Who are You Talking to? Breaching Privacy in Encrypted IM Networks

Muhammad U. Ilyas
Department of EE, SEECS
National University of Sciences & Technology
Islamabad – 44000, Pakistan
usman.ilyas@seecs.edu.pk

M. Zubair Shafiq, Alex X. Liu*
Department of CSE
Michigan State University
East Lansing, MI – 48824
{shafiqmu,alexliu}@cse.msu.edu

Hayder Radha
Department of ECE
Michigan State University
East Lansing, MI – 48824
radha@egr.msu.edu

***Abstract*—We present a novel attack on relayed instant messaging (IM) traffic that allows an attacker to infer who's talking to whom with high accuracy. This attack only requires collection of packet header traces between users and IM servers for a short time period, where each packet in the trace goes from a user to an IM server or vice-versa. The specific goal of the attack is to accurately identify a candidate set of top-k users with whom a given user possibly talked to, while using only the information available in packet header traces (packet payloads cannot be used because they are mostly encrypted). Towards this end, we propose a wavelet-based scheme, called COmmunication Link De-anonymization (COLD), and evaluate its effectiveness using a real-world Yahoo! Messenger data set. The results of our experiments show that COLD achieves a hit rate of more than 90% for a candidate set size of 10. For slightly larger candidate set size of 20, COLD achieves almost 100% hit rate. In contrast, a baseline method using time series correlation could only achieve less than 5% hit rate for similar candidate set sizes.**

I. Introduction

The proliferation of online social networks has attracted the interest of computer scientists to mine the available social network data for developing behavior profiles of people. These profiles are often used for targeted marketing [7], [20], [21], web personalization [16], and even price discrimination on e-commerce portals [11], [15]. Recently, there has been increased interest in more fine-grained profiling by leveraging information about people's friendship networks. It has been shown that information from people's friendship networks can be used to infer their preferences and religious beliefs, and political affiliations [2], [6], [12], [22].

There has been a lot of research on de-anonymizing people's friendship networks in online social networks such as Facebook, MySpace, Twitter [4], [8]. Surprisingly, little prior work has focused on de-anonymizing people's friendship link in instant messaging (IM) networks. IM services – such as Yahoo! Messenger, Skype, IRC, and ICQ – are popular tools to privately communicate with friends and family over the Internet. IM networks are different than other online social networks in various respects. For example, in contrast to online social networks, communication among users in IM networks is synchronous in nature and messages between two communicating users are routed through relay servers of the IM service provider.

The goal of this paper is to identify the set of most likely IM users that a given user is communicating with during a fixed time period. Note that packet payloads in IM traffic are encrypted; therefore, payload information cannot be used for the identification. Therefore, to infer who a user is talking to, we will rely only on the information in packet header traces. Packet header traces contain information such as timestamp, source IP address, destination IP address, source port, destination port, and protocol type, and payload size of each packet. It is noteworthy that each packet in the IM traffic has as its source and destination IP addresses of a user computer and an IM relay server (or vice versa). At no point do two users exchange packets directly with each other, *i.e.*, there are no packets in which the two communicating users' IP addresses appear in the same packet. For this attack, we assume that IM service acts neutral, *i.e.*, it neither facilitates the attacker nor actively participates in providing anonymity to the users using non-standard functionality. Our specific goal is to accurately identify a candidate set of top-k users with whom a given user possibly talked to using only the information available in packet header traces.

A natural approach to tackle this problem would be to match header information of packets entering and leaving IM relay servers. However, simply matching header information of packets entering and leaving IM servers is not feasible due the following reasons. First, a user may be talking to multiple users simultaneously. Second, IM relay servers typically serve thousands of users at a time. Third, the handling of duplicate packets that are the result of packet losses followed by re-transmissions. Forth, the handling of out-of-order packets. Finally, the handling of variable transmission delays, which are introduced by the IM relay servers.

In this paper, we propose a wavelet-based scheme, called COmmunication Link De-anonymization (COLD), to accurately infer who's talking to whom using only the information available in packet header traces. Wavelet transform is a standard method for simultaneous time-frequency analysis and

* Alex X. Liu is the corresponding author of this paper. The majority work of Muhammad U. Ilyas was conducted while he was a postdoctoral researcher at Michigan State University, East Lansing, Michigan, United States.

978-1-4799-1270-4/13/$31.00

helps to correlate the temporal information in one-way (*i.e.* user-to-server or server-to-user) traffic logs across multiple time scales [10]. Wavelet analysis allows decomposition of traffic time series between a user and an IM relay server into several levels. All levels are associated with a coefficient value and contain different levels of frequency information starting from low to high. The original traffic time series can be reconstructed by combining all levels after weighing them with their respective coefficients. COLD leverages the multi-scale examination of traffic time series provided by wavelet analysis to overcome the aforementioned technical challenges. Given two candidate time series between an IM relay server and two users, COLD computes correlation between the vectors of wavelet coefficients for both time series to determine whether these users talked to each other.

We evaluate the effectiveness of COLD on a Yahoo! Messenger data set comprising of traffic collected over 10, 20, 30, 40, 50 and 60 minute periods. We also compare COLD's performance to a baseline time series correlation (TSC) scheme, which represents the state of the art. The effectiveness is quantified in terms of hit rate for a fix-sized candidate set. The results of our experiments show that COLD achieves a hit rate of more than 90% for a candidate set size of 10. For slightly larger candidate set size of 20, COLD achieves almost 100% hit rate. In contrast, a baseline method using time series correlation could only achieve less than 5% hit rate for similar candidate set sizes.

We summarize the major contributions of this paper as follows.

1) We define an attack for breaching communication privacy in encrypted IM networks using only the information available in packet header traces.
2) We propose COLD to to infer who's talking to whom using wavelet based multi-scale analysis.
3) We conducted experiments using a real-world Yahoo! Messenger data set to evaluate the effectiveness of our proposed approach.

Paper Organization: The rest of this paper is organized as follows. Section II summarizes the related work. A detailed description of attack scenarios is provided in Section III. Section IV provides details of the proposed attack. In Section V, we present the evaluation results on a real-world Yahoo! Messenger data set. Possible evasion techniques and their countermeasures are discussed in Section VI. Finally, Section VII concludes the paper.

II. Related Work

In this section, we provide details of the research work related to our study. To the best of our knowledge, no prior work has reported a successful attack to breach users' communication privacy in encrypted IM networks using only the information available in packet header traces. However, there is some relevant work in the area of mix network de-anonymization. We discuss it and other related studies below.

