This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2011

A Novel Packet Size Based Covert Channel Attack against Anonymizer

Zhen Ling*‡, Xinwen Fu† Weijia Jia‡, Wei Yu§ and Dong Xuan¶
*Southeast University, Nanjing 211189, P. R. China. zhenling@seu.edu.cn
†University of Massachusetts Lowell, Lowell, MA 01854, USA. xinwenfu@cs.uml.edu
‡City University of Hong Kong, Kowloon, Hong Kong SAR. wei.jia@cityu.edu.hk
§Towson University, Towson, MD 21252, USA. wyu@towson.edu
¶The Ohio State University, Columbus, OH 43210, USA. xuan@cse.ohio-state.edu

***Abstract*—Anonymizer is a proprietary anonymous communication system. We discovered its architecture and found that the size of web packets through Anonymizer are very dynamic at the client. Motivated by this finding, we investigated a novel packet size based covert channel attack, against the anonymity service. In the attack, one attacker manipulates the web packet size between the web server and Anonymizer and embed signal symbols into the target traffic. An accomplice at the user side can sniff the traffic and recognize the secret signal. We developed intelligent and robust algorithms to cope with the packet size distortion incurred by Anonymizer and Internet. We developed several techniques to make the attack harder to detect: (i) We pick up right packets of web objects to manipulate in order to preserve the regularity of the TCP packet size dynamics; (ii) We adopt the *Monte Carlo* sampling technique to preserve the distribution of the web packet size despite manipulation. We have implemented the attack over Anonymizer and conducted extensive analysis and experimental evaluations. It is observed that the attack is highly efficient and requires only tens of packets to compromise the anonymous web surfing. The experimental results are consistent with our theoretical analysis.**

***Index Terms*—Anonymizer, Covert Channel, TCP dynamics.**

I. Introduction

Anonymizer is a commercial anonymous communication system. In this paper, we present a novel covert channel attack that may drastically degrade the Anonymizer service. This covert channel exploits the varying size of packets through Anonymizer and is one type of active traffic analysis [1], [2], [3], [4]. Such active attacks can reduce the false positive rate significantly and don't require massive traffic training required in passive traffic analysis attacks [5], [6].

We will present the first exposure of the Anonymizer architecture, which consists of anonymizing server and client. The server consists of reverse proxy/NAT, SSH server and HTTP proxy, while the client software is a SSH port forwarding configuration tool. We found that the size of HTTP packets through Anonymizer is very dynamic and random at the client. Motivated by this finding, we designed the novel covert channel attack against the Anonymizer service. In this attack, the attacker between the malicious web site and the victim client can embed a secret message into the packet size variation of target traffic. This attacker can be the owner of the malicious web server or one manipulating (repacketizing) the traffic between the web server and Anonymizer server. Without loss of generality, we use the former case as the example in this paper. An accomplice at the client side can sniff the traffic and recognize the secret message. Given the small size of the Anonymizer network, such sniffing is feasible to organizations or people with modest power. To cope with packet size distortion caused by Anonymzier and Internet (e.g., packet padding, packet merging, limited TCP buffer and various MTU), we design intelligent and robust detection algorithms to recover the message. In this way, the anonymity service provided by Anonymizer is compromised.

The attack can be made hard to detect. (i) To attack a HTTP session, we repacketize the web traffic into virtual web objects, and modulate secret messages bits into the size of last packets of these virtual web objects. The last packet of a web object is denoted as the least significant packet for brevity and clarity. The size of a least significant packet is very dynamic in comparison with other packet sizes. Modulation of successive packets to carry message bits will disrupt TCP packet size dynamics (as illustrated in Figure 7), which can be measured by Hurst parameter from R/S plot [7], [8]. This least significant packet based covert channel approach can preserve TCP regularity and self-similarity (as illustrated in Figure 8) while the attacker can control the number of virtual objects to control the number of message bits. (ii) To preserve the size distribution of web packets of virtual web objects, we apply the *Monte Carlo* sampling technique to carefully sample the empirical cumulative distribution function (ECDF) of the least significant packet size of real web objects. This requires the input of the *Monte Carlo* method should be random and uniformly distributed. To this end, we first encrypt the message. The generated ciphertext bits are uniformly distributed and encoded into k-ary symbols. A k-ary symbol can then be mapped to a packet size by a *Monte Carlo* sampling technique.

