Encryption Is Not Enough: Inferring user activities on KakaoTalk with traffic analysis

Kyungwon Park and Hyoungshick Kim

Department of Computer Science and Engineering,
Sungkyunkwan University, Korea
{kyungwon,hyoung}@skku.edu

Abstract. Many people started being concerned about their privacy in delivering private chats, photographs, contacts and other personal information through mobile instant messaging services. Fortunately, in the majority of mobile instant messaging services, encrypted communication channels (e.g., using the SSL/TLS protocols) are used by default to protect delivered messages against eavesdropping attacks. In this paper, however, we show that encryption is not enough. For example, in a real world service named KakaoTalk, many users' online activities can effectively be identified with 99.7% accuracy even though traffic is encrypted. We present a practical traffic analysis attack using a supervised machine learning technique.

Keywords: Traffic analysis, Traffic classification, Mobile instant messenger, KakaoTalk

1 Introduction

The popularity of mobile instant messaging services has grown immensely over the past few years. The majority of smartphone users today are using a mobile instant messenger as their main communication tool rather than phone calls. However, as mobile instant messaging services have become the most dominant communication medium, many users also begins to become concerned about their privacy in using those services because their personal information such as private chats, photographs, contacts and user profiles can be exposed. Therefore, the communications via mobile instant messengers would have become one of the most attractive targets for eavesdroppers. For example, according to the Snowden's leaks [10], some government agencies have tracked users' online activities by intercepting their communications and eavesdropping them. Therefore, for user privacy, it is necessary to secure the communication channels between an instant messenger server and its clients. Fortunately, the majority of mobile instant messaging services have already provided a secure and authenticated communication channel (e.g., using the SSL/TLS protocols) by default to protect their users from eavesdroppers.

In practice, however, payload encryption is not enough; we can see that encrypted traffic can be vulnerable to traffic analysis attacks in many applications. Conti et al. [3] presented a framework to classify user activities on

online social networks and email applications for smartphone, such as Facebook, Twitter, and Gmail by analyzing their encrypted traffic. We extend their framework to the traffic analysis of a mobile instant messing service named KakaoTalk (`http://www.kakao.com/talk/en`) which is the most widely used instant messaging service in Korea. According to a recent report [1], the number of KakaoTalk users had surpassed about 48 million global users. Although Coull and Dyer [4] already showed that several mobile instant messaging services are vulnerable to traffic analysis attacks, they only considered coarse-grained classification for detecting message types (e.g., control, image, and text) at high level. Unlike the previous study, we presented a classification method to infer users' detailed online activities in KakaoTalk even with encrypted traffic. Our key contributions can be summarized as follows:

- We proposed a framework to infer users' online activities on a mobile instant messaging service. The proposed framework was implemented with a supervised machine learning technique based on a hierarchical clustering to classify users' online activities.
- We evaluated the accuracy of the proposed inference attacks with a mobile instant messing service named KakaoTalk. The experimental results showed that the proposed classification method (with 986 clusters) is capable of achieving about 99.7% accuracy.

The rest of this paper is organized as follows. In Section 2, we define the threat model in this work and introduce the important user activities on KakaoTalk. Then, we present the proposed framework for inference attacks with traffic analysis in Section 3. In Section 4, we analyze the experimental results about the accuracy of the proposed inference attacks. In Section 5, we suggest some reasonable defense methods to mitigate such inference attacks with traffic analysis. Some related work is discussed in Section 7. Finally, we conclude in Section 8.

2 Inferring User Activities on KakaoTalk

Previously, instant messengers were only used for exchanging text messages with other people. However, the latest instant messaging services have also served as a platform for voice calling, exchanging photographs or other files, social networking and even distributing games.

For using those features, users should send control commands and/or data via Internet. In this section, we first define the attackers' goals under the given assumptions about their capabilities and then briefly introduce the important user activities on KakaoTalk.

