SUBROSA 2 : AN EXPERIMENTAL EVALUATION OF TIMING ANALYSIS ATTACKS AND DEFENSES IN ANONYMITY SYSTEMS

by

PAYAP SIRINAM

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
August 2012

Copyright © by PAYAP SIRINAM 2012

All Rights Reserved

To my family for their blessings

and The Royal Thai Air Force Academy who made me who I am.

ACKNOWLEDGEMENTS

I would like to express my profound appreciation to my supervising professor Dr. Matthew Wright for constantly motivating and encouraging me, and also for his invaluable advice and devotion during the course of my thesis study. I wish to thank Dr. Vassilis Athitsos and Dr. Gergely Zaruba for their interest in my research and for taking time to serve on my thesis committee.

I would also like to extend my gratitude to Royal Thai Air Force Academy for providing financial support for my M.S. studies. I am especially grateful Vishal Gupta with his good discussion and comments as well as all ISEC lab members for good moral. I also wish to thank DETER, a public facility for medium-scale repeatable experiments in computer security in which I have mainly implemented and evaluated my thesis.

Finally, I am grateful to all the teachers who have taught me during the years I have spent in school, first in Thailand, and finally in the Unites States. I would like to thank Assoc.Prof.Dr. Prasong Praneetpolgrang for encouraging and inspiring me to pursue graduate studies.

August 06, 2012

ABSTRACT

SUBROSA 2 : AN EXPERIMENTAL EVALUATION OF TIMING ANALYSIS ATTACKS AND DEFENSES IN ANONYMITY SYSTEMS

PAYAP SIRINAM, M.S.

The University of Texas at Arlington, 2012

Supervising Professor: Matthew Wright

A circuit-based low-latency anonymous communication service such as Tor helps Internet users hide their IP addresses and thereby conceal their identities when communicating online. However, this kind of service is vulnerable to timing analysis attacks that can discern the relationship between incoming and outgoing messsages in order to find correlations between them. The attacker can use this information to reveal the idenity of the internet users without knowing the IP addresses concealed in the anonymous communication services.

Dependent link padding (DLP) is a scheme propsed to enable anonymity systems to resist these attacks. However, DLP adds high overhead from dummy packets in the network systems, resulting in poor quality of service.

We have developed a Tor-like experimental evaluation platform called *SubRosa 2* for studying and investigating the overall dummy packets overhead on each scheme that is used to prevent timing timing analysis attacks. We have developed our platform on real distributed networks by using the DETERLab network testbed, which is a public facility for medium-scale repeatable experiments in computer security. In our experiments, we evaluated DLP and reduced overhead dependent link padding (RO-DLP). Furthermore, We compared these schemes to a recently-proposed technique called selective grouping (SG) that aims to further reduce overhead from dummy packets in the padding algorithms at the cost of some anonymity.

Through evaluations of the whole anonymity systems, we validated that RO-DLP could significantly reduce dummy packet overhead and enable larger number of users to be protected from timing analysis attacks in comparision to DLP implementation. We also showed that SG could practically reduce the network overhead with a lower ratio of dummy packets overhead reduction than the previous work proposed. We also deeply investigated the factors and causes to explain the lower ratio of reduction when we implemented SG on the real distributed networks. Furthermore, we performed the partial implementation of SG on some mix nodes with a circuit to compare the results with full implementation of SG. Finally, we showed that SG could enable larger numbers of users participated in the systems when compared with DLP and RO-DLP without SG.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

There have been a number of anonymity systems proposed for anonymous network communications such as Tor [1], I2P [2] and AN.ON [13]. Tor is one of the most commonly used that allows people and groups to improve their privacy and security on the internet. Individuals use Tor to keep a communication on the internet from tracking them. Journalists use Tor to communicate more safely with whistleblowers and dissidents. Non-government organization (NGOs) use Tor as the medium to their organization website without notifying that they are working with that organization [6]. Tor seeks to frustrate attackers from linking communication partners, or from linking multiple communications to or from a single user [1]. Tor provides anonymity to internet users by concealing users' location as well as other forms of identifying factors such as Internet Protocol (IP) address.

In low-latency communication, an anonymous system can implement defenses such as reordering of messages, delaying messages, batch processing and so on. In contrast with interactive communication that has to meet strict latency requirements, the defenses applied in non-interactive communication are not viable options. Due to this limitation, it makes the communication vulnerable to timing attacks, in which an attacker examines the timings of packets moving through the system and then finds the correlation between the sender and the receiver in order to disclose their identities [3]. In order to protect the network against these attacks, some defenses have been proposed such as independent link padding (ILP) [7], dependent link padding (DLP)

[4], reduced overhead dependent link padding (RO-DLP) [5], and selective grouping (SG) [14].

1.1 Contribution

To investigate the related-performance results on each algorithm in the real-world applications, we have developed Tor-like experimental evaluation platform called *SubRosa 2* to evaluate the efficiency of implementing given algorithms including DLP, RO-DLP and SG. In the last algorithm, we would like to show that this algorithm can practically decrease overhead from dummy packets in the real distributed network with real amount of traffic by looking at a whole system.

In chapter 2, we describe the background context for our works including the basic knowledge of low-latency anonymity systems, timing analysis attack, and defenses against these attacks. Chapter 3 details experimental evaluation platform called SubRosa 2. The main objectives of our work are to implement the given dummy padding schemes in the real distributed network and to validate the traffic overhead of their implementations as well as evaluate the result to find constraints and limitations. For validation, we conducted several experiments using the *UMass* network trace to reflect the real human behaviors in the real anonymity systems. In chapter 4, we evaluated and made a discussion on the experimental results. We also explain some limilations and constraints. Finally, we conclude our works by considering all outcomes of experimental results in chapter 5.

CHAPTER 2

BACKGROUND

In this chapter, we describe low-latency anonymity systems, timing attacks, and defenses used against such attacks. We also describe examples of experimental evaluation platforms.

2.1 Low-Latency Anonymity Systems

Low-latency anonymity systems enable users to communicate in a manner that is untracable by adversaries [10]. They are designed based on the idea of mixes, in which users connect to the internet via a chain of proxies with sophisticated protocols to hide their identities from attackers. The most common anonymity systems that are well-implemented in interactive communications are Tor [1], Anonymizer [18], Web MIXes [16] and I2P [15].

Tor is a distributed overlay network mainly designed to anonyize TCP-based applications. Its design is based on the idea of mixes to create a private network pathway, in which the user's software or client increamentally builds a circuit of encrpted connection through relays on the network. The circuit is extended one hop at a time, and each relay along the way knows only which relay gave it data and which relay it is giving data to. No individual relay knows the complete path that a data packet has taken [6]. The Tor network is operated by volunteers from all over the world. They are running their machines called oninon routers (ORs) [1]. When the Tor client runs an onion proxy (OP) locally. This OP will fetch directories, establish circuits by randomly selecting nodes from the set of guard nodes, middle nodes and exit nodes for builing the path and handling connections from user applications.

Anonymizer is an internet privacy company. It offers a variety of consumer information security services [18]. The anonymizer proxy server is one of the services which protects the anonymity of users. If somebody uses their service for illegal activities, they will disclose the customer's information to the authorities.

Web MIXes [16] is the anonymity system designed on the MIX-based system for anonymous and unobservable real-time Internet access. It adds a mechanism on the mix to preserve the anonymity of users by adding dummy packets when an active client become idle. Moreover, it uses a ticketing mechanism for user authentication to prevent flooding attacks and provides the current level of protection information to the users. This system is operated by JAP (Java Anon Proxy) on the client-side and MIXes and cached-proxy on the server-side. The users connect to MIXes through a JAP anonymous tunnel (MIX-cascade) to acquire the anonymous communication [17].

I2P is another anonymity system with the aim to support the efforts of trying to build a more free society by offering them an uncensorable, anonymous, and secure communication system. I2P is a development effort producing a low latency, fully distributed, autonomous, scalable, anonymous, resilient, and secure network. I2P is building a peer to peer network that takes advantage of the anonymity and security of mixnets, the performance, scalability, and resilience of distributed hash tables, and the global interoperability of the Internet based on the Kademlia algorithm [19]. Communication between individuals does not need to expose the location or identity of those communicating to each other or to a third party attempting to monitor their activity, even if the third party has unlimited resources dedicated to doing so.

2.2 Timing Analysis Attacks

A timing attack is one of the significant threats against anonymity systems that support low-latency applications and attempts to prevent the privacy of users [2,7]. The attacker tries to examine the timing patterns of messages moving through the system to find correlations. Timing attacks can be broadly classified into two categories: passive and active [7].

In a passive timing attack, the attacker collects and observes network traffic information in term of packets timing. Based on inter packet delays (IPDs) which is the difference between the two consecutive packets flowing on the incoming and outgoing with respect to a mix node, the attacker can statistically correlate them with patterns in other traffic that it observes. If he can discern between the user and the destination, he can thereby link the two and finally disclose the identiy of the user [8].

In an active timing attack, the adversary collects and observes network traffic by inserting the specific timing pattern such as delaying, injecting or dropping packets into the traffic as it passes through routers under his control [9]. This technique is called watermarking, in which the attacker can observe and find the correlation of incoming and outgoing stream which already have the specific timing pattern previously added by the adversary. However, watermarking attacks can be prevented by tracing back through stepping stones and replacing the distorting watermarks with original one [4].

