SELECTIVE GROUPING ALGORITHM FOR LOW LATENCY ANONYMOUS SYSTEMS

by

VISHAL GUPTA

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 ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2012

To my parents for their prayers, blessings and for their consistent support

ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance and support of my supervisor Dr Matthew Wright. I would also like to thank Dr. Vassilis Athitsos and Dr. Gergely Zaruba for taking time to serve on my thesis committee. I would also like to thank my student colleagues in the iSec lab for always being ready to help. Finally, I thank my parents for their unconditional love and support all this time.

April 30, 2012

ABSTRACT

SELECTIVE GROUPING ALGORITHM FOR LOW LATENCY ANONYMOUS SYSTEMS

VISHAL GUPTA, M.S.

The University of Texas at Arlington, 2012

Supervising Professor: Dr. Matthew Wright

Low latency anonymous communications are prone to timing analysis attacks. It is a technique by which the adversary can de-anonymize the user by correlating packet timing patterns. A recent proposal to stop these attacks is called Dependent Link padding. However, it creates high dummy packets overhead in the network. In this work we propose selective grouping, a padding scheme that protects users in an anonymity system from those attacks with minimal overhead. The aim is to decrease overhead by dividing users in different groups while maintaining good anonymity. The key idea of our approach is to group clients with similar timing patterns together by providing a strict delay bound. We ran simulation experiments to test the effectiveness of these techniques and to measure the amount of extra network congestion. We have also statistically analyzed bursty traffic in the network by using the mean and standard deviation of inter packet delays over a fixed duration. The result of bursty traffic analysis added one more dimension to the count of packets for grouping clients efficiently. To analyze anonymity, we ran a statistical disclosure attack against our selective grouping defense. We performed extensive sets of experiments to find a threshold value at which selective grouping achieves good profiling without adding excess dummy packets. We show that selective grouping is very effective at resisting timing analysis attacks and are still able to provide good anonymity with minimal overhead added to the network.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

Many anonymity systems such as Tor [1], and AN.ON [14] have been proposed for real-time communication processes such as web browsing, file sharing, and online chatting. The goal of these systems is to mask the true identity of the sender and the receiver, and make them practically unlinkable. In contrast with non-interactive communication such as email, an anonymous system cannot implement defense techniques like delaying messages, batch processing or reordering of messages precisely because of its interactive nature. This is because all interactive applications must meet strict latency requirements and so delaying packets is not a viable option for achieving anonymity. In addition, packets cannot be routed through different paths each time because most applications require a Transport Control Protocol (TCP) connection. These practical limitations make the communication system vulnerable to timing attacks, in which the attacker observes the timestamps of the packets entering and exiting the mix circuit, and correlates who is communicating with whom [2]. In a timing attack, the attacker observes the traffic pattern without controlling all the mixes in the circuit. The attacker captures packet timings for a small fraction of the network and then performs statistical correlation to link an initiator with a responder. To protect the network against these attacks, some well known defenses are independent link padding(ILP) [3], dependent link padding (DLP) [4], and reduced overhead dependent link padding (RO-DLP) [5].

ILP is the most basic padding algorithm in which the server adds packets in constant intervals to the incoming streams and makes the output streams identical for all clients. The biggest drawback of ILP is that the output pattern is always pre-determined, regardless of input stream.This leads to high overhead from dummy packets and substantial delay in the network. To overcome this problem, Wang et al. [4] and Venkitasubramaniam et al. [6] proposed the DLP algorithm, which provides the same anonymity level as of ILP with the use of fewer dummy packets. It dynamically adapts the rate of adding dummy packets with the change of incoming data, which decreases overhead in the network. However, this decrease is still not substantial enough to see implementation in the real world.

1.1 Contribution

To decrease overhead substantially, we propose *selective grouping*, an extension of DLP that protects anonymous systems from timing attacks with minimal use of dummy packets. The basic approach of this algorithm is to divide users into different groups with similar packet sending rates while maintaining reasonable anonymity. For grouping, we analyzed different algorithms such as *k-means* clustering, density based clustering, and sequential selection. To measure the systems anonymity in the presence of selective grouping, we observed the effectiveness of the *statistical disclosure attack* (SDA) in our simulation[7]. The SDA is a type of intersection attack in which the attacker tries to find all the recipients of a targeted user (Alice) in the anonymous network. The attacker under the global passive adversary model records all incoming and outgoing messages over a period of time. By taking the difference of the observations when client is active and when client is inactive, the attacker gets the client contribution towards its recipients[7].

In Chapter 2, we describe the background context of our work, which includes detailed information of low-latency anonymity systems, timing analysis attacks against them, and defenses against these types of attacks. We also describe clustering algorithms that could be used in our selective grouping algorithm. In Chapter 3, we outline the system and attack models that we use to explore the defenses against timing analysis attacks. The most important contribution of our research is described in Chapter 4, which includes details of our selective grouping algorithm for low-latency anonymous systems. The main objective of our work is to protect anonymity by making timing attacks impractical for an attacker, while maintaining a reasonable amount of dummy overhead in the network. For validation, we conducted several experiments using the *UMaas Network Trace* and explained the outcome of our results in Chapter 5. In Chapter 6, we concluded our idea of selective grouping by considering all outcomes of experimental simulations.

CHAPTER 2

BACKGROUND

In this chapter, we describe low-latency anonymity systems, timing analysis attacks, and the defenses used against them. We also describe the clustering algorithm which is used to develop our selective grouping algorithm.

2.1 Low-Latency Anonymity Systems

Low latency anonymity systems are designed around the idea of mixes, in which users connect to the Internet anonymously via a chain of proxies to hide their identity from potential eavesdroppers. The most common anonymity systems of this type are Tor [1], I2P [8], Web MIXes [9], and Anonymizer [10].

Tor is a commonly used anonymity system operated by volunteers from all around the world and the machines that run Tor are called Onion Routers [1]. Journalists use Tor for safe communication, nonprofit and business organizations use it to allow their workers to conceal their identity, and even individuals use Tor for socially sensitive communication [1] . Tor is composed of a client software and a network of servers. It provides online anonymity by hiding information about user's locations as well as other identifying factors such as IP address.

I2P is a similar anonymity system also with the aim to provide secure and anonymous communication. The network of I2P consists of routers that work as unidirectional inbound and outbound virtual paths, which is called tunnel routing.