A. Mix Network De-anonymization

In the area of mix network, several studies have used correlation techniques for de-anonymization. However, most of these studies are limited to computing temporal correlation between traffic of two user-network links to find user-user links. Furthermore, de-anonymization of mix networks is fundamentally different from our problem in the following two aspects. First, mix network de-anonymization techniques require traffic logs from multiple points inside a mix network. In contrast, this study treats IM relay servers as a black box. Second, the size of user populations in mix network de-anonymization studies is only of the order of tens or hundreds. However, in real-life IM networks, thousands of users can simultaneously communicate with other users; therefore, presenting a more challenging problem. In [17], Troncoso and Danezis build a Markov Chain Monte Carlo inference engine to calculate probabilities of who is talking to whom in a mix network using network traces. However, they log network traces from multiple points in a mix network and the maximum network size studied in their paper is limited to 10. In [23], Zhu *et al.* compute mutual information between aggregate inflow and outflow traffic statistics to decide if two users are talking to each other in a mix network. Similar to this study, they also log traffic from the edges of a mix network. However, their proposed approach requires traffic logs for longer time durations. In this paper, we compare the results of COLD and the method proposed by Zhu *et al.* [23].

B. Social Network De-anonymization

There is also some related work in the field of social network de-anonymization. Narayanan and Shamitkov developed an algorithm to utilize sparsity in high-dimensional data sets for de-anonymization [13]. Later they developed a user re-identification algorithm that operated on anonymized social network data sets [14]. Other related studies use group membership information to identify users in a social network [19], [22]. IM networks also fall under the broader category of online social networks; however, our problem and the nature of the data available to us is different from those tackled in the aforementioned papers. These studies focus on user identification using mainly topological information; whereas, we focus on link identification using dynamic user communication traffic.

III. Problem Description and Attack Scenarios

In this section, we first provide a summary of architectural details of IM services. We then provide the details of information available from traffic traces logged near IM relay servers. Finally, we describe two scenarios in which traffic can be logged for link de-anonymization.

A. IM Service Architecture

We first describe the architecture of a typical IM service. Consider the scenario depicted in Figure 1 where two users v_1 and v_2 are communicating with each other via an IM

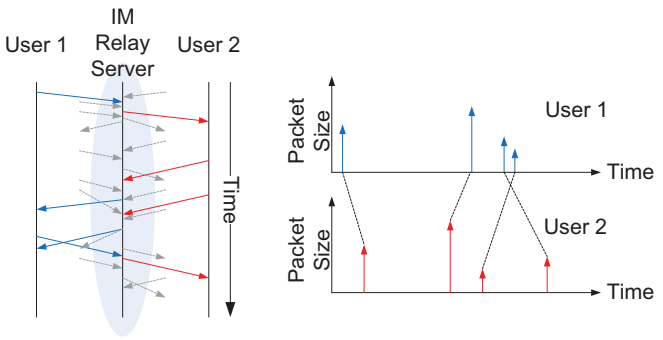
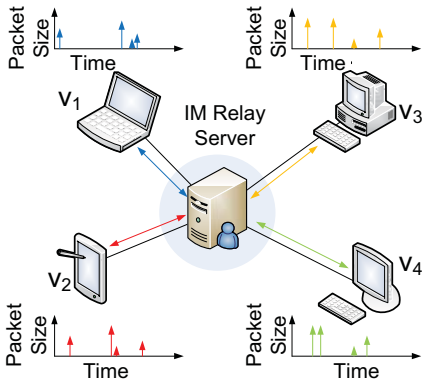


Fig. 1. Transforming logged traffic traces to user traffic signals.

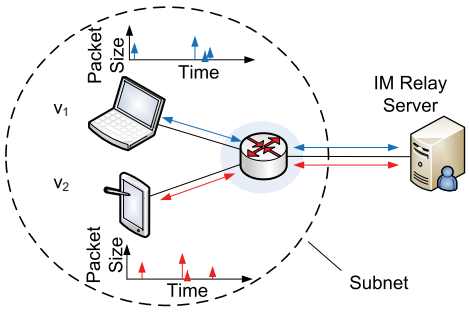
service. When v_1 sends a message to v_2, the source IP address in packets containing this message correspond to v_1 and the destination IP address correspond to the IM relay server. These packets are received by the IM relay server after a random delay. After receiving a packet from v_1, the IM server first looks up the IP address of v_2. It then creates new packets with its IP address as source and IP address of v_2 as destination. These packets containing message from v_1 are then relayed by the IM relay server to v_2 and have the same contents. This process incurs an additional delay after which the new packet reaches v_2.

The network traffic logged near the IM relay server only contains header information because the packet payload contents are not useful due to encryption. The statistics recorded by the well-known traffic logging tools like Cisco's NetFlow include IP addresses, port numbers, protocol, packet size, and timestamp information [3]. As mentioned before, IP addresses are used to identify individual users of the IM service. IM traffic is filtered from rest of the traffic using a combination of protocol and port number information. We are left with only aggregated packet sizes and timestamp information for each flow. A logged entry for a flow is an aggregation of packets which may be sent to or received from the IM server. Due to aggregation, information about the direction of flow is lost for individual packets. Therefore, we make a realistic assumption that the direction information is not available in the logged traffic. An example of a similar publicly available data set is the Yahoo! Network Flows Data [1].

Referring to Figure 1, each flow in the data set comprises of information about incoming and outgoing packets between an IM relay server and a user. Furthermore, individual users can be distinguished based on IP addresses in the IM traffic. In Figure 1, traffic exchanged between v_1 and the IM relay server is represented by blue arrows and traffic exchanged between v_2 and the IM relay server is represented by red arrows. The timestamps and packet sizes are both discrete and in units of milliseconds. The packet sizes are typically recorded in bytes. The resulting signal for each flow is discrete in both time and amplitude as shown in Figure 1. These sparse time domain traces of network traffic are referred to as *traffic signals* from now-onwards. It is interesting to simultaneously analyze traffic signals for both users v_1 and v_2. Note that every entry in v_1's



(a) Collecting all incoming and outgoing traffic from IM relay server



(b) Collecting all incoming and outgoing traffic near border gateway routers of an organizational network

Fig. 2. Two attack scenarios

traffic signal has a time-shifted (time delayed or advanced) matching entry of equal magnitude in v_2's traffic signal. These matches between each pair of entries are marked by broken lines joining traffic signals in Figure 1. Matching entries across both traffic signals may not have the same order due to random end-to-end delays. For example, 3^{rd} message flow entry in v_2's trace appears as 4^{th} entry in v_1's trace in Figure 1.

