Preventing Wi-Fi Privacy Leakage: A User Behavioral Similarity Approach

Xiuping Han*, Zhi Wang*, and Dan Pei†
*Department of Computer Science and Technology, Graduate School at Shenzhen, Tsinghua University
†Department of Computer Science and Technology, Tsinghua University
Email: {hxp15@mails., wangzhi@sz., peidan@}tsinghua.edu.cn

***Abstract*—Mobile devices adopt probe requests to discover nearby Wi-Fi access points (APs) and set up fast Wi-Fi connections. *Preferred Network Lists* (PNLs) are used to store the lists of connected Wi-Fi APs in the past. Previous studies have shown that such mechanism can lead to serious privacy leakage, for example, attackers can infer users' identity information and movement histories. In this paper, we investigate the privacy issue and propose a data-driven protection strategy. First, we conduct extensive measurement studies based on 27 million users associating with 4 million Wi-Fi APs in 4 cities. We show that probe requests can be used to identify and profile users. Despite that some actions have been taken to reduce privacy leakage (e.g., MAC address randomization), users' PNLs can still be inferred by attackers. Second, we propose a novel privacy protection method, in which users' PNLs are "blurred" by adding *faked* SSIDs generated using a collaborative filtering algorithm, such that nearby users' PNLs are similar to each other. Finally, we evaluate the performance of our design using real-world Wi-Fi association traces. Our trace-driven simulation shows that the refined PNLs can effectively protect user privacy and ensure fast Wi-Fi connection at the same time.**

I. Introduction

In recent years, 802.11 wireless LAN (Wi-Fi) has become a fundamental infrastructure. To enable fast Wi-Fi connectivity, mobile devices maintain Preferred Network Lists (PNLs) that contain Service Set Identifiers (SSIDs) of Wi-Fi hotspots connected to before. In the *active scan mode*, these SSIDs are sent to APs via probe request frames during the Wi-Fi association. According to [1], mobile devices can send as many as 50 probe requests per second in which 98% of the packets contain SSIDs. Attackers can utilize wireless sniffer tools to intercept the emitted probe requests in Wi-Fi channels, thus acquire the SSIDs of users' previous connected APs.

The SSID information in the probe requests can cause serious privacy leakage. First, SSIDs usually contain semantic information that can be used to infer the places a user has been to, e.g., workplaces related SSIDs like "Corp. XXX net", and travel destinations like "HK Airport wifi". Chernyshev et al. claim that 49% of the SSIDs are identifiable and potentially provide some information about the owners of the devices such as past visited locations or even names [2]. Second, previous studies show that user preference [3], user identification [4] and mobility trajectories [5] can be inferred from the combination of SSIDs in their PNLs. In our study, we analyze two large Wi-Fi association record datasets and discover that for 54.03% of the users, up to 50% of SSIDs in their PNLs reveal potential important semantic information.

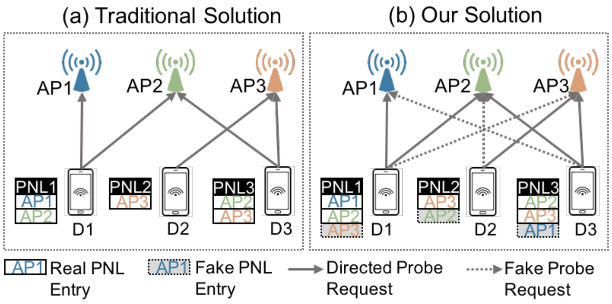


Fig. 1: Illustration of protection strategy.

To protect user privacy in Wi-Fi networks, two types of protection strategies have been commonly adopted. First, reducing the SSIDs sent in probe requests. Bonné et al. [6] designed a strategy to limit the amount of SSIDs emitted from mobile devices. Such approaches usually require modification to the current Wi-Fi protocols [7]. Second, using randomized MAC addresses. For example, iOS 8 uses MAC address randomization in the scan phase such that attachers cannot find the real MAC addresses of the users [8].