It uses the Kademlia algorithm [11] to distribute routing and contact information securely. Unlike other anonymity systems, it provides anonymity to both sender and receiver in P2P communication. For example, it can be used to host a website and also to send HTTP requests to that website. The Anonymizer is another type of anonymity system which keeps users untraceable on the Internet. It uses a trusted proxy server and encryption techniques to hide a user's identity from rest of the world.

The Web MIXes [9] is a system which provides anonymity for real time Internet communication. It works on a modified mix concept which adds dummy packets when an active client become idle. It also uses a ticketing mechanism for user authentication to avoid flooding attacks and provides feedback to users about their current level of protection. The system consists of JAP (Java Anon Proxy) on the client-side and MIXes and cache-proxy on the server-side. The users connect to MIXes through JAP anonymous tunnel (MIX-cascade) to provide anonymous communication [9].

2.2 Timing Analysis Attacks

A timing analysis attack [2] is a technique in which the adversary tries to de-anonymize the user by collecting relevant packet information. The adversary observes incoming and outgoing packet timestamps and correlates them to find the client and server's identity. Sometimes due to network jitter packets are drop in between, which creates a problem for the adversary to map the client's identity. To improve this concept, Levine et al. [2] proposed a cross correlation (CC) technique which neglects this error rate and matches incoming and outgoing streams. Timing analysis attacks are broadly classified as active and passive timing attacks, according to the attack model's capability.

In a passive timing attack, the adversary statistically correlates the incoming and outgoing packets with respect to the mixes over a period of time on the basis of inter-packet delay(IPD). This is the time difference between two consecutive packets that generates a unique timestamp pattern for every client.

In active timing attacks, the attacker does not merely observe IPDs, but rather tries to manipulate the incoming traffic by inserting timing patterns into the traffic as it passes through routers under his control known as stepping stones [12]. This includes introducing delays or injecting/dropping packets, which creates a specific timestamp pattern that can be observed in the output stream. This technique is called watermarking, which helps in correlating incoming and outgoing streams. However, watermarking attacks can be prevented by tracing back through stepping stones and replacing the distorting watermarks with the original watermarks [4].

2.3 Timing Analysis Defenses

Timing analysis defenses are proposed to protect anonymous systems such as Tor [1], Web-Mixes [9], ISDN-Mixes [13] and Pipenet [14] from timing attacks. These systems send messages at a constant rate which makes all outgoing streams identical. However, in the case of jitter or a sudden drop of packets, the system becomes vulnerable to correlation of input and output streams. Some notable defenses to protect these systems from timing attacks are : defensive dropping, independent link padding (ILP), dependent link padding (DLP) and reduced overhead dependent link adding (RO-DLP).

To overcome timing analysis attacks, Levine et al. [2] proposed a defensive dropping defense, in which, clients generate dummy packets in addition to their real packets. The mixes are instructed to drop these dummy packets whenever required to make the output timing pattern exactly similar across all clients. With respect to scalability, this algorithm can be implemented for multiple mixes, then collectively drop a set of packets.

ILP adds dummy packets according to a predefined time schedule into the incoming data stream to create the same output pattern for all clients [3]. ILP is performed at a constant rate *i.e.* all packets are sent at a fixed interval of time regardless of any delay [13]. This method can result in a long delay between packets until they reach their scheduled destination. If some of the clients connected to the network suddenly start sending packets at a higher rate, then the constant padding algorithm drops most of the packets because of the constant time interval. Also, if the independent link padding algorithm follows a strict bound delay, then most of the real packets in the network will be dropped too. Thus, two major drawbacks of this method are that the output pattern is always pre-determined regardless of input stream, and this algorithm uses an enormous amount of bandwidth for dummy packets. This pre-determined output pattern increases the count of dummy packets even when no user packets exist in flow, as the anonymous system continues to add dummy packets to maintain the output pattern. Moreover, it is shown that links padded by such a constant rate schedule are still susceptible to traffic analysis as the variance of packet timing may be correlated to the system loading [15].

Another method that is used apart from ILP involves random padding of dummy packets. In this algorithm, the server adds dummy packets according to the Poisson process [3], with the limitation that the server needs to know the average sending rate to work efficiently [4]. It is also difficult to vary the padding rate with respect to time because doing so can leak information about the clients to potential adversaries.

The DLP [4, 6] algorithm provides anonymity to the users from timing analysis attacks with the use of a strict delay bound. DLP claims to provide equivalent anonymity in comparison to ILP algorithm while using fewer dummy packets. It dynamically adapts to the change in traffic rates of incoming data, offering a reduced packet drop rate. Moreover, there is a direct relationship between anonymity and sending rates of incoming packets with different arrival times. When the incoming flows are in the Poisson distribution, the minimum sending rate is $O(logm)$ to provide full anonymity for m users flows. Their results for Pareto traffic distribution show that the rate of cover traffic converges to a constant value when the number of users tends to infinity. These findings were based on the real Internet traces to demonstrate the effectiveness of DLP.

DLP also uses a heuristic dropping algorithm to control the sending rate when the packet count increases drastically within the network. To this end, a token utility (u) is defined as $u = d/|F|$, where d is the number of non-dummy packets sent by the token and $|F|$ is the size of the incoming flow set [4]. A token is used only if its utility is larger than the given threshold U. If the token is not used due to low activity, then all packets scheduled at this token will be dropped if its delay bound is not met.

The ILP sends output packets in fixed intervals and therefore is not flexible to the real time dynamic change of the incoming traffic rate; DLP addresses this problem. In the DLP simulations, Wang et al. [4] assumed that the attackers are capable of monitoring all incoming and outgoing packets in the network. However, the attacker cannot correlate the timing patterns because the packets are encrypted and have the same timestamp. To make their simulation more realistic, the authors also performed matching and watermarking attacks [16], which further validated the findings from the previous simulation.

The RO-DLP is an extension of DLP that further reduces the dummy packet overhead to an acceptable level in an anonymous network. The link encryption is a feature of most anonymous networks deployed for the purpose of hiding correspondence between routers. RO-DLP successfully protects these links between the multiplex circuits from timing analysis, used to substantially reduce the number of dummy packets. This method is preferred over DLP because it can provide the same level of security against external adversaries. Thus RO-DLP performs significantly better in comparison to DLP while reducing the overhead imposed by dummy traffic.

2.4 Statistical Disclosure Attack

The SDA is a well known method in which an attacker is able to know who is talking to whom in anonymous mix network. In SDA, the attacker focuses on finding probable recipients for a particular user. In this approach an eavesdropper analyzes incoming and outgoing packets for a number of rounds. Rounds are classified as the count of messages sent by Alice and by the background clients. In every round the attacker records the count packets in an observation vector. The attacker sums up all observation vectors and subtracts Alice's contribution from that. Finally, in the resultant vector, the highest probability values correspond to the most likely of Alice recipients[7].

