Exploiting Mobile Social Behaviors for Sybil Detection

Kuan Zhang†, Xiaohui Liang‡, Rongxing Lu*, Kan Yang†, and Xuemin (Sherman) Shen†
†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada
‡Department of Computer Science, Dartmouth College, Hanover, NH, USA
*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email:{k52zhang, k62yang, xshen}@bbcr.uwaterloo.ca; xiaohui.liang@dartmouth.edu, rxlu@ntu.edu.sg

***Abstract*—In this paper, we propose a Social-based Mobile Sybil Detection (SMSD) scheme to detect Sybil attackers from their abnormal contacts and pseudonym changing behaviors. Specifically, we first define four levels of Sybil attackers in mobile environments according to their attacking capabilities. We then exploit mobile users' contacts and their pseudonym changing behaviors to distinguish Sybil attackers from normal users. To alleviate the storage and computation burden of mobile users, the cloud server is introduced to store mobile user's contact information and to perform the Sybil detection. Furthermore, we utilize a ring structure associated with mobile user's contact signatures to resist the contact forgery by mobile users and cloud servers. In addition, investigating mobile user's contact distribution and social proximity, we propose a semi-supervised learning with Hidden Markov Model to detect the colluded mobile users. Security analysis demonstrates that the SMSD can resist the Sybil attackers from the defined four levels, and the extensive trace-driven simulation shows that the SMSD can detect these Sybil attackers with high accuracy.**

I. Introduction

Sybil attackers can manipulate an enormous number of identities to profit from services without offering sufficient contribution [1], [2] and compromise the effectiveness of the distributed or peer-to-peer systems [3]. For example, Sybil attackers in online social network (OSN) may spread spam, maliciously mislead the overall popularity, or violate normal user's privacy by forging a large number of fake identities. In mobile social network (MSN), e.g., FireChat, Fon11, Groovr, etc., users directly exchange their social information via mobile devices in the local area or among the crowd. However, Sybil attackers can frequently change pseudonyms to repeatedly broadcast the same/similar information, such as social recommendation, traffic condition, etc. From the perspective of the encountered users, all the similar information seems to be from different senders, which may mislead the encountered user's opinions and preferences. Since mobile Sybil attackers can be merged into the crowd or rapidly move with unpredictable trajectories, it is intractable to detect them in mobile environments.

Extensive research efforts [4], [5], [6] have been put on Sybil detection by using social graph or community detection, while some works [3], [7], [8] investigate the network characteristics, such as wireless channel characteristics, or utilize cryptography to detect Sybil attackers. However, mobile users cannot easily detect Sybil attackers in mobile environments due to some limitations, e.g., the lack of strong social relationships, dynamical-changing mobility, limited detection capabilities, etc. Firstly, it is difficult to establish the social graph for mobile users, due to that: a) mobile users sometimes may not have tight social relationships with others in the physical proximity; and b) their mobility is highly dynamic so that the social connections may not be maintained for a long time. Without a stable social graph, some traditional social-graph based Sybil detection schemes cannot be directly applied in mobile environments. Secondly, mobile Sybil attackers usually act as normal users and merge into the normal user's crowd or social community, which may disrupt the community-based Sybil detection. Thirdly, mobile users have poor knowledge about others, and lack of powerful detection capabilities, such as storage, computation, etc. To alleviate mobile user's resource consumption, one of the promising solutions is to introduce the cloud server to assist mobile users for the data storage and computation. However, as the cloud server may not be fully trusted by the mobile users, critical security and privacy concerns are raised at the same time. Furthermore, the collusion among mobile users would augment the attacker's capabilities and significantly reduce the detection accuracy. Therefore, it is critical to take these challenges into account when developing mobile Sybil detection.

In this paper, we propose a Social-based Mobile Sybil Detection (SMSD) scheme to detect Sybil attackers according to their abnormal contacts and pseudonym changing behaviors in mobile networks. Intuitively, since the Sybil attackers frequently change their pseudonyms to cheat other users, we investigate the contact statistics of the used pseudonyms and detect Sybil attackers by comparing the contact statistics of pseudonyms from normal users and that from Sybil attackers. Due to the limited storage and computation capabilities of mobile users, we utilize cloud servers to store and process the large volume of user's contact information, alleviating the burden of mobile users. The SMSD also addresses the collusion attacks and resists cloud data modification when employing the cloud server for mobile Sybil detection. Specifically, our main contributions are three-fold.

Firstly, we investigate the characteristics of user's mobile social behaviors, i.e., pseudonym changing and contact. Then, we propose a social-based mobile Sybil detection scheme to detect mobile Sybil attackers according to their abnormal pseudonym changing behaviors.

Secondly, to alleviate the mobile user's storage and computation burden, we exploit the cloud server to store and preprocess the user's contact data. With powerful storage and

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

computation capabilities, the cloud servers can assist to detect the Sybil attackers, which significantly reduces the storage and computation overhead of mobile users.

Thirdly, we propose a learning assisted SMSD scheme (LSMSD), i.e., semi-supervised learning with Hidden Markov Model, to resist the collusion of mobile users. The LSMSD can utilize a small number of labeled data for training and adapt to the unlabeled data. We also adopt a ring structure to collect the mobile user's contact signatures associated with the bi-directional Hash chain, which can protect user's contact information (e.g., encountered user, contact time, time order, etc.) from being modified by semi-trusted cloud servers.

The remainder of the paper is organized as follows. We review the related works in Section II. The system model, attacker model and design goals are introduced in Section III. Then, we present the details of the proposed Sybil detection scheme in Section IV, followed by security analysis and performance evaluation in Sections V and VI, respectively. Finally, Section VII concludes the paper.

II. Related Works

Sybil detection schemes have received considerable attentions and been widely studied in recent years [9]. With the "social network" or social graph [10], [11], [12], Yu et al. [4] propose a social network based Sybil detection scheme, named SybilGuard, by using random walk [13] to detect Sybil users. Similarly, SybilLimit [5] provides the near-optimal guarantees by leveraging the intersections on edges rather than vertex (node), and short random routes with multiple independent instances of random walk algorithm. Different from the social network based Sybil detection schemes, Wang et al. [14] study the clicking and browsing patterns of OSN users and observe that the Sybil users have different behaviors compared with the normal users. With these above observations, they develop an unsupervised learning scheme to detect online Sybil attackers. In addition, Wang et al. [15] exploit crowdsourcing and social Turing tests for a distributed Sybil detection scheme.

Due to the limited detection capabilities of mobile users, mobile networks, such as MSNs, and VANET, are vulnerable to Sybil attacks [16]. Quercia et al. [3] enable mobile users to match their communities and detect the ones in the Sybil community. Similarly, Chang et al. [16] address the Sybil detection in MSNs via community matching, where communities are pre-defined and stored by individual mobile users. Using cryptography techniques, Lin [8] proposes an efficient local Sybil resistance (LSR) scheme to mitigate zero-day Sybil vulnerability in vehicular peer-to-peer network. With the group signature technique [17], once the number of a user's signatures on the same event is larger than 2, these signatures can be easily linked by the local users. Then, the vehicle user who generates the signatures would be detected as a Sybil attacker. Liang et al. [7] investigate trustworthiness among mobile users and propose a trustworthy based service evaluation scheme (TSE), which enables the review sharing in service-oriented MSNs and restricts the Sybil attackers' behaviors via group signature techniques. There are also some Sybil detection schemes using the network feature differences between Sybil attackers and normal users, such as channel characteristics [18], mobility [19], [20], and trust [21].

