Selective Cross Correlation in Passive Timing Analysis Attacks against Low-Latency Mixes

Titus Abraham
Dept. of Computer Science and Engineering
The University of Texas at Arlington
Email: titus.abraham@mavs.uta.edu

Matthew Wright
Dept. of Computer Science and Engineering
The University of Texas at Arlington
Email: mwright@cse.uta.edu

***Abstract*—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. A recently proposed defense called adaptive padding makes cover traffic more effective by ensuring that statistically unusual gaps between packets are partially filled in with dummy packets.**

In this work, we propose Selective Cross Correlation (SCC), an attack that an eavesdropper could employ to de-anonymize users despite the use of adaptive padding. The main insight of our approach is that, with the defense, the timings at one end of the stream are effectively a subset of the timings at the other end of the stream. By considering the network conditions, an appropriate correlation window can be found and used to effectively remove the cover traffic, thereby enabling us to correlate both ends of the stream. We have conducted real network experiments and have found that SCC greatly improves attacker effectiveness over prior techniques against the defense. With SCC, the attacker is nearly as successful as when no defense is applied. This attack demonstrates the need for more robust defenses against statistical timing attacks.

I. Introduction

Anonymity systems, such as Tor [5] 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, 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*, which prevents any single proxy from being able to observe user behavior. 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.

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 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. While such threats are not actually part of the Tor attacker model [5], we believe them to be reasonable attackers worth addressing. In particular, an attacker with several high-bandwidth nodes could add them to the Tor network and thereby capture the first and last node of a substantial fraction of all paths [1]. Whether through eavesdropping or control of some mixes, 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* [12]. In adaptive padding, dummy packets (padding) are added to the stream by the mixes. When the user's traffic rate is low, the mixes increase the padding rate to prevent correlation. Shmatikov and Wang demonstrated in simulation that this technique was very effective against eavesdroppers.

A. Contributions

In this paper, we first describe the continuing importance (§II) of investigating passive timing analysis in low-latency mixes. We then describe a model (§III) in which these attacks and defenses can be studied. Our most important contribution is to describe and evaluate a new timing analysis technique based on *Statistical Cross Correlation*, or SCC (§IV). The main insight of this technique is that, even with padding, the incoming stream's timings are preserved and the resulting outgoing stream consists of cover traffic superimposed on the

original stream. A filter 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 employ a basic statistical correlation to determine whether the streams actually match.

To validate this technique, we conducted experiments (§V) in the DETER network testbed, using the SubRosa system for evaluating timing analysis attacks and defenses [3]. Our results (§VI) show that a passive attacker can correlate the streams with high accuracy despite the use of adaptive padding.

II. Background

In this section we discuss timing analysis attacks on low-latency anonymity systems and defenses against timing-based attacks.

A. 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 (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 [11] can be used to determine the similarity of two streams and thereby link the sender with the reciever. This idea was improved by Levine et al. [8] by applying cross correlation (CC) over packet counts to improve error rates. CC is a measure of the similarity of two wave forms [9]. In this technique, each stream is split into non-overlapping windows of equal sizes and packet counting is used to apply cross correlation. This method works with reasonable error rates in the presence of multiple constant-rate input streams under the condition that the network exhibits jitter and packet drops before the first mix. This is true for most Internet connections, as has been shown in real network experiments [3]. 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 [6].

An *active adversary* actively perturbs the incoming traffic by manipulating packet delays or by injecting or dropping packets. This technique, also called watermarking [13], is used by an adversary to uniquely identify streams. Houmansadr et al. use both watermarking and a timing analysis technique called selective correlation in an active attacker model [7]. However watermarking can be detected [10] and thereby passive timing analysis attacks remain relevant and important to measure and improve anonymity systems [7].

B. 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 [8]. 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 [8].

1) Adaptive padding: Web traffic is bursty; 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 [12] is a defense in which intermediate mixes insert dummy packets into the streams to reduce the number and size of 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 statistical distribution, the packet is forwarded and a new inter-packet arrival is calculated. If a packet does not arrive upon expiry, 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 the 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.

III. Model

To analyze the threats posed by timing analysis attacks on low-latency mixes we study a Tor-like network topology where 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.

We also assume an *eavesdropper* model for the attacker. In this model, the attacker can monitor the incoming stream for the first mix and outgoing stream of the last mix. The two streams are then correlated to compromise the anonymity of the system. This is slightly more challenging for the attacker than if the attacker has compromised the first and last mixes.

IV. Selective Cross Correlation

In this section we discuss the main insight of selective cross correlation, followed by a description of how it works and how we measure its effectiveness.

Selective cross correlation is a two-step process that consists of preprocessing the incoming and outgoing streams, which allows cross correlation to then be applied. Even with padding, packet timings of the incoming streams are preserved in the outgoing streams. The resulting outgoing stream effectively consists of cover traffic superimposed on the original stream. In the first step of SCC, this property is exploited to identify and remove cover traffic.