2.5 Clustering Algorithms

Clustering algorithms are used to group a data set of size n, with d dimension. *K-means* clustering is the most common algorithm used to perform this operation in which n clients with d dimensional data are partitioned into k clusters. This includes the Euclidian *k-medians* in which clients near to their medians are considered to be in the same cluster. It should be noted that optimal *k-means* clustering is considered an NP hard problem. One of the most famous heuristic approaches for solving k-means clustering is to perform many iterations to get better clusters [17, 18]. This approach was first proposed by Lloyd and is called *Lloyd's algorithm*.

Lloyd's algorithm doesn't address the initial selection of the k centers. *Forgy* and *Random* partitions are commonly used choices for the initial selection of these k observations. The forgy algorithm randomly chooses k observations and uses them as the initial means for input data. In contrast, random partitioning divides all n clients randomly into k clusters and then rearranges them on the basis of the nearest mean. Lloyd's algorithm is used mostly in statistical analysis because of its simplicity and flexibility. However, it is quite slow because of the high number of iterations needed to compute nearest neighbors.

CHAPTER 3

MODEL

In this chapter, we describe system and attack models to study timing attacks in low latency anonymous communication.

3.1 System Model

Tor [1], I2P [8], Web MIXes [9], and Anonymizer [10] are the most commonly used anonymity systems that preserve identity of the sender and the receiver. To analyze timing analysis attacks we consider Tor like network with the use of selective grouping algorithm. However, selective grouping works effectively for all types of low-latency anonymity systems. Tor system consists of mixes which adds dummy packets with respect to the defense used in network. In our simulations, we consider only unidirectional traffic from user to responder but it can also work for duplex communication.

3.2 Attack Model

For passive timing attacks we considered *eavesdropper and compromised mix* attack model. In eavesdropper attack model the attacker monitors incoming and outgoing packets from entry and exit node as shown in Figure 3.1. In passive timing attacks, the adversary analyze the IPD between two packets of incoming and outgoing stream and de-anonymize the identity of users. In active timing attacks, the attacker manipulates the incoming traffic by adding watermarking bits and observes them in outgoing stream. The other type of attack model is *compromised mix* model.

In this, attacker compromise first and last mixes in the circuit and manipulate them to perform timing analysis attack as shown in Figure 3.2. Selective Grouping works effectively for passive attacks and modify timing pattern of outgoing streams, so that the attacker cannot able to correlate incoming and outgoing timing patterns.

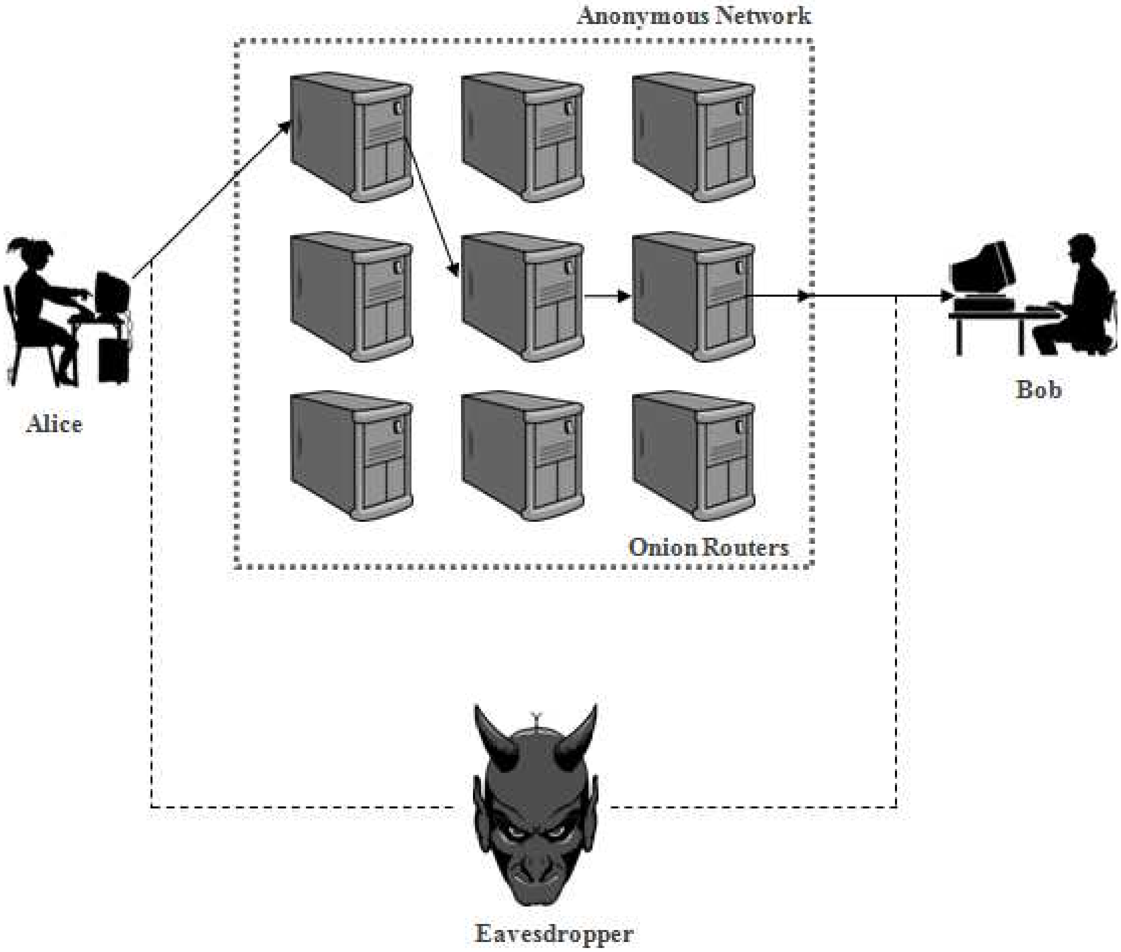

Figure 3.1. Eavesdropper Model.