2.1 Threat Model

Attacker's Capabilities We consider a passive attacker (e.g., a government agency) who can monitor traffic between users' mobile devices and the messaging

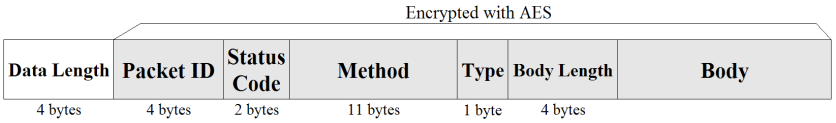


Fig. 1. LOCO packet structure

service provider. We assume end-hosts are trusted and not under the control of the attacker. However, the attacker can capture the network traffic and store it for future use (e.g., analyzing its characteristics in an offline manner).

KakaoTalk uses a proprietary protocol over TCP/IP (called `LOCO`). Each packet is encrypted with AES [5] after a handshake between client and server (see Figure 1). As for the encrypted traffic, we assume a computationally bounded adversary running in a polynomial time, and is not capable of breaking the encryption algorithm without knowing the key(s) used for packet encryption; this assumption is reasonable since breaking advanced encryption algorithms such as AES is computationally infeasible for the most powerful supercomputers.

Attacker's Goal The attacker's goal is to infer a user's online activities on KakaoTalk by analyzing the network packets and communication patterns between the KakaoTalk messenger server and the KakaoTalk application running on the user's mobile device — users' private activities can be monitored. For example, a user's stalking behaviors (i.e., viewing other users' profiles) can be uncovered. The effectiveness of an inference attack is evaluated with the results of a multi-class classification for users' activities. In the next subsection, we present the target user activities on KokaoTalk in detail.

2.2 User Activities on KakaoTalk

KakaoTalk has many useful features. For example, a user sets his (or her) profile (and status message), views friends' profiles, and registers new friends based on either their KakaoTalk IDs or phone numbers [7, 8]. Especially, the automatic friend registration feature can be misused for enumeration attacks [7, 8].

Typical user activities on KakaoTalk are relevant to (1) communicating with others (including friends); (2) managing friends, and (3) customizing the user's application. In this paper, we are only interested in identifying some sensitive user activities in (1) and (2) categories because those activities have dominated in KakaoTalk users' actual behaviors, compared with (3). We briefly explain each user activity, respectively, that we looked at (see Table 1). The first four activities ("`Receive a message`", "`Join a chat room`", "`Leave the chat room`", and "`Send a message`") are in the category (1); and all the remaining activities are in the category (2).

Table 1. User activities on KakaoTalk that we considered in the inference attacks

Activity	Description
Join a chat room	Join a chat room from the list of chat rooms.
Leave the chat room	Leave the chat room that you joined first.
Receive a message	Receive a message such as text, photograph and emoticon from another user.
Send a message	Send a message such as text, photograph and emoticon to others.
Add a friend	Add a new user into the friend list.
Hide a friend	Hide a specific friend from the friend list. After hiding the friend, the friend is not displayed in the friend list anymore.
Block a user	Block a specific user. After blocking the user, the blocked user cannot send (and receive) messages to (from) the requesting user.
Unblock a blocked user	Unblock a specific blocked user from the list of blocked users. After unblocking the user, the unblocked user can send (and receive) messages to (from) the requesting user.
Re-add a blocked friend	Re-add a blocked friend into the friend list again. When a blocked friend is unblocked, this option is available.
View a user's profile	View a specific user's profile (i.e., the user's status message and profile picture).
Synchronize friend list	Synchronize the friend list with the requesting user's contacts stored in the mobile device.

3 Inference Attack Framework

In this section, we describe the proposed framework to infer users' activities on KakaoTalk with traffic analysis.

The proposed framework first filters out all the packets except the packets for the KakaoTalk application with several filtering rules in order to focus on inferring interesting activities on KakaoTalk (see Section 3.1). Next, given a sequence of packets, our problem can be reduced to a multiclass classification problem for identifying a KakaoTalk user' activity with the sequence of packets (see Section 3.2).