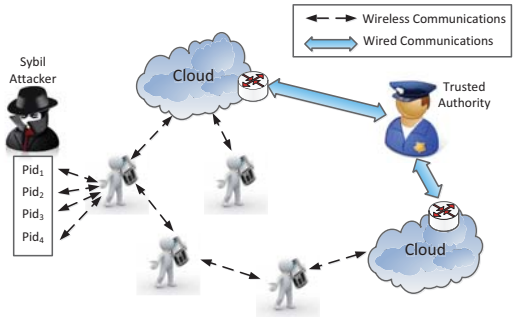


Fig. 1. System Model

In summary, some existing mobile Sybil detection schemes either rely on the pre-defined communities among users, or adopt cryptography techniques to restrict Sybil attackers. However, the Sybil attacker would act similarly as normal users to disrupt mobile Sybil detections. Furthermore, some online Sybil detection schemes cannot be directly applied in the mobile network. To this end, we study the relation between mobile user's contacts and pseudonym changing behaviors and propose the mobile Sybil detection scheme balancing the trade-off between the detection accuracy and overhead. Furthermore, the cloud server is introduced into our detection system to alleviate mobile user's resource consumption.

III. System Model and Design Goal

In this section, we first introduce the system model and attacker model, and then identify the design goals in details.

A. System Model

In our system, there are basically three entities, i.e., trusted authority, mobile users and cloud servers as shown in Fig. 1.

- **Trusted Authority (TA)** bootstraps the whole system, and generates secret keys to mobile users. Furthermore, the TA audits the mobile users' data stored in cloud servers. After a Sybil attacker is detected, the TA can revoke his identities and update the revocation list.
- **Mobile Users** take smart phones or portable communication devices to bi-directionally communicate with each other. User u_i should first register to the TA for identity and secret keys which are used to generate session keys, pseudonyms, and signatures. After the registration, u_i adopts pseudonyms to prevent its real identity from being exposed.
- **Cloud Server (CS)** is a semi-trusted entity in the system. The CS has powerful storage and computing capabilities and is deployed in the local area. Furthermore, the CS can directly communicate with mobile users and collect their data.

B. Attacker Model

According to the Sybil attacker's capabilities, we define Sybil attackers in four levels.

1) *General Sybil Attackers* (Level-1): Sybil attackers, denoted as $\mathcal{A}_s$, exist in mobile environments to compromise the

normal users and launch Sybil attacks to maliciously produce biased information to others [9]. $\mathcal{A}_s$ adopts pseudonyms to hide his real identity u_s and repeatedly sends the similar information or spam to normal user u_i. As a result, u_i would consider all the same information from different senders, and u_i's preference may be manipulated by $\mathcal{A}_s$.

2) *Sybil Attackers with Forged Contact* (Level-2): A Sybil attacker $\mathcal{A}_s$ would forge his contact information to benefit $\mathcal{A}_s$ during Sybil detection. In other words, a large number of contact records could provide the evidences of changing pseudonyms. As such, $\mathcal{A}_s$ would maliciously generate an extensive number of fake contact information associated with his pseudonyms to increase the pseudonym changing frequency. Then, $\mathcal{A}_s$ could disrupt the mobile Sybil detection.

3) *Sybil Attackers with Mobile User's Collusion* (Level-3): Mobile users may collude with Sybil attackers to illegally provide fake contact information and disrupt Sybil detection. The colluded users can generate valid signatures on the inexistent contact with the Sybil attackers, even though they have not met each other. As a result, the Sybil attackers can provide valid contact signatures to others and disrupt the Sybil detection.

4) *Sybil Attackers with Collusion of Cloud Servers* (Level-4): Even though the CS can honestly follow the protocol, but it is a semi-trusted entity. If a CS is compromised or colludes with the Sybil attacker $\mathcal{A}_s$, the CS could either add some fake contact information for $\mathcal{A}_s$, or modify and delete the normal user's contact information to increase the false detection rate.

C. Design Goals

1) *General Mobile Sybil Detection*: When a Level-1 Sybil attack maliciously changes his pseudonyms to launch attacks, the scheme should detect this from his mobile social behaviors.

2) *Unforgeability*: The proposed scheme should be able to prevent attackers from forging the contact information, since the forged contact would disrupt the Sybil detection. The encountered users should exchange unforgeable information (e.g., signatures of the contact) to the other user, and keep the integrity of contact information.

3) *Resistance to Collusion of Mobile Users*: The proposed scheme should be able to resist the collusion of mobile users. It is critical to find out the forged inexistent contacts when mobile users collude.

4) *Resistance to Collusion of Cloud Servers*: The data stored in the cloud server should not be added, modified or deleted by CSs (i.e., Level-4 attackers). The modified data should be detected by other entities, e.g., the TA and mobile users.

IV. The Proposed SMSD Scheme

In this section, we propose the SMSD scheme to detect the four levels of Sybil attackers. Mobile users collect the contact signature from each encountered user, which is used to support the pseudonym changing, as shown in Fig. 2. Collecting the contact information from mobile users, the detector (i.e., the CS) can distinguish Sybil attackers from the normal users according to the abnormal pseudonym changing and contact behaviors. The CS helps to store mobile user's

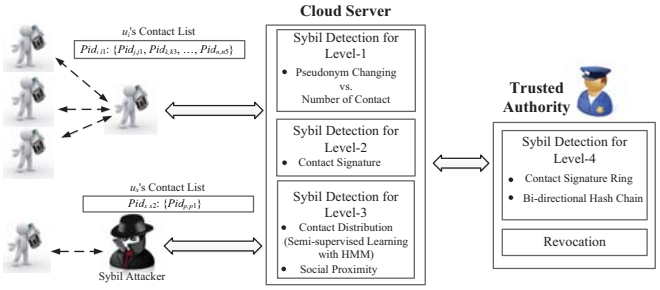


Fig. 2. Overview of SMSD

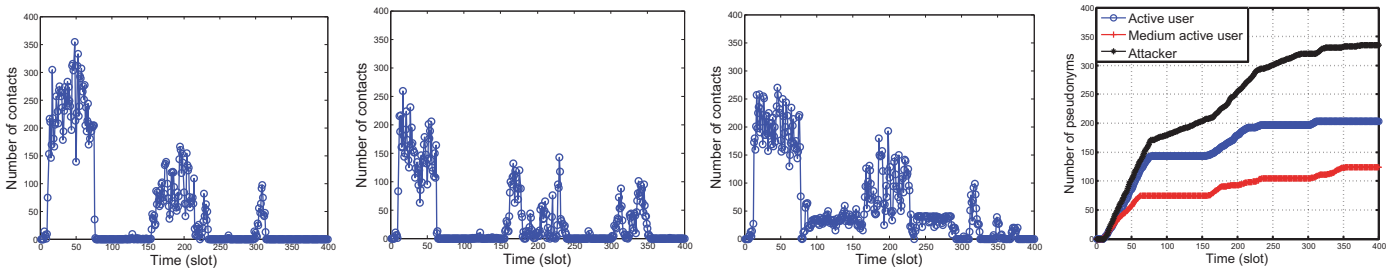
contact signatures, which considerably reduces their resource consumption. Using semi-supervised learning with Hidden Markov Model (HMM), we propose a novel detection scheme to distinguish the abnormal contact distribution and detect the colluded mobile users.

