message. These include delaying messages so that they are sent in batches, adding fake messages (*dummy messages*), and reordering the batch randomly. The attacker would (as far as we know) have to

compromise every mix along the path to trace a message, thus making it very difficult to defeat the system. In *low-latency mixes*, however, such defenses are impractical. Most applications require TCP connections; packets simply cannot be sent through a different circuit every time. Further, the latency requirements of these systems generally prevents much delaying of packets.

This lack of defense means that low-latency mixes are subject to *timing analysis*, a type of attack in which the attacker observes the timing of the user's packets entering and exiting the circuit. If the packet timings match closely, they are likely to be the same. This means that the attacker can link the user's identity (seen entering the circuit) with her traffic (seen exiting the circuit) without needing to control all of the mixes on the circuit. In this work, we focus on timing analysis attack and defense in low-latency mixes.

More specifically, we consider an attacker whose goal is to link the *initiator*, the user who sends her traffic along a given circuit, with the *responder*, the Web site or other receiver at the end of the circuit. Here the attacker could use a set of compromised mixes to observe a subset of the network traffic. Similarly, an eavesdropper may be able to observe a fraction of all the traffic. In either case, the attacker can capture packet timings and employ statistical correlation to link an initiator with a responder. To combat this attack, several defenses have been proposed to remove statistical correlations and make it more difficult for the attacker. In particular, both the initiator and the mixes can add or remove dummy packets to confuse an attacker attempting to correlate the streams.

One such defense is *adaptive padding* [4]. In adaptive padding, dummy packets (padding) are added to the stream by the mixes. When the user's traffic rate is low, the padding rate increases to prevent statistical correlation based on *gaps* between packets. Shmatikov and Wang demonstrated in simulation that this technique was

very effective against eavesdroppers. Another such defense is *defensive dropping* [5]. This defense is employed for constant rate traffic in which cover traffic is generated from the clients. The cover traffic is dropped at intermediate mixes according to specific drop rates. This defense has been tested in experiments in the PlanetLab testbed and has found to be effective for compromised mix model [6].

1.1 Contributions

In Chapter 2, we describe the continuing importance of investigating passive timing analysis against low-latency mixes. We then describe a model in which these attacks and defenses can be studied in Chapter 3. Our most important contribution is to describe and evaluate a new attacker technique based on *Statistical Cross Correlation*, or (SCC) in Chapter 4. The main insight of this technique is that, with either defense, the timings at one end of the stream are a subset of the timings at the other end of the stream. A timing window based on the network jitter observed by the attacker is used to remove the cover traffic. Having removed the cover traffic, the attacker can essentially employ a basic statistical correlation to determine whether the streams actually match.

To validate this technique, we conducted experiments in the DETER network testbed, using the SubRosa system for evaluating timing analysis attacks and defenses [6] which is detailed in Chapter 5. Finally, in Chapter 6 we present our results to show that a passive attacker can correlate the streams with high accuracy despite the use of adaptive padding or defensive dropping.

CHAPTER 2

BACKGROUND

In this chapter, we first describe low-latency anonymity systems, then describe timing analysis attacks on these systems, and finally describe defenses against timing attacks.

2.1 Low-Latency Anonymity Systems

The earliest mixes were suitable for high latency applications such as electronic mail. More recently, there has been work on several low latency mix networks, such as Web MIXes [7], Tor [1] and Crowds [8]. These systems are targeted at applications such as secure shell (SSH), VoIP, and web browsing, which require real-time responses. We now describe how each of these systems works.

Web MIXes is a system for anonymous real-time Internet access [7]. The system has Java Anonymous Proxy (JAP) on the client side, the MIXes and the cache proxy on the server side. The user, instead of directly connecting to a web server, uses JAP to communicate via a logical chain of MIXes that is called "MIX-cascade" [7] and uses layered encryption thereby providing anonymity.

Tor is the second generation onion router and it aims to resist eavesdroppers and limited insider attacks by distributing each transaction over several nodes in the network [1]. In Tor for a client to establish communication with a destination it chooses a set of mixes o form a path till destination. This path is called as a circuit. Client streams are divided into fixed size cells and encrypted layer by layer so that each Tor server can only know its predecessor and its successor in the circuit. This

ensures that even if one of the Tor servers is compromised, the users' anonymity is preserved.

Crowds operates by grouping users into a large and geographically diverse group that collectively issues requests on behalf of its members [8]. Clients are members of a peer to peer network. Client requests are routed to a random member of the crowd called *jondo*. A *jondo* may either submit the request to the web server (the destination) or forward it another *jondo*. This ensures that neither the web server nor other crowd members can identify the source of the request.

2.2 Timing Analysis Attacks

A *passive adversary* attempts to de-anonymize users by collecting packet information over a period of time. A simple method to link users is to statistically correlate their incoming and outgoing streams based on the inter-packet delay or IPD (the time difference between the arrival of two consecutive packets). However this task is made difficult due to network jitters and packet drops.

Packet counting [9, 10], as the name suggests, is a technique that involves counting the number of packets to determine similarity of two streams to link the sender with the reciever. This idea has been improved by Levine et al. [5] by applying cross correlation (CC) over packet counts to improve error rates. CC is a measure of similarity of two wave forms as a function of a time-lag applied to one of them [11]. A stream is split into non-overlapping windows of equal sizes and packet counting is done to apply cross correlation. This method works with reasonable error rate in the presence of multiple constant rate input streams under the condition that the network exhibits jitter and drops some packets before they arrive at the first mix. This is true for most Internet connections, as has been shown in real network experiments [6].

Statistical analysis by using sample variance or sample entropy for IPD can be used even when traffic has been padded with constant interval times of packets [12].

An *active adversary* can observe as well as actively perturb the incoming traffic by means of packet delay or introduction or dropping packets. This technique, called watermarking [13], is used by an adversary to uniquely identify streams. Watermark-based correlation is designed to be robust against timing perturbation as it uses multiple randomly selected IPDs to encode bit information [13]. Houmansadr et al. use selective correlation to remove watermarking in an active attacker model [14]. The outgoing stream has a one-to-one relation with the watermark and is used to filter out the watermarked packets. However watermarking can be detected [15] and thereby passive timing analysis attacks remain relevant and important to measure and improve anonymity systems [14].