3.1 Traffic Filtering

Collecting Packets with IP Address We found that the KakaoTalk servers have used four fixed IP addresses for the messaging service. Therefore, we can selectively capture only the desired packets for the KakaoTalk application with their IP addresses.

Filtering out Unwanted Packets As a next step, we carefully filtered out unnecessary packets such as acknowledgement and retransmitted packets; acknowledgement packets have a zero-length payload; and damaged packets (usually initiated by a time-out) are re-transmitted. We also need to particularly consider sending a large file (e.g., when a multimedia message is delivered); the large file must be broken up into several pieces since TCP protocol used on Ethernet limits the MSS (Maximum Segment Size) to 1460 bytes by default; however, even when more than one packet is generated per message, those packets are only relevant to a user's activity. Therefore, we regard a sequence of consecutive packets whose size is 1460 bytes as a single packet of 1460 bytes.

Separating Sequences of Packets Since an adversary cannot exactly identify when a sequence of packets per user's activity on KakaoTalk is started, it is also challenging to determine a reasonable time threshold for separating packet sequences of different user activities. In general, if the arrival time of two subsequent packets is larger, the respective packets are considered as packets for different user activities. Therefore, separating sequences from the captured traffic is equivalent to setting cuts in the packet sequence and aggregating all packets which are within these borders to one sequence. According to the observation in the previous study [13], we selected a threshold of 4.43 seconds as the optimal parameter value for separating packet sequences of different user activities.

3.2 Classification of User Activities

For inferring user activities with packet sequences, we use a supervised learning method that requires a labeled training dataset; each sample of the dataset is labeled as a user activity.

Interestingly, whenever we repeatedly performed the same user activity on KakaoTalk, the generated packet sequences were not always identical. That is, a user activity can be mapped to several different packet sequences. To effectively manage such situations, we selected a Random Forest [2] classifier with a tree constructed by a hierarchical clustering method [6].

For efficient grouping of packet sequences generated by the KakaoTalk application, we opted for an agglomerative hierarchical clustering algorithm; similar sequences of packets are grouped together in the same cluster, while dissimilar sequences of packets are assigned to different clusters. Agglomerative hierarchical clustering provides a *nested sequence of clusters* with a single and all-inclusive cluster at the top and single-point clusters at the bottom, unlike partition-based clustering techniques such as k-means. This allows us to control the desired number of clusters based on cluster relationships to maximize the accuracy of the classification algorithm.

The key parameter in agglomerative hierarchical clustering algorithms is the clustering criterion function used to determine the pairs of clusters to be merged at each step. In most agglomerative algorithms, this is accomplished by selecting the most similar (or closest) pair of clusters. Many cluster selection schemes have

been developed for computing the similarity between clusters. They mainly differ in how they update the similarity between existing and merged clusters. For example, the single linkage scheme [9] measures the distance of two clusters by computing the shortest distance between objects in the two clusters while the complete linkage clustering scheme uses the maximum distance between objects in the two clusters. In this paper, we used the average linkage scheme which is widely used in traffic analysis [3]. That is, the distance between two clusters A and B is given by

$$d(A,B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} distance(a,b)$$

where A and B are clusters in a dataset.

In order to calculate the $distance(a,b)$ between two sequences of packets $a \in A$ and $b \in B$ for two clusters A and B, we used Dynamic Time Warping (DTW) [12] which is a well-known technique to measure the similarity between time series objects.

In order to speed up the classification process, the representative object of each cluster called *leader* is selected. Given a cluster A containing the sequences of packets $\{a_1, \cdots, a_n\}$, the leader is elected by selecting the sequence a_i that has minimum overall distance from the other objects of the cluster which is defined as:

$$leader(A) = \underset{a_i \in A}{\operatorname{argmin}} \Big(\sum_{a_j \in A} distance(a_i, a_j) \Big)$$

After electing a leader for each cluster, we executed the Random Forest [2] algorithm with labeled leaders. For a given sequence s, we try to find the cluster C that minimizes the distance between s and the leader of the cluster $leader(C)$ and identify the user activity for s as the label of $leader(C)$.