2.3 Defenses Against Timing Analysis Attacks

There have been defenses proposed on the anonymity systems to prevent timing attacks such as defensive dropping, independent link padding(ILP), dependent link padding(DLP), reduced overhead dependent link padding(RO-DLP) and selective grouping(SG) [14]. They can overcome these attacks depending on the models of attack and real network systems.

2.3.1 Independent Link Padding (ILP)

ILP is the defense, in which all flows of traffic in the network are padded according a pre-defined rate [6,10]. The timing pattern of output stream will be exactly same with the rest of clients since the timing and rate of packets in outgoing flows is not dependent of the timing pattern and rate at the input i.e. all packets are scheduled to send at the constant time intervals regardless of a given delay [10]. As the consequence, an adversary cannot find the correlation between incoming and outgoing packets. This method is impractical and might cause some problems since if the traffic flows in the network being routed are bursty (e.g., web-browsing traffic) i.e. if the some of the clients send the packets at the rate lower than pre-defined padding rate, it will result in adding long delay between packets till they reach their schedule. On the other hand, if some of the clients send packets at a high rate suddently, then the constant padding algorithm will drop most of packets due to the the pre-defined interval. Bandwidth consuming is one of major drawbacks of this method. This is primary due to the fact that the output pattern is always padded regardless of constant rate of time e.g. even if there is no packet coming into the mix node, it will always pad dummy packets regardless of the given rate. With this, the algorithm uses enormous bandwidth for dummy packets. Moreover, it is shown that links padded by constant rate schedule are still vulnerable to traffic analysis as the variance of packet timing be correlated to the system loading [11].

Moreover, ILP is implemented in another way by adding dummy packets using the Poisson process [9]. However, the limitation of this implementation is that a server needs to know average sending rate to perform efficiently [4]. Due to this implementation, it is also difficult to vary the padding rate with respect to sending rate becuase the adversaries can use the client information to find the correlations from the variety of this rate.

2.3.2 Defensive Dropping

Defensive droping is a generization of *partial-path cover traffic* and generizes the idea of *partial-route padding*. In this defense, The client adds duumy packets within the network traffic along with the real packets and mark it to be dropped at any intermediate mix nodes at random [7]. With this, if a mix node is an honest participant, it will strictly drop the dummpy packet rather than send it out to the next mix nodes. Droping packet will occur whenever the mixe nodes are required to make the output timing pattern exactly similar across all clients. If defensive droping defense is randomly performed with sufficiently large frequency, the adversaries will not be able to find the correlations [3].

2.3.3 Dependent Link Padding (DLP)

Due to the drawbacks of ILP scheme, DLP schemes can be implemented to overcome these problems [4,11]. To protect user flows from a matching attack, the DLP dynamically adds dummy packets regardless to the rate of incoming data flow resulting in reducing number of dummy packets and packets drop rate. Moreover, DLP applies Δ time implementation to save dummy packets by delaying incoming packets on Δ interval time before processing forwarding and dummy packets generating processes as shown in Figure 2.1. This method provides full anonymity via manipulating the outgoing packet timing patterns similar to the rest of clients. Furthermore, there is a common relationship between anonymity and sending rate of incoming packets

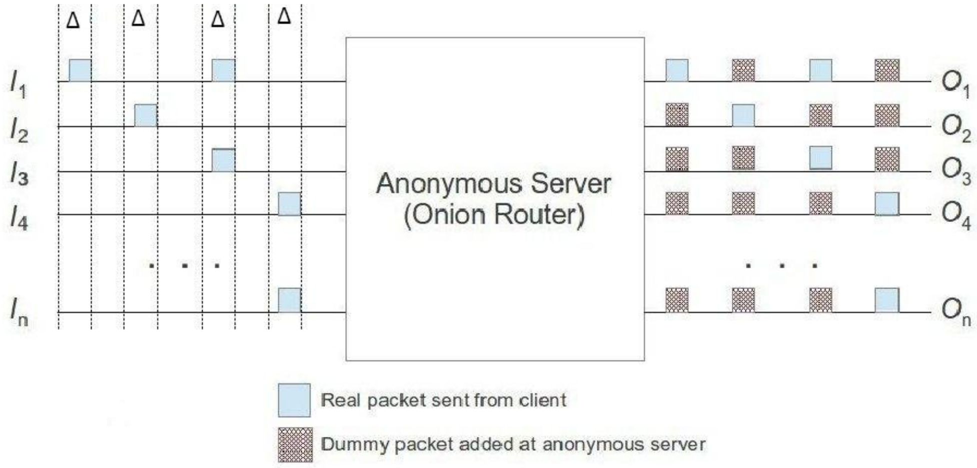


Figure 2.1. DLP implementation with I incoming flows and O output flows.

with different arrival rate distribution. When flows are in the Poisson process, the minimum sending rate is log_m for the system to reach the full anonymity level for m users flows. The rusults for Pareto distribution flows show that the sending rate will become a constant when the number of flows reach to infinity.

DLP can also implement a heuristic dropping to control the sending rate when the number of clients flows is extremely large with in the network. To implement this algorithm, the author defines token utility for a token as $u = \frac{d}{|F|}$, where d is the number of message packets sent by the token and lFl is the size of the incoming flow set of DLP algorithm. With this, it is easy to see that $\frac{1}{|F|} \leq u \leq 1$, as each token scheduled by DLP will send only one packet for each flow [4].

2.3.4 Reduced Overhead Dependent Link Padding (RO-DLP)

RO-DLP [6] is an advanced implementation of DLP and can significantly reduce the dummy packet overhead to an efficiently applicable number in anonymity systems and can maintain the same level of anonymity when we compare this method with a regular DLP. In the original DLP algorithm [4][12], nodes pad every outgoing circuit

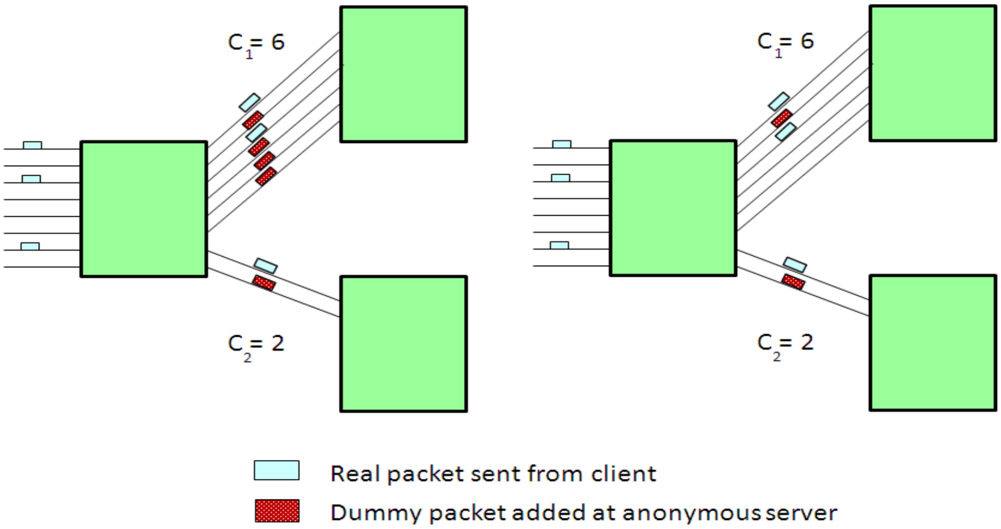

$C_1 = 6$
$C_2 = 2$
$C_1 = 6$
$C_2 = 2$
Real packet sent from client
Dummy packet added at anonymous server

Figure 2.2. Original DLP (left) and RO-DLP (right).

in the same way without considering whether or not some circuits are being multiplxed over the same link. In the anonymous network, link encryption is commonly implemented to hide the correspondence of cell to circuits within a link. In Figure 2.2, it compares the simple DLP to RO-DLP and clearly shows that RO-DLP can reduce the amount of dummy traffic sent over links that multiplex several circuits, while preserving the same level of security against global external adversaries that do not control nodes.

The intuition behind this scheme is the following, Given that at time t the node forwards R_t cells, it is enought to send R_t cells over links that contain a number c_i of circuits that is larger than R_t. Let us consider a node n that routes C circuits over L links and let c_i denote the number of circuits multiplexed over the same link l_i. Initially, RO-DLP schedules a cell for each of C outgoing circuits, as in DLP. Thus, at time t a set of C cells are scheduled, of which R_t correspond to cells that are being forwarded, and C - R_t are dummy cells generated by node n. RO-DLP can removes r_i dummy cells from link l_i [5] as folows:

$$r_i = \begin{cases} 0 & \text{if} \quad c_i \leq R_t \\ c_i - R_t & \text{if} \quad c_i > R_t \end{cases}$$

The attacker observes the number of cells arriving at node n and can predict the number of R_t of cell that will be forwared at time t. When $c_i > R_t$ cells are sent over link l_i, the adversary has sufficient information to know that(at least) c_i - R_t of these are dummy cells generated by n and thus these do not provide any additional protection.