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

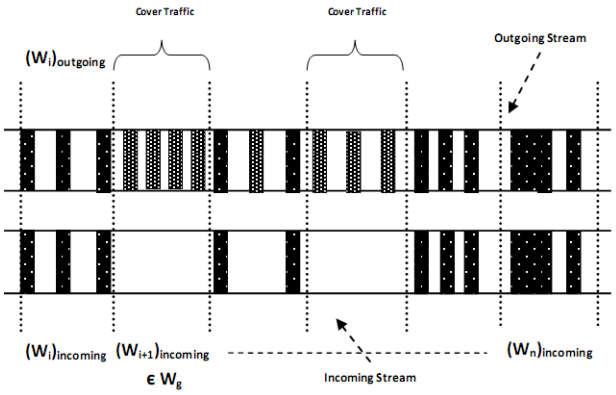

Fig. 1. Selective Cross Correlation

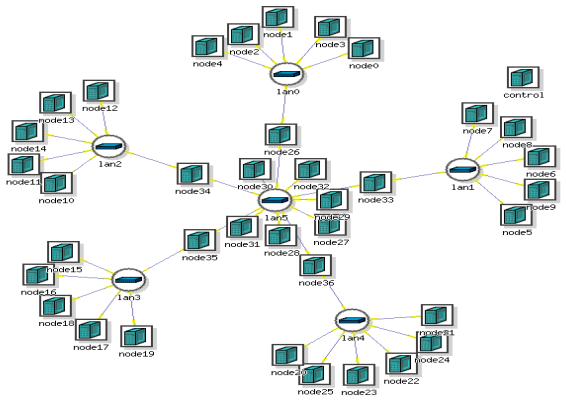

Fig. 2. Network topology in DETER

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 window. This process is repeated for the entire length of the streams. All suspected 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. 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.

As shown in Figure 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 $(W_i)_{\text{incoming}} \in W_g$.

A. 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 out by SCC. We show in Section(§ VI-B) that these constitute a small fraction of the traffic and SCC can filter most of the cover traffic. Thus, correlation can be still done very easily for web traffic.

V. Experiment Design

In this section, we describe our experimental setup, then we describe the traffic generation, and the defenses tested.

A. SubRosa

SubRosa is a system for studying timing analysis methods and defenses [3]. Unlike prior simulation studies [8, 12], SubRosa allows us to experiment with real network traffic. This platform emulates a Tor-like network having three components: Client, Mix, and Sink. The Client acts as the user which is responsible for generating data on the network. The Sink acts as the recipient. This platform has been designed to collect timing information as observable by an attacker. Each 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 route the traffic on the selected path and add adaptive padding if configured to do so.

B. DETER

DETER [4] 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 2. We chose a flower topology, with the petals hosting the Clients and the core hosting the Mixes. Since the petals have different cross traffic from each other, both core and petal nodes have independent cross traffic.

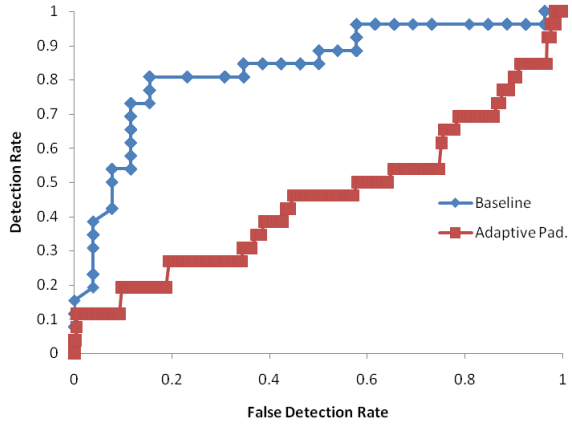

Fig. 3. **Eavesdropper Model:** ROC for CC

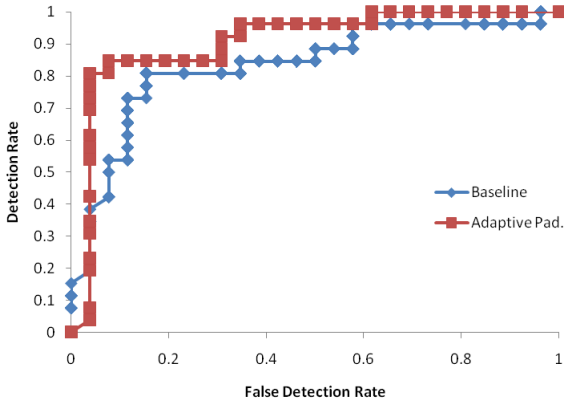

Fig. 4. **Eavesdropper Model:** ROC for SCC

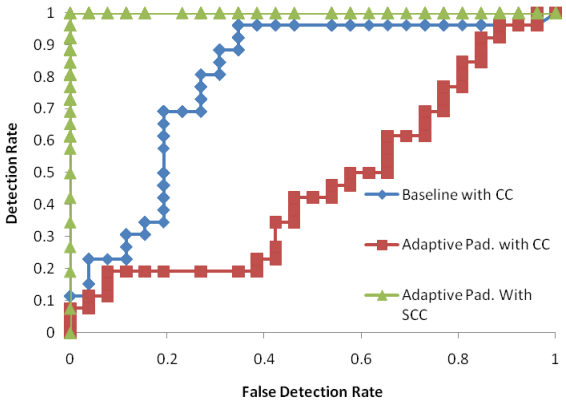

Fig. 5. **Ten minutes of traffic:** SCC and CC against adaptive padding in the eavesdropper model