The limitation of the previous studies is that users always send real SSIDs in their PNLs via the probe requests (as illustrated in Fig. 1(a)), thus real SSID information can always be obtained by attackers, more or less. An intuitive idea is to *fake* some SSIDs in a user's PNL and let the mobile device broadcast both faked SSIDs and real SSIDs in probe requests (as illustrated in Fig. 1(b)), such that an attacker will not be able to differentiate these SSIDs.

Though the idea seems simple, the challenges for its design and practical implementation are as follows. i) How do we generate the new SSIDs to refine one's PNL, from the large space of millions of valid SSIDs? ii) How do we refine PNLs for users who move across different locations?

Our answers to these questions are a set of strategies designed to refine users' PNLs. Our contributions can be summarized as follows.

- We carry out large-scale measurements to study the privacy issue caused by probe requests and semantic SSID information. Our measurement insights are as follows. (1) As many as 90% of the users have unique SSID sets leaked from their probe requests, indicating that a large fraction of Wi-Fi users can be "identified" by attackers. (2) Users whose PNLs are similar to nearby users' are less likely to be identified. (3) Users whose PNLs are similar are usually located close to each other, indicating that referring SSIDs from such nearby users to refine the PNL in our design is promising.

- Based on our measurement insights, we propose to add faked SSIDs to users' PNLs according to users' behavioral similarity of Wi-Fi association to maximize the PNL similarity between a user and nearby users. Mobile devices send out both real and faked SSIDs, which disguises users' profiles and protects user privacy.
- We propose to use a collaborative filtering (CF) based algorithm, by "recommending" unconnected SSIDs to be added from different locations over time, in a sense that the PNLs of users who are similar to the user will be referred. Our experiments show that the refined PNLs protect user privacy.

The rest of the paper is organized as follows. We review related works in Section II. In Section III we show the measurement results on privacy leakage in probe requests and the feasibility of our design. Section IV describes our behavioral similarity based PNL refinement design. We verify its effectiveness in Section V. Section VI concludes the paper.

II. Related Work

A. SSID Information Leakage

In recent years numerous researches have been conducted to analyze multiple aspects of SSID information leakage, such as privacy, social networks, human behaviors, etc. The usage of information extracted from SSIDs includes identifying user devices [4], [9], inferring social relationships [5], [10], profiling user preferences [3], etc. Although these studies shed some light on the applications and analysis of SSID information, they fail to propose any protection strategies. Sniffing SSIDs can still be used to launch several attacks against users.

B. Current Protection Strategy

Regular privacy protection solutions are not suitable for this problem, for example, k-anonymity [11], in which SSIDs as attributes cannot be deleted or generalized. To solve this specific privacy leakage problem, several previous efforts have been devoted to study Wi-Fi probe requests and the protection. Bram Bonné et al. [6] design a system to prevent smartphones from sending out SSIDs that are out of range when smartphones are in deep sleep mode. Lindqvist et al. [7] propose a new AP discovery protocol by adopting cryptographic challenge-responses on top of probe requests, which however incurs significant cost.

Besides, several vendors have implemented MAC address randomization. iOS 9 adds MAC address randomization to its devices during the scan phase [12]. Android 6.0 uses randomization for background scans if the driver and hardware support it [13]. Microsoft and Linux also support randomization [8]. However, they all implement their own variants of MAC address randomization since a specification on MAC address randomization does not yet exist, which raises the question whether their implementations actually guarantee privacy. Meanwhile, even with MAC address randomization, mobile phones still emit probe request frames with real SSIDs and real MAC addresses under some specific situations [8].