We implemented this novel covert channel attack against Anonymizer and performed extensive theoretical analysis and real-world experiments over Anonymizer. The attack achieves high detection rate with *very low false positive rate*. The experimental results are consistent with our theoretical analysis. To the best of our knowledge, the attack presented in this paper is the first exploiting the Anonymizer architecture and degrading its anonymity via packet size based covert channel. It is simple, efficient, and hard to detect. Compared with related attacks [1], our attack requires just tens of packets to achieve high

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

detection rate and low false positive rate.

The remainder of this paper is organized as follows: We explore the components of both Anonymizer server and client, and report our finding that size of web packets in Anonymizer network is very dynamic in Section II. In Section III, we introduce the covert channel based on least significant packets. Extensive experimental results are presented in Section IV. We review related work in Section V and conclude this paper in Section VI.

II. Exploration of Anonymizer

In this section, we first present the Anonymizer architecture discovered by our passive inspection of traffic into and out of Anonymizer. We have replicated the discovered Anonymizer architecture in a lab environment and is able to use the Anonymizer client software to browse the web through the lab Anonymizer servers. This verifies our discovery. We then show that the size of web packets in the Anonymizer client is very dynamic.

A. Architecture

The *Total Net Shield service* (*TNS*) from Anonymizer [9] is a commercially available anonymizing service. It is claimed that *TNS* (used with Anonymizer alternatively later for brevity and clarity) protects personal information by hiding a source computer's identity. Figure 1 shows its three basic components: (i) *Anonymizer Client*: The client runs commercial software to anonymize the client data to the server. (ii) *Anonymizer Server*: It consists of a reverse proxy/Network Address Translation (NAT) server, several SSH (Secure Shell) port forwarding servers, and proxy servers. (iii) *Application Server*: It runs TCP applications such as web service.

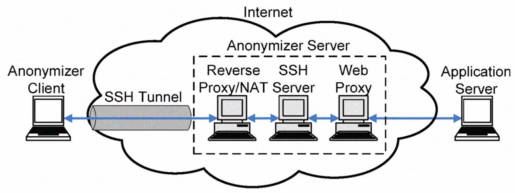

Fig. 1. Architecture of Anonymizer Network

1) Components of Anonymizer Server: The Anonymizer server consists of three components: one reverse proxy/NAT server, several SSH servers and web proxies. The reverse proxy/NAT server dispatches inbound client traffic to the SSH servers. For load balancing, the reverse proxy/NAT uses a cluster of SSH servers and web proxy servers. For content privacy and communication anonymity, the client TCP traffic of POP3, SMTP, FTP and HTTP is encrypted and sent to port *22* of a SSH server via a SSH tunnel as shown in Figure 1. The traffic is then decrypted and forwarded to port *80* of a web proxy. At last, the proxy server forwards the traffic to the destination. The reverse traffic follows the same path in the reverse order.

2) Components of Anonymizer Client: The client establishes a SSH connection to the SSH port forwarding server. The default cipher is AES-CBC with a 256-bit key (*AES256-CBC*). The default MAC is *HMAC-SHA1*. The client software sets the default domain name of Anonymizer SSH server and port as *cyberpass.net* and *22*, respectively. For surfing (the focus of this paper) anonymously, the browser should use the local Anonymizer proxy. Users can manually configure the SSH server, local listening port, SSH port forwarding destination, proxy server, encryption algorithm, and MAC algorithm in the client software.

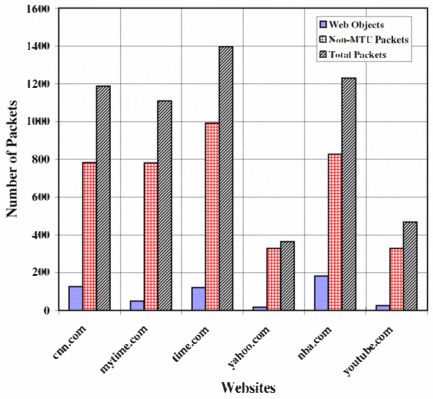

Fig. 2. Number of Web Objects, Non-MTU Packets and Total Packets after Decryption at Anonymizer Client

B. Dynamic Packet Size of HTTP Traffic in Anonymizer

Using the Anonymizer client, we surfed a number of websites, including CNN, Yahoo, YouTube, and others and captured a large number of web packets out of the SSH tunnel at the Anonymizer client. In Figure 2, we show the number of web objects, non-MTU packets and total packets in the web traffic, respectively. We can see that the number of non-MTU packet with random size is much larger than the number of web objects. We also found that in a sequence of packets from one web object, a number of non-MTU packets are randomly located in different places.