A. Social-based Mobile Sybil Detection

As the pseudonym technique [22] is widely applied in the mobile network, on one hand, it can protect user's real identity from identified and linked by others; on the other hand, the use of pseudonyms may hinder the Sybil detection since mobile users cannot easily link the Sybil identities only based on pseudonyms. Generally, a Sybil attacker $\mathcal{A}_s$ aims to maliciously produce the biased information and convince the normal users. If $\mathcal{A}_s$ uses the same pseudonym to send the same information to user u_i multiple times, u_i would detect them as the spam. If $\mathcal{A}_s$ changes his pseudonyms in a high frequency and sends the same information to u_i multiple times, these information might be originated from different users from the perspective of u_i. As a result, u_i's preference or decision might be impacted by $\mathcal{A}_s$. Therefore, it is of paramount importance to ensure mobile users to legitimately change pseudonyms only if they are encountered many other users.

Generally, users adopt period based pseudonym changing (PPC) strategy and k-anonymity based pseudonym changing (kPC) strategy. For the PPC, a normal user u_i can change his pseudonym after a specific period (or time window) $\mathbb{T}_s$. When using a pseudonym pid_{i,i_p} with a longer duration than $\mathbb{T}_s$, u_i should change it since pid_{i,i_p} is exposed for a long time and might be identified or linked by others. With the PPC, normal users cannot change their pseudonyms frequently if $\mathbb{T}_s$ is properly selected. The drawback of the PPC is that u_i cannot adjust the period according to the number of contacts. Alternatively, the kPC enables the normal user u_i to change his pseudonym pid_{i,i_p} when k-anonymity [22] is violated. In other words, after pid_{i,i_p} is used more than TH times (TH is a pre-defined threshold), pid_{i,i_p} should be changed. Note that it is possible for u_i to change his pseudonym pid_{i,i_p} in a high frequency, if pid_{i,i_p} meets many users (e.g., more than TH users) within a short period. However, u_i would not always change pseudonyms in such a high frequency in reality.

To understand the relation between contact and pseudonym changing behaviors, we investigate the Infocom06 trace [23], which is a real human trace with 78 mobile users attending a conference within four days. We collect the contacts and



(a) Contact of an active user (b) Contact of a medium active user (c) Contact of an attacker (d) Pseudonym changing among normal users and Sybil attackers

Fig. 3. Observations on contact and pseudonym changing between normal users and Sybil attackers

pseudonym changing behaviors from active user (with the highest number of contacts), medium active user (with the average number of contacts), and Sybil attacker (Note that similar to [3], we randomly select users from the trace as attackers, and set the pseudonym changing frequency higher than normal users) in Fig. 3. Time is divided into small time slots (i.e., 10 minutes for each). In Fig. 3(d), Sybil attackers adopt more pseudonyms under the similar mobility (and contact) of normal users as shown in Fig. 3(a) and 3(b). Normal users change their pseudonyms in the appropriate way. In contrast, Sybil attackers may sometimes normally change their pseudonyms to act as normal users, and abnormally change their pseudonyms when launching attacks.

In this paper, we adopt the mobile user's contact information including the encountered user's pseudonym and the number of contacts as the evidence for mobile users to support their pseudonym changing behaviors. The contact between two users with pseudonyms pid_{i,i_p} and pid_{j,j_q} is denoted as $\mathbb{C}_{i_p,j_q} = (pid_{i,i_p}, pid_{j,j_q}, t)$, where t is the encountered time. We utilize kPC as user's pseudonym changing strategy. The detailed Sybil detection steps are shown in Alg. 1. After the basic Sybil detection, the detector reports the Sybil attacker $\mathcal{A}_s$'s pseudonym and the corresponding contact list to the TA.

Algorithm 1 SMSD

1: **Input**: a mobile user u_i with pseudonym pid_{i,i_p}, an initialized contact list $\mathcal{CL}_{i,i_p}$, and pseudonym changing threshold TH
2: **Output**: $\mathcal{CL}_{i,i_p}$ and Sybil detection
3: **while** $|\mathcal{CL}_{i,i_p}| < TH$ ($|\mathcal{CL}_{i,i_p}|$ denotes the number of items) **do**
4: **if** pid_{i,i_p} is encountered with another user pid_{j,j_q} **then**
5: They generate $\mathbb{C}_{i_p,j_q} = (pid_{i,i_p}, pid_{j,j_q}, t)$.
6: pid_{i,i_p} adds $\mathbb{C}_{i_p,j_q}$ into $\mathcal{CL}_{i,i_p}$.
7: **end if**
8: **end while**
9: u_i changes pid_{i,i_p} to $pid_{i,i_{p+1}}$.
10: **Sybil Detection**:
11: Having $\mathcal{CL}_{i,i_p}$, the detector first checks if (1) $|\mathcal{CL}_{i,i_p}| < TH$ and (2) $\mathbb{T}_{p-1} < t_1 < \cdots < t_j < \cdots < t_n < \mathbb{T}_p$. Here, $\mathbb{T}_{p-1}$ and $\mathbb{T}_p$ are starting and ending time of pseudonym pid_{i,i_p}
12: **if** Both (1) and (2) are not guaranteed at the same time **then**
13: pid_{i,i_p} is maliciously used. u_i is a Sybil attacker.
14: **else**
15: pid_{i,i_p} is legitimately used.
16: **end if**

B. Contact Signature with Aggregate Verification

Since the pseudonym pid_{i,i_p} (belonging to $\mathcal{A}_s$) with fewer contacts would be detected as a Sybil attacker, the Level-2 Sybil attacker $\mathcal{A}_s$ may forge the contact information and provide fake contacts to the detector such that $\mathcal{A}_s$ could be merged into the crowd of normal users. To resist Level-2 Sybil attacker, we propose a contact signature scheme. As an evidence of the contact, the contact signature is generated by each pair of the encountered legitimate mobile users. We adopt a variant of aggregate signature [24] to reduce the overall signature size and the verification overhead.