2.3.5 Selective Grouping (SG)

According to the original DLP implementation, an alogrithm has to pad dummy packets to generate the same output pattern to provide the full anonymity level. In the consequence, it adds very high overhead on the network system causing the performance problem. However, in the real world, the sending rate of network traffic is various according to the different communication procols as well as the users' behavior i.e. file sharing users have a very high sending rate in comparison to the users who are chatting with their friends. This variation in sending rate directly results in the number of dummy packets added into the systems.

Vishal Gupta and Mathew Wright proposed a selective grouping padding algorithm (SG) to mitigate this problem by developing this algorithm from the idea of DLP with adding the implementation of clustering algorithms. SG will be able to decrease the dummy packet overhead by catagorizing the users according to a delay bound parameter into different groups while maintaining good anonymity [14]. Figure 2.3 shows that when this algorithm is implemented, it can reduce the number of dummy packets added into the system. As you can see, the way to add the dummy

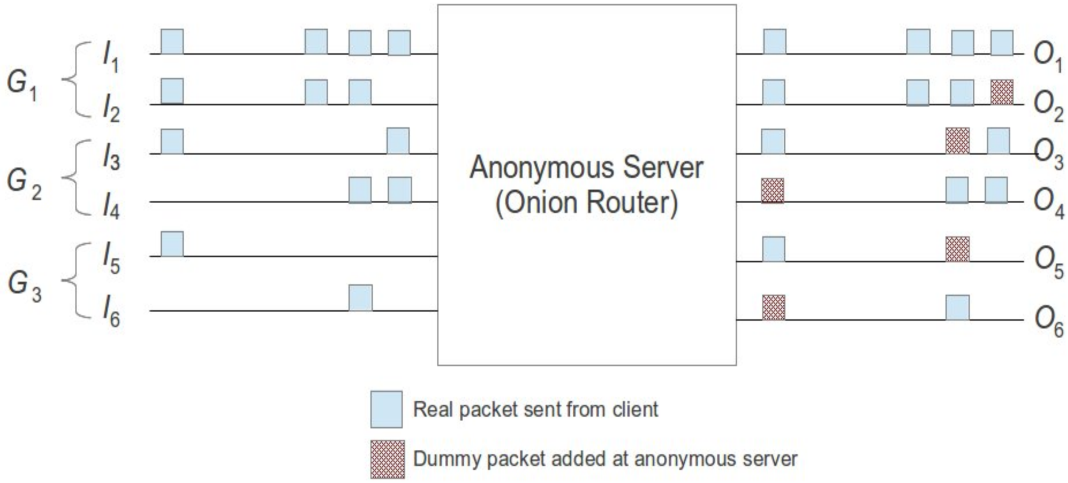


Figure 2.3. Selective Grouping Implementation.

packets are independent among the different groups can help reduce the dummy packets overhead in network resulting from the different of sending rate of clients.

The authors performed experiments by using *UMass* traces to simulate with the real web-browsing traffic timing information. Moreover they performed experiments using different clustering algorithms and density distributions by conducting extensive simulation experiments to find a threshold value at which selective grouping achieves good profiling without adding excess dummy packets.

For grouping, the authors have mostly used sequential clustering and k-means clustering algorithms in selective grouping. In this algorithm, They firstly calculated count of packets of each user within specific interval window. Once calculation of packets is completed, then they count the client's real packets in the current cycle and sort the clients on the basis of packets count as the hash key and split them sequentially.

K-means clustering with d dimension is another method in which all N clients are partitioned in K clusters, in which each client belongs to the cuslter of nearest mean [20]. In the simulations, they used *2* dimensions, which are packet count and standard deviation of each client. Moreover, they also considered minimum group size and divided larger groups into smaller size.

2.4 Experimental Evaluation Platforms

In order to study the practical implementation on the ideas proposed for the anonimity systems, experimental evaluation platforms have been created to validate the results of experiments with the main objective to answer the questions such as " what is the effectiveness of the algorithms in the real-distributed networks ? ", " what is the additional problem that might occur during implement the proposed solution in the real world systems ? " and so on. There have been recent experimental platforms proposed for the reasons above such as SubRosa [7], NS-2 [13], ExperimenTor[20], Tempura[24] etc.

2.4.1 SubRosa

SubRosa is timing analysis on the Internet studying platform focusing on timing analysis attacks and defenses in low-latency anonymity systems [7]. The researchers presented results of experiment on PlanetLab, a globally distributed network tested. The experiments emphasized on validating the major conclusion obtained by prior simulation studies. The authors also proposed a new lightweight defense, γ-buffering, and showed the limitations of this approach. They also implemented a defensive dropping defense with 20% and 50% drop rate. Moreover, They introduced spike analysis which is a new timing analysis technique taking advantage of unusual delays in a network flow to reduce errors over prior techniques.

The fundermental idea of his implementation is to model SubRosa on Tor-like systems but choose UDP as the transport layer protocol which is more flexible rather than TCP. SubRosa is a simple application for collecting timing data and does not use encryption.

2.4.2 Network Simulator 2(NS-2)

NS-2 is a discrete event simulator targeted at networking research [13]. It provides substantial support for simulation of TCP routing and multicast protocols over wired and wireless (local and satellite) networks. It can be used for many purposes such as measuring overhead and delay in overlay networks.

2.4.3 ExperimenTor

ExperimenTor is created to be a large scale Tor network emulation toolkit and testbed [20]. The primary objective for this testbed is to conduct Tor research in a manner that preserves realism while protecting live users' privacy. Moreover, it performs experiments to improve the network's resilience to attacks and enhance its performance. However, the author mentioned some interesting viewpoints about the challenges of building a Tor testbed. First, to modeling live tor network is difficult becuase it is essential to accurately model the distribution of Tor router bandwidth that is available for entry guard, middle node, and exit node routers. In addition to accurate Tor router models, it is also important to accurately model Tor client behavior. Second, it needs large-scale network emulation. Network emulation platforms such as Emulab [21] and DETER [22] have the limitation in term of scalibility. Finally, it should run native Tor and application code. Rather than re-implementing specific components of Tor, the authors wished to run the unmodified and complete Tor code in order to provide a higher degree of realism.

2.4.4 Tempura(Tor MultiPath Routing)

Tempura is the testbed created to study the implementation of multipath routing for Tor which a traffic splitting algorithm that forwards a client's individual cells down multiple circuits that share a common exit Tor router [24]. With this, the testbed measures the throughput of each constituent circuit and assigns traffic to each proportion to its observed throughput. According to the measurement, it showed that Tempura can improve approximately 30% in expected download time for web browsers who use Tor brige. This reseach also show that there is no significant impact on users' security or anonymity.

CHAPTER 3

EXPERIMENTS

We created an experimental evaluation platform working on a real distributed network to implement given schemes, collect experimental results, and evaluate them. The experiments were conducted on the DETER testbed with stratified network topology. We call this platform *SubRosa 2*.

3.1 DETER

DETER is a testbed advancing cyber security research practices, by extending the methods, technology, and infrastructure required for scientific development of cyber-defense technology [23]. Researchers can access the DETER remotely from hundreds of institutions wordwide and perform their experiments usings a wide set of network and computing infrastructures. DETER enables the researchers to create their project lab environment and remotely access to the DETER's user interface for experimenter workbench suporting and helping users manage experiments through their complete life cycle as well as develop, configure, and manipulate collections of nodes and links with nearly arbitrary network topologies [24].

3.2 SubRosa 2

3.2.1 Overview

We designed an experimental platform to emulate the behavior of a Tor-like network over UDP which is connectionless transport. We do not implement encryption features used in the secure systems, as this would make it harder to validate and evaluate the information collected. The primary objective of our platform is to collect data on performance such as the number of dummy packets added and number of packets dropped.

Our platform was written entirely in *Python* and consisted of various componets: clients, mix_nodes, sinks, multi_clients and a central_server. We describe each component in terms :

- **client** is the client application. It plays an important role in creating the circuit on each client and reading the timing information from the *UMass* traces to manipulate the sending rate of packets.
- **mix_nodes** acts as the guard nodes, middle nodes and exit nodes. It is responsible for receiving, padding dummy packets with respect to each algorithm and forwording them to the next nodes in DETER. Moreover, it keeps the essential data for evaluating the results.
- **sinks** is the destination of the data.
- **multi_clients** simulatates multiple virtual clients. Due to the limiation of number of available nodes, we had to virtually implement multiple clients on each physical node.
- **central_server** controls the experiment by sending the pre-processing information to corresponding nodes in the experiments, starting and terminating the experiments.