These observations can be reasoned as follows: (i) The Anonymizer server, i.e., SSH server, repacks the web packets. Note that the size of normal packets transmitted between the web server and Anonymizer server is the MTU size. However, the Anonymizer server (i.e., SSH server) will add a SSH header and such padding increases the packet size. Hence, the repacked web packet will be larger than the MTU size. These packets will then be split, and the split packets will be merged with other SSH packets. These split packets will be repacked at the SSH client and this results in the non-MTU packets. (ii) Network dynamics and performance of the Anonymizer may incur those non-MTU packets as well. If the network between web server and Anonymizer server is congested, the TCP sliding window at the web server may not be large enough and a MTU packet will be delivered. If Anonymizer server is not overloaded at that moment, such a packet will be sent to the client promptly. When this occurs, non-MTU packets will be generated.

In summary, the decrypted packet size observed at the client shows a large percentage of non-MTU packet size because of the Anonymizer client software's packet handling and Internet traffic dynamics. It will be hard for the client to detect the attack in Section III by investigating whether non-MTU packets appear within the transmission of a web object.

III. Packet Size Based Covert Channel

Because of web traffic packet size dynamics at the Anonymizer client shown in Section II-B, the packet size variation can be explored for a covert channel over Anonymizer to compromise the anonymity service. In this section, we first introduce the least significant packets and then present the basic idea and workflow of the attack. We discuss some practical issues and present our solutions at last.

A. Least Significant Packets

Normal HTTP packets (not through Anonymizer) can be roughly categorized into two classes. A *Class I* packet is defined as the largest packet in normal HTTP traffic, i.e., 1500 bytes in case of Ethernet, including an IP header of 20 bytes and a TCP header of 32 bytes. The size of HTTP content in the TCP payload is 1448 bytes. A *Class II* packet has a size less than 1500 bytes. Such packets are usually generated by the "tail" of a web object, i.e., the last packet when the web object is downloaded. If the web object size with the HTTP header is w bytes, the size of the web object "tail" is $(w \ mod \ 1448) + 20 + 32$ bytes. In this paper, we denote *Class II* packets as *least significant packets* for brevity and clarity.

By analyzing the traffic from 30 well known web sites including CNN, Yahoo and YouTube, we obtain the the empirical cumulative distribution function (ECDF) of the size of raw HTTP packets in *Class II* as shown in Figure 3. The raw HTTP packet size does not include the IP header and the TCP header. Also, the ACK packet is ignored because of its zero length. The mass probability function (MPF) of the least significant packets is $\{p_1^{'}, p_2^{'}, \ldots, p_m^{'}\}$ and the corresponding packet sizes are $\{pc_1^{'}, pc_2^{'}, \ldots, pc_m^{'}\}$. $p_i^{'}$ is the probability of the packet size $pc_i^{'}$. Therefore, the ECDF of the least significant packet size can be formalized as follows,

$$F_{lsp}(pc_i^{'}) = P(x \leq pc_i^{'}) = p_1^{'} + p_2^{'} + \ldots + p_i^{'}. \quad (1)$$

B. Basic Idea of Covert Channel over Anonymizer

Without loss of generality, we assume that the attacker is between a malicious web site and Anonymizer server and will embed a secret message into the target traffic packet size variation. This attacker can be the owner of the malicious web server or one manipulating (repacketizing) the traffic between the web server and Anonymizer server. We use the former case to introduce the attack in this paper.

The basic idea of this attack is as follows. An attacker at the malicious web site controls the reverse proxy to embed a secret message into the web traffic. An accomplice of the attacker sniffs the traffic at the client side and determines if that client has received the traffic embedded with the secret message. The message can be represented as a sequence of symbols (for example, "0000" to "1111") and one symbol corresponds to one packet size. We virtually generate web objects with different sizes, repacketize the web traffic, and choose appropriate size of a least significant packet for a message symbol.

To make the covert channel hard to detect, we need to preserve the least significant packet size ECDF in Figure 3. This is the criterion for mapping a symbol to a least significant packet size. A classical way for preserving an ECDF is *Monte Carlo* sampling. In our case, we divide the y axis [0, 1] of the ECDF into equal segments such as the 16 segments in Figure 4. The 16 segments corresponding to symbols from "0000" ("0") to "1111" ("F"). Therefore, one symbol can be uniformly mapped to a few packet sizes along the x axis of the ECDF. We need to guarantee that one symbol corresponds to at least one packet size. To preserve the ECDF, Monte Carlo sampling requires the message has uniformly distributed symbols. To achieve this, we encrypt the message first and transmit the cyphertext over the covert channel. A strong cipher generates uniformly distributed symbols in the ciphertext [10].