Let $\mathbb{G}$ and $\mathbb{G}_1$ be additive cyclic groups with the same prime order q, and P is the generator of $\mathbb{G}$. $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{G}$, and $\mathsf{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ are two cryptographic hash functions. Let e be a bilinear pairing, where e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ [24] between $\mathbb{G}$ and $\mathbb{G}_1$ exists under two conditions: 1) for any random numbers $a, b \in \mathbb{Z}_q^*$, $e(aP, bP) = e(P,P)^{ab}$; 2) $e(P,P) \neq 1$. Taking a security parameter κ as input, a probabilistic algorithm outputs a tuple (q, $\mathbb{G}_1$, $\mathbb{G}$, e, P, H, H_1) as the system parameters to the public.

• ***Initialization***: A user u_i receives a series of pseudonyms pid_{i,i_1}, pid_{i,i_2}, ..., pid_{i,i_n}. Each pseudonym pid_{i,i_p} is assigned with the corresponding secret key pair $\mathsf{SK}_{i,i_p} = (sk_{i_p,0}, sk_{i_p,1})$, where $sk_{i_p,0} = s_{i,i_p}\mathsf{H}(pid_{i,i_p}||0)$, and $sk_{i_p,1} = s_{i,i_p}\mathsf{H}(pid_{i,i_p}||1)$. $s_{i,i_p} \in \mathbb{Z}_q^*$ is selected by u_i. The public key is $\mathsf{PK}_{i,i_p} = s_{i,i_p}P$.

• ***Contact Signature***: When two users pid_{i,i_p} (i.e., u_i) and pid_{j,j_q} (i.e., u_j) are encountered, pid_{i,i_p} generates the contact as $\mathbb{C}_{i_p,j_q} = \{pid_{i,i_p}, pid_{j,j_q}, t\}$. pid_{i,i_p}'s signature of the contact between pid_{i,i_p} and pid_{j,j_q} at time t is

$$\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,j_q}) = (pid_{j,j_q}, \omega_{i_p}, \theta_{i_p}) \tag{1}$$

$$\begin{cases} \omega_{i_p} = r_{i_p}\mathsf{H}(pid_{j,j_q}) + sk_{i_p,0} + c_{i_p}sk_{i_p,1} \\ \theta_{i_p} = r_{i_p}P \end{cases} \tag{2}$$

where $c_{i_p} = \mathsf{H}_1(t||pid_{i,i_p}||pid_{j,j_q})$, and $r_{i_p} \in \mathbb{Z}_q^*$ is a random number. Finally, pid_{i,i_p} sends $\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,j_q})$ to pid_{j,j_q} as the unforgeable signature to prove the contact $\mathbb{C}_{i_p,j_q}$.

• ***Verification***: When receiving the contact signature from the encountered user, pid_{j,j_q} verifies its authenticity as

$$\begin{aligned} e(\omega_{i_p}, P) \stackrel{?}{=} & e(\theta_{i_p}, \mathsf{H}(pid_{j,j_q}))\cdot \\ & e(\mathsf{H}(pid_{i,i_n}||0) + c_{i_p}\mathsf{H}(pid_{i,i_n}||1), \mathsf{PK}_{i,i_p}). \end{aligned} \tag{3}$$

TABLE I
COMPARISON OF COMPUTATION COMPLEXITY

	Sign	Verification
S	$C_{H_p} + 3C_M$	$3C_{H_p} + 3C_p + 2C_M$
S_{agg}	$N \cdot C_{H_p}$+$3N \cdot C_M$	$(2N+1) \cdot C_{H_p} + (N+2) \cdot C_p + (N+1)C_M$

If Equation 3 holds, the received signature is valid; otherwise, it is forged or invalid. Then, pid_{j,j_q} replies $\mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q}) = (pid_{i,i_p}, \omega_{j_q}, \theta_{j_q})$ to pid_{i,i_p}. These signatures can be stored and used as the evidence of user's pseudonym changing.

• ***Aggregate Authentication***: When pid_{j,j_q} changes his pseudonym to $pid_{j,j_{q+1}}$, u_j collects all the contact signatures related to pid_{j,j_q} and sends them to the CS for Sybil detection. As the increasing number of encountered users, the size of signatures correspondingly increases. To reduce the communication and computation overhead of authentication, we adopt aggregate authentication. First, u_j aggregates the signatures $\mathsf{Sign}_{agg} = (\Omega_{agg}, \Theta_{agg}, pid_{j,j_q})$ of $(pid_{1,1_a}||pid_{2,2_b}||\dots||pid_{i,i_p}||\dots||pid_{n,n_x}, t_1||t_2||\dots||t_i||\dots||t_n, pid_{j,j_q})$ where

$$\Omega_{agg} = \sum_{i=1}^{n} \omega_{i_p}, \Theta_{agg} = \sum_{i=1}^{n} \theta_{i_p}. \quad (4)$$

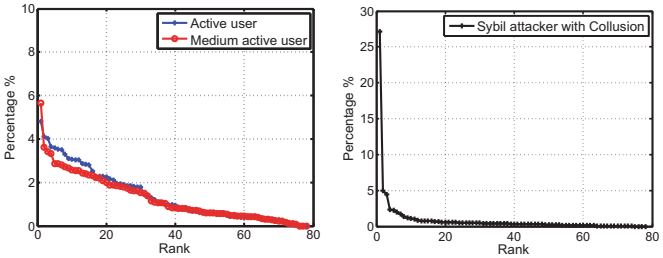
Then, u_j sends the aggregate signature Sign_{agg} to the CS for authentication.

To verify pid_{j,j_q}'s aggregate signature, the CS checks

$$e(\Omega_{agg}, P) \stackrel{?}{=} e(\Theta_{agg}, \mathsf{H}(pid_{j,j_q})) \cdot \prod_{i=1}^{N} e(\mathsf{H}(pid_{i,i_p}||0) + c_{i_p}\mathsf{H}(pid_{i,i_p}||1), \mathsf{PK}_{i,i_p}).$$

If it does not hold, some of pid_{j,j_q}'s contact signatures are forged by pid_{j,j_q} or other mobile users. Note that during each contact, pid_{j,j_q} should check the validity of the received signatures at the beginning. In other words, every stored contact signature by pid_{j,j_q} should be valid by pid_{j,j_q}'s verification. The forged signatures would be forged by pid_{j,j_q}. Therefore, the CS could directly detect the Level-2 Sybil attacker.

The contact signature may increase the communication, computation and storage overhead, which is crucial to mobile users. We adopt the cloud server to replace mobile users as the detector. We show the computation complexity in Table I, where C_{H_p} is map-to-point Hash operation, C_M is multiplication, and C_p is pairing operation. Our aggregate signature scheme can significantly reduce the verification overhead. If a mobile user u_i holds all the contact signature, u_i should provide his historic contact signatures for other user's detection. It would directly expose his past pseudonyms, while the authentication overhead exponentially increases as u_i meets more users. In the SMSD, the CS takes u_i's contact signatures and verifies once for each pseudonym. Then, the CS signs a receipt for the successful detection to u_i. Thus, u_i can adopt this receipt to prove his completed detection other than authenticating his past pseudonyms to every user.