4 Experiments

We present how the experiments were taken in Section 4.1 and evaluate the performance of our implementation in Section 4.2.

In order to analyze the network traffic for the KakaoTalk application running on an Android device, we used a PC (with running 64-bit Ubuntu) connected to the same network by configuring it as a NAT router and tried to capture all network traffic from the KakaoTalk application on the Android device by using the libpcap library (see Figure 2). In our experiments, we used the KakaoTalk v4.8.4 running on the Android 4.1.2 version, and equipped with a non-congested 100 Mbit/s WiFi connection to a LAN that was connected to the Internet.

4.1 Dataset Collection

In June 2015, during the course of a day, we collected 1100 objects of packet sequences by repeatedly simulating 11 KakaoTalk activities (100 per activity) that we are interested in (see Table 1 in Section 2.2).

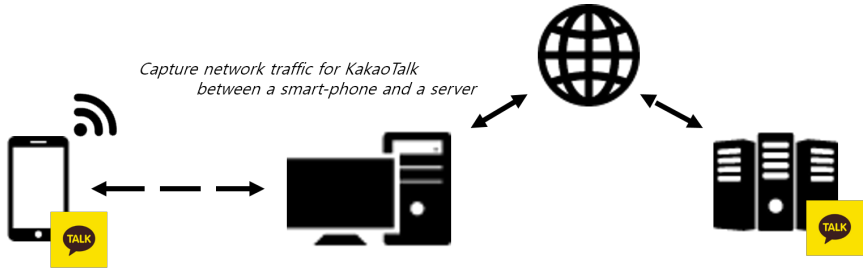
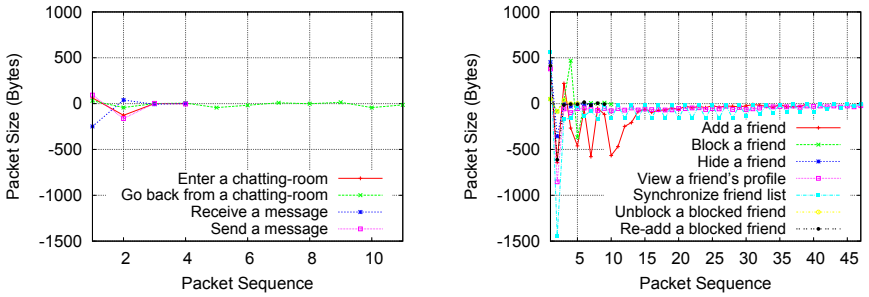


Fig. 2. Experimental environment



(a) Communicating with others (b) Managing friends

Fig. 3. Packet sequence of behaviors

In the proposed framework, the simulated user activities were generated in a random manner to reduce the bias associated with specific data. For example, when we mimic *sending a message* to a friend, a friend is randomly selected and then a randomly generated (short) message is sent to the friend.

In most of user activities, we can see nearly unique patterns of packet sequences by analyzing the sizes of packets in sequence from the captured traffic dataset. Figure 3 demonstrates the average packet size of kth packets for each user activity. The results were presented by the two categories of "communicating with others" and "managing friends" (see Table 1 in Section 2.2). In those figures, the y-axis represents the packet size; positive values represent outgoing packets while negative values represent incoming packets.

Through the analysis of those patterns, we can find some interesting observations. For example, unlike the other user activities, a packet sequence for "`Receive a message`" has a negative integer at the first packet in the session (see Figure 3(a)). In fact, it is obvious that this activity is triggered by an incoming packet. Also, we can see that the activities for "communicating with others" are completed within at most 12 packets; this short-term session behavior is totally different from some activities in the category of "managing friends", such as "`View a user's profile`" and "`Synchronize friend list`" which typically require more than 40 packet interactions on average (see Figure 3(b)).