B. Attack Scenarios

We now consider two different scenarios in which traffic information necessary for the proposed attack can be obtained.

1) Scenario #1: Near IM relay servers: The first scenario assumes the capability to monitor incoming and outgoing traffic of an IM relay server or server farm. Figure 2(a) shows four users v_1, v_2, v_3 and v_4 connected to an IM relay server. The shaded circular region around the IM relay server marks the boundary across which network traffic is logged. For the scenario depicted in Figure 2(a), v_1 is communicating with v_2 and v_3 is communicating with v_4. Traffic signals for all users that are obtained after pre-processing their traffic flow logs. For each flow represented in a user's traffic signal, a corresponding flow entry can be observed in the traffic flow log. The IM relay servers also introduces a random delay between the time a message arrives at the IM relay server and the time it is relayed to the other user. Therefore, there will be a mismatch in the timestamps of the occurrences of a message in communicating users' traffic signals.

2) Scenario #2: Border gateway: The second scenario assumes that all IM users communicating with each other are located in the same network. Many organizations, such as universities, connect to external networks and the Internet through one or more gateway routers. The incoming and outgoing traffic has to pass through a small number of gateway routers. In this scenario, it is possible to collect flow logs near the gateway routers of an organizational network. Figure 2(b) depicts the above-mentioned scenario. Here, v_1 and v_2 are in the same network and are communicating with each other via an IM relay server. All incoming and outgoing traffic of the network passes through the border gateway router near which it can be logged. The region near border gateway router is represented by the shaded region in Figure 2(b). The traffic signals obtained from pre-processing the flow logs have the same characteristics as described for the first scenario.

IV. COLD: COMMUNICATION LINK DE-ANONYMIZATION

In this section, we present the details of our proposed method (COLD) to detect communication links between users in IM networks. We first introduce the overall architecture of COLD. We then provide details of each of its modules. Finally, we provide an easy-to-follow toy example of COLD on a small set of three IM users.

A. Architecture

Figure 3 shows the overall architecture of COLD. As mentioned in Section III, the logged traffic traces are separated for all users based on IP address information. These user-wise separated traffic traces are further pre-processed and converted to traffic signals. The traffic signals for all users are stored in a database. Note that traffic signals of users may span different time durations. To overcome this problem, we use zero-padding so that the lengths of traffic signals are consistent for all users. After this pre-processing, wavelet transform is separately applied to all users' traffic signals [10]. We then construct feature vectors for all users using the computed wavelet coefficients. Now, to compare two given users, we compute the correlation coefficient between their constructed feature vectors. Finally, the values of the correlation coefficient are sorted to generate the candidate set. The details of all modules of COLD are separately discussed below.

B. Details

1) Discrete Wavelet Transform: After pre-processing the traffic traces, we compute the discrete wavelet transform (DWT) of each user's traffic signal. This step is performed in the wavelet decomposition module shown in Figure 3. The wavelet transform enables us to conduct a simultaneous time-frequency analysis. A traffic signal is decomposed into multiple time series, each containing information at different scales that range from coarse to fine. A time series at a coarse scale represents the low frequency or low pass information regarding the original time series. Likewise, a time series at a fine scale represents the high frequency or high pass information regarding the original time series. This allows us to compare traffic patterns of users at multiple time scales.

We have to select an appropriate wavelet function for our given problem. Since we are processing traffic signals of a large number of users, we want to select an efficient wavelet type. For our study, we have chosen the Haar wavelet function for wavelet decomposition [9]. We have chosen the Haar wavelet function because it is simple and is computationally and memory-wise efficient. Furthermore, the wavelet transform can be applied for varying decomposition levels to capture varying levels of detail. Choosing the optimal number of decomposition levels is important because this may lead to suppressing relevant and critical information that might be contained in one or more levels of the wavelet decomposition. Below, we discuss the method to select the optimal number of decomposition levels.

2) Choosing the Optimal Number of Decomposition Levels: Let $D \in \mathbb{Z}^+$ denote the optimal number of decomposition levels. Different methods have been proposed in the literature to select the optimal number of decomposition levels. In this paper, we have used Coifman and Wickerhauser's well-known Shannon entropy-based method to select the optimal number of decomposition levels [5]. We applied this method to traffic signals of all users and then selected the optimal decomposition level at the 95th percentile. Now that we have selected the optimal number of decomposition levels, we can apply the wavelet transform on user traffic signals.

3) Coefficient Feature Vector: Once we have obtained the wavelet coefficients after applying the wavelet transform to a user's traffic signal, we need to convert them to a standard feature vector so that we can compare users' signals. Let $\mathcal{F}_X$ denote the feature vector of a user X. The coefficients that contain high frequency information are more numerous and such coefficients are assigned lower weights. Similarly, the coefficients that contain low frequency information are fewer and are assigned higher weights. The time signal corresponding to level 1 of the wavelet decomposition represents the coarsest features containing low frequency information, and level D refers to the highest level describing the most detailed features containing high frequency information. The level D feature coefficients are assigned weight 1, the level $D-1$ coefficients are assigned weight 2, etc., and the level 1 coefficients are assigned weight 2^{D-1}. In general, the level d features are assigned a weight of 2^{D-d-1}. To produce the standard feature vector in which each coefficient is given the appropriate weight, we replace each coefficient by a vector of its copies of length equal to its weight, *i.e.* a wavelet coefficient of decomposition level d is replaced by a vector containing 2^{D-d-1} copies. This is equivalent to using the undecimated wavelet transform of users' traffic signals. By following this procedure, the total length of the feature vectors of all traffic signals becomes consistent.