2.3 Timing Analysis Defenses

Although buffering and reordering packets as in high-latency mixes, would certainly remove most of the statistical properties of a stream, latency requirements make these defenses infeasible. Web mixes and ISDN mixes use constant rate traffic along the entire path [5]. Ideally, when all participating nodes send identical constant-rate traffic, streams are indistinguishable from each other. However, network jitter, packet drops, and mix-induced delays make these mechanisms vulnerable to cross-correlation attacks [5]. Some of the recent defenses are based on addition/removal of cover traffic which changes the statistical properties of the outgoing stream from the incoming stream thereby making the stream correlation difficult. We now describe three defenses: Defensive dropping, Adaptive padding and γ buffering.

Defensive dropping is a form of *partial-path cover traffic* where cover traffic is dropped at random intermediate mixes [5, 10]. In this scheme the client constructs

dummy packets that are marked to be dropped by one of the intermediate mixes. If there is sufficient randomness in the choice of the drop there would be very less correlation between the incoming stream and the outgoing stream. This defense has been recently analyzed [6] for timing analysis attack.

Web traffic is bursty, and as a result we tend to find gaps in the streams. Similar to watermarking, statistical properties can be applied on the pattern of these gaps to uniquely identify streams. Adaptive padding [4] is a defense in which intermediate mixes insert dummy packets into the streams to reduce these statistically unlikely gaps in the stream without adding any latency. Each mix calculates a statistical distribution of inter-packet arrivals based on previous observation of the clients' stream. When a packet arrives before the expected arrival time, calculated based on a statistical distribution, the packet is forwarded and a new inter-packet arrival is calculated. If a packet does not arrive upon expected arrival time, the gap formed is repaired by sending dummy packets. Subsequently calculated inter-packet arrivals will be extended to avoid clustering the estimates to small values. This method ensures that much of the inherent entropy shown by the clients would be removed.

Shmatikov and Wang show that cross correlation fails when packets are inserted by the mixes into the client streams. Thus, the packet count at the window of the outgoing stream is considerably different from that of the incoming stream.

γ buffering [6] is a technique for buffering traffic that can be used to undermine traffic analysis attacks in low-latency anonymous communications systems. For γ, a parameter set by the system designer and p, the number of incoming connections, systems buffers at least γ^*p packets before sending the packets as a batch. This technique is designed to maintain low latencies at the cost of some cover traffic and can also be adapted for different levels of allowable latencies.

2.4 SubRosa

SubRosa is a system for studying timing analysis methods and defenses [6]. Unlike prior simulation studies [5, 4], SubRosa allows us to experiment with real network traffic. This platform emulates a Tor-like network with three components: Client, Mix, and Sink. The Client acts as the user and is responsible for generating data on the network. The Sink acts as the recipient. This platform is designed to collect timing information as observable by an attacker. The Client initially chooses a random sequence of Mixes and it initiates a circuit building process. After this process, the Clients then start generating traffic according to the configuration of the experiment. The Mixes apply the defenses according to the system configuration for the given experiment.

CHAPTER 3

MODEL

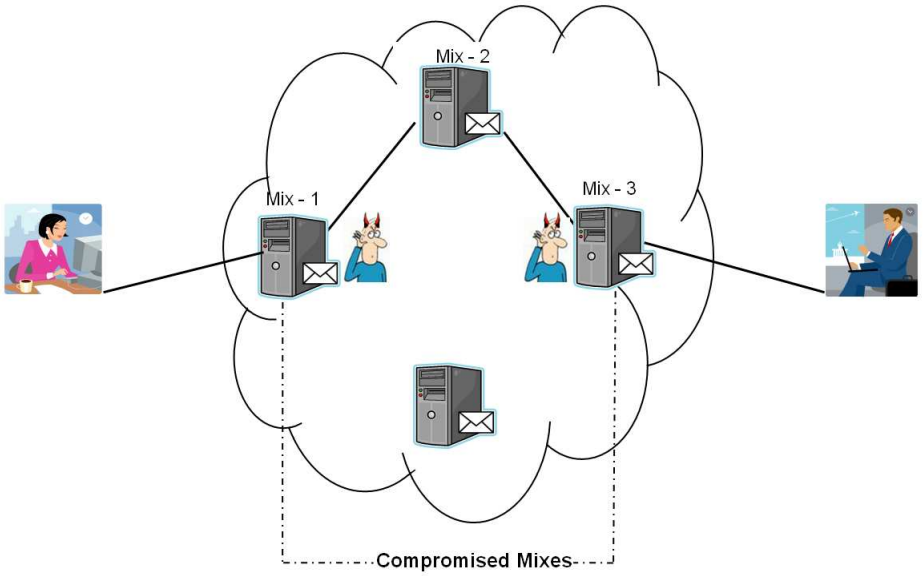
In this chapter, we present the system and the attacker model to analyze the attacks and defenses.