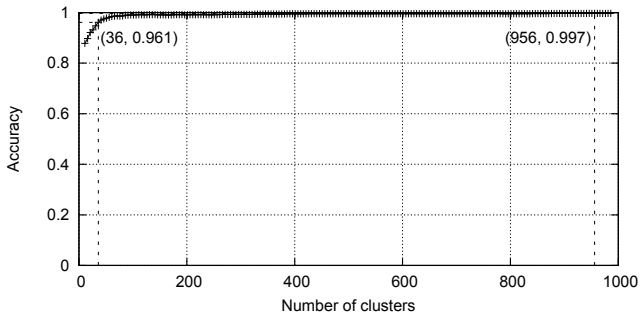


Fig. 4. Classification results with number of clusters

4.2 Classification Results

To evaluate the performance of the proposed classification method in Section 3, we use the *accuracy* measure that is the proportion of correctly classified activities. In general, the accuracy measure is acceptable if the test data is balanced.

For classification, we used a 10-fold cross-validation where the training samples are partitioned into 10 equal-sized blocks and each block in turn is then used as test data for the classifier generated from the remaining nine blocks. We analyzed the performance of the proposed classification method with varying the number of clusters from 11 to 990. This is because we consider the only 11 activities (see Section 2.2) and the 90% of 1100 samples is 990. Figure 4 shows the accuracy of the proposed classification method with number of clusters.

As we can expect, the accuracy of the proposed classification method overall was improved as the number of clusters increased. We were finally capable of achieving an accuracy of 0.997 with 956 clusters. Using that multiclass classifier with 956 clusters, we analyzed the accuracy results of individual user activities. The results are presented in the confusion matrix (see Figure5; darker the color, higher the accuracy of the classification method for each activity). From looking at the confusion matrix, we can see that all user activities were very accurately identified even though some activities for "`Hide a friend`" were misclassified as '`Receive a message`". For more detailed analysis, in addition to the accuracy results, we analyzed the other classification measures (*precision*, *recall*, and *F-measure*). For each activity i, those measures are defined as:

- **Precision**: the proportion of activities classified as activity i that actually are i activities;
- **Recall**: the proportion of i activities that were accurately classified;
- **F-measure**: the harmonic mean of *Precision* and *Recall*; (2**Precision***Recall*) / (*Precision*+*Recall*).

The evaluation results are shown in Table 2. We can still see that all activities could be classified well even though we achieved a relatively low precision and F-measure for "`Hide a friend`".