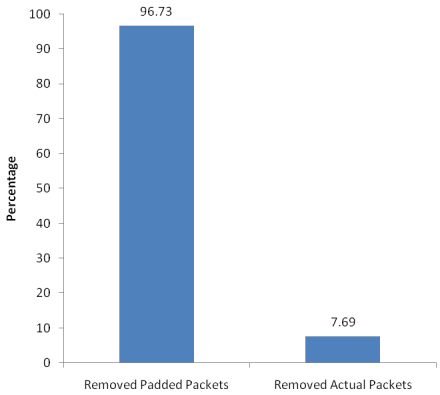

Fig. 6. **Eavesdropper Model:** Average percentage of padding and actual packets removed by SCC

C. Traffic Generation

In our experiments, 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 the University of North Carolina at Chapel Hill. We isolated HTTP streams in the streams using the destination port number.

In the Internet, there is cross traffic with different characteristics and a variety of protocols. To generate such cross traffic we used SEER, a traffic generator tool of DETER. We categorized traffic conditions as low, medium, and high average cross traffic (20 Mbps, 50 Mbps, or 80 Mbps) for a 100 Mbps link capacity.

D. Defenses Tested

We ran experiments with 26 Clients and five Mixes, where each of the Clients randomly choose three of the Mixes as a part of the circuit. Timing analysis is done based on 1 minute run of the experiments.

1) Baseline: Clients generate constant-rate traffic in which packets get generated 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 [5].

2) Adaptive padding: We implemented Adaptive Padding on Sub Rosa, based on the description of Shmatikov and Wang [12].

VI. Results and Discussion

We now show the relative effectiveness of CC and SCC against the defenses in the eavesdropper model. 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 [12]. 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 [12]. Thus, we tested CC and SCC against adaptive padding with short-lived connections, where the observation time was set to one minute.

A. Eavesdropper Model with CC

We now present result from experiments in the eavesdropper model. 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 very effective against cross correlation, as the eavesdropper does not know which packets are padding. Confirming the results of Shmatikov and Wang [12], the statistical properties of the outgoing stream are not easily linked to the incoming stream, resulting in .48 error rate as shown in Figure 3.

B. Eavesdropper Model with SCC

We next tested SCC in the eavesdropper model. Figure 4 shows that SCC achieves a very high detection rate against adaptive padding with very low false positives. The success rate is nearly as good as in the compromised mix scenario. The key to SCC's effectiveness is its ability to remove padding packets from the outgoing stream. In Figure 6, we see that SCC was able to remove 96% of the padding, while incorrectly removing only 7.7% of non-padding packets.

We also evaluated the defenses for a ten minute trial in the eavesdropper model. Figure 5 shows that CC for baseline and adaptive padding results in a near constant detection rate for increasing durations of data available for sampling. On the other hand, the detection rate of SCC increases as it gets more data. This can be attributed to the the nature of SCC – the longer a stream, the longer the pattern that can be identified and extracted, resulting in improved detection.

VII. Conclusion

In this paper, 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 adaptive padding, 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.

VIII. Acknowledgements

We greatly appreciate the staff and volunteers who operate the DETER testbed for making our experiments possible. We would also like to thank the anonymous reviewers for their comments that helped improved the paper. This work was supported in part by the National Science Foundation under award numbers CNS-0549998 and CAREER award CNS-0954133. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References

[1] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against Tor. In *Proc. WPES*, 2007.

[2] O. Berthold, H. Federrath, and S. Köpsell. Web mixes: a system for anonymous and unobservable Internet access. In *Proc. Intl. Workshop on Designing Privacy Enhancing Technologies*, 2001.

[3] H. Daginawala and M. Wright. Studying timing analysis on the Internet with SubRosa. In *Proc. PETS*, 2008.

[4] DETER. http://www.deterlab.net/.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. USENIX Security Symposium*, 2004.

[6] X. Fu, B. Graham, R. Bettati, and W. Zhao. On effectiveness of link padding for statistical traffic analysis attacks. In *Proc. IEEE ICDCS*, 2003.

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

[8] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright. Timing attacks in low-latency mix systems (extended abstract). In *Proc. Financial Cryptography*, 2004.

[9] Alexander M. Mood, Franklin A. Graybill, and Duane C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill Companies, 1974.

[10] P. Peng, P. Ning, and D. S. Reeves. On the secrecy of timing-based active watermarking trace-back techniques. In *Proc. IEEE S&P*, 2006.

[11] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. In *Proc. ESORICS*, 2003.

[12] V. Shmatikov and M.-H. Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In *Proc. ESORICS*, 2006.

[13] X. Wang and D. S. Reeves. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In *Proc. ACM CCS*, 2003.