In term of network topology, There have been topologies implemented in original DLP version such as *free router* and *cascade* but these two topologies cause the problems due to scalability issues, feedback effects [5] and the degree of anonymity provided. With feedback effects, this kind of failure occurs in the *free route* networks topology both with DLP and RO-DLP, and it provokes dummy traffic to be generated even in the absence of real traffic. This case leads to infinite padding overhead. The

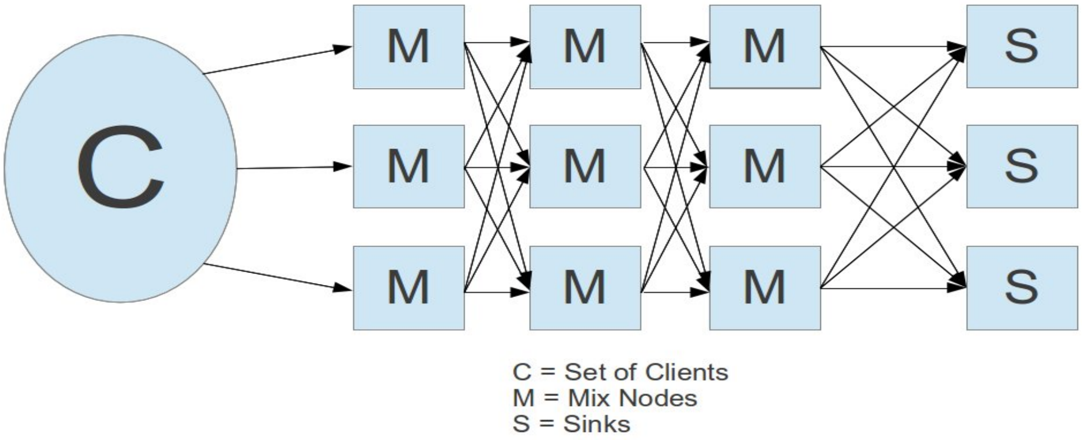


Figure 3.1. Data flowing in the system.

way to handle this problem is to implement stratified network topology in which nodes are divided into the set of guard, middle and exit nodes such that any guard node connects to any middle node and any middle node to any exit one.

Data flow in onion router networks is shown in Figure 3.1. C represents a set of clients particating in the experiment, M represents the mix nodes and S represents the sinks. The client selects the path with length equal to three from the set of nodes including guard nodes, middle nodes and exit nodes and then starts the circuit building process. Figure 3.2 shows five steps involved in the circuit building as described below :

Step 1 the client start establishing the connection with the guard node.

Step 2 extends the connection to the middle node on the path already created at the guard node on step 1.

Step 3 continue extending the connection to the exit node on the same path previously created by using guard node and middle node as a relay.

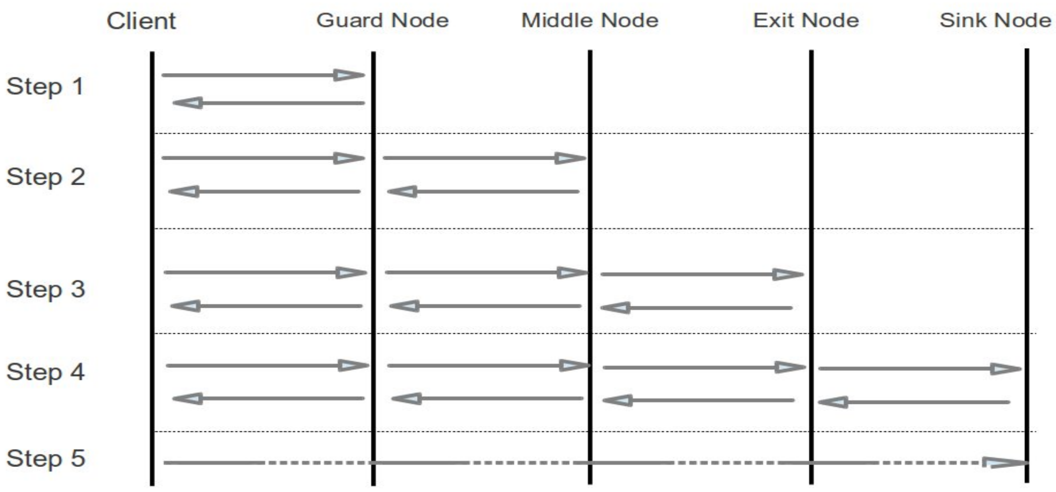


Figure 3.2. Circuit Building Process.

Step 4 set up the connection to the destination(sink) through previous path created.

Step 5 finally, circuit is established, and the client will use this path to forward packets through guard node, middle node, exit node alog the way to the sink node.

In our experiments, we created all circuits participated in each experiment before hand in order to make sure that we could start the circuit together on each experiment. This is due to the fact that the circuit creation process takes time and during the experiment, we have a large number of clients with high volume of traffic. It has a high chance that the circuit creation process cannot finish on time before sending data process starts. As we observed from circuit creation process, the time spends for establishing the connection will vary from 1.5 ms - 1 s as number of clients in the system increases. In order to overcome the variation of circuit creation process time, we created circuits on each round before hand and make them active at a designated round.

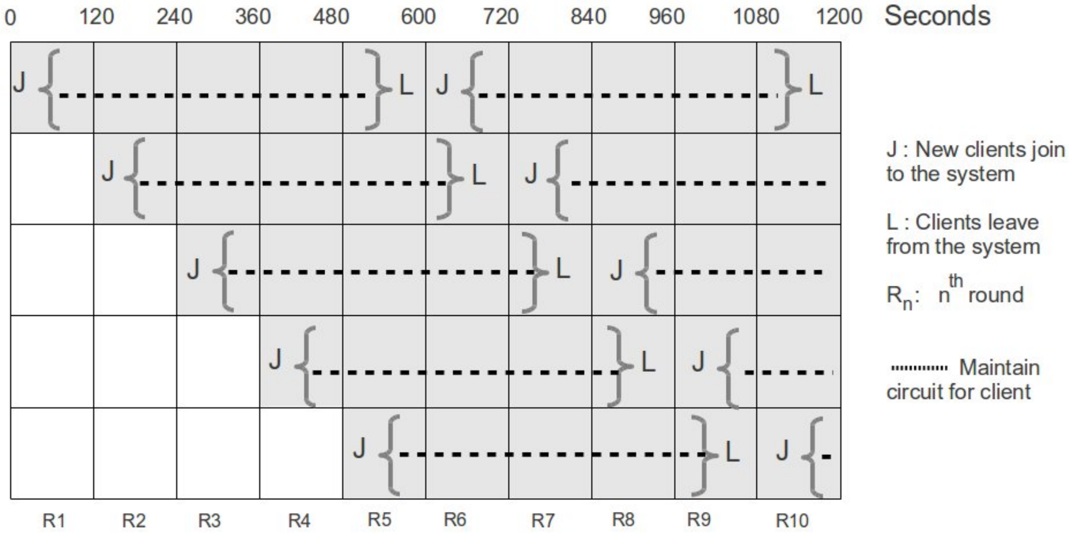

0
120
240
360
480
600
720
840
960
1080
1200
Seconds
J
L
J : New clients join to the system
L : Clients leave from the system
R_n: n^{th} round
Maintain circuit for client
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10

Figure 3.3. Experiment design for joining and leaving of clients on each round.

3.2.2 Methodology

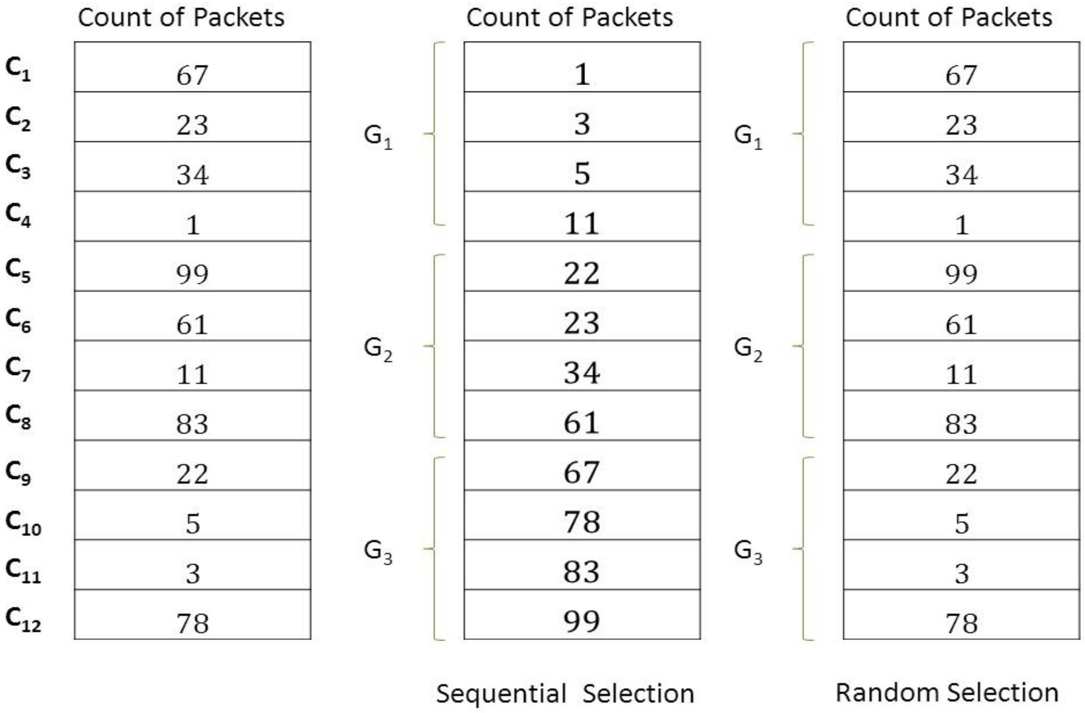
In our experiments, we implemented each algorithm on DETER with path length equal to three. The network topology implemented was stratified network in order to prevent the *free route* effect.