4) Correlation: After applying the wavelet transform and post-processing coefficients to a user X's traffic signal, we obtain a feature vector denoted $\mathcal{F}_X$. To compare the feature vectors $\mathcal{F}_X$ and $\mathcal{F}_Y$ for two users X and Y, we have to

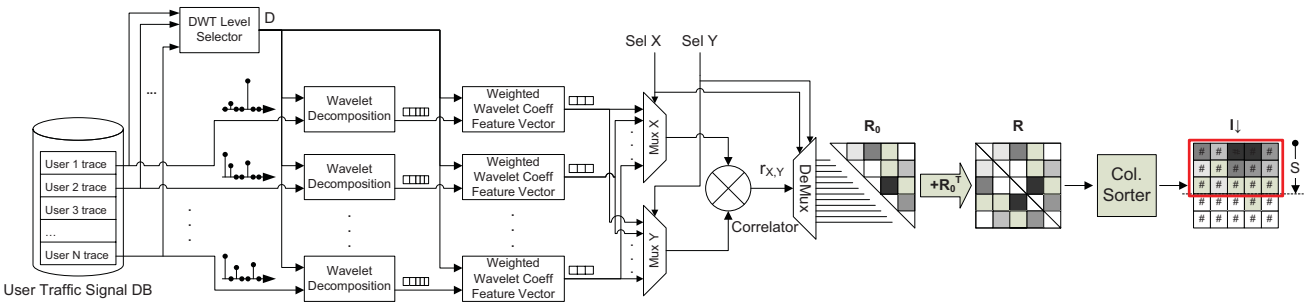


Fig. 3. COLD architecture

compute their correlation. The sample correlation coefficient $r_{X,Y}$ of two discrete signals $\mathcal{F}_X$ and $\mathcal{F}_Y$, both of length L, is defined as,

$$r_{X,Y} = \frac{\sum_{i=1}^{L}(\mathcal{F}_X(i) - \overline{\mathcal{F}_X})(\mathcal{F}_Y(i) - \overline{\mathcal{F}_Y})}{(L-1)s_X s_Y}. \quad (1)$$

Here, $\mathcal{F}_X(i)$ is the ith element of the feature vector $\mathcal{F}_X$, $\overline{\mathcal{F}_X}$ is the sample mean of its elements, and s_X is the sample standard deviation of its elements. The values of the correlation coefficient lie in the closed interval $[-1, 1]$. The correlation coefficient values close to zero indicate no correlation; whereas, the values close to 1 and -1 respectively highlight strong correlation and anti-correlation. For this study, we only consider the magnitude of the correlation coefficient and discard its sign. After computing the correlation coefficient for all pairs of users, we get the upper triangular correlation matrix $\mathbf{R}_0$. $r_{i,j}$ is written into the ith row and the jth column of the correlation matrix $\mathbf{R}_0$. Conceptually, this correlation matrix is similar to the adjacency matrix of a weighted graph. We add to $\mathbf{R}_0$ its transpose to obtain $\mathbf{R}$

5) Candidate Set Generation: After obtaining the correlation matrix $\mathbf{R}$ whose elements are in the range of $[0, 1]$ we need to generate, for each node, a sorted list of nodes in decreasing order of probability of communicating. This is done by sorting the node indices in descending order of correlation coefficients in every column of $\mathbf{R}$. The resulting matrix will have the same size as $\mathbf{R}$ and is labeled $\mathbf{I}\downarrow$. Suppose that the S most likely users that are communicating with user i is required. Then the user IDs contained in the top S rows of the i-th column of $\mathbf{I}\downarrow$ is the sorted list of users i is most likely communicating with.

C. Example

We now provide a easy-to-follow toy example of COLD on three users (A, B, and C) in an IM network. Users A and B are communicating with each other while user C is not communicating with either user A or user B. Figure 4 shows the traffic signals for all three users. The traffic signals of users A and B are visibly similar to each other and significantly different from the traffic signal of user C. However, if we directly compute the correlation coefficients of users' time signals we get $r_{A,B} = -0.0046$, $r_{B,C} = -0.0053$, and $r_{A,C} = -0.0053$. Equivalently, the correlation matrix is:

$$\mathbf{R} = \begin{pmatrix} 1 & 0.0046 & 0.0053 \\ 0.0046 & 1 & 0.0053 \\ 0.0053 & 0.0053 & 1 \end{pmatrix}$$

This indicates that directly correlating users' traffic signal time series is not accurate.

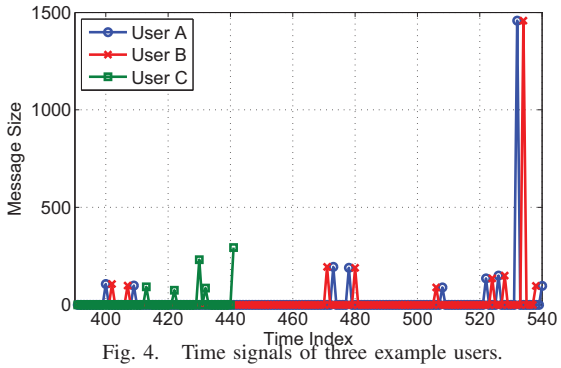


Fig. 4. Time signals of three example users.

Let us now obtain the feature vectors for all users using the wavelet transform. Figure 5 shows the coefficient feature vectors for users A, B, and C. Note that the feature vectors at lower indices contain coarse grain or low frequency information. We observe significant similarity between the lower indices of the feature vectors of users A and B. Now when we compute the correlation coefficients of users' feature vectors we get $r_{A,B} = 0.7042$, $r_{B,C} = 0.0743$, and $r_{A,C} = 0.0742$. Equivalently, the correlation matrix is:

$$\mathbf{R} = \begin{pmatrix} 1 & 0.7042 & 0.0742 \\ 0.7042 & 1 & 0.0743 \\ 0.0742 & 0.0743 & 1 \end{pmatrix}$$

This clearly indicates the superiority of COLD (with the wavelet-based feature vectors) attack method compared to the direct correlation of users' traffic signals.

V. Experimental Results

In this section, we first describe the data set used for evaluating COLD, then define evaluation metrics, and finally present evaluation results.

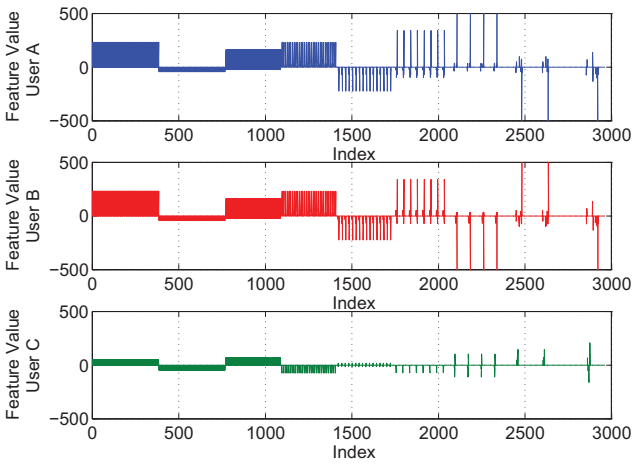


Fig. 5. Wavelet feature vectors of three example users.