(a) Contact rate distribution of normal users (b) Contact rate distribution of the Sybil attacker

Fig. 4. Comparison of contact rate distribution between normal users and Sybil attacker

C. Learning Assisted Mobile Sybil Detection

Level-3 Sybil attacker $\mathcal{A}_s$ may collude with other mobile users or attackers to disrupt the general Sybil detection via maliciously generating valid signatures to prove the contacts with $\mathcal{A}_s$. These inexistent contacts between $\mathcal{A}_s$ and the colluded users increase the total number of $\mathcal{A}_s$'s contacts and "validate" his abnormal pseudonym changing. To this end, we propose Learning assisted SMSD scheme (LSMSD) to detect Level-3 Sybil attackers to enhance the basic SMSD. Specifically, the LSMSD basically consists of three steps: contact rate distribution, semi-supervised learning with Hidden Markov Model (HMM) and social proximity evaluation.

• ***Contact Rate Distribution***: To detect the collusion of mobile users, we first analyze the contact rate distribution with other users. Specifically, when two users are frequently encountered, they are in local vicinity. If two users are colluded, they would have a very high contact rate with each other. Meanwhile, they only have regular or limited contact rate with any other user as shown in Fig. 4. The percentage in y-axis is about the contact number for each pair of encountered users. The contact rate distribution could be approximated to an exponential distribution. The detector, e.g., the CS, the TA or other trusted party, can put the contact distribution into sequences for semi-supervised learning with HMM.

• ***Semi-supervised Learning with HMM***: We propose a semi-supervised learning scheme with Hidden Markov Model (HMM) to detect the collusion of mobile users. First, we utilize an ergodic k-class HMM to analyze the abnormal contact distribution. k is the amount of abnormal states in HMM, while there would be multiple normal states. In the initialization, there is only one state NS_0, which is a central state in HMM as shown in Fig. 5. There are l normal states and k abnormal states. NS_0 denotes the basic normal state, and could be obtained by training from a certain number of contact distribution samples. For the ground truth data, we select user's contact distribution during daytime and night time to adjust user's different mobilities and social behaviors.

A set of parameters θ^* of normal state HMM model is obtained from maximizing the likelihood of the ground truth

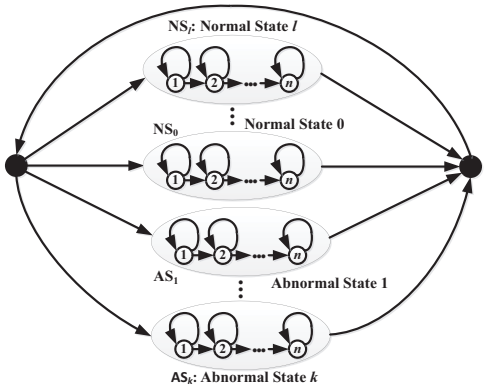


Fig. 5. Hidden Markov Model

sequence (i.e., contact distribution) $\{c_1^{(l)}, \cdots, c_{N_l}^{(l)}\}$ as

$$\theta^* = \arg\max_{\theta} \prod_{j=1}^{N_l} P(c_j^{(l)}|\theta). \tag{5}$$

We assume that each HMM state follows the Gaussian Mixture Model (GMM), which can be estimated by standard Expectation-Maximization (EM) algorithm [25]. The concrete steps of semi-supervised learning algorithm with HMM are stated in Alg. 2.

In the adaptation phase, we utilize Maximum A Posteriori (MAP) [26] scheme to adjust the normal state model to a certain abnormal state model for training on the abnormal state model. While, the original normal state model is also trained by adapting the non-outlier segments. We select θ^* to maximize posterior probability density as

$$\theta^* = \arg\max_{\theta} P(\theta|C) = \arg\max_{\theta} P(C|\theta)P(\theta). \tag{6}$$

With GMM, the model is adapted according to the new weight, mean and variance, denoted by w_i', μ_i' and σ_i', respectively.

$$\begin{aligned} w_i' &= \frac{1}{L}\sum_{j=1}^{L} P(i|c_j,\theta),\ \mu_i' = \frac{\sum_{j=1}^{L} c_j P(i|c_j,\theta)}{\sum_{j=1}^{L} P(i|c_j,\theta)} \\ \sigma_i' &= \frac{\sum_{j=1}^{N} P(i|c_j,\theta)(c_j-\mu_i')(c_j-\mu_i')^T}{\sum_{j=1}^{L} P(i|c_j,\theta)}. \end{aligned} \tag{7}$$

The adaptive parameters can be updated as

$$\begin{aligned} \widehat{w}_i =& \beta \cdot w_i + (1-\beta)\cdot w_i' \\ \widehat{\mu}_i =& \beta \cdot \mu_i + (1-\beta)\cdot \mu_i' \\ \widehat{\sigma}_i =& \beta \cdot (\sigma_i + (\widehat{\mu}_i - \mu_i)(\widehat{\mu}_i - \mu_i)^T) \\ &+ (1-\beta)\cdot(\sigma_i' + (\widehat{\mu}_i - \mu_i')(\widehat{\mu}_i - \mu_i')^T). \end{aligned} \tag{8}$$

Here, w_i, μ_i and σ_i are the previous weight, mean and variance. Note that β is the adapter factor to balance the new parameters and the previous ones. When β becomes larger, the new parameters would contribute more in the adapted model.

Algorithm 2 LSMSD

1: **Input**: A set of N_l labeled contact distributions $\mathbb{C}^{(l)} = \{c_1^{(l)}, \cdots, c_{N_l}^{(l)}\}$, N_u unlabeled contact distributions $\mathbb{C}^{(u)} = \{c_1^{(u)}, \cdots, c_{N_u}^{(u)}\}$.
2: **Output**: Trained HMM $\Theta = \{\theta_1, \cdots, \theta_K\}$ where $K = l + k$. The classification of the contact distributions $\mathbb{NS} = \{NS_1, \cdots, NS_l\}$ and $\mathbb{AS} = \{AS_1, \cdots, AS_k\}$.
3: Step 1 (**Supervised training**): Given the ground truth data, estimate the central state NS_0 with $\theta^* = \arg\max_{\theta} \prod_{j=1}^{N_l} P(c_j^{(l)}|\theta)$.
4: **while** The number of iterations is small than that of abnormal states **do**
5: Step 2 (**Outlier**): Having a sliding window ω, split the contact distribution into Ω segments with overlapping.
Select the outlier with the smallest likelihood
$s^* = \arg\min_{s} \left(\arg\max_{s} \prod_{j=1}^{\Omega} P(c_j^{(l)}|s) \right)$.
Label this contact distribution as an abnormal state.
6: Step 3 (**Adaptation**): A new abnormal state model AS_i is adapted from the general model via the abnormal detection. The normal state model is adapted from the general model by using the other segments.
7: Step 4 (**Boundary**): Determine the boundary of states
8: Step 5 (**New Outlier**): Select a new state model with the smallest likelihood in the adaptive normal state model as an outlier.
9: **end while**

In step (4) of Alg. 2, we follow the sophisticated Viterbi decoding [25] to determine the boundary of states.