Figure 5 illustrates the workflow of the covert channel attack. Please refer to our technique report [11] for details of the two procedures of embedding a message into target traffic and recovering the message from the target traffic.

IV. Evaluation over Anonymizer

In this section, we use real-world experiments to demonstrate the feasibility and effectiveness of the covert channel attack based on least significant packets (LSPs). All the experiments were conducted in a controlled manner over the commercial Anonymizer and we experimented on TCP flows generated by ourselves to avoid legal issues. Please refer to our technical report [11] for theoretical analysis of the performance of the covert channel based traceback attack based on least significant packets. We have derived detection rate and false positive rate formulas and investigated what factors impact the attack effectiveness.

A. Experiment Setup

Figure 9 illustrates the experiment setup. We deploy a malicious web site and a reverse proxy on Campus A. The web server and reverse proxy are installed on a single computer. The web server is Apache/2.2.11 (Linux) and the reverse proxy is Pen [12]. Two other computers are deployed on Campus B. All computers are running Fedora Core 11 operating system. One computer acting as a client is connected to a wireless access point and its traffic is not encrypted over the network. The web browser is Firefox 3.5. We configure Firefox to not cache data. The other computer is used as a sniffer to record the size of packets destined to the client computer with source TCP port *22*.

We modified the code of the reverse proxy *Pen* to manipulate web packet size and implement the encoding algorithm. For the verification purpose, we downloaded the real-world web pages from CNN.com and deployed our own "CNN" web server to simulate a malicious web site. By configuring the reverse proxy *Pen*, we map the reverse proxy port *8080* to the HTTP server port *80*. Hence, the reverse proxy can forward the packets for the web server. At the client side, the SSH client connects to the commercial Anonymizer server by the command *"ssh -L 80:cyberpass.net:80 username@cyberpass.net -N"* in the console with appropriate password. We configure the browser HTTP proxy as "127.0.0.1:80". In this way, we use Firefox to browse the web server via the remote reverse proxy port "8080" and fetch the web objects via Anonymizer.

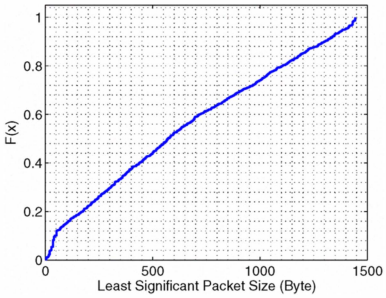

Fig. 3. ECDF of Least Significant Packet Size

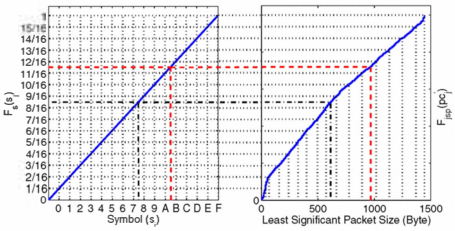

Fig. 4. Mapping between the Symbols and the Least Significant Packet Sizes

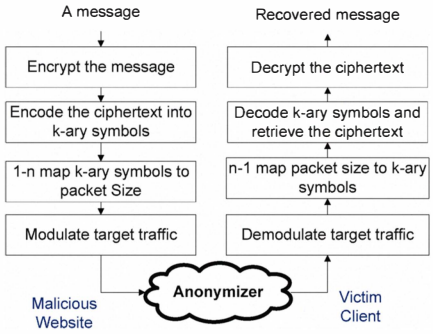

Fig. 5. Workflow of Encrypted Covert Channel Attack based on Least Significant Packets