3.1 System model

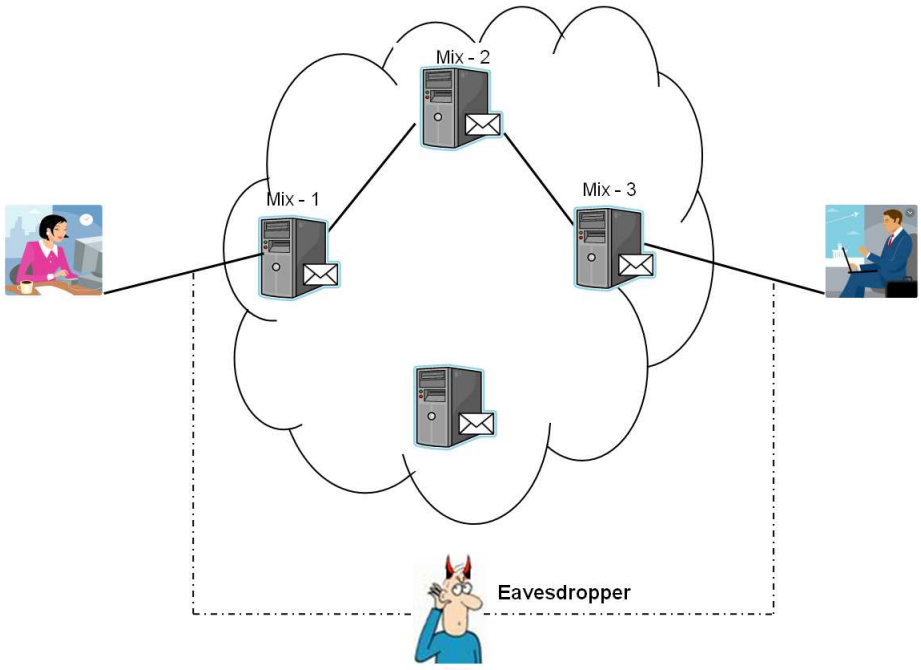
To analyze the threats posed by timing analysis attacks on low-latency mixes, we consider Tor-like network topology in which users send traffic through randomly selected mixes. Users may use cover traffic (constant rate cover) while the mixes may add cover traffic depending on the defense used. For timing analysis, we consider traffic only from the user to the responder. Similar attacks and strategies can be applied in the return direction.

3.2 Attacker model

For passive timing analysis attacks, we consider two types of attacker models – *compromised mixes* and *eavesdropper*. In the compromised mixes model as shown in Figure 3.1(a), the initial and final mixes in the circuit are compromised and can collaborate to perform a timing analysis attack. This allows the attacker to link the identities of the sender and the receiver [16]. The outgoing stream of the first mix is correlated with the incoming stream of the last mix. The *eavesdropper* model of attack assumes that the attacker can monitor the incoming stream for the first mix and outgoing stream of the last mix as shown in Figure 3.1(b). The two streams are then correlated to compromise the anonymity of the system.



(a)



(b)

Figure 3.1. Attacker Model (a) Compromised mixes (b) Eavesdropper.

CHAPTER 4

SELECTIVE CROSS CORRELATION

In this chapter, we discuss the main insight used in selective cross correlation (SCC), followed by a description of how SCC works and how we measure its effectiveness.

As described in Chapter 2.3, injection of packets by adaptive padding or dropping of cover traffic by defensive dropping makes cross correlation ineffective. However, in both the defenses, actual packet timings are preserved. In adaptive padding, the resulting outgoing stream effectively consists of cover traffic superimposed on the original stream, whereas in defensive dropping, the output stream (actual traffic) is a subset of the input stream (actual traffic with cover traffic). SCC exploits this property to identify and remove cover traffic. We also can generalize SCC for any defenses that involve either insertion of cover traffic or removal of cover traffic.

Selective cross correlation was first applied in unmasking of multifocal ERG [17] in vision research. We apply SCC to network streams by extracting the sequence of timing values from both the incoming and outgoing streams. The subset stream would be taken as the input stream while the superset stream would be taken as the output stream. We divide the two streams into non-overlapping windows of time, the *filter window* W of size s and count the number of packets received during each window interval. We compare the count of packets in the incoming stream window to the corresponding outgoing stream window. Absence of packets in the incoming stream window is an indicator of the presence of cover traffic in the outgoing stream

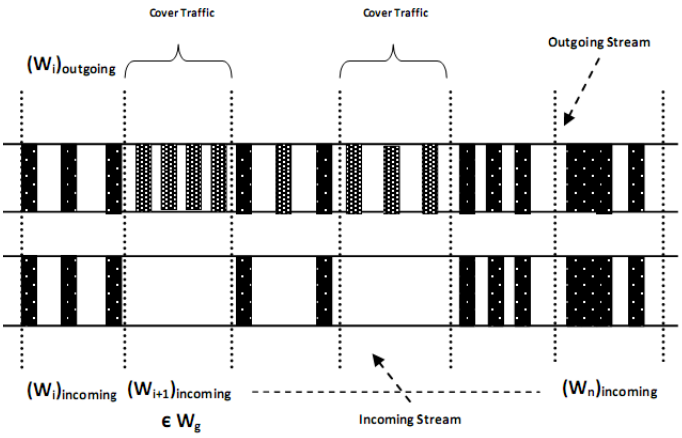


Figure 4.1. Selective Cross Correlation.

window. This process is repeated for the entire length of the streams. All suspect cover traffic is filtered.

Cross correlation is then applied on the filtered stream and the input stream to calculate the correlation value κ. If κ is below a threshold value, the streams are considered unrelated. Since adaptive padding is used for web traffic, we tend to find gaps in the stream that may be padded by the mixes. We use a sliding filter window in SCC of size $s \in \{0.35, 1.0\}$ seconds to remove cover traffic. The minimum filter window should be twice the average jitter present in the network so as to avoid removing actual packets. In defensive dropping, however since the packets are being generated at a constant rate we choose the filter window according to the rate.

As shown in Figure 4.1, $(W_i)_{\text{incoming}}$ and $(W_i)_{\text{outgoing}}$ represent the i^{th} filter window in the incoming and outgoing streams, respectively. A given filter window W_i is a composition of actual user traffic, cover traffic, and gaps (W_g) in the traffic. The resulting output stream after the removal of cover traffic (i.e. filtered stream) is denoted by $fStream$.

Algorithm 1 Selective Cross Correlation

$fStream = \emptyset$

$(W_{\text{incoming}}) \leftarrow$ Windows of the incoming stream

$(W_{\text{outgoing}}) \leftarrow$ Windows of the outgoing stream

for $(W_{in,i}, W_{out,i}) \in W_{\text{incoming}} \times W_{\text{outgoing}}$ **do**

  if Num-Packets $(W_{in,i}) > 0$ **then**

    $fStream = fStream \cup p; p \in W_{out,i}$

  end if

end for

As described in Algorithm 1, the cover traffic is removed in outgoing stream by eliminating $(W_i)_{\text{outgoing}}$ where $(W_i)_{\text{incoming}} \in W_g$.