TABLE I
DATA SET STATISTICS

Time	Duration	Users	Messages	Sessions
8-8:10a	10 mins	3,420	15,370	1,968
8-8:20a	20 mins	5,405	33,192	3,265
8-8:30a	30 mins	7,438	53,649	4,661
8-8:40a	40 mins	9,513	75,810	6,179
8-8:50a	50 mins	11,684	99,721	7,669
8-9a	60 mins	13,953	126,694	9,264

A. Data Set

We collected a data set from Yahoo! Messenger IM network to validate our proposed approach. To keep the volume of logged data manageable, the users of Yahoo! Messenger were filtered by geographic location and restricted to the New York City area. This data set consists of traffic logs of Yahoo! Messenger user activity over a period of 60 minutes from the greater New York area, between 8 a.m. to 9 a.m. Using this data set, we create six data sets that are the subsets of the entire data. These consist of data over the only the first 10, 20, 30, 40, 50 and 60 minutes, *i.e.* from $8-8:10$ a.m., $8-8:20$ a.m., $8-8:30$ a.m., $8-8:40$ a.m., $8-8:50$ a.m. and $8-9$ a.m. To gauge the effect of the duration over which a data set is collected we evaluated our proposed COLD scheme on all six data sets. Table I lists, along with the time of day and duration, the number of logged users, number of messages exchanged between them, and the number of instant messaging sessions included in each data set.

The collected data is divided into two parts: input data and ground truth data, to systematically evaluate our proposed approach. Both data sets were collected with the assistance of Yahoo! and are described in the following text.

1) Input Data: The input data consists of *user-to-server* traffic traces that were collected similar to the scenario described in Figure 2(a). Figure 6 plots the volume of traffic logged in these traffic traces. The figure on top plots number of bytes per second against time. Similarly, the plot in the bottom figure plots the traffic volume in packets per second for the same period of time.

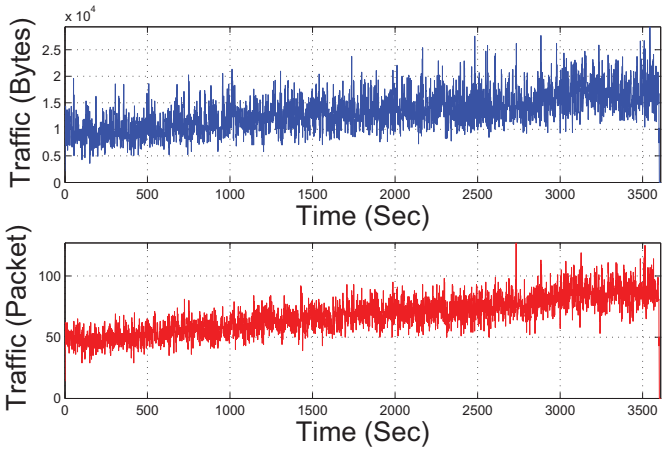


Fig. 6. Time series plot of traffic volume, in bytes and number of packets, over the entire 60 minute time period from $8-9$ a.m.

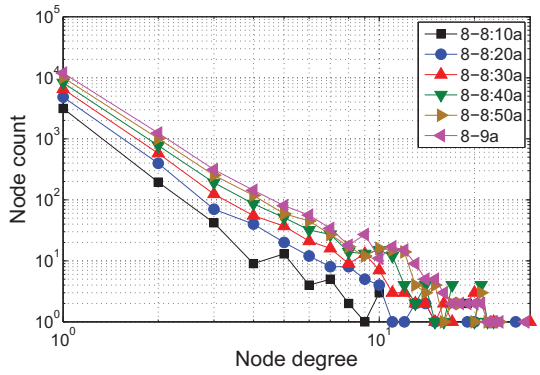


Fig. 7. Node degree distribution in our Yahoo! Messenger data set.

2) Ground Truth Data: The verification data contains a record of the actual *user-to-user* connections resulting from conversations between users. Therefore, the verification data contains the ground truth for given problem. Our proposed COLD scheme attempts to recreate the link structures between users contained in the verification data by only using information in the input data. Figure 7 is a plot of the degree distribution of users observed in the verification data collected over 10 and 60 minute time periods. The distribution is approximately linear on log-log scale over the range of degrees from 1 to 9 for the 10 minute data, and from 1 to 11 for the 60 minute data.

B. Evaluation Metrics

Let V denote the set of Yahoo! Messenger users $v_1, v_2, \ldots, v_N$. Furthermore, let E denote the set of actual communication links $u_1, u_2, \ldots, u_M$ of size M between N users captured in the verification data. Then $G(V, E)$ is the graph of users (or vertices) connected by the communication links (or edges) between them. Recall that the goal of the attack is to detect communication links $\widehat{U}$ that estimates the actual set of communication links in the verification data U. The graph $\widehat{G}(V, \widehat{U})$ is the outcome of the scheme that constitutes the attack. In the rest of this section, we compare

our proposed COLD scheme with the baseline time series correlation (denoted by TSC here onwards). A graph that is obtained using COLD will be denoted by $\widehat{G}_C(V, \widehat{U}_C)$. A graph obtained using TSC is denoted by $\widehat{G}_T(V, \widehat{U}_T)$.

Consider the subset of vertices with degree δ in a graph $\widehat{G}\left(V, \widehat{U}\right)$ obtained using either schemes. Now consider a candidate set C_i of size $S \geq \delta$ for every vertex v_i of degree δ. The candidate set C_i of a vertex v_i contains S vertices most likely to be v_i's neighbors, as determined by the COLD or TSC. We also define a neighborhood function denoted by $\Gamma_G()$. $\Gamma_G(v_i)$ returns the set of vertices in the graph G that are connected to vertex v_i. Furthermore, we define the node hit rate of a vertex v_i as the fraction of vertices in $\Gamma_G(v_i)$ that are also elements of candidate set C_i of size S. The node hit rate of vertex v_i is denoted $h_i(S)$ an is defined formally as follows.

$$h_i(S) = \frac{|\Gamma_G(v_i) \cap C_i(S)|}{|\Gamma_G(v_i)|} \tag{2}$$

The node hit rate can take values in the range of the closed interval $[0, 1]$. We also define the hit rate $H_{\widehat{G}}(S, \delta)$ for degree δ vertices of a graph $\widehat{G}\left(V, \widehat{U}\right)$ as the average of their node hit rates $h_i(S)$ when candidate set sizes are S.

$$H_{\widehat{G}}(S, \delta) = \frac{\sum_{i=1, \delta_i=\delta}^{N} h_i(S)}{n_d} \tag{3}$$

Here n_d is the number of vertices in $\widehat{G}$ of degree δ. Just like the node hit rate, the hit rate can take values in the range of the closed interval $[0, 1]$.