III. Measurement studies

A. Dataset

We base our measurement studies on two datasets, a Wi-Fi connection record dataset and a point of interest (PoI) dataset, collected by *Tencent Wi-Fi Manager*, a crowdsourced Wi-Fi association App, in 4 metropolises of China in one month. The App is used to help users to associate to nearby Wi-Fi hotspots and record the detailed information of each association session, including anonymous user ID, SSID, BSSID, etc. Since PNLs are simply the lists of previously connected Wi-Fi, they can be represented using the SSIDs collected by the App. In this paper, we use the records of connected Wi-Fi in one month as users' PNLs. We use the records of one day as the SSID sets that attackers may obtain, since the size of these SSID sets matches that of real-world SSID sets collected from probe requests [2], [10]. It is possible to collect the real SSID set, but it is hard to get the entire PNL list, which is needed in the paper. Thus SSID sets are generated from real PNL lists instead. In our study, the length of PNLs is within 20 and the size of SSID sets is within 4 for over 97% users.

The Wi-Fi connection dataset contains 250 million Wi-Fi association session records of 27 million users. The PoI dataset contains the location information of 4 million APs, including the longitude, latitude, specific street address and location type. There are totally 16 location types in our dataset, such as hospitals, shopping districts, hotels, etc. In this paper, we use the number of SSID location types in PNLs to represent the profiling degree of users [3].

B. Motivation

Users can be profiled and identified by probed SSIDs, which is verified by previous studies. Based on the measurements, we find that over 90% users contain no more than 6 location types in their PNLs, and 54.90% users only contain one or two location types, which indicates that users' profiles are lack of diversity and these locations in their PNLs usually correspond to home and work locations [14], thus causing some unexpected dangerous privacy leakage. However if we add more SSIDs with different location types, for example, adding the SSID *GolfClub* to a PNL that contains *Campus, zoo-wifi*, the PNL becomes more diversified and is less likely to infer the user's real preference and identity information. Thus the idea of adding faked SSIDs to users' PNLs is effective to disguise users' profiles.

We analyze the two datasets and try to estimate the privacy leakage caused by the identifier. Since an attacker can only collect probe request frames in a limited area, a user can be "identified" if it has a unique SSID set compared with others in the area. We find that 54.60% of users can be identified

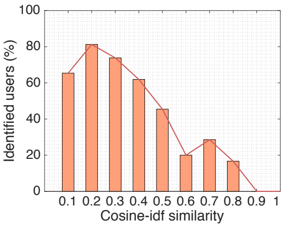


Fig. 2: Similarity between users and their siblings vs. probability that users are identified by probed SSIDs.

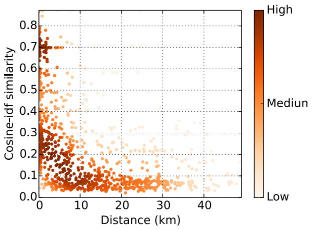


Fig. 3: Relation between distance and similarity between users and their siblings.

only based on the SSID sets that attackers obtain. People leak more SSIDs as time span increases, causing the percentage of identified users to increase to 90.58%.

C. Feasibility Analysis

The leakage of user personal information (e.g. user identification and preference) is due to the differentiation of their PNLs. An intuitive idea is making PNLs on every mobile device identical via adding enough faked SSIDs, thus probing can not leak individuals' preferences or tracks. Nevertheless the number of different APs that can be stored in mobile devices' PNLs is limited and the entries are updated dynamically [15]. Hence the idea above is impractical.

However, we can reduce the difference of users' PNLs instead of eliminating it, which will reduce the possibility of being identified by SSID sets that attackers may obtain. To verify this, we analyze the relation between probability of being identified and similarity of PNLs in Fig. 2. We define an index, Cosine-idf similarity [5], to measure the similarity between PNLs of users, which takes into account both the intersection of PNLs and the popularity of SSIDs. The Cosine-idf similarity is defined as follows:

$$C(S_u, S_v) = \frac{\sum_{s \in S_u \cap S_v} (\log(\frac{1}{f_s}))^2}{\sqrt{\sum_{s \in S_u} (\log(\frac{1}{f_s}))^2} \sqrt{\sum_{s \in S_v} (\log(\frac{1}{f_s}))^2}}, \quad (1)$$

