Social Closeness Based Clone Attack Detection for Mobile Healthcare System

Yanzhi Ren*, Yingying Chen*, Mooi Choo Chuah†
*Dept. of ECE, Stevens Institute of Technology † Dept. of CSE, Lehigh University
Castle Point on Hudson, Hoboken, NJ 07030 Bethlehem, PA 18015
{yren2, yingying.chen}@stevens.edu *chuah@cse.lehigh.edu*

***Abstract*—The explosion of the usage of mobile devices and their rapid deployment of sensing technology provide users with the ability to sense the world. The collected sensing data from the mobile device enabled network can be mined not only for extracting social communities which reflect close relationships or similar behavior patterns among people but also for supporting a broad range of applications including mobile healthcare systems. However, the mobile networks are vulnerable to clone attacks, in which the adversary replicates the legitimate nodes and distributes the clones throughout the network to start a variety of insider attacks. The traditional detection approaches can not address such kind of insider attacks in the mobile device network. (ychen: too vague.) In this paper, we propose a social community based method that exploits the social relationships to detect the clone attack launched to undermine a mobile device enabled disease propagation control framework. We define a new metric called community betweenness, which considers both the community and neighboring information of mobile users. We further propose three methods based on community betweenness to facilitate the detection of the clone attack. An analytical analysis on threshold setting when using community betweeness is provided. We evaluate our clone attack detection scheme through extensive simulations using a trace-driven approach by utilizing data sets collected from mobile phones. The results confirm that our method can detect clone attacks efficiently with high detection ratio and low false positive rate. This strongly indicates the feasibility of exploiting the social community information derived from mobile sensing data for detecting the clone attacks.**

I. Introduction

The popularity of the mobile wireless devices is growing significantly in our daily lives. In particular, with the rapid deployment of sensing technology in mobile devices, the collected sensing data can be comprehensive enough to be mined not only for the understanding of human behaviors but also for supporting a broad range of applications. For instance, the Bluetooth device-discovery software running in a mobile device allows it to collect information from other Bluetooth devices nearby. Thus, it is convenient to exploit the mobile devices equipped with Bluetooth technology to discover the encounter events between people such that their social relationships can be derived and analyzed. More importantly, the discovered social relationships can be used to extract social communities [1], [2], which reflect close relationships or similar behavior patterns among people, to assist in the development of applications in various domains, ranging from monitoring/tracking to healthcare applications. One important application of the social community structures is to facilitate the study for controlling disease spread [3], [4] in healthcare domain. [3] experimented how to construct human contact networks by deploying a senor network with hundreds of nodes, which contributed to determine the best intervention strategies for controlling disease spread. Whereas [4] studied to build a mobile phone enabled social community based framework to reduce the rate at which an infectious disease spreads.

However, wireless devices may easily be captured by adversaries and unlimited number of clones of the compromised nodes can be deployed. Since the cloned devices have legitimate IDs and have access to security keys and other credentials, they can participate in the wireless network as a legitimate node. Thus, cloned devices can launch a variety of insider attacks. Such insider attacks may not affect the network performance, however they can have significant impact on the pervasive applications supported by wireless networks. For example, in a high school scenario introduced by [3], the clone attacker can monitor a significant fraction of the users' behaviors around the clone devices or even jam legitimate signals from benign users to corrupt the accurate discovering of the encounter events. A more aggressive and smart attacker could corrupt the intervention strategies for controlling disease spread in [3], [4] and causing continual disruption to the system: an adversary distributes the cloned devices in the network and let these replicas contact with more other users, which makes him look like to have high risk of being infected by the disease or have the ability to spread the disease out to many people. Such action can increase its chance of being selected as the user for receiving the vaccine shots or attract more vaccine shots to be delivered to its physical proximity. Therefore, clone attack is a severe destructive insider attack and effective schemes for detecting the clone attack are needed.