4.1 Effectiveness of SCC

The effectiveness of SCC can be measured by the amount of correct filtrations done on the outgoing stream. Due to network jitter, a packet of $(W_i)_{\text{incoming}}$ may fall into $(W_{i+1})_{\text{outgoing}}$, which would be removed by SCC if $(W_{i+1})_{\text{incoming}} \in W_g$. Similarly a packet of cover traffic of $(W_i)_{\text{outgoing}}$ can also fall into $(W_{i+1})_{\text{outgoing}}$ which would not be removed by SCC if $(W_{i+1})_{\text{incoming}} \notin W_g$. There may be windows W_i for which *mixed traffic* is present and not filtered off by SCC. We show in Chapter 6.4 that these constitute a small fraction of the traffic and that SCC can filter most of the cover traffic. Thus, correlation can be still done very easily for web traffic.

CHAPTER 5

EXPERIMENT DESIGN

In this chapter, we first describe the test bed that we used for our experiments. Then we describe how we generated traffic for our experiments, and parameters used for the defenses tested.

5.1 DETER

DETER [18] is a network security test bed that provides a controlled environment in which users can setup network topologies and run a variety of computer security experiments. The network topology used for the experiments is shown in Figure 5.1. We chose a flower topology as we could represent the petals as LANs and the core as the Internet. The clients are nodes present in the LANs and mixes are nodes present in the Internet. Since the LANs have cross traffic different from each other both core and petal nodes have independent cross traffic.

5.2 Traffic Generation

In the experiments conducted, Clients generate constant rate traffic for baseline and web traffic for adaptive padding. For constant rate traffic, packets were generated every 100 milliseconds. For the user generated web traffic, we used HTTP traces from the National Laboratory for Applied Network Research (NLANR) [**?**]. We also used traces from University of North Carolina at Chapel Hill. To segregate HTTP traces, we isolate HTTP streams using the destination port number. In the Internet, there is cross traffic with different characteristics and a variety of protocols. To generate

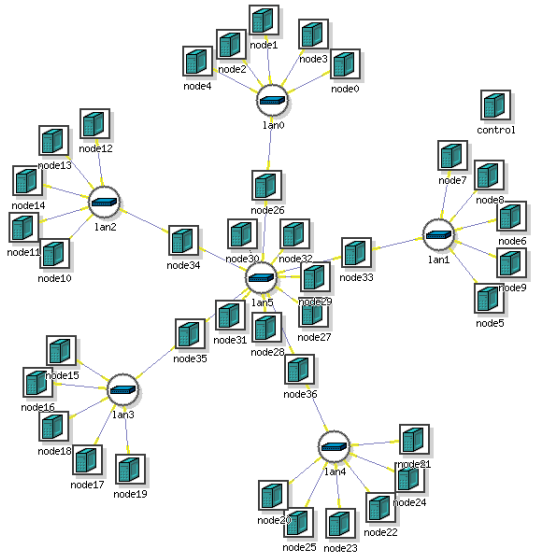


Figure 5.1. Network topology in DETER.

this cross traffic we used SEER, a traffic generator tool available in DETER. The link capacity in DETER is 100 Mbps. The Mixes have cross traffic varying from 50-85 Mbps on their links. In the flower topology, every petal has independent cross traffic varying from 65 to 95Mbps. Figure 5.2(a) shows different cross traffic for every petal in the topology and Figure 5.2(b) shows the individual cross traffic of the mixes. Each graph of the figure shows the incoming and outgoing traffic present on the link of the nodes.

5.3 Defenses Tested

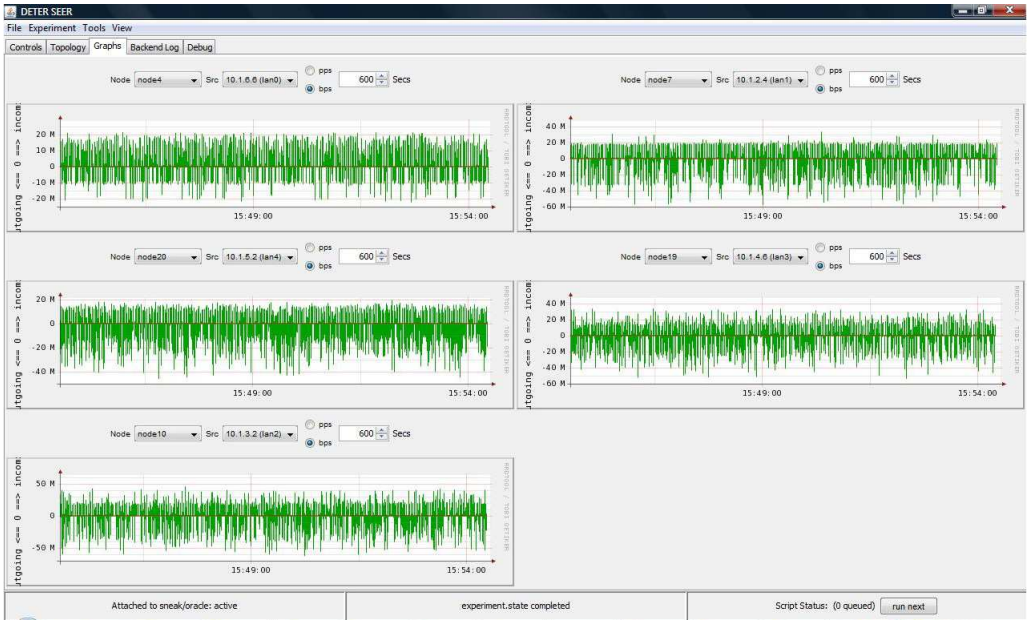
We ran experiments with 26 Clients and 5 Mixes, where each of the Clients randomly chose three of the Mixes as a part of the circuit. Timing analysis is done based on 10-minute runs and one minute of the experiments, since Tor uses the same circuit for all connections happening in a period of ten minutes [19].