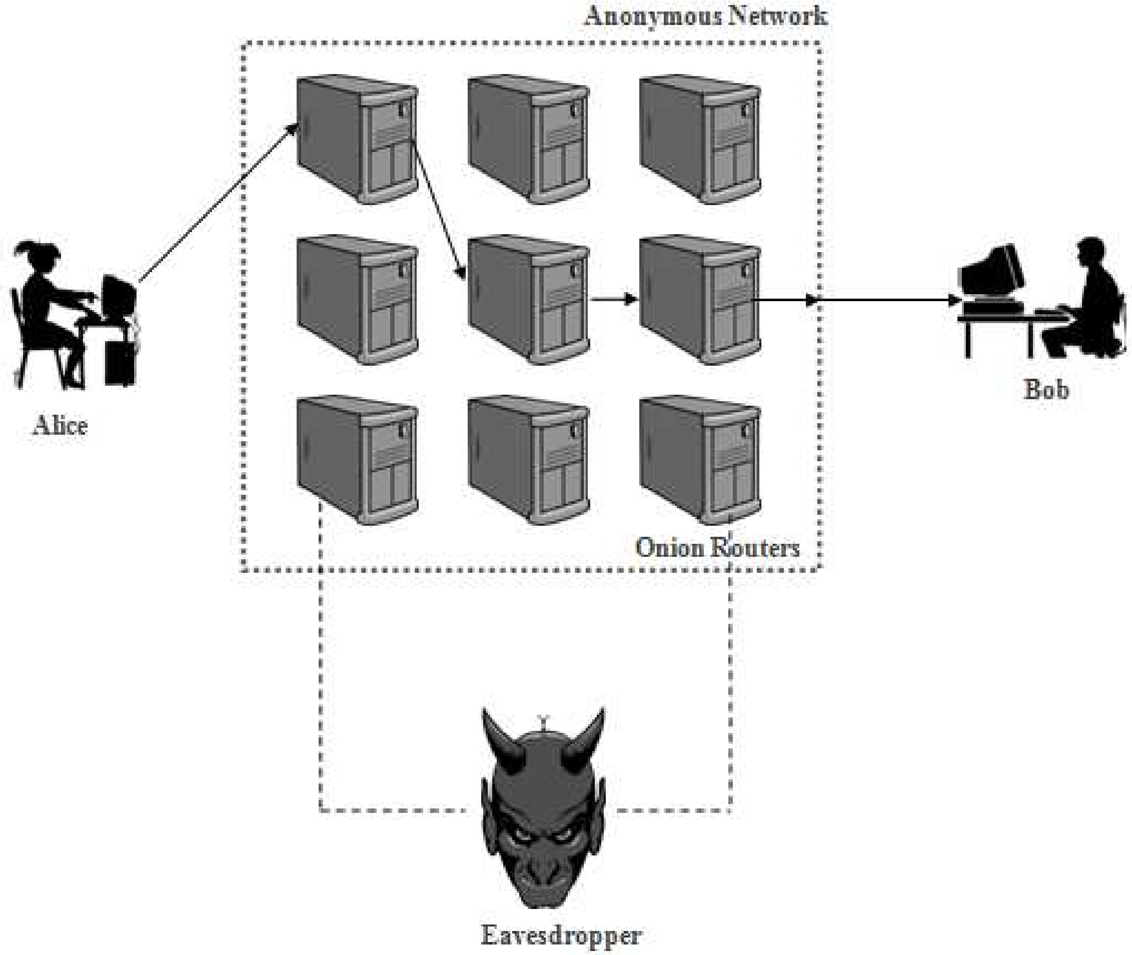

Figure 3.2. Compromise mix Model.

CHAPTER 4

SELECTIVE GROUPING

In their original DLP paper [4, 6], Wang et al. and Venkitasubramaniam et al. proposed an algorithm that generates the same output stream of packets for all clients by achieving full anonymity. However, in the real world, the sending rate of clients varies drastically due to the use of different communication protocols such as VoIP or P2P. File sharing users have a very high sending rate in comparison to the users who use systems like Tor for VoIP calls. This variation in sending rate results in addition of more dummy packets to flows with a lower sending rate to make them in-line with other high rate flows.

To overcome this problem we propose a Selective Grouping padding algorithm which protects anonymous system from timing attacks with minimal dummy overhead. Selective grouping is an extension of the Dependent Link padding algorithm with the use of clustering algorithms. The goal of the Selective Grouping is to decrease overhead by splitting users in different groups while maintaining good anonymity. It involves grouping clients with similar timing patterns by using a delay bound parameter. We analyzed different groping methods such as 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, we have mostly used Sequential clustering and k-means clustering algorithms in selective grouping.

In sequential clustering we count packets of each user in specific time interval and then divide them sequentially after sorting them with respect to the packet count. This packet count value directly relates to the length of circuit life time (cycle). Selective grouping performs analysis on each life cycle and its duration varies with every simulation. Once the cycle duration becomes fixed for a simulation, SG counts the client's real packets in the current cycle. Before completion of a cycle it sorts the clients on the basis of packet count as the hash key and splits them sequentially. In the next cycle these clients follow previous cycle's grouping and send packets by considering only clients who exist in their respective group. To maintain minimum level of anonymity in every group we have introduced a condition of minimum group size. If the group size becomes very low then the eavesdropper can easily de-anonymize identity of a user but if group size becomes too high then it increases the dummy overhead in network. We also observed that approximately 50% of the total clients joins network newly after each cycle. So to make all clusters almost equal in size and to maintain anonymity we split new clients randomly in minimum group size. Each set is then processed with the DLP algorithm to make the output pattern exactly similar with respect to each client.

We also statistically analyzed bursty traffic in the network by using mean and standard deviation of inter-packet delays (IPD). If one client sends bursty traffic then it can affect all other clients of that group which can lead to more overhead. To overcome that problem we proposed to use standard deviation of IPDs. If standard deviation value is high then it means high bursty traffic. So, in sequential selection with standard deviation, we first performed the complete sequential selection process and then filtered all bursty traffic clients into one new group, which decreases overhead substantially.

In k-means clustering all n clients with d dimension are partitioned in k cluster, in which each client belongs to the cluster of nearest mean [18]. For our simulations we used 2 dimensions, which are packet count and standard deviation of each client. In k-means, we also consider minimum group size and divided larger groups in smaller size.

We consider an anonymous server which has n clients with different packet transmission rates depicted in Figure 4.1. For maintaining full anonymity, the anonymous server requires adding dummy packets to make all output pattern the same. However, due to the variation in sending rates, DLP requires a lot of dummy packets to make all output pattern same. Figure 4.2 shows that, if we split clients in three groups with different sending rate then we can drastically decrease dummy packets in the network.