The LSMSD requires a small amount of ground truth data for training, which significantly reduces the training overhead and is suitable for mobile Sybil detection. In HMM, we also establish l normal states which leverages from the active users to inactive ones, from the daytime to the night time. With the adaption on the HMM, the LSMSD can improve the learning accuracy with a large amount of unlabeled data.

• ***Social Proximity Evaluation***: Although the LSMSD is effective to detect the collusion by classifying the abnormal contact distribution, it might produce false detection if some normal users always stay together. To address this issue, we exploit the social community to facilitate the LSMSD. Actually, if two users frequently contact each other, they would have some specific social relationships, such as colleagues, social friends, neighbors, etc. All these features can be extracted as social communities. Each user u_i maintains a social community vector $\overrightarrow{SC_i} = [1, 0, 0, \ldots, 1, 0]$. We define the social proximity $SP_{i,j}$ between u_i and u_j as

$$SP_{i,j} = \frac{|\overrightarrow{SC_i} \bigcap \overrightarrow{SC_j}|}{|\overrightarrow{SC_i} \bigcup \overrightarrow{SC_j}|} \in [0,1]. \tag{9}$$

According to the detection results in the LSMSD, the user pair (u_i, u_j) with the social proximity $SP_{i,j} < SP$ can be labeled as the Level-3 Sybil attackers. Here, we define SP as the social proximity that normal friends should have. Therefore, the LSMSD can be further enhanced with the social assistance.

D. Ring Structure of Contact Signature

In the aforementioned subsections, we present the solutions to resist Level-1, Level-2 and Level-3 Sybil attackers via the relation between contacts and pseudonym changing behaviors, aggregate signatures on contacts, and the contact rate distribution, respectively. A condition is that the CS honestly follows

all the procedures and cannot be compromised. In reality, the CS is semi-trusted entity as indicated in Section III, and is possibly compromised. Therefore, we adopt a ring structure of contact signature to resist the deletion and modification on the contact data sent to the CS.

Before uploading the contact list to the CS, each mobile user should form his contact list in a ring structure where each item cannot be removed or modified by others.

1) u_i first initializes the contact list $\mathcal{CL}_{i_p}$ for pid_{i,i_p}. When u_i begins to use a pseudonym pid_{i,i_p} at time t_0, the contact list is $\mathcal{CL}_{i_p} = \{\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p})\}$, where $\mathbb{C}_{i_p,i_p} = (pid_{i_p,i_p}, pid_{i_p,i_p}, t_0)$.

2) When pid_{i,i_p} meets pid_{j,j_q} at t_1, u_i obtains the contact signature $\mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q})$, and updates the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, \mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q})\}$, where $R_1 = (pid_{i,i_p}, t_0, \mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p}))$.

Similarly, when another user pid_{l,l_r} is encountered with pid_{i,i_p} at t_2, pid_{l,l_r} sends the contact signature $\mathsf{Sign}_{\mathsf{SK}_{l,l_r}}(\mathbb{C}_{i_p,l_r})$ to pid_{i,i_p}. pid_{i,i_p} then updates the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, R_2, \mathsf{Sign}_{\mathsf{SK}_{l,l_r}}(\mathbb{C}_{i_p,l_r})\}$, where $R_2 = (pid_{j,j_q}, t_1, \mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p}))$.

3) pid_{i,i_p} recursively builds the ring structure following step 2). When u_i changes pseudonym pid_{i,i_p} at t_N, u_i finalizes the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, R_2, \ldots, R_N, \mathsf{Sign}_{\mathsf{SK}_{i_p}}(\mathbb{C}'_{i_p,i_p})\}$, where $\mathbb{C}'_{i_p,i_p} = (pid_{i_p,i_p}, pid_{i_p,i_p}, t^*)$ and $t^* = \mathsf{H}_1(t_0||t_1||\ldots||t_N)$.

In addition, u_i generates a contact order list $\mathcal{CO}_{i,i_p} = \{CO_0, CO_1, \cdots, CO_N\}$. Let H_2 and H_3: $\{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be two cryptographic hash functions. u_i adopts pid_{i,i_p} in the duration $[t_0, t_N]$, and has contacts at $\mathbb{T} = \{t_1, t_2, \cdots, t_{N-1}\}$. For the n-th contact, $CO_n = \mathsf{H}_1(h_n||pid_{j,j_q})$, where pid_{j,j_q} is the encountered user. Here, $h_n = \mathsf{H}_1(\mathsf{H}_2(h_{n+1}||t_{n+1}) \oplus \mathsf{H}_3(h_{n-1}||t_{n-1}))$ where $n \in [1, N-1]$. As such, a bi-directional Hash chain is established, where the forward seed is $h_0 = \mathsf{H}_1(t_0)$ and backward seed is $h_N = \mathsf{H}_1(t_N)$.

The contact signatures form a closed ring, while the established bi-directional Hash chain guarantees the order of every contact time. The contact list should be synchronized with the contact order list to ensure the integrity of the contact information provided by mobile users.

V. Security Analysis

In this section, we analyze the security properties of the SMSD scheme according to the attacker model in Section III.

A. General Mobile Sybil Detection (Level-1)

The SMSD scheme can resist general Sybil attack since mobile user's contact and pseudonym changing behaviors are correlated. If a Level-1 attacker $\mathcal{A}_s$ launches the attack, i.e., frequently changes pseudonyms, it is difficult for $\mathcal{A}_s$ to collect sufficient contacts to prove the correctness of changing pseudonyms within the short period. The behavior difference between normal users and Level-1 attackers can directly reflect their primary purposes of participating in the mobile network.

Even though $\mathcal{A}_s$ sometimes performs as a normal user and does not change his pseudonym frequently, the other pseudonyms used within a short period or few contacts can also be identified. The SMSD can also restrict Level-1 Sybil attack's malicious behaviors and lead to the higher resource consumption to launch the attack. Thus, it is applicable to practical applications, and performs effective detection in mobile environments.