Clients generate constant-rate traffic in which packets get generated at every 100 milliseconds and sent to the destination through the Mixes. No defense against timing analysis is employed in the Mixes, just like in real Tor nodes [1].

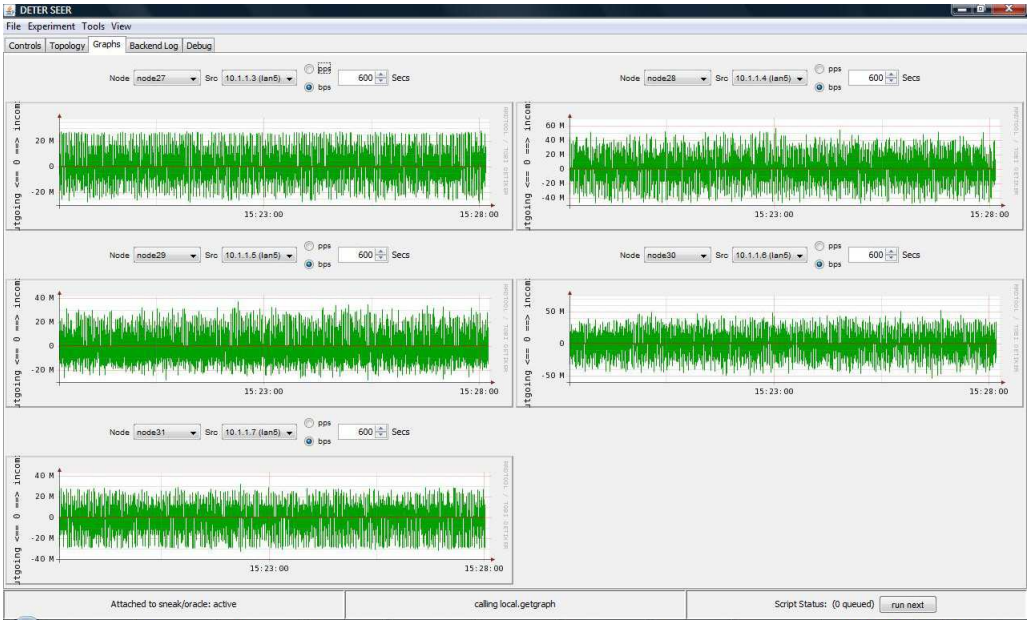
Defensive dropping is initiated by the Clients. Clients at random would mark the nodes to be dropped at intermediate Mixes. The drop command is set at the header of the packet. Each of the intermediate nodes inspects its header and drops the packet if the drop command is set. Drop Rates of 20% and 50% have been used in the experiments.

We implemented adaptive padding on Sub Rosa. Though the first Mix does repairs all the gaps, intermediate Mixes also provide adaptive padding so that, in theory even a compromised Mix won't be able to de-anonymize the stream. In adaptive padding , after a packet has been received , a new expected inter-packet interval is selected only from higher bins, ie. those associated with long intervals [4]. This collection of bins are known as High-Bins Set (HBS). After the previously choosen expected inter-packet interval has expired without receiving packets and a dummy packet has been sent, the next interval is selected from lower bins, ie. those associated with shorter intervals known as Low-Bins set (LBS). These HBS and LBS can be configured based on observed characteristics. In our experiments we have LBS as bins ranging from 1-11 and HBS above bin 11.

In γ buffering, γ is multiplied with the number of active circuits on the node to obtain the number of packets to buffer, before queuing it on the send queue. γ is a configurable parameter of the Mixis and is initialized at the start of the application. We have choosen γ to be 40 based on previously conducted experiments on Planet Lab [6].



(a)



(b)

Figure 5.2. Traffic Configuration in DETER (a) Client (b) Mixes.

CHAPTER 6

RESULTS AND DISCUSSION

We now show the relative effectiveness of CC and SCC against the defenses in both attacker models. For evaluating the detection results, we use Receiver Operator Characteristic (ROC) curves. In an ROC curve, the x-axis is the false positive rate and the y-axis is the detection rate; the curve varies according to the correlation threshold. Different ROC curves are plotted on same graphs for relative comparisons.

Adaptive padding, is meant to only provide effective defense for short-lived connections, e.g. one minute long [4]. In general, as the amount of data grows, the greater the statistical pattern that can be observed by the attacker for his attacks; Shmatikov and Wang argue that few practical defenses can be expected to hide all patterns for long periods of time [4]. Thus, we also tested CC and SCC against adaptive padding and other defenses with short-lived connections where the observation time was set to one minute.

6.1 Compromised Mix Model with CC

We first present results from the compromised model with cross correlation for both 1 minute traffic and 10 minutes traffic as shown in Figure 6.1.

Baseline is based on constant cover traffic and modified only by network jitter and packet drop. Thus, the statistical property of the outgoing and the incoming streams remain almost same, resulting in a 95% detection rate in the 10 minute traffic. We however see a detection rate of 85% in 1 minute traffic which can be

attributed to the lesser statistical properties that can be sampled from the shorter stream.

Adaptive padding is also ineffective in this model. As adaptive padding is based on filling up the gaps in bursty web traffic, most of these gaps are filled at very first mix (compromised mix) as shown in Figure 6.2. Thus, the statistical properties of the outgoing stream of the first mix and the incoming stream of the last mix would be identical. This results in better detection 6.1(a) as compared to the baseline, since adaptive padding is for web traffic, which is inherently bursty. In the 10 minute traffic we see a 100% detection rate due to more statistical properties we can sample from.

In the case of 20% and 50% defensive dropping, intermediate mixes are responsible for dropping packets. As a result, the statistical properties of the incoming and outgoing streams for these mixes are distinct. In partial buffering we see that with time correlation between streams decrease which can be attributed to the buffering mechanism followed by this defense. Both defensive dropping and partial buffering shows a .49 error rate for the 10 minute traffic.