B. TCP Packet Size Dynamics in Attacks

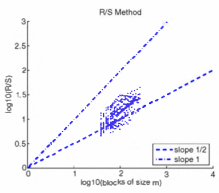

Fig. 6. R/S Plot of Original Packet Sizes

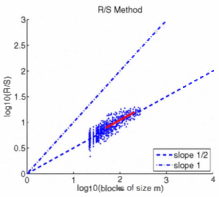

Fig. 7. R/ Plot for Simple Attack

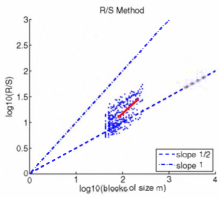

Fig. 8. R/S Plot for LSP Covert Channel

Figure 6 is the R/S plot ([7], [13]) of an original CNN web packet size sequence. The Hurst parameter is much greater than 0.5. Therefore, the web packet size is self similar. Figure 7 illustrates the R/S plot of the packet size sequence for the simple attack embedding a message by changing size of successive web packets from CNN. It can be observed that *Hurst* parameter is around *0.5* and implies pure randomness. The simple attack destroys the TCP packet size dynamics and is easier to detect. Figure 8 is the R/S plot of the packet size sequence from CNN for the least significant packet (LSP) based covert channel attack. The Hurst parameter is also much greater than 0.5. The least significant packet based covert channel preserves web packet size self-similarity and is hard to detect.

C. Detection Rate

To validate the accuracy of the attack using least significant packets, we let the client browse our replicated web pages 30 times. At the reverse proxy, we generate a message and encrypt it with RC4 in counter mode. We then derive a sequence of symbols of length 20. Note that we generate HTTP objects of different size and calculate the location of the least significant packets, i.e. the location of our symbols. When the target web traffic arrives at the reverse proxy, we choose the symbol location, vary the read buffer and embed the symbols into the target traffic. At the client side, Sniffer records the SSH packet size by removing the MAC (IEEE 802.11) header, IP header and TCP header.

To evaluate the false positive rate of the attack, we let the client browse our replicated web page 30 times via Anonymizer. However, no symbol is embedded into the traffic at the reverse proxy in these cases. We refer to the traffic without symbols as clean traffic. We then use the same detection algorithm proposed in [11] to detect 20 random symbols from the clean traffic and calculate the false positive rate.

Figure 10 illustrates the relationship between the detection rate and the delay interval, as well as the threshold S_t in [11]. Notice that S_t is the threshold to restrain the difference between the real symbol packet position and its predefined position. From Figure 10, we have a few observations. First, the best value of S_t is around 8. This can be reasoned as follows: The smaller S_t may not detect packets carrying symbols. The larger S_t may erroneously pick packets that do not carry a symbol, but has the same size as the packet carrying a symbol. As a result, the detection algorithm cannot correctly recognize the later symbols. Second, the detection rate increases dramatically when the delay interval increases. This matches our analysis in [11] very well. The detection rate approaches 100% when the delay interval is 350ms and the threshold S_t is 8. These results validate that the attack using least significant packets can significantly degrade the degree of anonymity service that Anonymizer promises.

Figure 11 illustrates the relationship between the detection rate and the delay interval, as well as the number of symbols. Figure 11 shows that the detection rate will decrease while the number of symbols increases, which is matched with our analysis in [11]. From this figure, we know that only tens of packets is needed for our covert channel attack. This observation confirms that the attack is highly efficient and can compromise the anonymous web surfing very fast.

We did not plot the false positive rate since in all the cases, the false positive rate approaches 0. This matches with our analytical results in [11] very well.

V. Related Work

There are a large number of related works on traffic analysis and covert channel. We only review the most related ones because of the space limit. Ramsbrock *et al.* [14] and Gianvecchio *et al.* [15] applied packet size based covert channel to Botnet and general network traffic. The TCP packet size dynamics was not considered in their work. We are the first to apply packet size based covert channel against the Anonymizer service and explored various packet size distortion by Anonymizer proxies. Ling *et al.* [4] proposed the cell counter based attack against Tor [16] that the attackers