Nevertheless, detecting clone attacks is not an easy task: the cloned devices own all the security information of the compromised device and thus they are able to pass most of the security check from being detected as a malicious device. In addition, a smart attacker can cheat

on other devices or even collude with each other to trick the other users into believing that they are legitimate. Exiting works against the clone attack in sensor network focused on preventing the attack rather than detecting techniques through key distribution schemes [5]. However, as pointed out in [6], most of these prevention schemes are not effective due to the clone devices also have legitimate information and they can report false information to cheat other devices. Recently, new techniques have been developed to actively detect the clone attack by utilizing the neighboring information [7], [8]. However, they can either be applied only to static networks or be used for online social networks, making them less suitable for mobile social networks.

Despite of these difficulties, the clone attack detection can be useful by exploiting the social communities of each node in mobile phone enabled social network. The social communities in [4] extracted from the mobile enabled social relationships, which reflect similar behavior patterns among people in a time period, can not only assist in the vaccine distribution, but can also be used to detect the clone attack. The main idea is that one user often stays with a fixed community of closeby users during a time period and such community of people often remains unchanged during that time period. However, the cloned users can break the rule that one user often resides within a fixed social community during one time period and they may belong to multiple distinct social communities at the same. For instance, a student takes a class from 10:00 AM to 11:30 AM, his/her neighbors often form a stable set and he/she resides with a fixed community unless getting out of the class area. However, altough cloned users may have the legitimate credentials as the original users, but they may not reside in the same social community as the original users. For instance, a original user may take a class by staying with his/her classmates while a user carrying mobile devices with cloned ID of the original user may do some shopping in the supermarket with family memers at the same time. Inspired by this idea, we conduct our research on the detection of the clone attack by exploiting the social communities.

In this paper, we design a social community based method that exploits the social relationships to detect the clone attack in the disease control framework proposed by [3], [4]. A metric called community betweenness is proposed by considering both the community and neighboring information of mobile users and the existence of a clone attack is identified by checking whether the community betweenness of each node is larger than a threshold. To find a suitable threshold in the clone attack detection, in this paper, we develop two methods, analytical and training based, to find the threshold of the community betweenness under certain confidence level. We experimentally evaluated our detection scheme through a trace-driven approach by using the MIT reality mining trace [9] and the SWIM trace [10]. The results showed that our scheme is highly effective for detecting the clone attack in mobile social network.

The rest of the paper is organized as follows. We first put our work in the context of current research in Section II. We then present our attack model in Section III. It describes the clone attack model used in this work. We next present our social community based detection scheme in Section IV. In Section V and Section VI, we introduce two methods for threshold setting and validate the feasibility of our proposed detection schemes by using datasets generated by a contact process model and collected from mobile phones, respectively. Finally, we conclude our work in Section VII.

II. Related Work

(hardware based method is not good) A straightforward method to thwart the clone attack is to prevent the attacker from extracting secret keys from a mobile node by equipping them with tamper-resistant hardware [6]. However, it may still be possible to bypass the tamper-resistant hardware to extract secret keys from capture nodes given enough time and computation ability even though the tamper-resistant hardware makes it much harder to accomplish this. Thus, it is more important to thwart the clone attack using some software-based countermeasures.

(these methods can only be used in static network) Several software-based clone attack detection schemes have been proposed for sensor networks [7], [11]. The main idea of these schemes is to let each node report its locations and attempt to find conflicting reports that one node resides in multiple locations. However, these existing methods in [12], [13] can only be applied to static networks and cannot be used for mobile nodes.

(these methods fail if the clone nodes collude) In [14]–[17], neighboring nodes vote for the the sanity of a given node based on their local observations. However, these schemes may fail when multiple cloned nodes collude. Furthermore, voting detection schemes lack the ability to detect clones if the number of replicas is large because of their collusion. A recent scheme [18] for clone attack detection in mobile environments proposed a solution in which each node records random numbers it has exchanged with the nodes which it encounters. If these two nodes contact again, they check the random number they have exchanged before. If this random number does not match, they report the presence of a possible clone attack. However, this method may also fail if the cloned nodes collude with each other and synchronize the random number.