6.2 Eavesdropper Model with CC

We now present result from experiments in the eavesdropper model with cross correlation for both 1 minute traffic and 10 minutes traffic as shown in Figure 6.3. Here the attacker can only collect timing information from the incoming stream of the first mix and the outgoing stream of the last mix.

In this model, adaptive padding is much more effective against cross correlation, as the eavesdropper does not know which packets are padding. Confirming the results of Shmatikov and Wang [4], the statistical properties of the outgoing stream are not easily linked to the incoming stream, resulting in .48 error rate. A noteworthy point is that both defensive dropping and partial buffering is performed only by intermediate

mixes, so effect of both the defenses in both the models remains the same. As a result in comparison to the compromised mix model, baseline, defensive dropping and partial buffering results are nearly identical.

6.3 Compromised Mix Model with SCC

We now present results from experiments for SCC in compromised model for both 1 minute traffic and 10 minutes traffic as shown in Figure 6.4.

As seen from the previous results, we see that the detection rate increases with more traffic with which more statistical properties can be sampled. We see that the detection rate for defensive dropping has increased from 49% to 65% and 45% in both the 10 minute and 1 minute traffic. Even the detection rate for adaptive padding has increased from 85% to 90% in the 1 minute traffic. This leap in detection rate can be attributed to the filtering of padding by SCC.

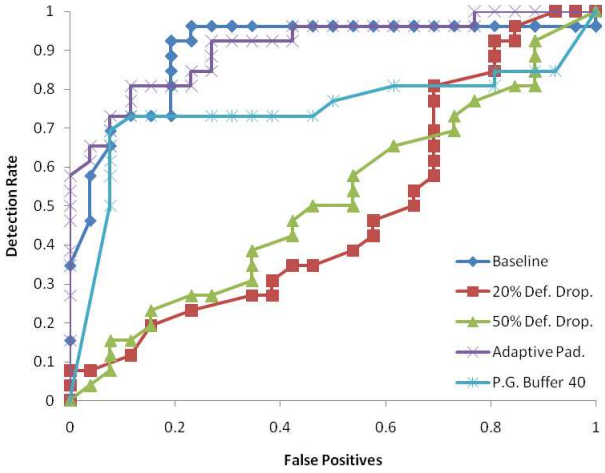
We also tested partial buffering with SCC. This defense being based on buffering makes SCC ineffective against it as there is no cover traffic to be removed and due to the buffering it removes many actual packets. Thus as expected we get a detection rate of 49% for partial buffering. Baseline remains identical as with CC results.

6.4 Eavesdropper Model with SCC

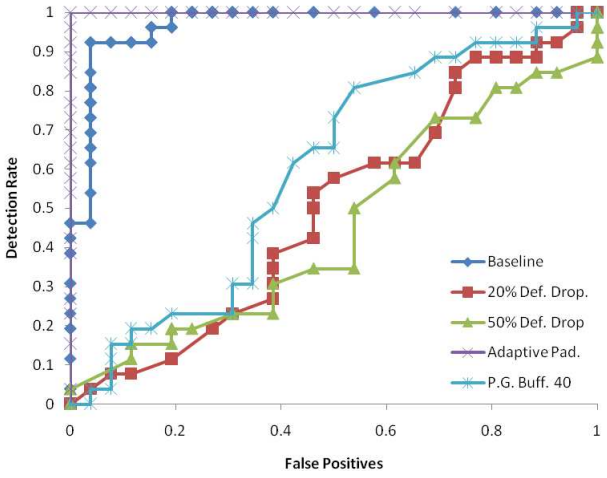
We next tested SCC in the eavesdropper modelfor both 1 minute traffic and 10 minutes traffic as shown in Figure 6.5.

SCC achieves a very high detection rate against adaptive padding with very few false positives. The success rate is nearly as good as in the compromised mix scenario. We see that the detection rate is 85% and 100% for 1 minute and 10 minute traffic respectively. The detection rate of SCC increases the more the data (in terms

of time) that is made available to it. This can be attributed to the the nature of SCC - the longer a stream, the greater the statistical properties that can be identified and extracted resulting in improved detection. The key to SCC's effectiveness is its ability to remove padding packets from the outgoing stream. In Figure 6.6, we see that SCC was able to remove 96% of the padding, while incorrectly removing only 7.7% of non-padding packets. Baseline, defensive dropping and partial buffering results remain identical as in the compromised model for SCC.



(a)



(b)

Figure 6.1. **Compromised Model:** ROC for CC with: (a) 1 minute observation (b) 10 minutes observation.