$$f_s = \frac{|U_s|}{|U|}, \quad (2)$$

where S_u is the SSID set (i.e. PNL) of user u, f_s is the popularity of SSID s, $|U|$ is the number of users U in the current area, and $|U_s|$ is the number of users who connected to Wi-Fi with SSID s before. If a user shares common SSIDs with another one, which indicates their similarity is greater than 0, we call them "neighbors". And the neighbor with the highest Cosine-idf similarity is called a "sibling".

In Fig. 2 we focus on the similarity of PNLs between users and their siblings. Users are divided into 10 groups by the similarity of PNLs between them and their siblings, from $[0, 0.1]$ to $(0.9, 1]$. We observe that the relation between probability and similarity is strong since users with high similarity to others are less likely to be identified by probed SSIDs. When the similarity is over 60%, 21.75% of users can be identified. In contrast, when the similarity is below 30%, the probability is 73.48%, i.e., over 3 times higher, which suggests that it is promising to increase the similarity between users' PNLs to reduce the possibility of being identified.

Note that even if two PNLs are the same, the corresponding SSID sets might still be different, since the SSID sets obtained by attackers are only a fraction of the PNLs and may be different each time. Thus it is difficult to measure the extent of information leakage directly. In the following paper, we provide an alternative approach, i.e., use the PNL similarity to represent the extent of privacy leakage. Since the more similar two users' PNLs are, the more likely these two users send out similar SSID sets in the same wireless network environment, and attackers are less likely to distinguish between these two users, i.e., less privacy information are leaked.

Due to the real-world constraint that attackers can only acquire limited SSIDs in real time in a certain location, we utilize the PNL information of users in the current location to refine users' PNLs. In Fig. 3, we present the density plot of the distance and similarity between users and their siblings. When the similarity is larger than 0.6, the distance is within 2 kilometers for over 75% users. We observe that most users who are similar to each other are spatially close, which confirms it is promising to refer faked SSIDs from PNLs of nearby similar users.

Additionally, previous studies claim that the dense-AP coverage and hidden APs are the main reasons for the delay in the active scan, not the length of the PNL [16]. Thus adding faked SSIDs to the PNLs will not affect user experience in Wi-Fi association.

To sum up, most users can be identified using part of their PNLs. The users who have high-similarity PNLs with their neighbors are less likely to be identified and are spatially close to their neighbors. Therefore it is promising to blur a user's PNL using nearby users' PNLs and make them more similar and thus protect their privacy.

IV. Wireless Privacy Protection Strategy

In this section, we first describe the problem of selecting faked SSIDs. Then we propose a CF-based protection algorithm to "recommend" faked SSIDs to be added to a user's PNL based on the relationship among users. The protection strategy is designed to operate on our Wi-Fi association App.

A. The Faked SSID Selection Problem

Our task is to find a set of SSIDs that can be added to a user's PNL and maximize the similarity between this user and its nearby users. The objective can be formulated as follows:

$$\underset{\mathcal{A} \subset S}{\text{maximize}} \quad \sum_{u \in U} \sum_{v \in U \setminus u} \mathit{Sim}(S_u + A_u, S_v + A_v), \quad (3)$$

subject to

$$\mathcal{A} = \bigcup_{u \in U} A_u \quad (4)$$

$$|A_u| \leq k, \forall u \in U \quad (5)$$

where A_u is the faked SSID set added to the PNL of user u. $\mathcal{A}$ is the set of all faked SSID sets and S is the set containing the SSIDs of all the users U. The number of SSIDs that are added to a user's PNL is limited to k to avoid causing too much operation cost and affecting user experience.