In general, let us consider n clients with sending rates r_n and m clients with sending rates r_m. If $(r_n > r_m)$, then according to DLP algorithm, the minimum dummy packets required to make all clients output pattern same is $(r_n - r_m) * m$. However, with a little compromise on anonymity and with the use of selective grouping, all these dummy packets can be removed from the network. Selective grouping clustering algorithm splits $(n+m)$ clients into two different groups of n and m clients according to their sending rate and then adds dummy packets if required in their respective groups.

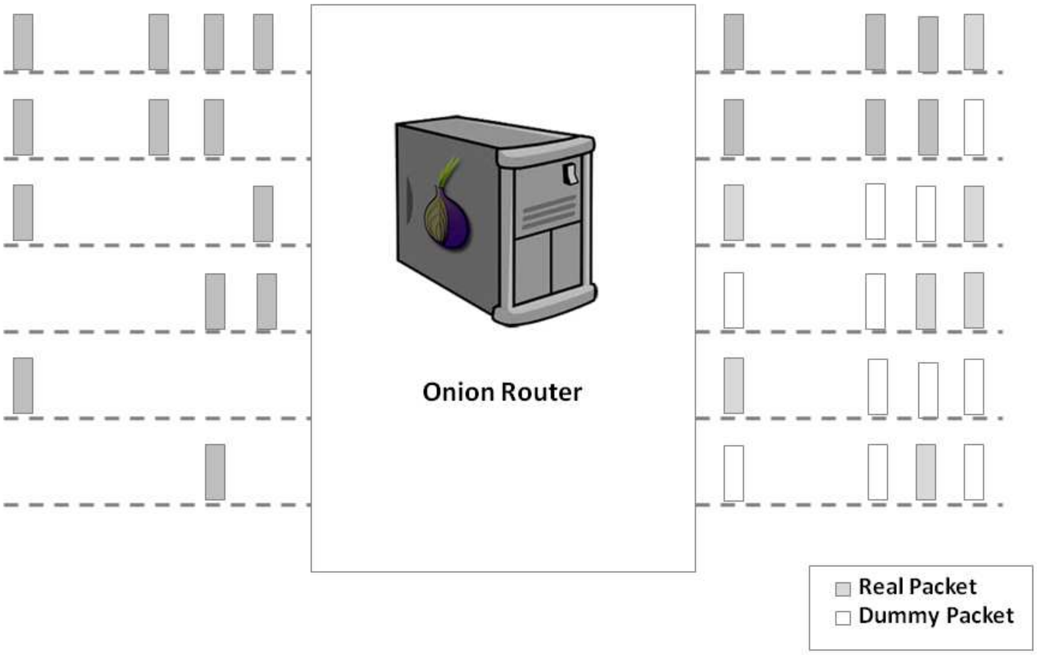

Figure 4.1. DLP algorithm - Input and output stream.

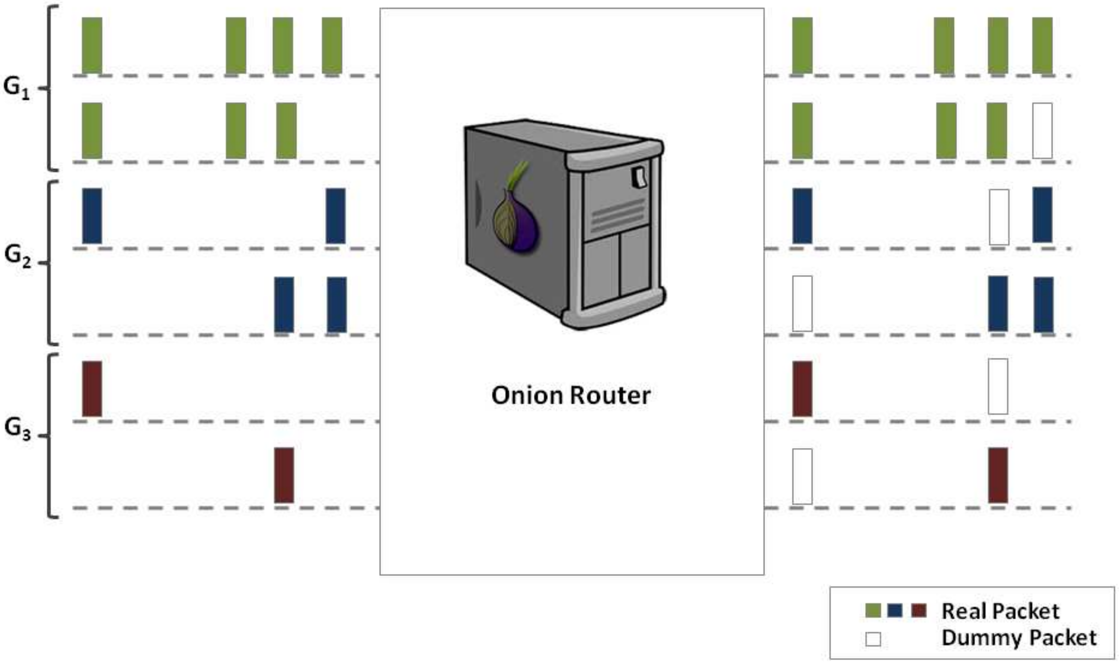

Figure 4.2. Selective Grouping - Input and output stream.

CHAPTER 5

RESULT AND DISCUSSION

In this chapter we describe the pre-processing of network traces for our experiments and to calculate dummy packet count in the network. We performed experiments using the mean algorithm to cluster users in separate groups with respect to their packet sending rate. We are also planning to run simulation using K-means anonymity algorithm and density clustering in future.

5.1 Variation of Number of Groups

In this simulation, we varied group size by modifying number of clusters and analyzed the effect of this variation on the dummy overhead in the network. We used selective grouping with standard deviation for grouping clients, and window lengths of 30 seconds and 60 seconds for each round. We retained 5000 active circuits at any time in the network. Figure 5.1 shows exponential decrease in dummy overhead with decrease in group size. For both 30 second and 60 second cycles, we observed an 11 fold decrease in the overhead for 20 groups (250 clients/group) in comparison to DLP. For 10 groups (500 clients/group), which is the standard number we have used in most of our other simulations, it shows a 6 fold decrease in overhead. However, this increase in groups results in little anonymity loss, which we discuss in section 5.5. In the all subsequent graphs, the real packet transmission rate in the network is 128 packets/second and the error bar represents the 95% confidence interval.