C. Results

We compute the hit rates achieved using COLD on the 10, 20, 30, 40, 50 and 60 minute data sets and compare them with the hit rates achieved by TSC. We further separate vertices by the number of packets they exchange over the duration of the data set, *i.e.* hit rates are computed separately for vertices exchanging $1-60$, $61-70$, $71-80$, $81-90$, $91-100$, $101-110$, and $111-120$ packets. As we observed in the degree distributions of nodes in figure 7, data sets for all six durations are dominated by nodes of degree 1. Therefore, in our evaluation we focus primarily on degree 1 vertices. Figures 8(a), 8(b), 8(c), 8(d), 8(e), and 8(f) plot the hit rates of degree 1 vertices as a function of set size S for COLD on 10, 20, 30, 40, 50, and 60 minute data sets, respectively. Within each figure, hit rates are segregated according to the number of packets users send and receive over the duration the data was collected. As these six figures consistently show, the hit rate reaches between 0.9 and 1.0 for users exchanging 71 or more packets over the duration of the data sets. In case of the 20, 30, 40, 50, and 60 minute data sets in Figures 8(b), 8(c), 8(d), 8(e), and 8(f), this set of users is further extended to those exchanging 61 or more packets. In the 10 minute data set in figure 8(a) users with 61-70 packets in their trace have a high hit rate of more than 0.80. However, the candidate set size S has to be increased all the way to 40 for the hit rate to reach close to 1.0. For users exchanging between 1-60 packets the hit rate starts out between 0.20 and 0.40. As the candidate set size is increased from 1 upward, the hit rate rises at a very similar rate in all six data sets.

We compare the accuracy of our proposed approach to that of the time series correlation (TSC) method. Similarly, figures 9(a), 9(b), 9(c), 9(d), 9(e) and 9(f) plot the hit rates of degree 1 vertices as a function of set size S for TSC on 10, 20, 30, 40, 50 and 60 minute data sets, respectively. The baseline TSC method, which represents the state of the art, fails to deliver sufficient performance to be useful for any conceivable application, across all six data sets. With one slight exception, TSC fails to achieve a hit rate of even 0.20 even for candidate set size of as large as 100. The only exception is the group of users exchanging between 71-80 packets in the 10 minute data set. However, even for this subset of users, TSC provides a hit rate of less than 0.30 at a set size greater than 70, *i.e.* at best, for users messaging with only one other person, in a set of 70 candidates TSC will include the actual instant messaging partner with a probability of only 0.30.

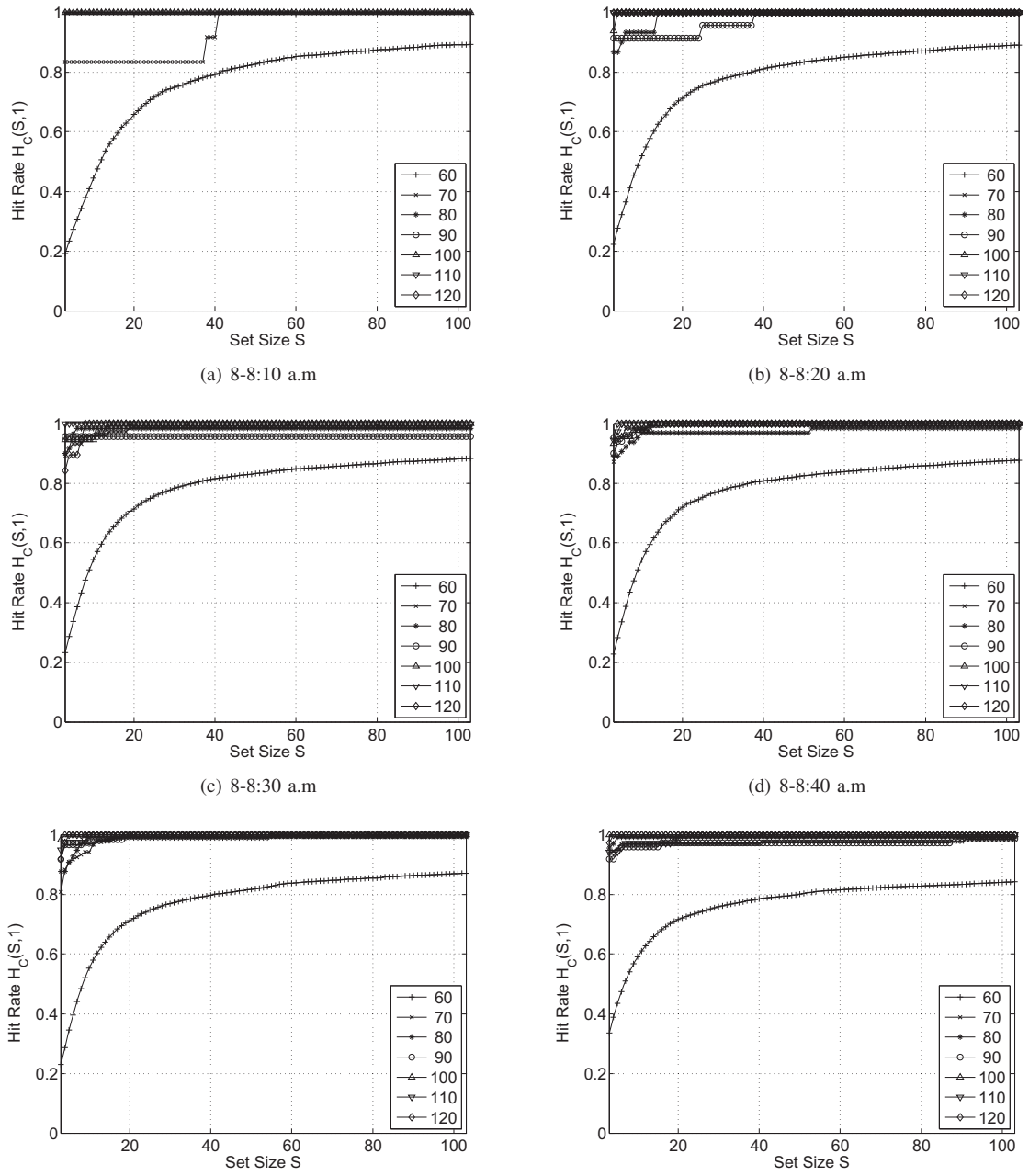
D. Discussions

These results provide us with several insights into the working of COLD. We separately discuss these insights in the following text.