B. Contact Unforgeability of Mobile User (Level-2)

Theorem 1. *The SMSD can prevent Level-2 attackers from maliciously forging contacts by using the contact signatures.*

Proof: During the contact between pid_{i,i_p} and pid_{j,j_q}, they sign on the contact event, including the encountered pseudonyms and the time, by using secret keys. Assume that the computational Diffie-Hellman problem in $\mathbb{G}$ is hard. The contact signature $\omega_{i_p} = r_{i_p}\mathsf{H}(pid_{j,j_q}) + sk_{i_p,0} + c_{i_p}sk_{i_p,1}$ and $\theta_{i_p} = r_{i_p}P$ are unforgeable, since $sk_{i_p,0}$, $sk_{i_p,1}$ and r_{i_p} are selected by pid_{i,i_p}. Without compromising other users, the forged contact signature ω_{i_p} and θ_{i_p} from $\mathcal{A}_s$ would be detected by the CS, since the CS aggregates $\mathcal{A}_s$'s contact signatures and verifies them at first. Therefore, the contact signature can validate the authenticity of the contact event, which is also the foundation of the SMSD. ■

C. Resistance to Collusion of Mobile User (Level-3)

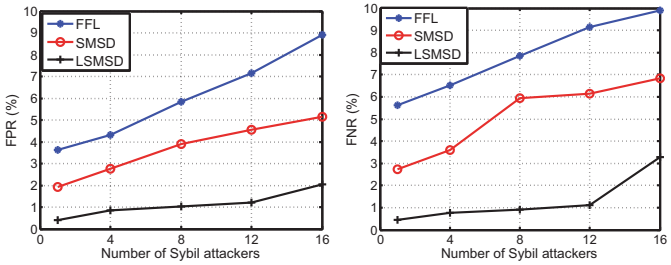
The LSMSD can resist the collusion of mobile users via the semi-supervised learning with HMM on the contact rate distribution. If one mobile user u_i colludes with the Sybil attacker $\mathcal{A}_s$, u_i generates valid signatures for some inexistent contacts with $\mathcal{A}_s$, such that $\mathcal{A}_s$ can change his pseudonyms early prior to the normal changing time point. $\mathcal{A}_s$ and u_i would have a large number of contacts, reflecting a high contact rate in their contact distribution. However, $\mathcal{A}_s$ would meet other users infrequently, such that $\mathcal{A}_s$ has lower contact rates with others. As shown in Fig. 4, normal users and Level-3 attackers have different contact distribution. Therefore, the proposed semi-supervised learning with HMM can classify the normal contact distribution and the abnormal one due to the collusion. The detection accuracy will be presented in Section VI.

As an enhancement of the LSMSD, social proximity is explored to assist the contact distribution. Since the colluded user may not have strong social connections with $\mathcal{A}_s$, but contacts frequently with $\mathcal{A}_s$, the colluded users can be classified according to their social relationships.

D. Resistance to Collusion of Cloud Server (Level-4)

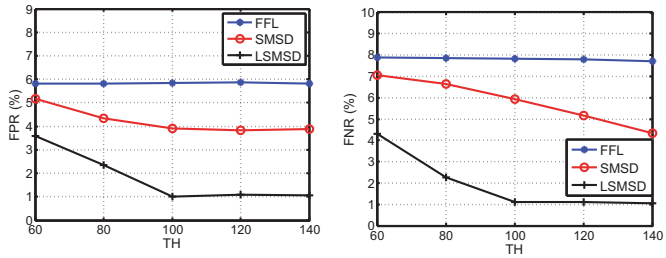
Theorem 2. *The CS cannot add, modify and remove the mobile user's contacts due to the contact signature ring and bi-directional Hash chain of contact order.*

Proof: Suppose the CS deletes the contact between pid_{i,i_p} and pid_{j,j_q} at t_j. $h_{j-1} \neq \mathsf{H}_1(\mathsf{H}_2(h_{j+1}||t_{j+1}) \oplus \mathsf{H}_3(h_{j-2}||t_{j-2}))$. Similarly, h_{j+1} cannot be recovered as well. As a result, the whole contact order list is invalid. Due to the forward and backward secrecy, the contact order list cannot be forged. If the CS modifies or adds contact signatures for any user, the detectors can find out the CS's malicious operations due to Theorem 1.



(a) False positive rate vs. the number of Sybil attackers (b) False negative rate vs. the number of Sybil attackers

Fig. 6. The impacts of the number of Sybil attackers



(a) False positive rate vs. TH (b) False negative rate vs. TH

Fig. 7. The impacts of TH (i.e., every user changes pseudonyms when the pseudonym meets more than TH users)

In the contact signature ring, if R_2 (e.g., from pid_{j,j_q}) is deleted, t^* cannot be calculated without t_2. Similarly, if the CS adds R_j^* into $\mathcal{CL}_{i_p}$, the contact signature ring cannot be synchronized with the order list. Therefore, the proposed contact signature ring and bi-directional Hash chain can protect the stored contact information from addition, modification and deletion by the CS. ■

In summary, the proposed SMSD scheme can resist the four levels of Sybil attackers defined in Section III.

VI. Performance Evaluation

In this section, we evaluate the performance of the SMSD based on the trace-driven simulation.

A. Simulation Setup

We use Infocom06 trace [23] with 78 mobile users during a four-day conference. Each mobile user carries a dedicated Bluetooth device, which can discover the surrounding users. Totally, $128,979$ contacts are recorded. We separate the entire contact data into two parts: 20% of data are the training set to produce mobile users' profiles (e.g., social communities), and the remaining data are used for the simulation.

We assign users with social communities according to the sociology theory. A complete graph G is built up, where each edge $E(u_i, u_j)$ weighted by the total number of contacts between two vertex u_i and u_j. We then refine the graph G with 78 vertices and $2,863$ edges by removing the edges weighted less than 100. Based on Bron-Kerbosch algorithm [27], we extract maximal cliques in G. A clique is a complete subgraph where every edge is high-weighted. According to the weight of each maximal clique, we select 100 social communities (i.e., cliques). The selected communities are used for simulation comparison on social connections.

B. Simulation Results

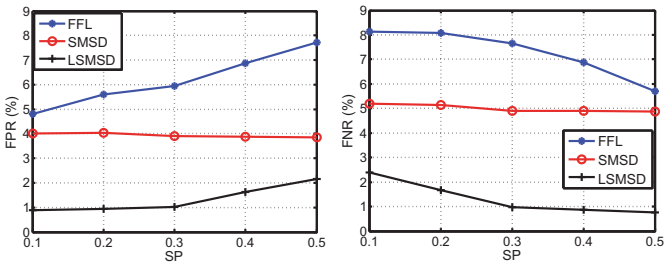
We compare the SMSD and LSMSD schemes with the Sybil detection scheme in [3], named FFL (Friend and Foe list). In FFL, mobile users detect attackers by comparing their social friend list. We determine the Sybil attackers similarly to [3], and select 1, 4, 8, 12, and 16 users as the attackers in our simulation. In addition, we utilize false positive rate (FPR) and false negative rate (FNR) as the metrics to evaluate the Sybil detection accuracy. A false positive detection results in a normal user being detected as an attacker, while a false negative indicates that a Sybil attacker being regarded as a normal user. We define $\mathsf{FPR} = P_f/(P_f + N) \times 100\%$ where P_f denotes the number of false positive detections, N is the amount of attackers. Similarly, $\mathsf{FNR} = N_f/(N_f + P) \times 100\%$ where N_f denotes the number of false negative detections, P is the amount of normal users. We run different schemes within 400 time slots for performance comparison. For the LSMSD, we set adapter factor $\beta = 0.5$ and select 100 contact distributions from different users as ground truth data.

In Fig. 6, we compare the FPRs and FNRs among FFL, SMSD and LSMSD as the increasing number of attackers. We set $TH = 80$ and $SP = 0.3$. The number of attackers has a larger impact on FFL compared with that on SMSD and LSMSD. The reason is that the increasing number of attackers introduces the larger challenge to the mobile users to detect Sybil attackers via their friend and foe lists. The number can affect SMSD and LSMSD since a large number of Sybil attackers can launch strong collusion attack and forge contact signatures to disrupt the SMSD and LSMSD. In the following results, we set 8 Sybil attackers in the network.