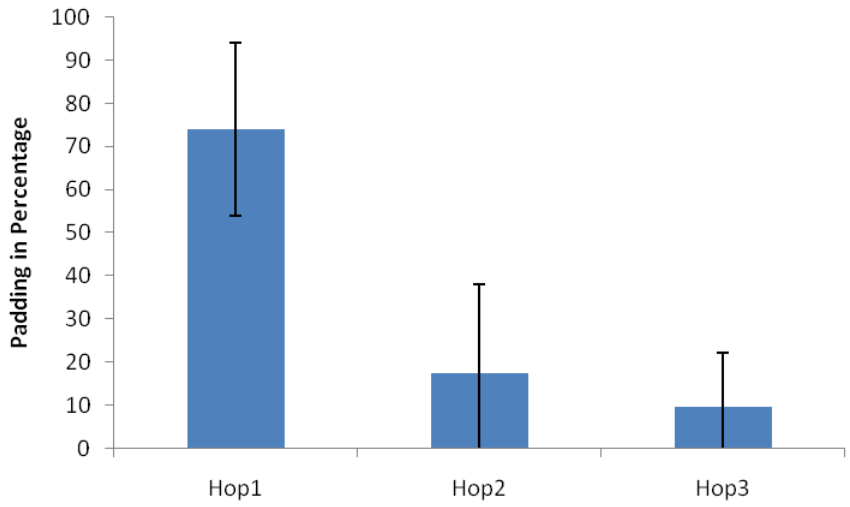
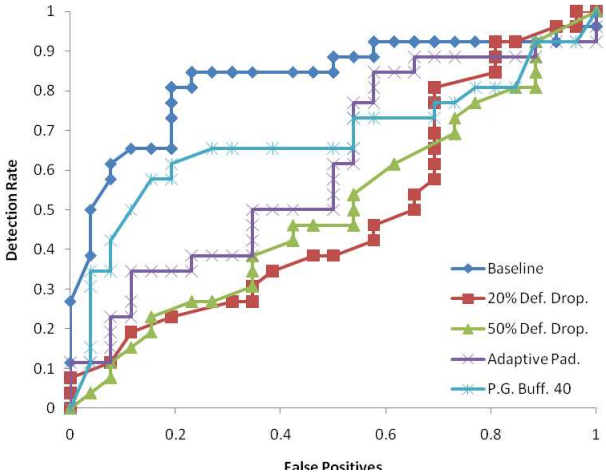
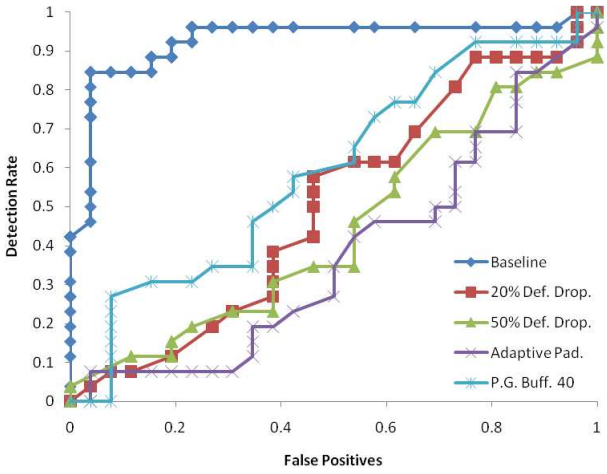


Figure 6.2. Average padding done by mixes in adaptive padding.

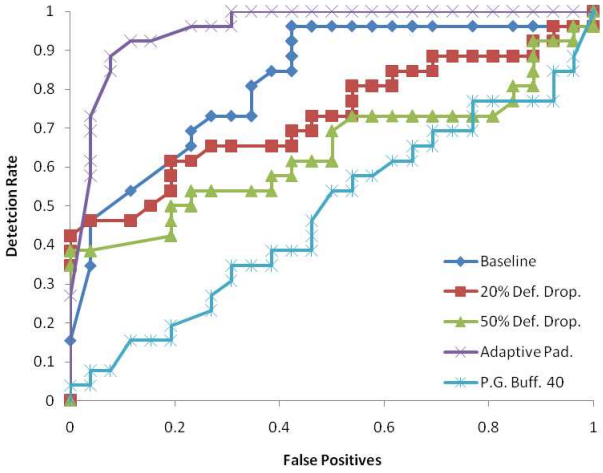


(a)

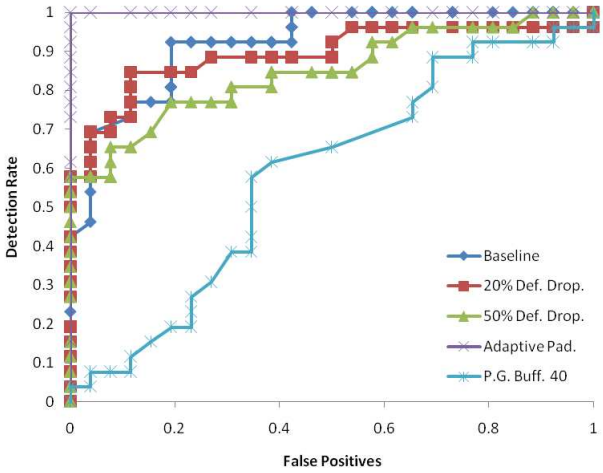


(b)

Figure 6.3. **Eavesdropper Model:** ROC for CC with: (a) 1 minute observation (b) 10 minutes observation.

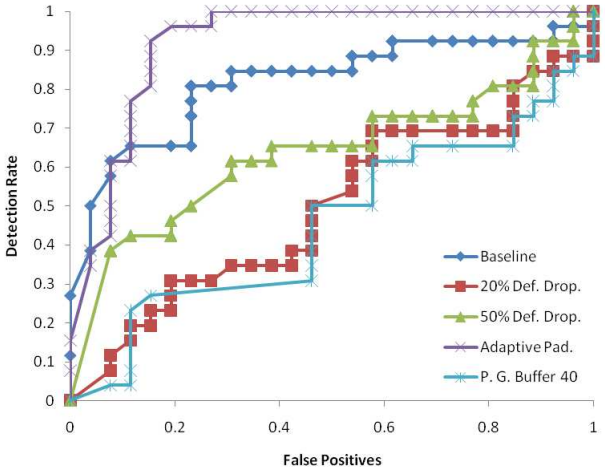


(a)

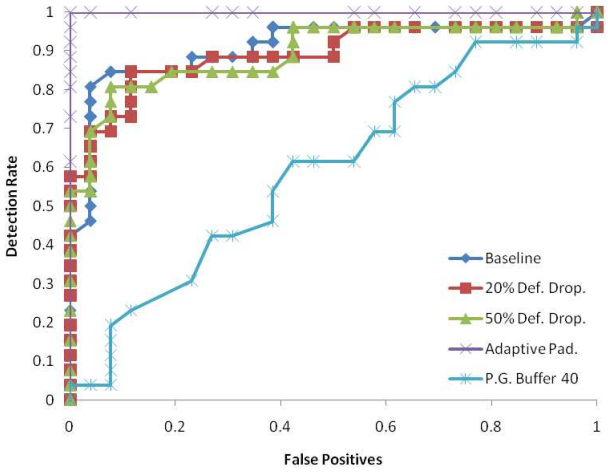


(b)

Figure 6.4. **Compromised Model:** ROC for SCC with: (a) 1 minute observation (b) 10 minutes observation.