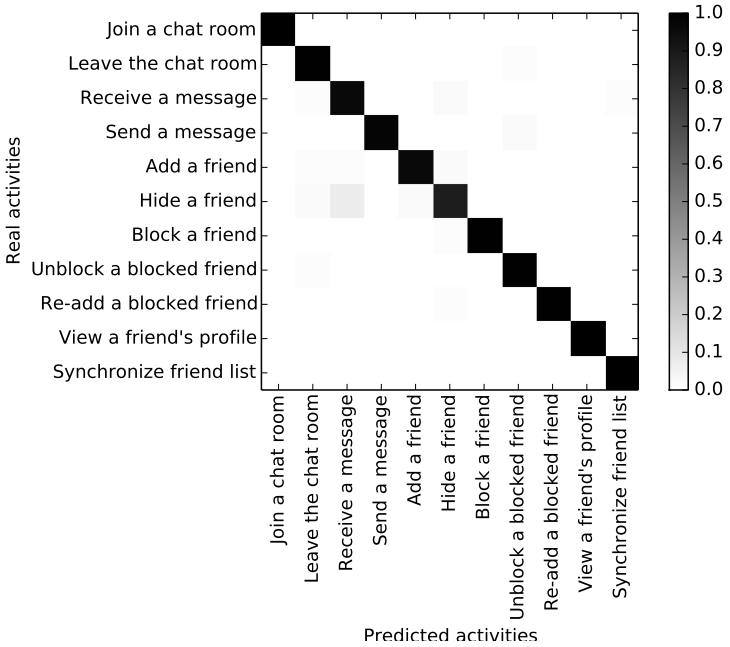


Fig. 5. Confusion matrix in 956 clusters

Table 2. Classification accuracy with 956 clusters

Behavior	Precision	Recall	F-measure	Accuracy
Join a chat room	1.000	1.000	1.000	1.000
Leave the chat room	0.990	0.952	0.971	0.995
Receive a message	0.960	0.923	0.941	0.989
Send a message	0.980	1.000	0.990	0.998
Add a friend	0.960	0.980	0.970	0.995
Hide a friend	**0.880**	0.931	**0.905**	0.984
Block a friend	0.990	1.000	0.995	0.999
Unblock a blocked friend	0.990	0.971	0.980	0.996
Re-add a blocked friend	0.990	1.000	0.995	0.999
View a friend's profile	1.000	1.000	1.000	1.000
Synchronize friend list	1.000	0.990	0.995	0.999

Although the multiclass classifier with 956 clusters produced the best classification results, it might be desirable to classify packet sequences with only a few clusters in terms of efficiency since the computation time of the classification increases (linearly) with the number of clusters used in the proposed method. Fortunately, we can see that an accuracy of 0.961 can still be achieved with a small number of clusters (e.g., 36 clusters), which is shown in Figure 4.

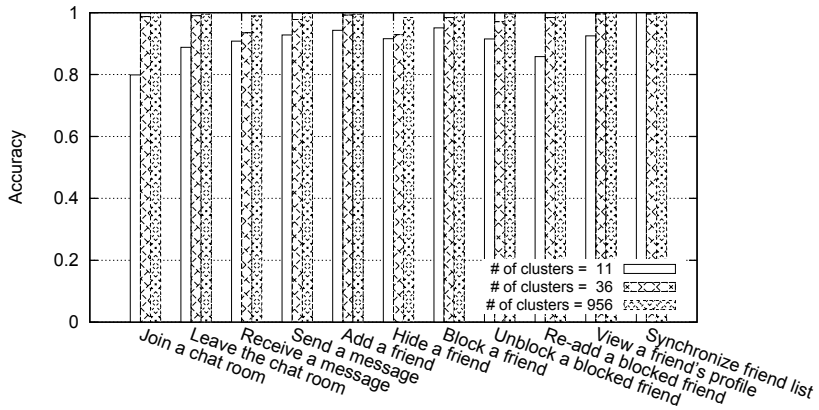


Fig. 6. Accuracy of the classification with 11, 36, and 956 clusters, respectively

To demonstrate the effects of the number of clusters for each activity, we analyzed the accuracy results with 11, 36, and 956 clusters, respectively. The results were shown in Figure 6. With 11 clusters only, some activities such as "`Join a chat room`" and "`Re-add a blocked friend`" could be misclassified with a significant rate.

5 Countermeasures

The proposed framework can be used to infer user activities even with encrypted traffic. The key idea is to use an activity's network characteristics (e.g., number of packets, size of each packet, directions of packets in the network session) as its signature. With this idea, we proved that a popular instant messaging service named KakaoTalk could be vulnerable to our traffic analysis attacks.

In practice, however, preventing such traffic analysis is not an easy problem. Most conventional protocols have been carefully designed to minimize communication overhead by adjusting packet flows.

To mitigate such traffic analysis, the simplest defense solution is to use a proper padding scheme for fixed size packets. Also, dummy packets might be inserted so that packet sequences have nearly identical patterns. However, those solutions will inherently introduce additional communication overhead. Those techniques should be treated with caution: the communication cost they impose on a system is likely to be nontrivial, and they may indeed be unacceptable in competitive markets.

6 Limitation

The primary aim of this work was to demonstrate the feasibility of using network characteristics in a session for identifying user activities. However, some of

the features (e.g., the time intervals between packets), which could significantly improve the accuracy, have not been considered in depth.

In evaluating the effectiveness of the proposed framework, we simulated user activities to collect labeled ground truth network traffic rather than using the actual users' packets. However, those stimulated user activities might significantly different from real users' activity patterns in several aspects. For example, when a large file is used for "`Send a message`", the number of packets and/or the size of each packet can be greatly increased. Also, we do not consider various message types such as text, emoticon, and image.