To describe the relation between users, we propose to model this problem as a social network. Two users are neighbors and connected if they share at least one common SSID. The neighborhood of a user u can be denoted as N_u^h and includes its neighbors up to h hops in the social network. For example, the 1-hop neighborhood N_u^1 of user u is composed of the users that share at least one common SSID with u, which is hereinafter referred to as "neighbors" if not specifically pointed out. In this paper, the average number of users in N_u^1 is 24.13. And the 2-hop neighborhood is composed of N_u^1 and the users that share common SSIDs with N_u^1.

To measure the similarity between user u and users in its neighborhood, we define *Sim*(S_u, S_v) as follows:

$$Sim(S_u, S_v) = \begin{cases} C(S_u, S_v), & \text{if } v \in N_u^1 \\ Ave(Sim(S_u, S_t) \times C(S_t, S_v)), & \text{otherwise} \end{cases} \tag{6}$$

where t, belonging to N_v^1, is the user who connects user u and user v. If two users are connected directly, the similarity between them is defined as the Cosine-idf similarity, otherwise defined as the average value of the products of similarity between users and user t. To measure the relevance between users and SSIDs, we define the weight $w_{u,s}$ as the importance of SSID s to user u. Based on the *tf-idf* scheme, $w_{u,s}$ is calculated as follows:

$$w_{u,s} = f_{u,s} \times \log(\frac{|U|}{|U_s|}), \tag{7}$$

where $f_{u,s}$ is the frequency that user u connected to Wi-Fi with SSID s. Since SSIDs that are connected frequently are usually important to users, the relevance between a user and a SSID is strong if the user connected to this SSID frequently and the number of users who connected to Wi-Fi with this SSID is small (e.g., the SSID of your home's AP).

B. Faked SSID Selection Based on Collaborative Filtering

Based on the definitions above, we propose a CF-based algorithm to select faked SSIDs from PNLs of nearby similar users. Inspired by the user-based CF, we can increase the similarity between users by "recommending" SSIDs from similar users. We calculate the relevance between users and SSIDs using the *tf-idf* scheme, and define a user's "rate" to a SSID as the *tf-idf* weight. A user can rate a new SSID that he/she has not connected to before, as follows:

$$R_{u,s} = \overline{w_u} + \frac{\sum_{v \in U}(w_{v,s} - \overline{w_v}) * Sim(S_u, S_v)}{\sum_{v \in U} Sim(S_u, S_v)}, \tag{8}$$

where $R_{u,s}$ is the score that user u rates a faked SSID s, and $\overline{w_u}$ is the average score that user u rates its original SSIDs. We take into account both the similarity between users and the importance of SSIDs to users. Important SSIDs, i.e., SSIDs with high $w_{v,s}$ values, are connected by user v frequently recently. These SSIDs are more likely to be sent out by user v, thus they can be added to the PNL of the target user u to increase their similarity. And SSIDs that are less important, are usually replaced by those that are most recently used [15].

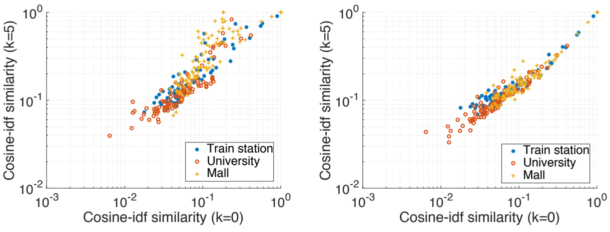
Algorithm 1: PNL Refinement Algorithm

```
Input: $U$, $S_u$ foreach $u \in U$
Output: $\mathcal{A}$
for each $u \in U$ do
    Calculate $w_{u,s}$ foreach $s \in S_u$;
    Calculate $Sim(S_u, S_v)$ foreach $v \in N_u$;
    $L_u = \phi$, $A_u = \phi$;
    $Q.enqueue(u)$, $\Gamma.append(u)$ // BFS;
    while $Q$ is not empty do
        $cur = Q.dequeue()$;
        for each $v \in N_{cur}^1$ & $v \notin \Gamma$ do
            for each $s \in S_v$ & $s \notin S_u$ do
                Calculate $R_{u,s}$ with Equation(8);
                $L_u.append(R_{u,s})$;
            $Q.enqueue(v)$;
        if $|L_u| \geq k$ then
            break;
    if $|L_u| == 0$ then
        $A_u \leftarrow$ the most popular $k$ SSIDs in $S$;
    else
        Sort $L_u$ in descending order;
        $A_u \leftarrow$ top-k SSIDs;
    $P_u^T \leftarrow S_u^{T-1} + A_u$;
$\mathcal{A} = \bigcup_{u \in U} A_u$;
return $\mathcal{A}$
```