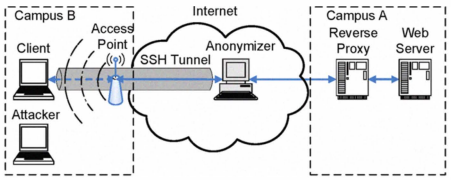

Fig. 9. Experiment Setup

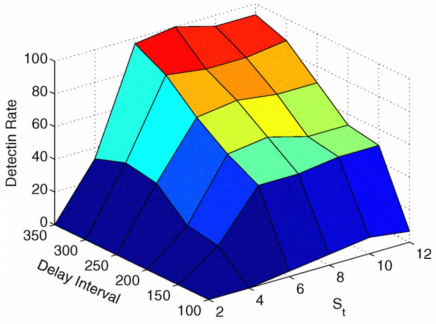

Fig. 10. Detection Rate vs. Delay Interval and S_t

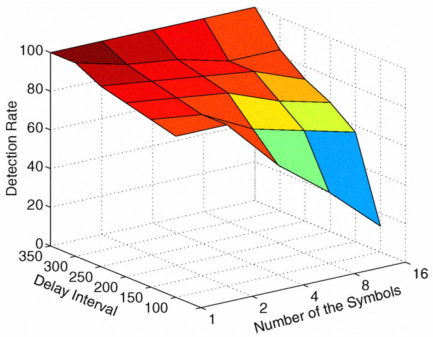

Fig. 11. Detection Rate vs. Delay Interval and Number of Symbols

embed a signal into the variation of cell counter of the target traffic by varying the counter of cells in the target traffic at the malicious exit onion router and did not explore weakness of Anonymizer.

VI. Conclusion

In this paper, we discovered the architecture of Anonymizer and investigated a novel covert channel attack based on least significant packet size variation to drastically degrade the anonymity service provided by Anonymizer. We developed several techniques that make the attack efficient, accurate, and hard to detect. In particular, we applied the *Monte Carlo* sampling technique to carefully sample the least significant packet size ECDF in order to preserve its distribution. We designed techniques to choose right packets of web objects in order to preserve the regularity of the TCP packet size dynamics measured by the *Hurst* parameter and R/S plot. All these efforts make the attack practical and more undetectable. We also designed intelligent and robust detection algorithms to recover the distorted symbols caused by Anonymizer and Internet traffic dynamics. Extensive analysis and experiments were conducted to validate the effectiveness and feasibility of the proposed attack. Our data show that the covert channel attack could dramatically and quickly degrade the anonymity service by Anonymizer. Defending against the proposed attack remains a challenging task. We plan to work with Anonymizer developers and investigate the solution in our future work.

Acknowledgment

This work was supported in part by USA NSF grants 0942113, 0958477, 0943479, 0907964, 0546668 and 0916584, by the Army Research Office (ARO) under grant AMSRD-ACC-R50521-CI, by Research Grants Council of Hong Kong SAR, No. (CityU 114609), CityU Applied R & D Funding (ARD-(Ctr)) No. 9681001, ShenZhen-HK Innovation Cycle Grant No. ZYB200907080078A, and NSFC under grants 61070222/F020802, 60903162, 60903161, 90912002 and 61003311, National Key Basic Research Program of China under grant 2010CB328104, China National Key Technology R&D Program under grants 2010BAI88B03, China Specialized Research Fund for the Doctoral Program of Higher Education under grant 200802860031, China National S&T Major Project under grant 2009ZX03004-004-04, Jiangsu Provincial Natural Science Foundation of China under grant BK2008030, Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201, and by Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under grant 93K-9. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor agencies.

References

[1] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of the IEEE Symposium on Security & Privacy (S&P)*, May 2007.

[2] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, May 2007.

[3] A. Houmansadr, N. Kiyavash, and N. Borisov, "Rainbow: A robust and invisible non-blind watermark for network flows," in *Proceedings of the 16th Network and Distributed System Security Symposium (NDSS)*, February 2009.

[4] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A new cell counter based attack against tor," in *Proceedings of 16th ACM Conference on Computer and Communications Security (CCS)*, November 2009.

[5] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix-based systems," in *Proceedings of Financial Cryptography (FC)*, February 2004.

[6] 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 (PET)*, May 2004.

[7] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended)," *IEEE/ACM Transactions on Networking*, vol. 2, February 1994.

[8] J. Beran, *Statistics for Long-Memory Processes*. Chapman & Hall, October 1994.

[9] Anonymizer, Inc., http://www.anonymizer.com/, 2010.

[10] J. Soto and L. Bassham, "Randomness testing of the advanced encryption standard finalist candidates," in *NIST IR 6483, National Institute of Standards and Technology*, 1999.

[11] Z. Ling, X. Fu, W. Jia, W. Yu, and D. Xuan, "A novel packet size based covert channel attack against anonymizer," Computer Science Department, Texas A&M University, Tech. Rep., 2010.

[12] "Pen," http://siag.nu/pen, 2010.

[13] R. G. Clegg, "A practical guide to measuring the hurst parameter," in *Proceedings of 21st UK Performance Engineering Workshop*, 2005.

[14] D. Ramsbrock, X. Wang, and X. Jiang, "A first step towards live botmaster traceback," in *Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID)*, September 2008.

[15] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-based covert timing channels: Automated modeling and evasion," in *Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection (RAID)*, September 2008.

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