First, there appears to be a very clear threshold value for the number of recorded packets beyond which the de-anonymization attack using COLD yields high hit rates. From the plots in figure 8 we observe that the hit rate for users containing more than 60 packets in their traffic traces is significantly higher, above 90%, even at very small candidate set sizes. On the other hand, the hit rate of users containing 60 packets or less in their traffic trace is significantly lower. This threshold value holds across all six data sets of different durations. More packet entries in traffic traces provide more points to match two communicating users' traces with each other. The greater number of data points also reduces the probability of a false match. Therefore, it is easier to identify communicating users that message each other more frequently.

Second, the hit rate of users, classified by the traffic they generate, is largely independent of the time duration over which the traces were collected. Rather, it is the actual number of message packets exchanged during that period that determines the hit rate. Hit rates for users exchanging the same number of packets over different periods of time are very similar. Therefore, we can state that we can identify two communicating users using COLD with great certainty as soon as they exchange more than 60 message packets.

Third, while we have already stated that the time period over which traffic traces are collected have only a weak effect on the hit rate. However, looking at the hit rate functions of users with $61-70$, $71-80$ and $81-90$ packets in their traffic trace across different data sets, we observe that the hit rate function rises close to 1.0 at a faster rate in data sets collected over longer durations.



(a) 8-8:10 a.m

(b) 8-8:20 a.m

(c) 8-8:30 a.m

(d) 8-8:40 a.m

(e) 8-8:50 a.m

(f) 8-9 a.m

Fig. 8. Hit rates of COLD for vertices of degree 1 in the (a) 10 minute data set, (b) 20 minute data set, (c) 30 minute data set, (d) 40 minute data set, (e) 50 minute data set, and (f) 60 minute data set.

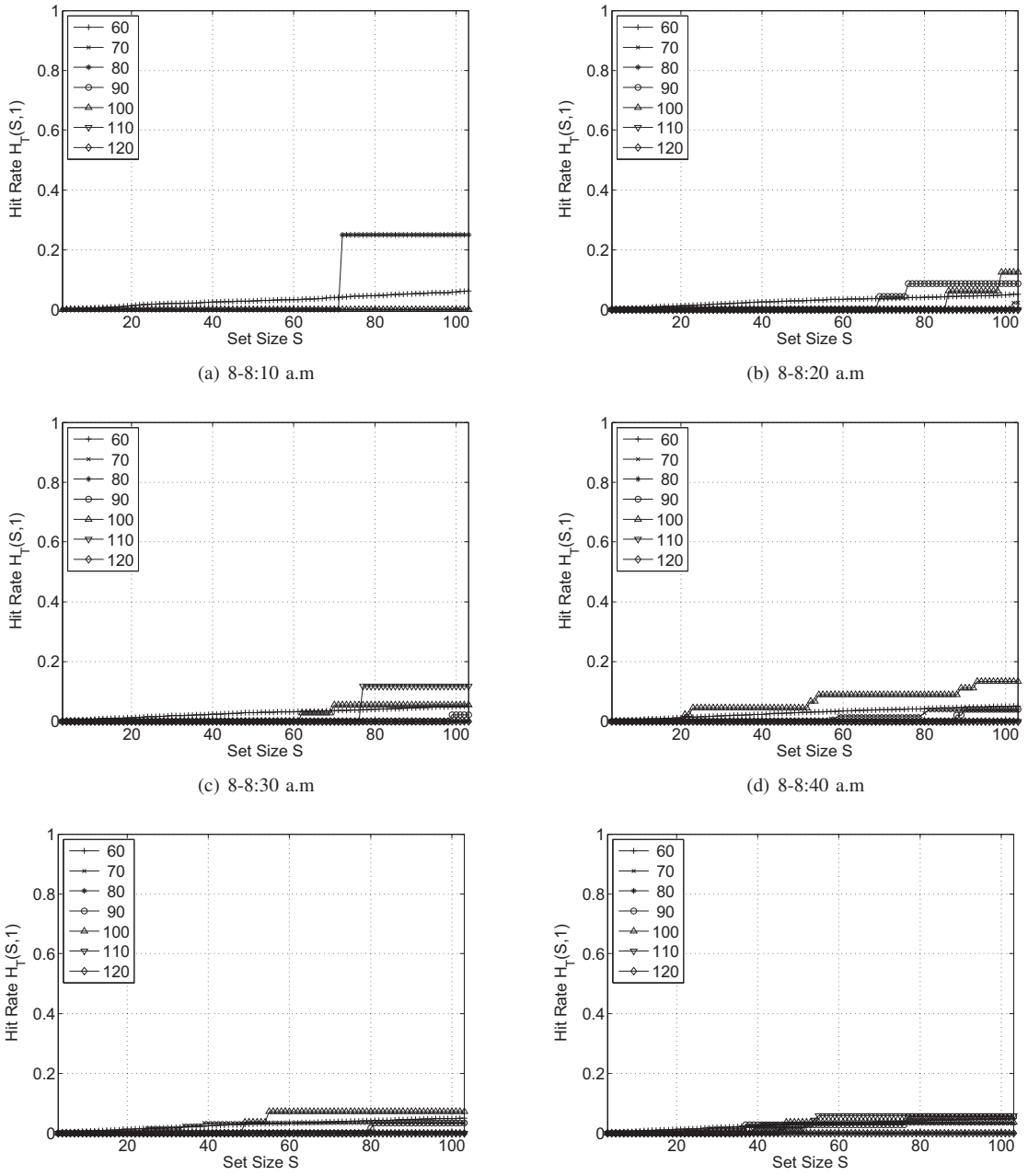
Fourth, judging by the time durations of the data sets (between 10-60 minutes), we conclude that the amount of data necessary to achieve a high hit rate by COLD can be collected in a relatively short period of time. Therefore, COLD does not require an extensive data collection effort to achieve high accuracy.

Finally, we observe that when TSC is applied to all data sets, the hit rate remains almost 0 for vertices of all traffic levels. This leads us to the conclusion that TSC is effectively unable to detect any communication links among users. We attribute this failure to the random phase delay of packet entries in traffic traces of two communicating users. These delays are a result of the bidirectional flow of traffic and jitter in the end-to-end delay.

VI. Evasion and Countermeasures

This section presents some possible techniques that an adversary may utilize to evade the de-anonymization attack by COLD. We also discuss countermeasures to such evasion techniques below.