(this method has large communication overhead) In [6], set operations are used to detect the clone attack: the network first is divided into subregions organized hierarchically. In each subregion, a cluster header is selected and it is in charge of reporting the list of

members in this subregion. The existence of cloned sensors can be detected by checking the intersection of the two member lists reported by the cluster heads. However, the total amount of data to be transferred is large and this scheme can also only work in static networks.

As discussed above, the existing solutions have their limitations for detecting clone attacks in mobile networks. Furthermore, little work has been done in detecting clone attacks in mobile phone enabled social networks.

III. System and Attack Model

The cloned devices can launch a variety of insider attacks and cause many damages in the network. To work through this paper, we studied how the attackers utilize the vaccine distribution process in healthcare system and disturb this process by launching the clone attack. In this section, we first briefly introduce the dynamic community extraction method and then describe the system model of the community based framwork for controlling of the disease propagation in [4], [19]. Finally, we describe our clone attack models, which consist of both the passive attack model and the active attack model in our work.

A. Dynamic Community Extraction

In general, people may belong to different social communities during different time periods. Furthermore, the same social communities can reappear again and again in the daily life of the community members. Thus, in [4], instead of directly extracting communities from the contact graphs, the authors proposed a dynamic community extraction method and they first extract the communities for each non-overlapping time period and then merge those communities with high similarity.

Particularly, they construct the contact graph $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$,...,$G_R = (V_R, E_R)$, which is a geometric representation of the relationship between people by counting the encounter events between them, for each non-overlapping time period: $[T_0, T_1]$, $[T_1, T_2]$,...,$[T_{R-1}, T_R]$. Then, they divide each contact graph $G_i = (V_i, E_i)$ into multiple communities by using the hierarchical clustering algorithm [1] and the modularity Q [20]. The authors then take a snapshot of the communities in each time period and a total of R snapshots are collected: $S_1, S_2, ..., S_R$. Each S_i contains a vertex set A_i. We assume that each A_i has been divided into k_i communities, which are represented as follows:

$$A_i = A_i^1, A_i^2, ..., A_i^{k_i} \tag{1}$$

Then, two extracted communities A_i^j and A_{i+1}^l are merged if they satisfy the following criteria to get M dynamic social communities $\{V_1, V_2, ..., V_M\}$:

$$\frac{|A_i^j \cap A_{i+1}^l|}{Max(|A_i^j|, |A_{i+1}^l|)} > \tau \tag{2}$$

B. Vaccine Distribution System Model

Instead of random vaccine distribution, targeting vaccination to a group of people with higher risk of infection can provide more effective control of an infectious disease propagation. Some existing work [4] utilize the existing infrastructure in cellular networks in the vaccine distribution: users who are subscribed to the cellular data plan recorded encounter events (which include discovered device IDs and timestamps) will be periodically sent back to a back-end server authorized by the service provider. A dynamic community extraction mechanism is run by the server and the extracted community information will be stored at the server and updated from time to time.

When a new disease is discovered in public, in [4], [19], the people who have high risk of being infected by the disease or have the ability to spread the disease out to more people will be identified as V_s. Thus, the vaccines will then be sent to the people within V_s to reduce the propagation of the disease.

C. Attack Model

Nevertheless, in practice, the disease control framework proposed in [3], [4], [19] is vulerable to the clone attack: it is easy for an adversary to capture mobile devices within the network and deploy unlimited number of clone devices. Since these replicas have legitimate access to the network, they can participate in the mobile social network and disturb the vaccine distribution process. We generalize the clone attacks into two categories: one is passive attack and another is active attack.

1) Passive attack: User based clone attack: The adversary can distribute their tempered devices with clone IDs to different users by selling on the websites. For example, they can create an account on eBay (http://www.ebay. com/) or Craigslist (http://www.craigslist.org/) and sell their cloned Bluetooth devices to customers from different social groups or even different areas. In such case, the tempered devices are also not limited to Bluetooth enabled mobile phones: such as Bluetooth earphone, Bluetooth enabled laptops, Bluetooth mouse and so on. The customers did not know the devices they received with cloned IDs and they will use it as usual in their daily life.