We designed the experiment to simulate the behavior of users that periodically join and leave to the system with 120 s window lengths for each round, and we assumed that the circuit lifetime is 10 minutes e.g. the circuit were started at time $t = 0$ s and terminated at time $t = 600$ s. We did experiment with 10 rounds of windows. The duration of our experiments was approximately 20 minutes each as shown in Figure 3.3. We collected the data on performance on each scheme including the number of the dummy packet overhead and a number of packets dropped . Moreover, we also figured out the limitations of the anonymity system in DETER in term of the maximum number of clients that can participate in the system without causing an unacceptable rate of dropping rate (1 %).

The client nodes were selected from DETER nodes. In this experiment, we used 25 physical node to virtually simulate required number of clients and used others 9 physical nodes to be the mix nodes and 3 physical nodes to be the sink nodes. The set of mix nodes were equally categorized into set of guard nodes, middle nodes and exit nodes. Experiments were conducted for DLP with 250 clients and for RO-DLP with 1,000 clients and packets were generate by using the *UMass* Trace.

Dummy packet padding algorithms are implemented on the mix nodes to perform dummy packet padding algorithms. Once the mix node receives packets from the networks, it would create the dummy packets and then forward them to the next nodes along with the incoming packets to prevent timing analysis attack as described in the introduction part.

We simulated circuit's connections by implementing multi-threading process. Each connection independently consumes its shared system's resource and creates its own network socket with its own port number. We assigned circuit label by using random number and determined the circuit life time. The circuit will be active when it receives the signal form *central_server* component which is an application running on experimental controller's node and automatically terminated according to the designated circuit life time. Due to our design, the participated clients would gradually increase and reach to the required number of clients of the experiment after starting the 5^{th} round and then continously maintain the total number of clients participating in the experiment by joining and leaving of clients as shown in Figure 3.3. The experiment would be terminated after finishing the 11^{th} round and keep collecting data from 1^{st} round to the 10^{th} round.



	Count of Packets		Count of Packets		Count of Packets
C_1	67	G_1	1	G_1	67
C_2	23		3		23
C_3	34		5		34
C_4	1		11		1
C_5	99	G_2	22	G_2	99
C_6	61		23		61
C_7	11		34		11
C_8	83		61		83
C_9	22	G_3	67	G_3	22
C_{10}	5		78		5
C_{11}	3		83		3
C_{12}	78		99		78

Figure 3.4. Grouping methods for SG.

3.2.3 Validation and Implementation

DLP and RO-DLP are implemented on the mix-nodes to validate the dummy packet overhead and their limitations in term of the maximum number of users that can participate in the system with acceptable rate of packets dropped and dummy packets overhead in the whole anonymity systems. We would like to extensively investigate the practical implementation of a recently-proposed scheme called SG presented by Vishal Gupta. He performed SG with DLP and showed that SG could significantly reduce dummy packets overheads. His experiments were performed on only one virtual mix node.

SubRosa 2 studied results of performing SG with DLP and RO-DLP on the real anonymity environment including the set of mix nodes as described on methodology part. We used three clustering methods including sequential selection, sequential selection with standard deviations and random selection. In sequential selection, we counted the incoming packets of each circuit in the first round of its circuit's life time. After the system had finished counting packets, it sorted them with respect to the packets count and grouped them into each group as shown in Figure 3.4. Each client follows previous cycle's grouping and sends packets by considering only clients existing in its respective group until the circuit is terminated. We also applied sequential selection with standard deviation on the systems by statistically analyzing bursty traffic in the network. If one client sends bursty traffic, it will affect all other clients of that group resulting in adding more overhead. In order to implement this method, we first performed sequential selection process and then filtered all bursty traffic clients by selecting the n clients who have top-ranked of high standard deviation values and add them into the new group, which substantially helps decrease overhead. We finally applied the random selection by randomly group all client into each group without using any statistical information.

Moreover, we extensively study the partial implementation of SG by performing it only on guard nodes for DLP and on middle nodes and exit nodes for RO-DLP along with regular DLP and RO-DLP without SG on middle nodes and exit nodes to evaluate the results in term of whether it could reduce the overhead.

3.3 Experimental Challenges

We created SubRosa 2 by using Python programing and running on the DETER testbed, There were challenges occurring during our development that worth considering. We faced the system limitations in term of the maximum number of thread allowed in DETER. With this problem, we overcame it by changing the configurations of linux operating system by reducing the stack size in *ulimit* command to be 512 KB. It would allow us to increase the maximum number of threads. Besides, we encountered the problem of network socket forwording. In DETER, we can use alias name to be network address for forwarding each packet to the destination in DETER e.g. *node1.sneak.isi.deterlab.net.* However, when we used this alias name to identify their addresses for each circuit especially with high volume of traffic, we confronted with the problem of having high number of packets dropped. This is due to the fact that every packet being forwarded has to be converted its alias name and use NS lookup to find the physical address. With this process, it takes some time and becomes vulnerable to be error with high volume of traffic. We solved the problem by using the exact IP addresses instead.

CHAPTER 4

RESULT AND DISCUSSION

In this chapter, we present the results on our experiments by showing the data on performance in terms of dummy packet count generated in the whole network and the maximum number of clients that can participate in the system before and after implementing SG. We performed experiments validating DLP, RO-DLP to establish the base line for comparisons and compared with the results of SG schemes with given number of clients. We also evaluate and make a discussion on the results.

4.1 Implementation of SG on DLP and RO-DLP

In this experiment, we implemented SG on DLP and RO-DLP dummpy packets padding algorithms. We collected data with different sizes of group on each clustering algorithm and analyzed the effect of this variation on the dummy packets overhead in the network. Due to the fact that the capability of the system to handle the maximum number of clients on DLP and RO-DLP is different(250 clients for DLP and 1,000 clients for RO-DLP), therefore we would like to seperately evaluate between them. DLP with SG were performed with 1 (Regular DLP), 5, 10, and 25 groups on each mix node in the anonymity systems. Figure 5.1 shows the gradual decrease in dummy overhead for SG implemented on DLP when we increased number of groups. We observed an 4 fold decrease in the overhead for 25 groups in comparison to Regular DLP without implemented SG. In term of various clustering algorithms, we observed that sequential selection with standard deviation could beter perform when compared with sequential selection with 5 % and random selection with 15 % respectively.

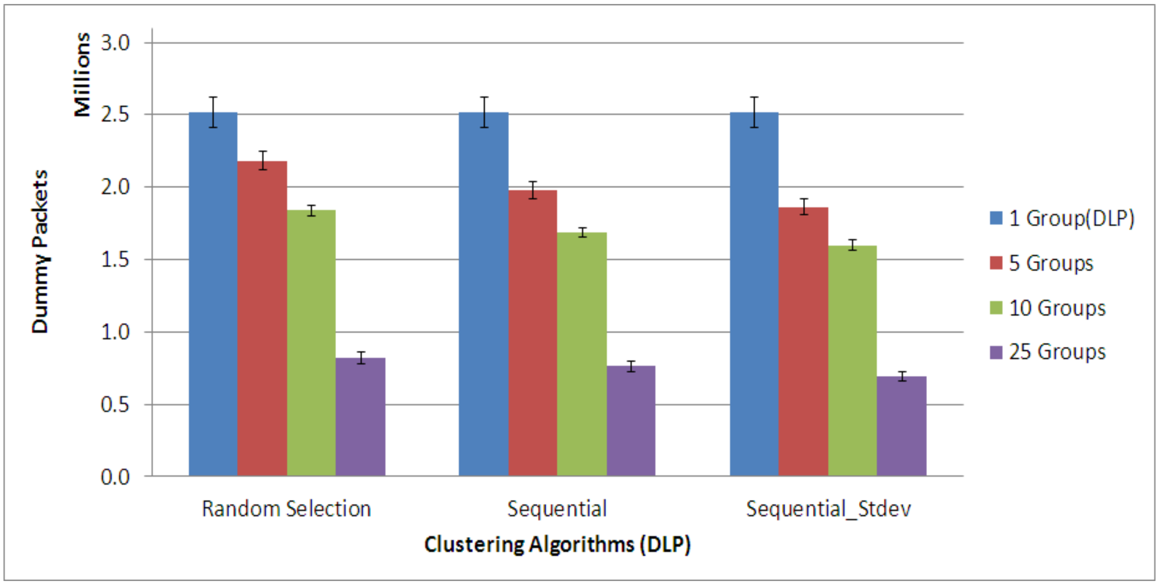


Figure 4.1. Different Number of Groups and Clustering Algorithms(DLP).

RO-DLP with SG could similary perform well as shown in Figure 5.2. With various group sizes (1, 5, 10, 20, 40, 100 groups) and clustering algorithms. We observed that sequential selection with standard deviation for 100 groups could decrease dummpy packets overhead with 3 fold decrease in comparision to RO-DLP. Furthermore, the results also show that sequential selection with standard deviation could best perform among clustering algorithms. It could substantially decrease 9 % in comparision to random selection. However, the result of DLP with SG and RO-DLP with SG are quite different from the previous SG's results proposed by vishal in term of the capablility of decreasing the overhead, which we discuss in section 4.2.