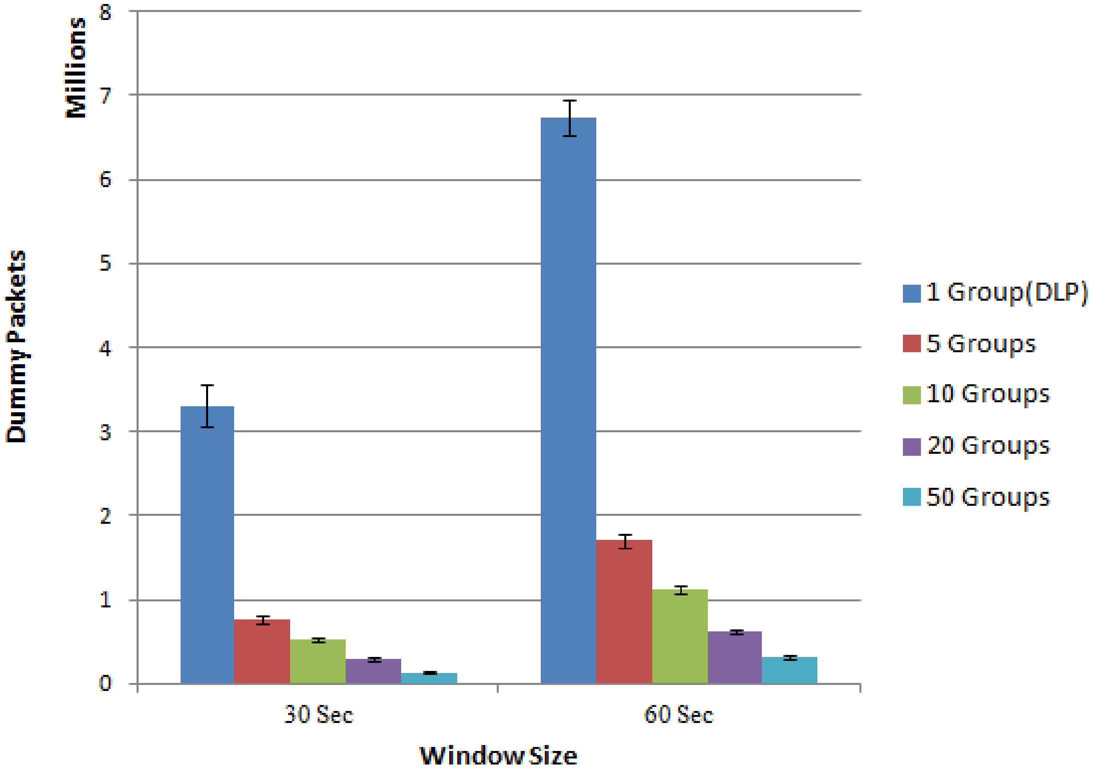

Figure 5.1. Different Number of Groups.

5.2 Variation of Clustering Algorithms

Here we have analyzed the effect of using various clustering algorithms in Selective Grouping on the dummy overhead. We performed runs for 30 and 60 second window lengths for each round with 10 groups (500 clients/group) in the network. For grouping we used k-means clustering, random selection of clients, and sequential selection with standard deviation. We observed that sequential selection with standard deviation performed very well with a 49% decrease in overhead in comparison to random selection for a 30 second window size, as shown in Figure 5.2. For the 60 second window size, overhead decreased by 46% which is also a substantial decrease in the absolute number of dummy packets.

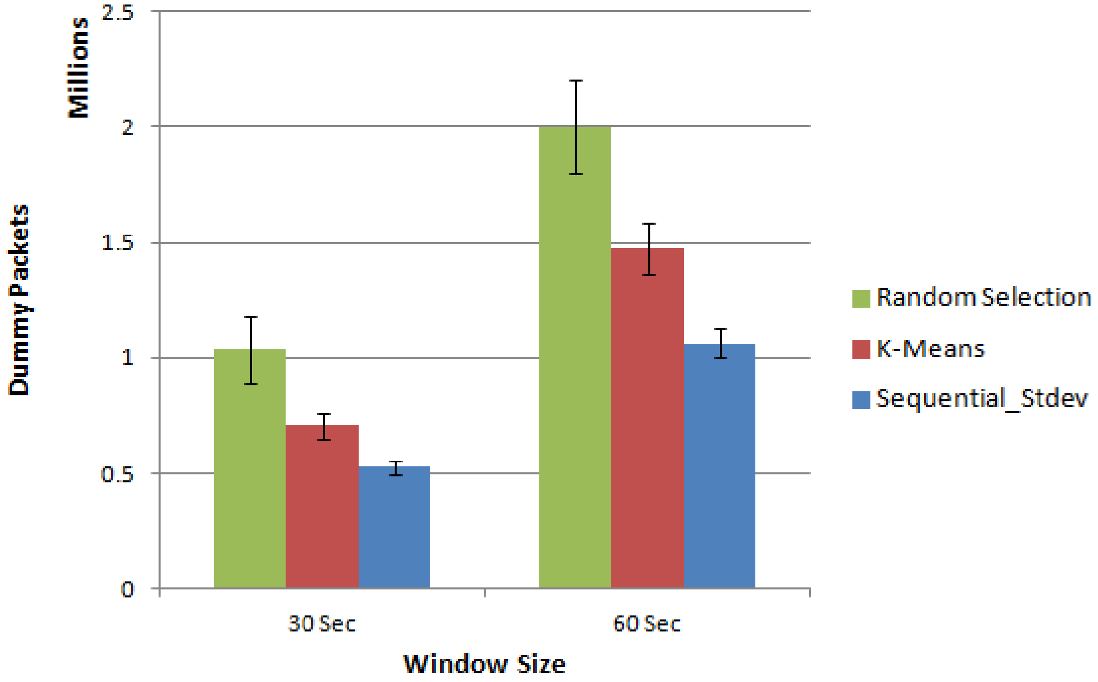

Figure 5.2. Different clustering algorithms.

5.3 Comparison of One and Two Dimension Algorithms

Next, we have compared the results of one dimension and two dimension algorithms. We used the packets count for the one dimension case, and for two dimensions we used both packets count and standard deviation of IPDs. We performed runs for 30 and 60 seconds window sizes with 10 groups (500 clients/group) in the network. For one dimension, we used sequential selection algorithm and k-means with one input field, and for two dimensions we used sequential selection with standard deviation and k-means with two input fields. In Figure 5.3 Our result shows that two dimensional algorithms gives more reduction in overhead compared to one dimensional algorithms. Sequential selection with standard deviation produce 18% better results then sequential selection.