Bluetooth IDs of the compromised devices have been duplicated and these replicas contacted with many users or appeared in multiple social communities of the network. In the vaccine distribution system, the vaccine shots are given according to the Bluetooth ID and its corresponding mobile phone number, thus it also increases the chance of a compromised user being selected as the user for receiving the vaccine shots from the clinic. The vaccine distribution process will be disturbed and the attacker can also receive the vaccine shots with high probability from launching this clone attack as well.

2) Active attack: Proximity based clone attack: The adversary can also distribute tempered devices to a group of users who share the similar interest with him. These users are often from different social communities and they can collude with the attackers.

In this situation, these clone users can attract more vaccine shots for their physical proximity: if one user is sick, other users who share the same cloned Bluetooth ID can also announce that they are sick when encountering with other users. According to the proposed vaccine distribution strategy, the users who have close relationship with the sick user will receive the vaccine shots from the clinic. Thus, the number of users for taking vaccines will increase dramatically. As a result, the number of vaccines needed for this physical region would also be increased dramatically and the more vaccines will be delivered to this area. This strategy is especially effective when there are the limited supply and relatively high cost of vaccine shots.

IV. Detection Method of Clone Attack

A. Overview

Our clone attack detection method is based on the observation that mobile device users typically belong to small number of stable communities. By stable, we mean a user's communities do not change rapidly after that user's community profile has been analyzed for sufficient period of time. However, when clone attacks occur, the number of communities a particular user belongs to can increase significantly but members within the communities that this cloned ID belong to may not see the same impact and hence one can design a clone attack detection scheme based on these observations. Specifically, we define a unique metric called community betweenness that allows us to identify nodes with memberships in large number of communities and flag them as potential nodes that may be under clone attacks.

The betweenness [2] has been studied in the past as a measure of the centrality and influence of nodes in networks. The betweenness of a node is defined as the number of shortest paths between pairs of other vertices which run through it. In order to find which nodes in a network are always "between" other pairs of vertices which are from different social communities, we generalize the concept of betweenness and define the community betweenness of a node as the number of certain kind of shortest paths between its neighboring communities that run through it. If clone nodes exist in the network, then almost all shortest paths between its neighboring communities must go through the clone nodes. Thus, the clone node will have high community betweenness and thus clone devices can be detected accordingly.

In this section, we first define some basic concepts, which is the foundation for computing the community betweenness. Then, we describe the three variants of community betweenness that we explore for clone attack detection. To help readers understand how our community betweeness values are derived, we present several illustrative examples. Finally, we present our clone attack detection scheme based on the defined community betweenness metric.

B. Preliminaries

In this section, we introduce some basic definitions on the shortest path and the node-centric set, based on which we can compute the community betweenness for each mobile device.

We consider a mobile network comprised of N devices and each node is assigned with a unique identifier. We assume that these devices contain interfaces e.g. bluetooth, WiFi etc that allow encounter events betweeen owners of devices to be recorded. For example, one can collect contact-based traces using bluetooth discovery software. The collected contact trace of each device can be divided into W trace files. We assume that each trace file consists of recorded encounter events that happened during the time period $T'_i = [t'_{i-1}, t'_i]$ $(i = 1, ..., W)$. As in Section III, we can mine such contact-based trace files to construct a contact graph. The constructed contact graphs $G'_1 = (V'_1, E'_1)$, $G'_2 = (V'_2, E'_2)$,..., $G'_W = (V'_W, E'_W)$, consists of nodes that represent mobile devices (and their owners) and edges that connect the nodes if some contact events happen between these two devices. The contact weights of the edges represent either cumulative contact times or the cumulative time periods of these encounters. Such a contact graph is a geometric representation of the relationship between people, will be built for each non-overlapping time period: $[t'_0, t'_1]$, $[t'_1, t'_2]$,...,$[t'_{W-1}, t'_W]$.