4.2 Evaluation and Discussion of Results of SG

According to the results proposed in the SG reseach, The author presented that SG implemented with DLP could exponentially reduce the overhead. On his research,

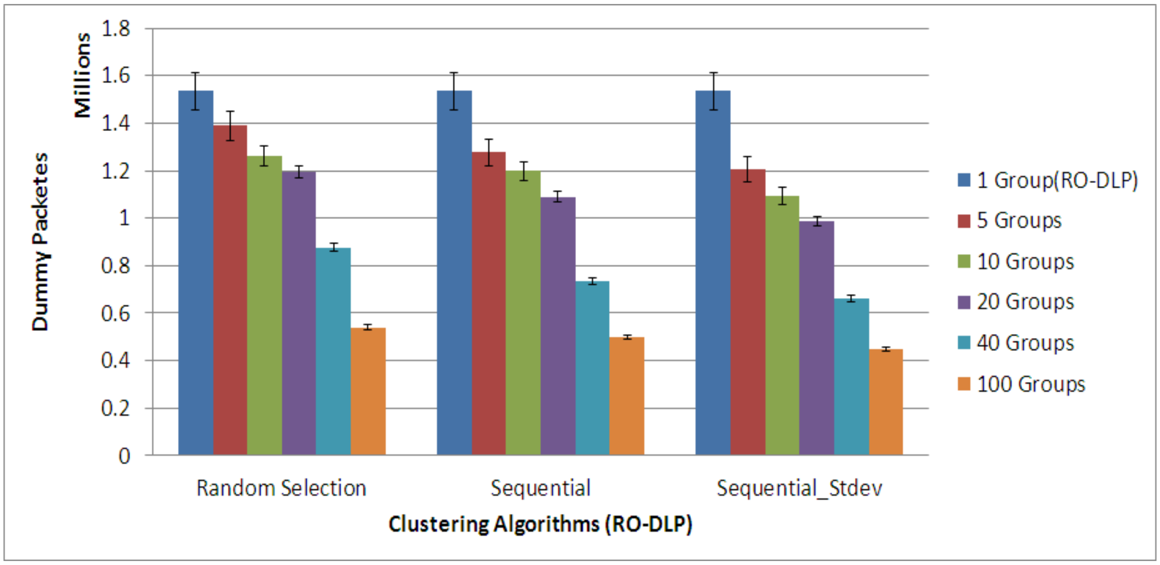


Figure 4.2. Different Number of Groups and Clustering Algorithms(RO-DLP).

he showed an 11 fold decrease in the overhead for 20 groups in comparision to DLP when implemented sequential selection with standard deviation.

In our research, the reduction on dummy packets overhead is less than what the SG's author could do. This is due to the fact that we performed experiments on the set of mix nodes including guard nodes, middle nodes, and exit nodes. Packets are sequentially forwarded on the paths along the way to the recipients via the set of mix nodes as showned in Figure 4.3. When packets pass through each mix node, the mix node will perform dummy padding algorithms to make a decision on how to add dummy packets according the pattern of incoming packets as described on the introduction part. The key point of this figure that could explain the lower capability of dummy packets decrease on our results is that SG counts the packets flowing on all circuits and use this information in order to divide them into groups with respect to the patterns of traffic (count of packets, standard deviation values, and random selection). With this, if SG could obtain the information that exactly reflected the real number of traffic from clients, SG would be able to perform very well. However,

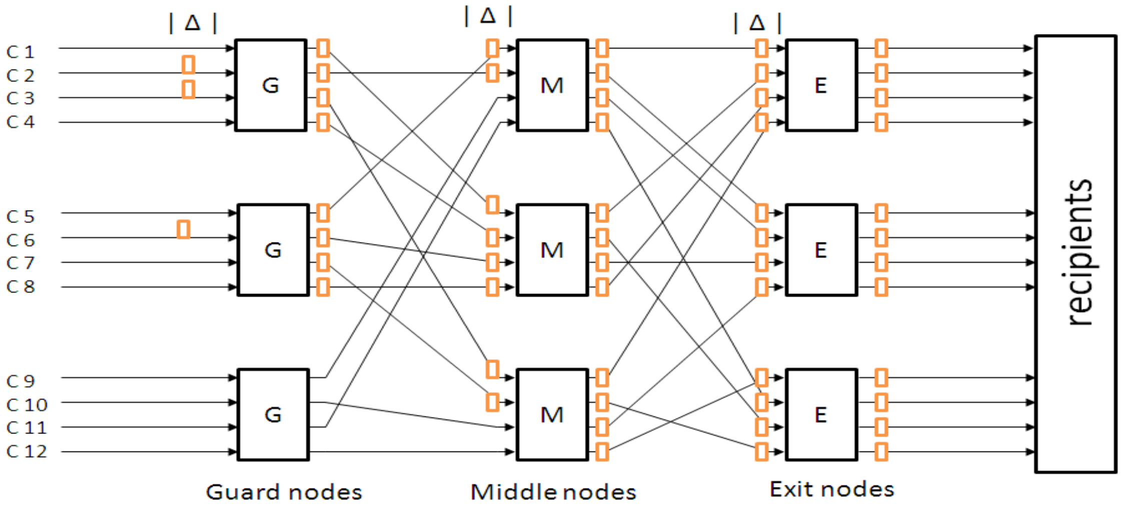


Figure 4.3. Packets flowing direction and Δ time implementation.

when we implemented SG on the real distributed network, the information used for making discusstion on grouping was distorted. As you can see from Figure 4.3-4.6, in the guard node, there is no problem on the packets count information becuase the incoming packets directly come from the clients and can reflex the exact real amount of network traffic on each circuit. The ratio of dummy packets decrease is similar to the previous work with roughly 11 times reduction. In the middle node, it receives the incoming packet containing packets sent from clients as well as dummy packets padded from the previous guard nodes. This circumstance makes the middle node performs SG by using the clients' traffic information which is not exactly reflected the network traffic of clients due to dummy packets padded on the previous nodes. This make the dummy packets reduction ratio decrease in comparision with guard nodes becuase SG could not correctly group clients into exactly-reflected traffic group. This situation also occurs in the exit node and becomes worst resulting in the number of dummy packets higher than incoming packets.

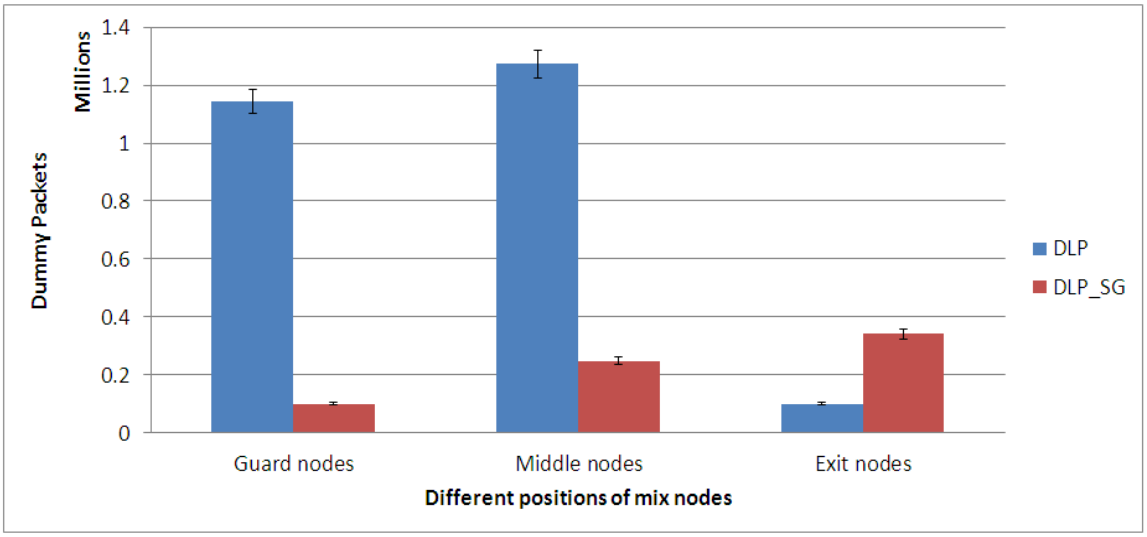


Figure 4.4. Comparision between Incoming and Dummy Packets on Different Mix Nodes Locations (DLP, DLP with SG).

Another factor worth considering is Δ time implementation. On both DLP and RO-DLP, we always determined time period(0.1 s) called Δ time. This implementation helps the mix nodes save dummy packets by delaying packets on Δ interval time before processing forwarding and dummy packets generating processes. In Figure 4.3 shows that the ratio between outgoing packets (reals packet and dummy packets) and incoming packets is highest at guard nodes and becomes lower at middle nodes and lowest at exit nodes respectively. This is due to the fact that the outgoing packets are generated on all circuits at the guard nodes and they will be the incoming packets of the middle nodes. Some patterns of packets will be the chunk of packets with in the same Δ time. With this, the dummy packets that have to pad at the middle nodes decrease. Finally, in the exit nodes sometimes do not have to add any dummy packets becuase all circuits on the same Δ time are fullfilled by the incoming packets generated from the middle nodes. This factor could help us explain the results on DLP and RO-DLP with SG.