1) **Evasion by using proxy or NAT.** An adversary may access instant messaging network behind a proxy or a NAT to bypass the detection by the COLD attack algorithm. However, in this scenario, COLD will still detect the external IP address, which appears in the traffic traces



(a) 8-8:10 a.m

(b) 8-8:20 a.m

(c) 8-8:30 a.m

(d) 8-8:40 a.m

(e) 8-8:50 a.m

(f) 8-9 a.m

Fig. 9. Hit rates of TSC for vertices of degree 1 in the (a) 10 minute data set, (b) 20 minute data set, (c) 30 minute data set, (d) 40 minute data set, (e) 50 minute data set, and (f) 60 minute data set.

collected outside the proxy or NAT. Once the external IP address is detected, our proposed approach will require additional traces collected inside the proxy or NAT to specifically pin-point the end-host.

2) **Evasion by IP spoofing.** An adversary may try to spoof source IP address to evade COLD. However, IP spoofing will not be successful because every end-user has to setup a connection with the IM relay server, which is not possible with spoofed IP address.
3) **Evasion by fragmentation/aggregation.** An adversary may try to break-down a large message into multiple smaller messages. However, fragmentation at the end-host into smaller packets will not adversely affect COLD because our approach relies on correlating the traffic traces that are collected entering and leaving the IM service. The smaller packets created due to fragmentation will appear the same in both sets of traffic traces. In fact, the increased number of packets would improve COLD's accuracy. On the other hand, an adversary may try to aggregate as many messages as possible into a single message to minimize the data available. However, the maximum packet size is limited by the IM service provider and maximum transmission unit (MTU) of the network.

4) **Evasion by changing packet sizes.** If an adversary tries to deliberately change packet sizes, *e.g.* by inserting garbage, they will appear the same in the two sets of traffic traces correlated by COLD. Therefore, changing packets sizes will not affect COLD.
5) **Evasion by random delays.** Adversaries may also add random small or long delays between their communications. The time delays introduced by end-host will not affect COLD because these delays appear the same in the two sets of traffic traces. In another scenario, random delays may be introduced by the IM network due to network congestion or other processing delays. These delays will affect COLD because they will be different across the two correlated traffic traces. However, COLD is robust to such delays as well because it utilizes binning techniques, which reduces their effect.
6) **Evasion by injecting noise packets.** Injecting random noise packets is unlikely to affect the accuracy of COLD as long as the noise packets follow the protocol utilized by the IM network. Packets that do not follow the protocol utilized by the IM network will be discarded by the IM network after sanity checks and will not appear in the second traffic trace collecting traffic exiting the IM network. To mitigate the effect of such noise packets, similar sanity checks can be deployed to check if the logged packets follow the protocol utilized by the IM network under study.
7) **Evasion by encryption.** Encryption is only applicable to the packet payloads and packet headers remain unaffected. The use of encryption cannot evade COLD because our proposed approach only utilizes fields in the packet header.

VII. Conclusions

In this paper, we present a novel attack to breach the privacy of IM communication services that allows an attacker to infer who's talking to whom with high accuracy. We proposed a wavelet-based scheme, called COLD, that allows us to examine and compare the time series of one-way (user-server) traffic logs at multiple scales. We evaluated the COLD attack algorithm using a real-world Yahoo! Messenger data set, which was specifically collected for this study. Our experimental results showed that COLD clearly outperforms the baseline time series correlation scheme.

Our proposed approach can also be applied to the related problems such as mix network de-anonymization. In the mix network de-anonymization problem, a set of mix servers can be treated as the black box and the traffic logs at the edges of the mix network can be correlated using COLD to detect communication links among end-users [18], [24].

Acknowledgement

The authors would like to thank Donald McGillen, Joy Ghosh, Andrew Large, and Kim Capps-Tanaka from Yahoo! Labs for collecting and providing the Yahoo! messenger traffic traces used in this study. This work is partially supported by the National Science Foundation under Grant Numbers IIS-0968495, CNS-0845513, CNS-1017588, and CNS-1017598, the National ICT R&D Fund of Pakistan, and the National Natural Science Foundation of China under Grant Number 61272546.

References

[1] Yahoo! network flows data, version 1.0. Yahoo! Research Webscope Data Sets.

[2] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *Recent Advances in Intrusion Detection*, 2010.

[3] B. Claise. Cisco systems NetFlow services export version 9. Wikipedia, the free encyclopedia, October 2004.

[4] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[5] R. Coifman and M. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2 Part 2):713–718, 1992.

[6] R. Heatherly, M. Kantarcioglu, and B. M. Thuraisingham. Preventing private information inference attacks on social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2012.

[7] X. Li. Informational cascades in IT adoption. *Communications of the ACM*, 47(4), 2004.

[8] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the12th International Conference on Information and Knowledge Management*, pages 556–559, New York, NY, USA, 2003. ACM.

[9] W. Lu and A. A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing*, 2009.

[10] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.

[11] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris. Detecting price and search discrimination on the internet. In *HotNets*, 2012.

[12] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2010.

[13] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[14] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[15] A. M. Odlyzko. Privacy, economics, and price discrimination on the internet. In *Fifth International Conference on Electronic Commerce (ICEC)*, 2003.

[16] E. Pariser. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think [Paperback]*. Penguin Books, 2012.

[17] C. Troncoso and G. Danezis. The Bayesian traffic analysis of mix networks. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[18] M.-H. W. V. Shmatikov. Timing analysis in low-latency mix networks: Attacks and defenses. In *European Symposium on Research in Computer Security (ESORICS)*, 2006.

[19] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[20] W.-S. Yang, J.-B. Dia, H.-C. Cheng, and H.-T. Lin. Mining social networks for targeted advertising. In *39th Annual Hawaii International Conference on System Sciences (HICSS)*, 2006.

[21] Y. Zhang, Z. Wang, and C. Xia. Identifying key users for targeted marketing by mining online social network. In *Advanced Information Networking and Applications Workshops (WAINA)*, 2010.

[22] E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *World Wide Web (WWW) Conference*, 2009.

[23] Y. Zhu, X. Fu, R. Bettati, and W. Zhao. Anonymity analysis of mix networks against flow-correlation attacks. In *IEEE Global Communications Conference (GLOBECOM)*, 2005.

[24] Y. Zhu, X. Fu, B. Gramham, R. Bettati, and W. Zhao. Correlation-based traffic analysis attacks on anonymity networks. *IEEE Transactions on Parallel and Distributed Systems*, 2009.