In Section III, we already assume that the network $[V, E]$ has been divided into M dynamic communities $\{V_1, V_2, ..., V_M\}$ which satisfies $V_1, V_2, ..., V_M \subset V$ and $V_1 \cup V_2 \cup ... \cup V_M = V$. Thus, by utilizing the community information $\{V_1, V_2, ..., V_M\}$, we define the following concepts for a vertex k in each contact graph $G'_i = (V'_i, E'_i)$ $(i = 1, ..., W)$:

Definition 1. Neighbor set $N(k)$: The immediate neighbors of a node k in the contact graph are defined as neigbor set $N(k)$.

Definition 2. Shortest path between nodes: The shortest path $P_s(k, d)$ between node k and node d is defined as the path with the shortest hop length for connecting node k and d.

Definition 3. Shortest path between node and node set: We denote the nodes in a set D as $\left\{d_1, ..., d_{\|D\|}\right\}$ and the shortest path $P_s(k, D)$ from a node k to node set D is defined as:

$$P_s(k, D) = \underset{P(k,d_i)}{\arg\min} \left(\left| P(k, d_i) \right|, d_i \in \left\{d_1, ..., d_{\|D\|}\right\} \right) \tag{3}$$

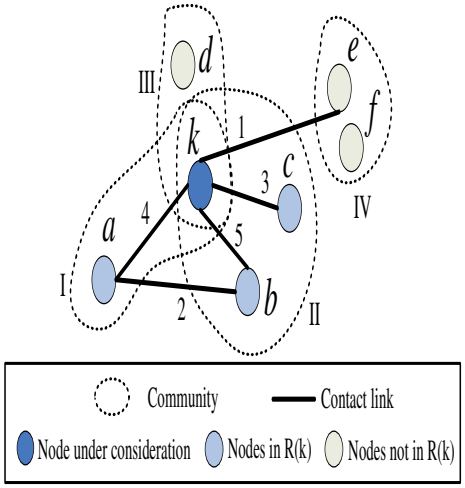


Fig. 1. An example of community betweenness computing.

Definition 3. Distance between node and node set: The distance between node k and node set D is defined as:

$$Dist\,(k, D) = |P_s\,(k, D)| \tag{4}$$

Definition 4. Node-centric set: Suppose node k belongs to M_k different social communities: $V_1, V_2, ..., V_{M_k}$. The node-centric set $R(k)$ of node k is defined as: $R(k) = N(k) \cap (\bigcup_{i=1}^{M_k} V_i)$. In other words, node k's node-centric set includes its neighboring nodes which also belongs to the same social community as k.

C. Community betweenness

1) Definition of Community Betweenness: Community betweenness of node k is defined as the number of certain kind of shortest paths between its neighboring communities that run through k. To compute the community betweenness, we also define the betweenness link weights in the contact graph, which is different from the contact weights introduced before. The community betweenness can then be computed by adding all the betweenness link weights between k and each node in $R(k)$ together. Depending on whether the application scenario cares for the frequency of encounter events or the shortest paths between nodes in $R(k)$, we define three variations on community betweenness according to how their shortest paths are defined. These three variations are namely (a) Contact Frequency Based, (b) Contact Duration Based, and (c)Shortest Path Based community betweenness values.

2) Methods on Betweenness Computation: The main differences the three variations are: (a) the existence of contacts betweeen users is used in contact frequency based betweenness, (b)cumulative contact times betweeen users are used in the contact duration based betweenness, while (c) the Shortest Path based betweenness considers the number of shortest paths between users from different communities. We illustrate our community betweenness computing flow by showing how to compute the betweenness of node k. From the definition of the community betweenness, we first consider the computing of k's betweenness link weights. Initially, the weight of each link between node k and nodes in $R(k)$ is 0 and the computing flow is as follows:

Contact Frequency Based (CFB), For a node $n \in R(k)$, we suppose that node n belongs to community V_j. If $\exists V_i$ ($n \notin V_i$, $V_i \neq V_j$) which satisfies that:

$$W_i = V_i \cap R(k) \neq \emptyset \tag{5}$$

We then start the procedures of computing the link weights L_{nk} between node k and n as follows:

If there exists one shortest path from node n to set W_i which also goes through node k ($k \in P_s(n, W_i)$), the betweenness link weight L_{nk} between node n and k will be increased by the number of nodes in W_i: $L_{nk} = L_{nk} + |W_i|$, otherwise it will not be changed.