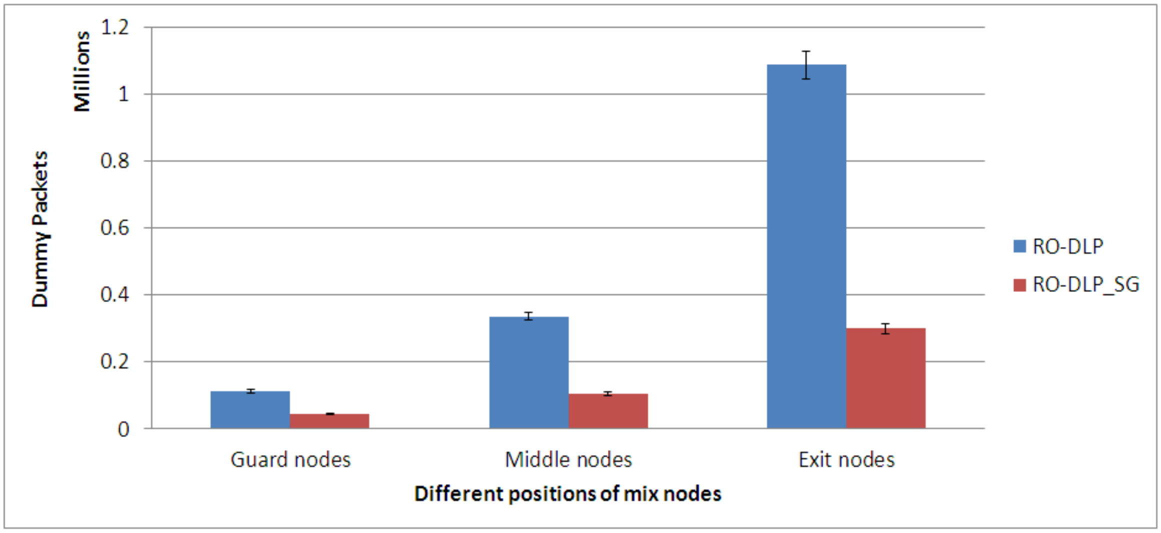


Figure 4.5. Comparision between Incoming and Dummy Packets on Different Mix Nodes Locations (RO-DLP, RO-DLP with SG).

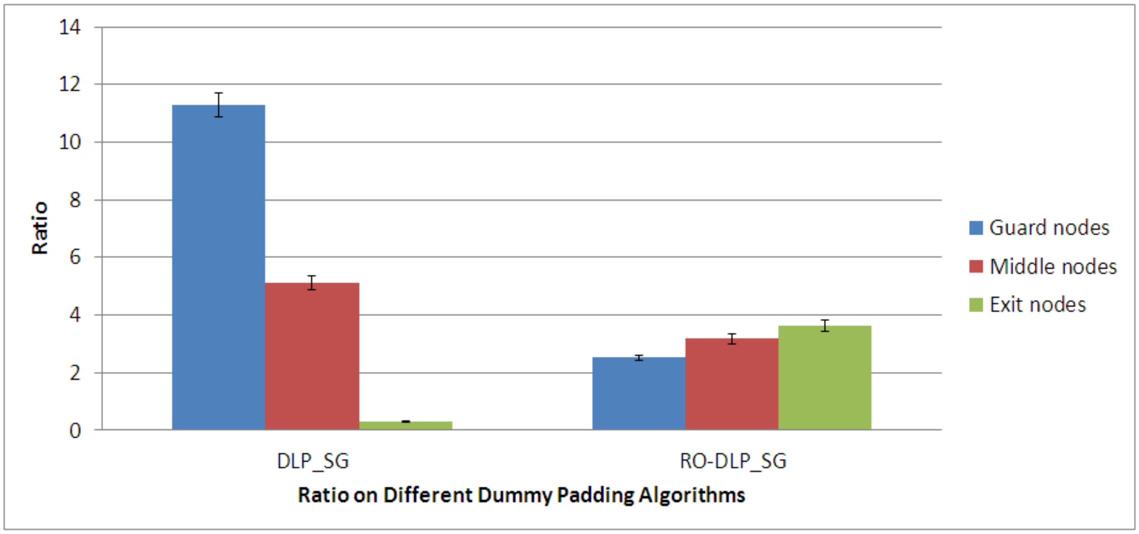


Figure 4.6. Ratio between Incoming Packets and Dummy Packets on Different Mix Nodes Locations.

In RO-DLP implemented with SG as shown in Figure 4.5-4.6, we observed different results in term of the capability of dummy packets decrease when compared with regular RO-DLP. The ratio of dummy packets reduction is also lower than DLP with SG. This is due to the fact that regular RO-DLP is the optimized vertion of DLP and already removed some dummy packets. When we implemented SG on RO-DLP, It could partially save the rest of dummy packets overhead and make the dummy packets reduction ratio lower than DLP with SG. Moreover, we observed the grudual increase of dummy packets reduction ratio on guard nodes, middle nodes and exit nodes sequentially is the reversed trend in comparison to DLP with SG. The intuition behind this results is that RO-DLP with SG partially pad the dummy packets including the links containing the real packets and other links containing the dummy packets. The outgoing packet from previous nodes could reflect the real amount of traffic with higher degree than DLP with SG. With this, there is a high chance that the expected pattern of traffic such as higher volume or bursty traffic will be grouped into the expected groups. Moreover, RO-DLP with SG could make use of Δ time implementation on the deeper levels of mix nodes which makes the middle nodes and exit nodes perform better than the guard nodes.

4.3 Partial Implementation of SG

In the previous section, it explained about the benefits of a Δ time implementation and the problems of a distorted grouping mechanism on deeper levels of mix nodes. We would like to investigate the partial implementation of SG on the mix nodes to investigate the results by changing the positions and number of mix nodes implemented by SG in order to try to mitigate the problems of the distorted grouping and gaining the benefit of Δ time. For DLP with SG, we implemented SG only on guard nodes and applying the regular DLP on the deeper levels. For RO-DLP, we

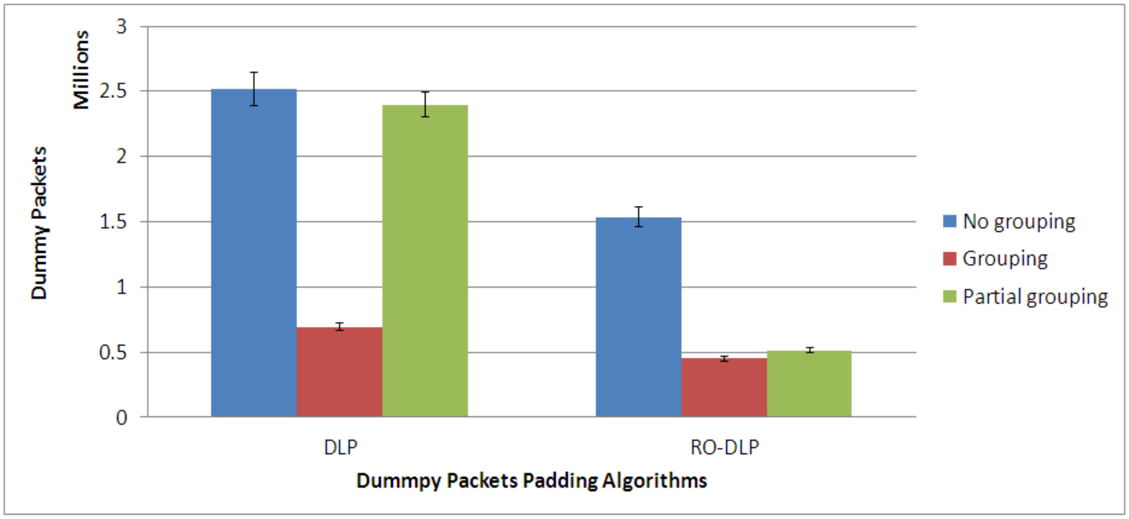


Figure 4.7. Partial Implementation of SG.

partially implement SG on the middle and exit nodes due to the higher availability of gaining the benefit of SG on RO-DLP and implement regular RO-DLP on the guard node. We observed that there is no significant decrease on dummy packets on partial SG on DLP as shown in Figure 4.7. Even if we could fully gain the benefit of SG on guard nodes, they still generate outgoing packets becoming the incoming packets for the middle nodes. The middle nodes then perform DLP and fully generate outgoing packets. We could gain the benefit of Δ time only on the exit nodes which is not enough to make overall overhead reduction significant when compared with DLP with SG. Furthermore, partial implementation of SG on RO-DLP could perform very well with nearly equal reduction ratio when compared with full implementation of SG.