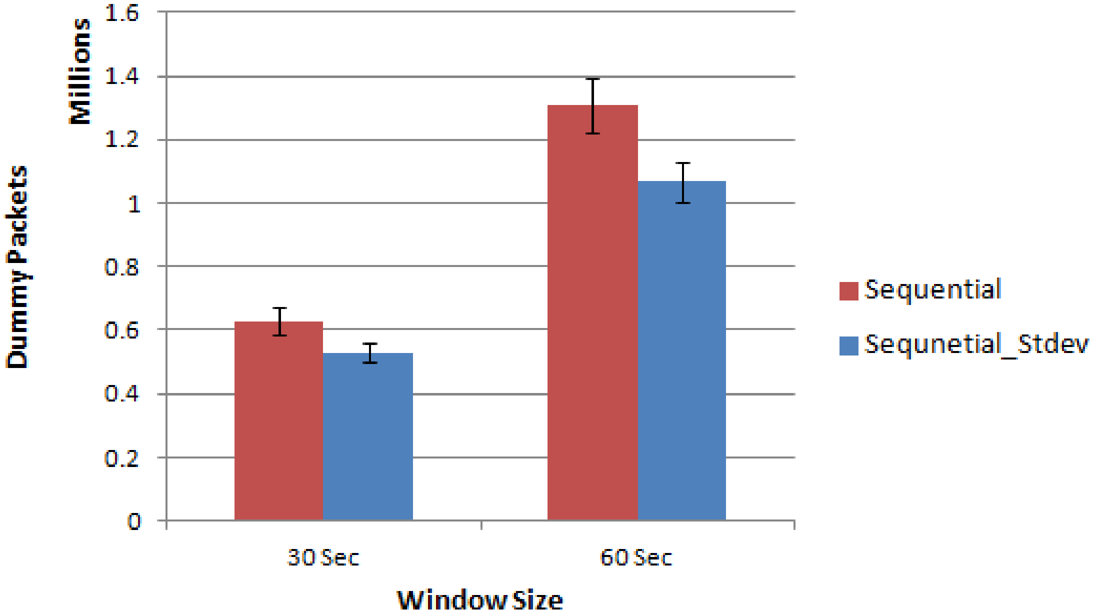

Figure 5.3. Comparison of one and two dimension algorithms.

5.4 Variation of Window size

To study the effect of window size, we varied the window size from 15 seconds up to 120 seconds and analyzed the dummy overhead per second. We used the sequential selection algorithm to perform these simulation with 10 groups (500 clients/group) in the network. We fixed the maximum value of active circuits to 5000 in the whole network. Our result shows that with an increase in window size, dummy overhead also increases as shown in Figure 5.5. We ran this simulation for 6 samples and observed a gradual increase for all samples. For a 60 second window, we observed an 80% increase in overhead in comparison to the 30 second window.

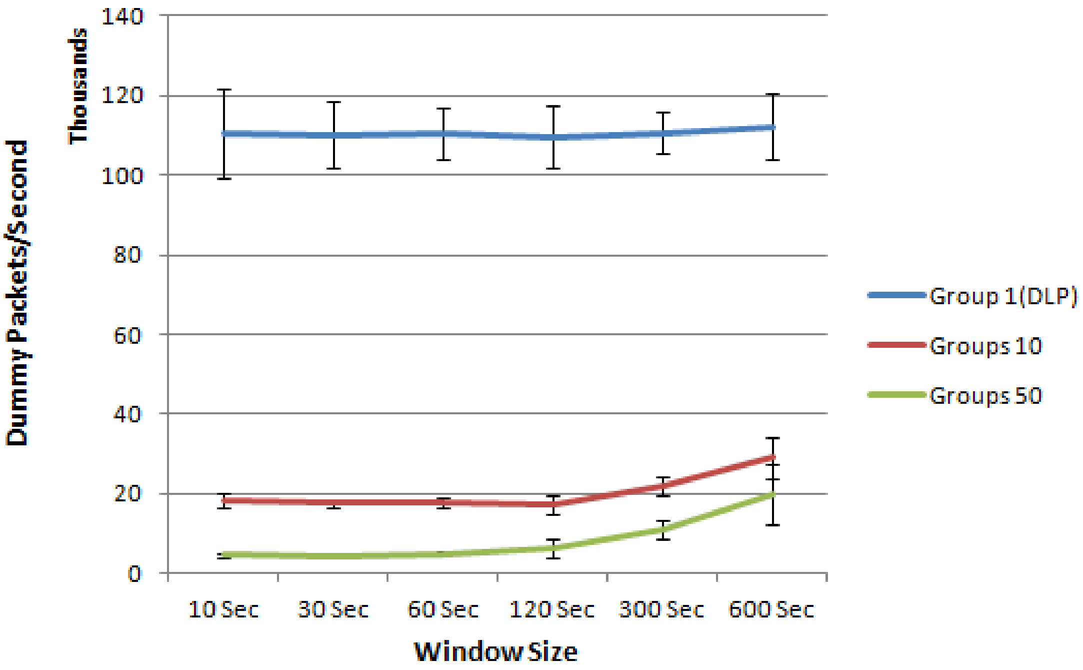

Figure 5.4. Different window size.