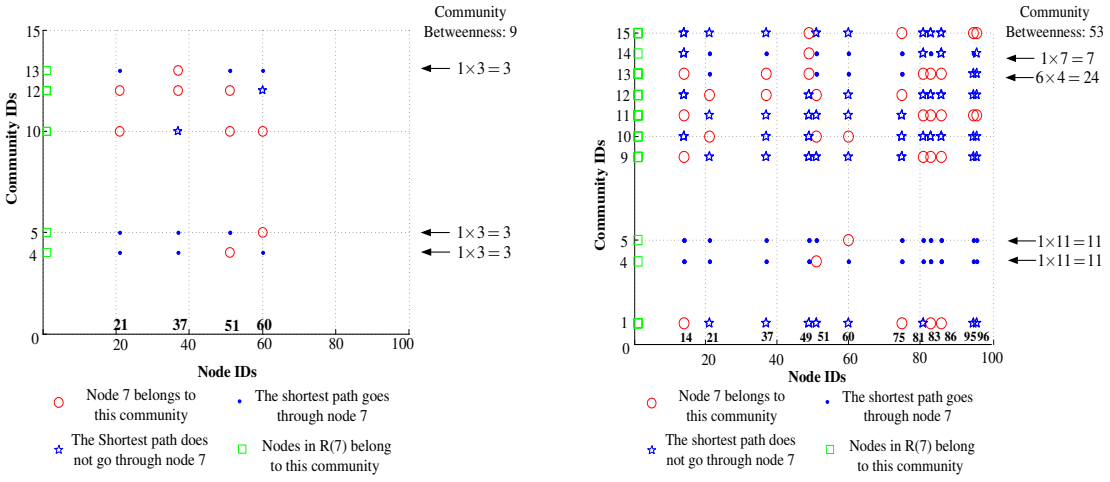
We repeat the above process until all of such node n and all of its V_i have been considered. Then, the community betweenness of node k can be computed by adding all the betweenness link weights together:

$$bet(k) = \sum_{i \in R(k)} L_{ik} \tag{6}$$

Next, we give an example to show how to compute the contact frequency based betweenness. An contact graph of node k is shown in Figure 1: nodes $\{a, k\}$, $\{k, c, b\}$, $\{k, d\}$ and $\{e, f\}$ belongs to communities (1)-(4), respectively and the communities are shown as dotted circles. From the definitions above we can get that k's neighboring node a, b and c are in node-centric set $R(k)$ because they belong to the same community with k.

We consider each node in $R(k)$ and compute their betweenness link weights between node k: node a belongs to community (1) and its shortest path to node set $W_2 = \{b, c\}$, which is the intersection of $R(k)$ and community (2) is path a-b and it does not go through node k. Thus, from the definition of link weights, the link weight between a and k should 0. Similarly, the weight link between b and k is also 0 because the shortest path between b and $W_1 = \{a\}$, the intersection of $R(k)$ and community (1), also does not go through node k. However, if we consider node c, its shortest path to set $W_1 = \{a\}$ goes through node k. Thus, the link weight between c and k will be increased by 1 because there is only one node a in set $\{a\}$. Now we already have considered each node in $R(k)$ and the computation process stops. Thus, the contact frequency based community betweenness of k should be $0 + 0 + 1 = 1$.

Contact Duration Based (CDB), the only difference between CFB and CDB is that CDB considers the



(a) Before clone attack

(b) After clone attack

Fig. 2. The illustration of computing the betweenness.

cumulative contact times between node n and k. Thus, we also use the definitions in CFB and only introduce how to compute the link weights L_{nk} between node k and n: If there exists one shortest path from node n to W_i which also goes through node k ($k \in P_s(n, W_i)$), the weight of link L_{nk} will be increased by $|W_i| \times w_{nk}$, otherwise the link weight will not be changed. The w_{nk} is the cumulative contact times between node n and k. We also repeat the above process until all of such node n and all of its V_i have been considered to compute the CDB betweenness.