4.4 Performance Measurements of SG

To study the effect of implementing SG on DLP and RO-DLP in term of performance, we studied the capability of increasing maximum number of clients that can

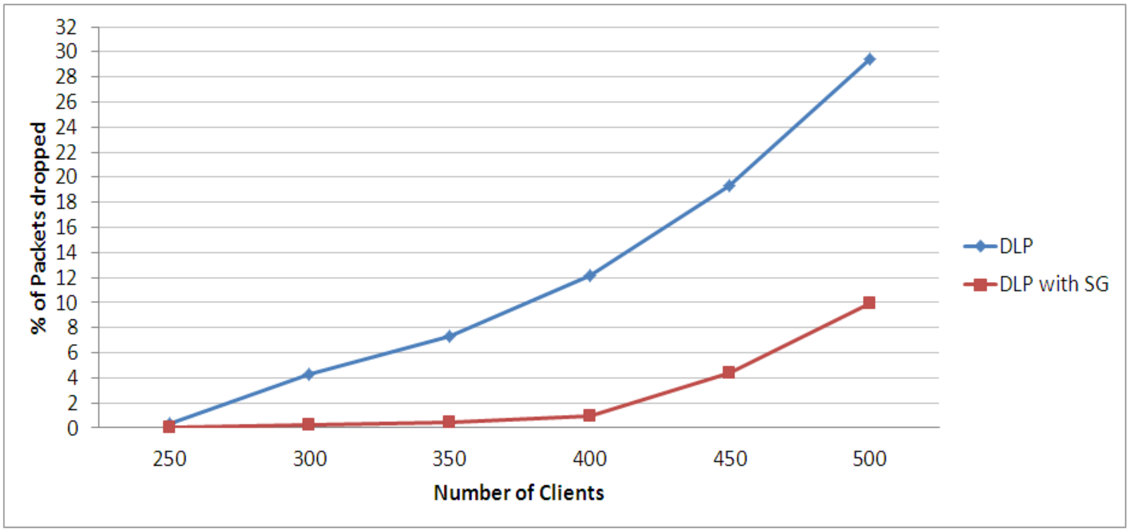


Figure 4.8. Percentage of Packets Dropped on Diffent Number of Clients(DLP).

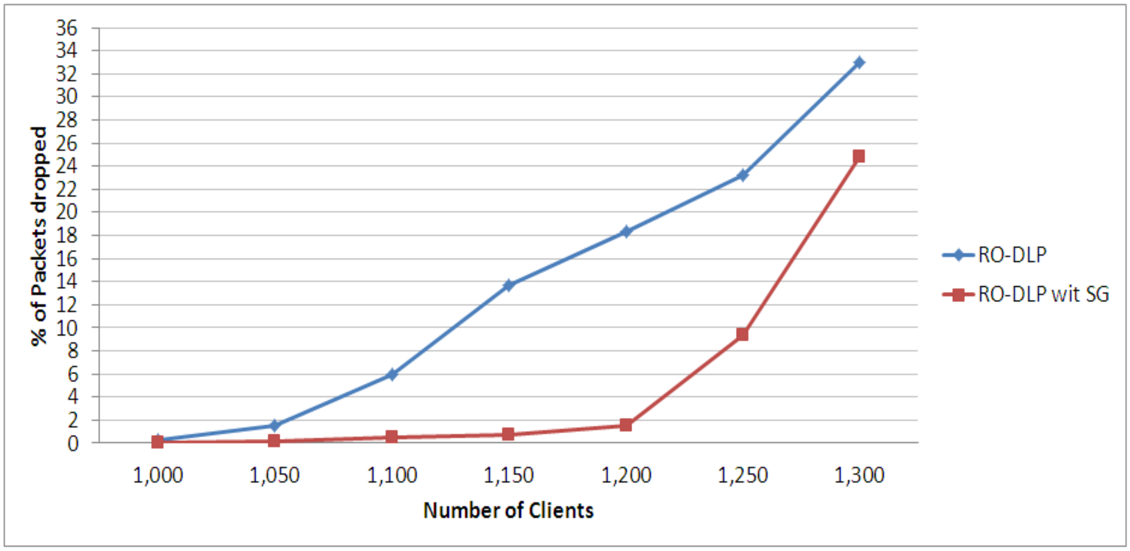


Figure 4.9. Percentage of Packets Dropped on Diffent Number of Clients(RO-DLP).

participate in the anonymity system. We used the average dropping packets percent on the whole systems calculated from the number of packets droped and the total number of network traffic. With this, the acceptable percentage of packets dropped is indicated to be less than 1 %. Our results shows that the maximum number of clients of DLP with SG is 250 clients with 0.332 % of packets dropped and could increase to 400 clients with 0.939 % of packets dropped when we implemented SG as shown in Figure 4.8. Besides, For RO-DLP, the results shows that the maximum number of clients of RO-DLP is 1,000 clients with 0.245 % of packets dropped which is higher than DLP and increases to 1,150 clients with 0.739 % of packets dropped when implemented with SG as shown in Figure 4.9.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we developed a Tor-like experimental evaluation platform called SubRosa 2 for studying and investigating the DLP and RO-DLP in term of dummy packets overheads and compared these to a recently-propsed technique called SG. The main purpose of our thesis is to implement these schemes in the real distributed networks containing the chain of mix nodes. We conducted several experimental simulations and showed that SG could substantially reduce the dummy packets overhead with lower rate of reduction in comparision to SG's author proposed on both DLP and RO-DLP due to the distorted grouping occuring in the deeper levels of mix nodes and Δ implementation effect. In term of clustering algorithms, the capability of reducing the dummy packets overheads is consistent with SG's author proposed. Sequential selection with standard deviation could provide better results in comparison to sequential selection and random selection respectively. Moreover, Our results showed that implementing SG on DLP and RO-DLP could increase the maximum number of clients participated in the whole systems with acceptable packets dropped rate. Finally, our work also need to be extensively researched in term of anonymity measurement in the future.

REFERENCES

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

[2] Project: AN.ON. http://anon.inf.tu-dresden.de/index.en.html

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

[4] W. Wang, M. Motani, and V. Srinivasan. Dependent link padding algorithms for low latency anonymity systems. In Proceedings of the 15th ACM Conference on Computer and Communications Security, 2008.

[5] C. Diaz, J. S. Murdoch, C.Troncoso. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS 2010), July 2010.

[6] Tor. http://www.torproject.org/about/overview.html

[7] H. Daginawala and M. Wright. Studying Timing Analysis on the Internet with SubRosa. In Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS 2008), July 2008.

[8] J. Feigenbaum, A. Johnson, P. Syverson. Preventing active timing attacks in low-latency anonymous communication (extended abstract). In Proceedings of PETS, 2010.

[9] P. Venkitasubramaniam, T. He, and L. Tong, Relay secrecy in wireless networks with eavesdroppers," in Proceedings of Allerton Conference on Communication, Control and Computing, 2006.

[10] A. Pfitzmann, B. Pfitzmann, and M. Waidner, ISDN-Mixes: Untraceable communication with very small bandwidth overhead, in Proceedings of GI/ITG-Conference Communication in Distributed Systems, 1991.

[11] X. Fu, B. Graham, R. Bettati, and W. Zhao, On effectiveness of link padding for statistical traffic analysis attacks, in Proceedings of IEEE ICDCS, 2003.

[12] P. Venkitasubramaniam and L. Tong : Anonymous networking with minimum latency in multihop networks. In: Proceeding of the IEEE Symposium on Security and Privacy, pp. 18-32. IEEE Computer Society, Los Alamitos (2008)

[13] NS2: AN.ON. http://www.isi.edu/nsam/ns/

[14] V. Gupta and M. Wright : Selective Grouping Algorithm For Low Latency Anonymous Systems, University of Texas at Arlington (2012).

[15] Invisible Internet Project: I2P http://www.i2p2.de/

[16] O. Berthold, H. Federrath, and S. Kopsell, Web mixes: a system for anoymous and unobservable internet access in Proceeding of International workshop on Designing privacy enchacing technologies: design issues in anonymity and unobservability, 2001, pp. 115-129

[17] P. Venkitasubramaniam and L. Tong : Anonymous networking with minimum latency in multihop networks. in Proceeding of the IEEE Symposium on Security and Privacy, pp. 18-32. (2008)

[18] J. Boyan, "The anonyizer - protecting user privacy on the web", 1997.

[19] P. Maymounkov and D. Mazieres, Kademlia : A peer-to-peer information system based on the xor metric, in Revised Papers from the First International Workshop on Peer-to-Peer Systems, 2002, pp. 53-65.

[20] J.B. MacQueen, Some methods for classification and anlysis of multivariate observation, in Proceedings of the 5th Berkeley Symposium on Mathematical Statistic and Probability, 1967, pp. 281-297

[21] K. Bauer, M. Sherr, D. McCoy and D. Grunwald, ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation , in Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test CSET 2011.

[22] Emulab. http://www.emulab.net.

[23] DETERlab testbed. http://www.isi.edu/deter.

[24] M. AlSabah, K. Bauer, T. Elahi and I Goldberg, The Path Less Travelled: Overcoming Tors Bottlenecks with Multipaths , University of Waterloo (2012)

[25] DETER Project. http://deter-project.org.

BIOGRAPHICAL STATEMENT

Payap Sirinam was born in Chiang Mai, Thailand, in 1985. He received his B.S. degree from Royal Thai Air Force Academy, Thailand, in 2008, in Computer Science. From 2008 to 2010, he was with the department of Computer Science , Royal Thai Air Force Academy as a lecturer. In 2011, he joined information security lab, University of Texas at Arlington. His current research interest is in the area of anonymity systems.