The algorithm of refining users' PNLs is presented in Algorithm 1. First it calculates the relevance between users and their SSIDs and the similarity between users. Users rate new SSIDs that they did not connect to before, sort them by descending order and add the top-k SSIDs to their PNLs. We use the breadth-first search to find the top-k faked SSIDs for each user. Here L_u is the list of scores that users rate the new SSIDs, and Γ is the set of users who have been "visited" by the algorithm. We use Q to denote the queue that stores the users to be searched. If the number of new SSIDs from users' direct neighbors N_u^1 is more than k, we will only traverse the PNLs of N_u^1. Otherwise, the PNLs of users' indirect neighbors (i.e. N_u^h, $h > 1$) will be searched until the top-k SSIDs are found.

Besides, if all SSIDs in S_u are unique and not connected by other users, the PNL will not be refined (i.e., the length of L_u is 0) since there are not similar PNLs. For this case, we add the most popular SSIDs in the area to the PNL. The refined PNL P_u of user u at time T is denoted as P_u^T and updated by the original PNL S_u^{T-1} and the added SSIDs A_u.

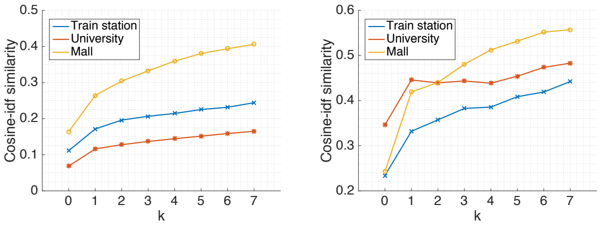
The procedure above describes a single update process. We classify the movements of users into two cases. In the one



(a) Our method.

(b) Random method.

Fig. 4: Comparison with the random selection method.



(a) Neighbors.

(b) Siblings.

Fig. 5: Impact of k on the similarity between users and their neighbors.

case users move from place to place, and in the other case, users are stationary at a place for a period but the contexts constantly change. We assume that users would connect to Wi-Fi initiatively when they arrive at a new place. During the association process, our system will execute the program above to refine PNLs. When users stay at a place and the contexts change, we need to update the PNLs periodically. We do not remove the real SSIDs from users' PNLs, only replace the old faked SSIDs with new faked SSIDs. This process does not affect the Internet connections even when mobile devices have connected to a Wi-Fi network.

In the practical implementation, our design can be implemented following a real-time steaming recommendation architecture [17]. Users' devices send out their PNLs and location updates to a centralized management server, which feeds back the "recommended" faked SSID lists calculated using our algorithm. The new SSIDs will then be added to the devices' PNLs, for example, by editing "/data/misc/wifi/wpa supplicant.conf" file in Android systems. To ensure scalability, the centralized server only updates the recommendation lists periodically.