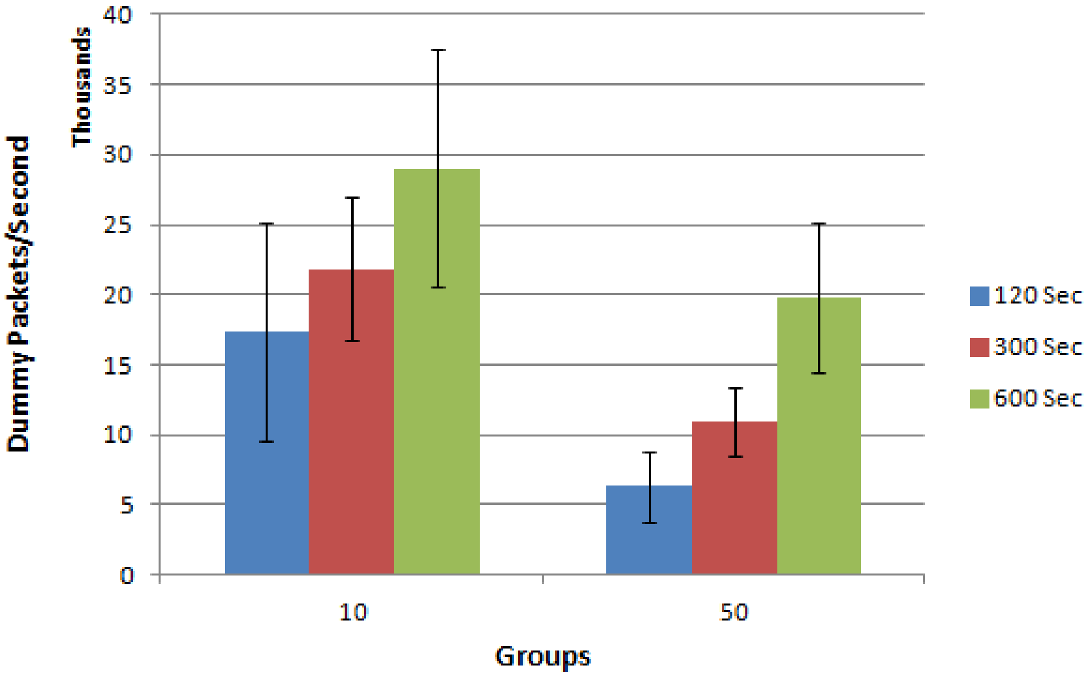

Figure 5.5. Window size Vs Groups.

5.5 Anonymity Analysis of SG by SDA

In this simulation, we measured anonymity of clients by varying their group size according to the selective grouping algorithm. We ran simulations for 32 Alice recipients by varying rounds from 100 to 5000, with total recipients at 5000. Our results show that anonymity of the user (Alice) increases as number of groups decrease, which is also shown in Figure 5.6. We also observed that with the increase in number of rounds, anonymity of Alice decreases. For a group count of 50 (100 clients/group), we observed a drastic fall in anonymity compared to a group count of less than 10 (500 clients/group).

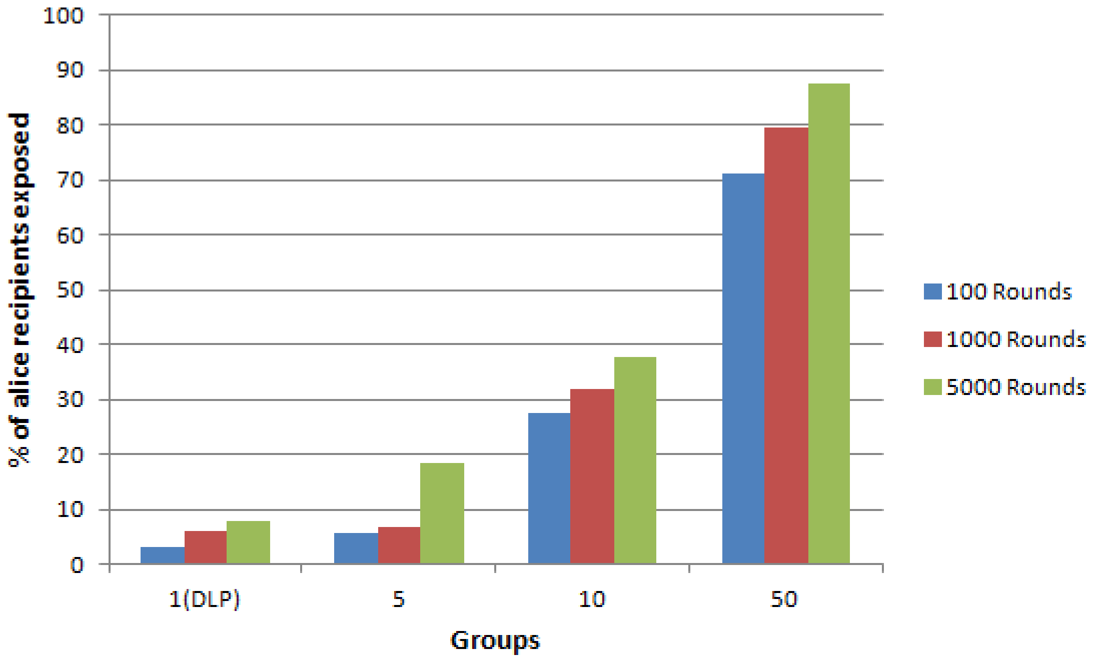

Figure 5.6. Anonymity analysis of selective grouping by SDA.

CHAPTER 6

CONCLUSION

In this thesis we proposed the selective grouping algorithm which effectively works against timing analysis attacks with minimal dummy overhead. The main aim of the selective grouping is to reduce dummy packets by profiling users while maintaining reasonable anonymity. We conducted several experimental simulations using different clustering algorithms to reduce network congestion. For measuring anonymity we performed statistical disclosure attacks against selective grouping. With this, we show that the defense used against timing analysis attacks needs more improvement on the factor of network congestion. Our selective grouping is one of those defenses which provides good anonymity with minimal overhead in the network.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13 th Usenix Security Symposium*, 2004.

[2] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, “Timing attacks in low-latency mix systems (extended abstract),” in *Procedings of the 8th international financial cryptography conference.*

[3] P. Venkitasubramaniam, T. He, and L. Tong, “Relay secrecy in wireless networks with eavesdroppers,” in *Proc. of 2006 Allerton Conference on Communication, Control and Computing*, 2006.

[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, pp. 323–332.

[5] C. Diaz, S. J. Murdoch, and C. Troncoso, “Impact of network topology on anonymity and overhead in low-latency anonymity networks,” in *Proceedings of the 10th international conference on Privacy enhancing technologies*, 2010, pp. 184–201.

[6] P. Venkitasubramaniam and L. Tong, “Anonymous networking with minimum latency in multihop networks,” in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008.

[7] G. Danezis and A. Serjantov, “Statistical disclosure or intersection attacks on anonymity systems,” in *Information Hiding*, 2005, vol. 3200.

[8] Jrandom, *Invisible Internet Project*, Apr. 2012. [Online]. Available: http://www.i2p2.de/

[9] O. Berthold, H. Federrath, and S. Köpsell, "Web mixes: a system for anonymous and unobservable internet access," in *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*, 2001, pp. 115–129.

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

[11] P. Maymounkov and D. Mazières, "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.

[12] J. Feigenbaum, A. Johnson, and P. Syverson, "Preventing active timing attacks in low-latency anonymous communication," in *Proceedings of the 10th international conference on Privacy enhancing technologies*, 2010, pp. 166–183.

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

[14] W. Dai, *Pipenet*, Apr. 2012. [Online]. Available: from http://www.weidai.com/pipenet.txt/

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

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

[17] E. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–780, 1965.

[18] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

BIOGRAPHICAL STATEMENT

Vishal Gupta was born in India in 1983, completed his Masters from UTA in 2012.