(a)



(b)

Figure 6.5. **Eavesdropper Model:** ROC for SCC (a) 1 minute observation (b) 10 minutes observations.

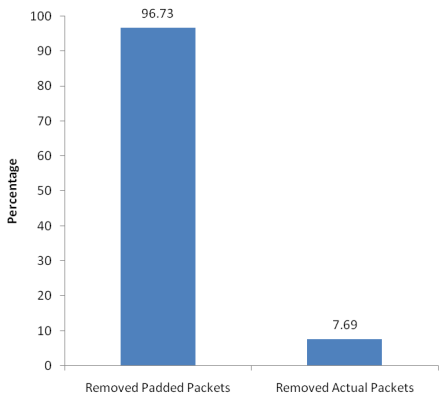


Figure 6.6. **Eavesdropper Model:** Effectiveness of SCC in percentage of packets removed.

CHAPTER 7

CONCLUSION

In this thesis, we proposed selective cross correlation (SCC), a novel technique that a passive attacker can use to de-anonymize users and link them with their communications. The technique works despite the use of defenses that involve insertion or removal of cover traffic, which was effective against other passive attacks and some active attacks. We conducted network experiments to show that the attacker can use SCC to get very good correlation results between the streams, even for short-lived connections. With this, we show that passive attacks against low-latency anonymity systems remain effective and important to study, and there is a pressing need to come up with new defenses against timing analysis.

APPENDIX A

SUBROSA CONFIGURATION

In this appendix, we present the configuration parameters for adaptive padding as used in SubRosa for the experiments, presented in this thesis.

Table A.1. **SubRosa configuration**: adaptive padding

Section	Parameter	Value	Default	Description
Server	SendBufferType	Integer	0	0 – No buffering, 1 – Fixed number of packets, 2 – packets based on γ multiplier, 3 – Adaptive padding
	AdaptiveCount	Integer	100	Count of IPDs to be read from the file
	AdaptiveDistributionFile	String	""	File having the IPDs
	DontPickAboveBin	Integer	10	For random low we shouldn't be picking above this Bin
	DontPickBelowBin	Integer	0	For random low we shouldn't be picking below this Bin
	DontPickAboveBinHigh	Integer	30	For random high we shouldnt be picking above this Bin
	DontPickBelowBinHigh	Integer	0	For random high we shouldnt be picking above this Bin
	RenewThreshold	Integer	100	No. of tries before new IPDs are read for client distribution
Client	PacketGenerationType	Integer	1	Packet Generation: 1 – Constant, 2 – Exponential, 3 – Web Traffic
	IPDDistribution	String	""	File having the IPDs for Web traffic

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 303–320.

[2] A. Pfitzmann, B. Pfitzmann, and M. Waidner, "Isdn-mixes: Untraceable communication with very small bandwidth overhead," in *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*. Springer-Verlag, 1991, pp. 451–463.

[3] D. Chaum, "Untraceable electronic mail,return address and digital pseudonyms," in *Communications of the ACM*, vol. 24. ACM, 1981, pp. 84–88.

[4] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *ESORICS*, 2006, pp. 18–33.

[5] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix systems (extended abstract)," in *Proceedings of the 8th International Financial Cryptography Conference (FC 2004), Key West, FL, USA, February 2004, volume 3110 of Lecture Notes in Computer Science*. Springer, 2004, pp. 251–265.

[6] H. Daginawala and M. Wright, "Studying timing analysis on the internet with subrosa," in *Privacy Enhancing Technologies Symposium*, 2008, pp. 133–150.

[7] O. Berthold, H. Federrath, and S. Köpsell, "Web mixes: a system for anonymous and unobservable internet access," in *International workshop on Designing privacy enhancing technologies*. New York, NY, USA: Springer-Verlag New York, Inc., 2001, pp. 115–129.

[8] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, pp. 66–92, 1998.

[9] P. S. Adam, A. Back, U. Mer, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity," in *Information Hiding (IH 2001)*. Springer-Verlag, LNCS, 2001, pp. 245–257.

[10] A. Serjantov and P. Sewell, "Passive attack analysis for connection-based anonymity systems," in *ESORICS*, 2003, pp. 116–131.

[11] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*. McGraw-Hill Companies, 1974.

[12] X. Fu, B. Graham, R. Bettati, and W. Zhao, "On effectiveness of link padding for statistical traffic analysis attacks," in *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS*, 2003, pp. 340–349.

[13] X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2003, pp. 20–29.

[14] A. Houmansadr, N. Kiyavash, and N. Borisov, "Rainbow: A robust and invisible non-blind watermark for network flows," in *NDSS*, 2009.

[15] P. Peng, P. Ning, and D. S. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in *IEEE Symposium on Security and Privacy*, 2006, pp. 334–349.

[16] P. Syverson and G. Tsudik, "Towards an analysis of onion routing security," in *In Workshop on Design Issues in Anonymity and Unobservability*. Springer-Verlag, LNCS, 2000, pp. 96–114.

[17] D. Keating, S. Parks, D. Smith, and A. Evans, "The multifocal erg: unmasked by selective cross-correlation," in *Vision Res*, vol. 42, no. 27,

2002, pp. 2959–68. [Online]. Available: http://www.biomedsearch.com/nih/multifocal-ERG-unmasked-by-selective/12450505.html

[18] DETER, http://www.deterlab.net/.

[19] Tor, http://www.torproject.org/overview.html.en/.

BIOGRAPHICAL STATEMENT

Titus Abraham was born in Kuwait, in 1982. He received his B.E.(Computer Science And Engineering) degree from Madurai Kamaraj University, India in 2004. From 2004 to 2008, he was with the Mobile Software Lab of LG Electronics in Seoul, South Korea as a Senior Software engineer. He has been a part of the iSec, the information Security Lab, from 2009. From May 2010, he will join Qualcomm as a Senior engineer. His current research interest includes 3GPP LTE and anonymous communication.