V. Experiments and Evaluation

A. Experiment Setup

To verify the effectiveness of Algorithm 1, three different areas are chosen as test scenarios: one train station, one university and one shopping mall. We first select the Wi-Fi APs inside these areas, based on their detailed addresses. We verify the selection by comparing the distance between these APs using their latitudes and longitudes and remove the outliers. Similar to the measurement experiments, we use the SSIDs collected in one month as users' PNLs. We randomly select one day within the month and filter users appeared at these areas in that day as our test groups. There are 76, 116 and 119 users in the three locations respectively. These users' PNLs are generated based on the previous records, along with the corresponding SSID sets in that day. We then verify the effectiveness of our method on reducing privacy leakage using these data. We apply Algorithm 1 on the selected data and add faked SSIDs to users' PNLs. The results of PNL similarity improvement are compared to a random selection method. We analyze the impact of the number of added SSIDs k and the length of original PNLs n. Finally we verify the effectiveness of disguising users' profiles.

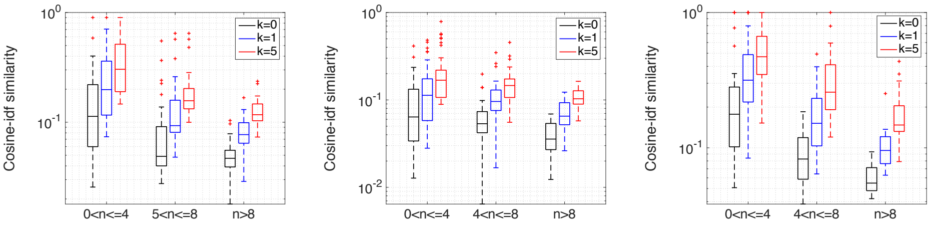
B. Experiment Results

We compare Algorithm 1 with the random selection method in Fig. 4. The x-axis and y-axis represent the similarity when $k = 0$ and $k = 5$ respectively. Note that the x-axis and y-axis are within different ranges. A higher y-axis value indicates a better improvement. We randomly select k SSIDs from the SSID dataset S in a specific location for each user and add them to PNLs in the random selection method. The Cosine-idf similarity between users and their direct neighbors is only improved from 0.11, 0.07, 0.16 to 0.16, 0.11, 0.20 respectively in the train station, university and mall when $k = 5$, indicating the random selection method has little effect on improving the similarity. In contrast, the similarity is improved to 0.23, 0.15, 0.38 respectively using our method.

In Algorithm 1, the number of SSIDs that can be added is constrained by k. Fig. 5(a) shows the different similarities between users and their neighbors over different k values. We observe that the similarity increases as k increases. When k increases from 0 to 1, the similarity increases most significantly. The average similarities are improved by 53.86%, 69.39% and 61.06% respectively in the train station, university and mall. And the average similarities are improved by 119.21%, 140.52%, 147.35% respectively when $k = 7$. The reason why users in the university have the lowest similarity is because they have more neighbors than others. The number of neighbors is 59.81 on average in the university, while 16.32 in the train station and 9.13 in the mall. Although users tend to have a rich number of neighbors in the university, the connections between users and most of their neighbors are weak, resulting in a small average similarity. That is also why the result of Algorithm 1 only has a small improvement compared to the random selection method. In Fig. 5(b) we observe that the similarity between users and their siblings is improved greatly.

Fig. 6 shows the relation between the similarity and the length of users' original real PNLs. Users are divided into 3 groups based on the length of their original PNLs n: $0 < n \leq 4$, $4 < n \leq 8$, $n > 8$. The average similarities between users in the first group originally are 0.19, 0.09, and 0.27 for the three different locations and are improved by 95.57%, 139.17%, and 103.41% respectively when $k = 5$. In the last group, they are improved by 154.86%, 166.75% and 200.97% respectively.

We present the number of location types of SSIDs in users' PNLs to demonstrate the effectiveness of disguising profiles. In



(a) Train station.

(b) University.

(c) Mall.

Fig. 6: Impact of n on the similarity between users and their neighbors.

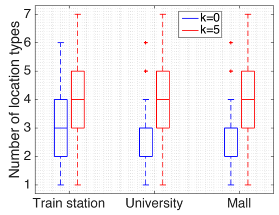


Fig. 7: Number of location types that APs in users' PNLs are related to.