In Fig. 7, we show the performance comparison by varying the threshold (number of contacts) of changing pseudonyms. We set $SP = 0.3$ for the LSMSD. When TH is small, e.g., $TH = 60$, the FPRs and FNRs of SMSD and LSMSD is not high. The reasons are two-fold: on one hand, attackers can also perform as normal users; on the other hand, when attackers launch Sybil attacks, each pseudonym still has a specific lifetime. A smaller TH results in a small gap with the number of contacts that an attacker has. When increasing TH, the gap becomes larger so that the FPRs and FNRs of SMSD and LSMSD significantly decrease.

As shown in Fig. 8, the social proximity threshold SP can only impact on LSMSD which detects Level-3 attackers. We set $TH = 80$. For FFL, we adopt SP as the threshold to befriend with others. When SP is large, the Sybil attackers would not befriend with normal users so that the FNR reduces. Meanwhile, a large SP prevents some normal users from befriending with others. Therefore, they may be detected as attackers, which increases the FPR.

For the users with high contact rate, the LSMSD can detect whether their contacts are forged or not according to the social proximity. If SP is small, the FNR increases since the colluded



(a) False positive rate vs. SP (b) False negative rate vs. SP

Fig. 8. The impacts of SP (i.e., social proximity)

attackers may have certain social connections. It is easy to achieve so that both normal users and Sybil attackers with high contact rate are likely detected as normal users. By increasing SP, the FNR drops, while the FPR increases. The reason is that the colluded users with high contact rate can hardly build very strong social connections with each other as $SP > 0.3$.

In summary, LSMSD performs better than SMSD since the Level-3 attackers can be detected by semi-supervised learning with HMM, which balances the training overhead and detection accuracy. Having the appropriate parameters, e.g., $TH = 100$, $SP = 0.3$, the four levels of Sybil attackers could be detected.

VII. Conclusion

In this paper, we have proposed a social-based mobile Sybil detection scheme to detect four levels of Sybil attackers with different attacking capabilities. We have investigated mobile user's pseudonym changing behaviors compared with that performed by Sybil attackers, and utilized contact statistics as the criteria of pseudonym changing for mobile Sybil detection. The security analysis demonstrates that the SMSD can resist four levels of Sybil attackers, while the extensive trace based simulation can validate the detection accuracy of the SMSD. The proposed SMSD scheme is a new paradigm in mobile environments, taking the advantages of powerful storage and computing capabilities in the cloud server, and starts a trend to distinguish Sybil attackers via mobile user's contacts and pseudonym changing. The semi-supervised learning with HMM can offer accurate detection with reasonable training overhead. For our future work, we will investigate the cooperation among mobile users to pool the contact statistics for fully distributed Sybil detection.

Acknowledgement

This work has been supported by a research grant from the Natural Science and Engineering Research Council (NSERC), Canada, the grant AOARD-144029, the grants NTU-SUG (M4081196) and MOE Tier 1 (M4011177) from Nanyang Technological University, Singapore, and the grant NSFC No. 61420106010 from China.

References

[1] Q. Lian, Z. Zhang, M. Yang, Y. Zhao, Y. Dai, and X. Li, "An empirical study of collusion behavior in the Maze P2P file-sharing system," in *Proc. of IEEE ICDCS*, 2007, pp. 56–66.

[2] K. Zhang, X. Liang, R. Lu, and X. Shen, "Exploiting multimedia services in mobile social network from security and privacy perspectives," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 58–65, 2014.

[3] D. Quercia and S. Hailes, "Sybil attacks against mobile users: Friends and foes to the rescue," in *Proc. of IEEE INFOCOM*, 2010, pp. 336–340.

[4] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, "SybilGuard: Defending against sybil attacks via social networks," *IEEE ACM Transactions on Networking*, vol. 16, no. 3, pp. 576–589, 2008.

[5] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A near-optimal social network defense against sybil attacks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 885–898, 2010.

[6] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *CoRR*, 2011.

[7] X. Liang, X. Lin, and X. Shen, "Enabling trustworthy service evaluation in service-oriented mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 310–320, 2014.

[8] X. Lin, "LSR: Mitigating zero-day sybil vulnerability in privacy-preserving vehicular peer-to-peer networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 237–246, 2013.

[9] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil Attacks and Their Defenses in the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[10] W. Wei, F. Xu, C. Tan, and Q. Li, "SybilDefender: Defend against sybil attacks in large social networks," in *Proc. of IEEE INFOCOM*, 2012, pp. 1951–1959.

[11] L. Shi, S. Yu, W. Lou, and T. Hou, "SybilShield: An agent-aided social network-based sybil defense among multiple communities," in *Proc. of IEEE INFOCOM*, 2013, pp. 1034–1042.

[12] N. Gong, M. Frank, and P. Mittal, "SybilBelief: A semi-supervised learning approach for structure-based sybil detection," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 6, pp. 976–987, 2014.

[13] T. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[14] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Zhao, "You are How You Click: Clickstream analysis for sybil detection," in *Proc. of USENIX*, 2013, pp. 241–255.

[15] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Zhao, "Social turing tests: Crowdsourcing sybil detection," in *Proc. of NDSS*, 2012, pp. 1–16.

[16] W. Chang, J. Wu, C. Tan, and F. Li, "Sybil defenses in mobile social networks," in *Proc. of IEEE GLOBECOM*, 2013, pp. 641–646.

[17] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proc. of CCS*, 2004, pp. 168–177.

[18] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Channel-based detection of sybil attacks in wireless networks," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 492–503, 2009.

[19] T. Zhou, R. Choudhury, P. Ning, and K. Chakrabarty, "P^{2}DAP - Sybil attacks detection in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 582–594, 2011.

[20] M. Mutaz, L. Malott, and S. Chellappan, "Leveraging platoon dispersion for sybil detection in vehicular networks," in *Proc. of PST*, 2013, pp. 340–347.

[21] J. Xue, Z. Yang, X. Yang, X. Wang, L. Chen, and Y. Dai, "VoteTrust: Leveraging friend invitation graph to defend against social network sybils," in *Proc. of IEEE INFOCOM*, 2013, pp. 2400–2408.

[22] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, "Fully anonymous profile matching in mobile social networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 641–655, 2013.

[23] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31)," Jan. 2006.

[24] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. EUROCRYPT*, 2003.

[25] D. Zhang, D. G. Perez, S. Bengio, and I. McCowan, "Semi-supervised adapted HMMs for unusual event detection," in *Proc. of CVPR*, 2005, pp. I: 611–618.

[26] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[27] C. Bron and J. Kerbosch, "Finding all cliques of an undirected graph (algorithm 457)," *Commun. ACM*, vol. 16, no. 9, pp. 575–576, 1973.