Finally, some activities might be more frequently appeared than the other activities in practice. In our experiments, however, we simply assumed that all user activities would be expected to occur equally. The performance of classification methods can typically be changed with a different distribution of samples.

7 Related Work

In conventional network environments, it has been already studied to identify online user activities. Zhang et al. [14] showed the risk of traffic analysis by identifying user applications with an accuracy of over 90% if the eavesdropping lasts for 1 minutes.

Because of the increased use of smartphones, it is also challenging to perform traffic analysis with applications on smartphones. Lee et al. [11] studied the characteristic of smartphone traffic by particularly analyzing the proportion of smartphone traffic from the entire network traffic in a campus network. Their study analyzed the traffic proportion of most popular 50 smartphone applications and found that KakaoTalk was the third ranked in the experiment.

Stöber et al. [13] also proposed a traffic analysis technique to identify specific traffic for a smartphone application. In their work, they showed network traffic generated from the most popular applications (Facebook, WhatsApp, Skype, Dropbox, and others) can successfully be identified with a probability of 90%. Conti et al. [3] developed a system to identify the specific actions that a user is doing on smartphone applications such as Gmail, Facebook and Twitter by using machine learning techniques. Similarly, Coulle et al. [4] focused on the traffic analysis of encrypted messages for instant messaging services such as WhatsApp, Viber, and Telegram to infer the message types and language used in chatting. Here, we extend their work by developing a framework to identify user activities on the KakaoTalk application in more detail.

8 Conclusion

In this paper, we proposed a framework to infer user activities in KakaoTalk by passively analyzing network traffic. From the experiment, we demonstrated that each activity generates a practically unique packet sequence and those packet sequences can be sufficiently used to infer user activities with 99.7% accuracy by using a supervised machine learning technique.

In future, we plan to evaluate the effectiveness of the proposed technique on a larger scale so as to further extend this attack strategy.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (No. 2014R1A1A1003707), the ITRC (IITP-2015-H8501-15-1008, IITP-2015-R0992-15-1006), and the IITP (2014-044-072-003).

References

1. KakaoTalk: number of monthly active users 2013-2015. `http://www.statista.com/statistics/278846/kakaotalk-monthly-active-users-mau/`, 2015.
2. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
3. Mauro Conti, Luigi V Mancini, Riccardo Spolaor, and Nino Vincenzo Verde. Can't you hear me knocking: Identification of user actions on Android apps via traffic analysis. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015.
4. Scott E Coull and Kevin P Dyer. Traffic Analysis of Encrypted Messaging Services: Apple iMessage and Beyond. *ACM SIGCOMM Computer Communication Review*, 44(5):5–11, 2014.
5. Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer, 2002.
6. Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
7. Eunhyun Kim, Kyungwon Park, Hyoungshick Kim, and Jaeseung Song. I've got your number: Harvesting users' personal data via contacts sync for the kakaotalk messenger. In *Proceedings of the 15th International Workshop on Information Security Applications*, 2014.
8. Eunhyun Kim, Kyungwon Park, Hyoungshick Kim, and Jaeseung Song. Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk. *Computers & Security*, 2015.
9. G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4):373–380, 1967.
10. Susan Landau. Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations. *IEEE Security & Privacy*, 11(4):54–63, 2013.
11. Sang-Woo Lee, Jun-Sang Park, Hyun-Shin Lee, and Myung-Sup Kim. A Study on Smart-phone Traffic Analysis. In *Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium*, 2011.
12. Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
13. Tim Stöber, Mario Frank, Jens Schmitt, and Ivan Martinovic. Who do you sync you are?: Smartphone Fingerprinting via Application Behaviour. In *Proceedings of the 6th ACM conference on Security and privacy in wireless and mobile networks*, 2013.
14. Fan Zhang, Wenbo He, Xue Liu, and Patrick G Bridges. Inferring users' online activities through traffic analysis. In *Proceedings of the 4th ACM conference on Wireless network security*, 2011.