Fig. 7, the numbers of location types are improved from 3.03, 2.70, 2.39 to 4.28, 4.02, 3.13 on average respectively when $k = 5$, indicating that users' profiles are more diversified, thus the original preferences are less likely to be profiled.

In a nutshell, the PNL refinement algorithm succeeds in improving the similarity between PNLs of users' devices in various situations. It is effective to prevent eavesdroppers from stealing real SSID information and breaching user privacy.

VI. Conclusion

During the Wi-Fi association process, user privacy is breached by active probing. In order to understand the threat, we present a detailed analysis of the SSID leakage based on two datasets that contain millions of Wi-Fi association records. To solve this problem, we propose a CF-based heuristic SSID protection algorithm to prevent attackers from inferring users' identification and profiles, according to users' behavioral similarity of Wi-Fi association. The algorithm disguises PNLs of users' mobile devices by adding faked SSIDs instead of limiting the amount of SSIDs or modifying wireless protocols, which incurs less deployment costs. Moreover, the algorithm reduces the possibility of users being identified and profiled by probed SSIDs and thus reduces the extent of privacy leakage. We evaluate the performance of the algorithm under different scenarios and the results confirm its effectiveness. The similarity between users and nearby users is improved by 61.44% even with one faked SSID, which indicates the extent of privacy leakage is significantly reduced.

Acknowledgement

This work is supported in part by the National Basic Research Program of China (973) under Grant No. 2015CB352300, the National Natural Science Foundation of China under Grant No. 61531006, 61472214, 61472210 and U1611461.

References

[1] J. Freudiger, "How talkative is your mobile device?: An experimental study of wi-fi probe requests," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '15, 2015, pp. 8:1–8:6.

[2] M. Chernyshev, C. Valli, and P. Hannay, "On 802.11 access point locatability and named entity recognition in service set identifiers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 584–593, March 2016.

[3] Y. C. Fan, Y. C. Chen, K. C. Tung, K. C. Wu, and A. L. P. Chen, "A framework for enabling user preference profiling through wi-fi logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 592–603, March 2016.

[4] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '07, 2007, pp. 99–110.

[5] M. Cunche, M. A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in wi-fi probe requests," *Pervasive and Mobile Computing*, vol. 11, pp. 56–69, 2014.

[6] B. Bonné, W. Lamotte, P. Quax, and K. Luyten, "Raising awareness on smartphone privacy issues with sasquatch, and solving them with privacypolice," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MOBIQUITOUS '14, 2014, pp. 379–381.

[7] J. Lindqvist, T. Aura, G. Danezis, T. Koponen, A. Myllyniemi, J. Mäki, and M. Roe, "Privacy-preserving 802.11 access-point discovery," in *Proceedings of the Second ACM Conference on Wireless Network Security*, ser. WiSec '09, 2009, pp. 123–130.

[8] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '16, 2016, pp. 413–424.

[9] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices," *Security and Communication Networks*, 2017.

[10] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13, 2013, pp. 265–276.

[11] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[12] K. Skinner and J. Novak, "Privacy and your app," in *Apple Worldwide Dev. Conf. (WWDC)*, June 2015.

[13] "Android 6.0 changes," https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html, 2015.

[14] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '11, 2011, pp. 145–156.

[15] A. D. Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests," in *The 35th Annual IEEE International Conference on Computer Communications*, ser. INFOCOM, April 2016, pp. 1–9.

[16] C. Xu, J. Teng, and W. Jia, "Enabling faster and smoother handoffs in ap-dense 802.11 wireless networks," *Comput. Commun.*, vol. 33, no. 15, pp. 1795–1803, 2010.

[17] Y. Huang, B. Cui, W. Zhang, J. Jiang, and Y. Xu, "Tencentrec: Real-time stream recommendation in practice," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 227–238.