Thus, in Figure 1, the weight links of CDB between node c and k should be $1 \times 3 = 3$. Similarly, the weight links between a, b and k are 0 and the contact duration based betweenness of k is 3.

Shortest Path Based (SPB), SPB considers the number of shortest paths between node pairs which belongs to different communities from $R(k)$ instead of considering the shortest paths between node and node sets in the former betweenness definitions. We also use the definitions in CFB and only introduce the computation of link weight L_{nk} between node k and n: If there exist M_i ($M_i > 0$) nodes in W_i and each of them (e.g., node m_s) satisfies that the shortest path from node n to m_s goes through node k ($k \in P_s(n, m_s)$), the weight of link L_{nk} will be increased by $|M_i|$, otherwise the link weight will not be changed. We also repeat the this process until all of such node n and all of its V_i have been considered to compute the SPB betweenness.

Similarly, in Figure 1, we consider each node in $R(k)$ to compute the links weights: node a in community (1) has only one shortest path a-k-c which go through node k to node c in W_2, which belongs to the intersection of $R(k)$ and community (2). Thus, the weight link between k and a should be 1. Similarly, the weight between k and c should also be 1. The link weight between k and b is 0 because we can not find a shortest path which go through k between b and a node from W_1, the intersection of $R(k)$ and a different community from community (1).

3) Feasibility Study of Betweenness: We also illustrate the community betweenness computing in real traces and demonstrate the changes of betweenness before and after launching the clone attack in Figure 2. We extract the encounter events in an 8-hr period from the MIT reality [9] tracs and use the contact frequency based betweenness as an example to illusrate how to compute community betweenness value and how the community betweenness changes before and after the launching of attack.

The IDs of nodes which encounters with node 7 in this 8-hr time period are listed in Figure 2. The x-axis denotes nodes' IDs and the IDs of the nodes that encounters with node 7 during this time period are highlighted above the x-axis. For instance, in Figure 2 (a), nodes 21, 37, 51, 60 have encountered node 7 in this time period and thus their IDs are listed. The y-axis shows the community IDs and the green squares denote the communities which the nodes in node-centric set $R(7)$ belongs to. Each red circle denotes the community that a node at the x-axis belongs to. The blue dots and blue stars show whether there only exists one shortest path between a node (shown in the x-axis) and a community (whose identifier is the y-axis value) which goes through node 7 or not. For instance, node 51, a neighboring node of node 7 in Figure 2 (a), belongs to communities 4, 10 and 12 which node 7 also belongs to. Figure 2 also shows via the blue dots that there exists the shortest path between node 51 and community 5 or 13 which goes through node 7. However, there does not exist a shortest path between node 51 and community 1 which goes through 7 and hence we see a blue star.

Thus, using the definition of the contact-frequency based community betweenness, we can compute the betweenness value introduced by a community by multi-

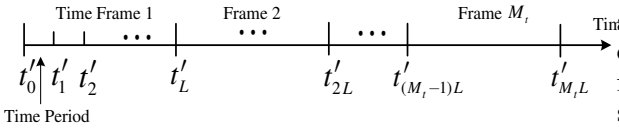


Fig. 3. The illustration of detection time frames

pying the number of red circles and blue dots. The total community betweenness can be computed by adding all the betweenness values from each row. In Figure 2, the contact frequency based community betweenness of node 7 before clone attack is $3 + 3 + 3 = 9$ and we also note that the community betweenness value of node 7 in that particular time period increases from 9 to 53 if a clone attack has been launched (by changing the encounter events of node 18 to be additional encounter events for node 7), which clearly shows that we can use this community betweeenness metric to detect the presence of a clone attack.

D. Detection Algorithm

As described in this section, the community betweenness of each node in contact graphs $G'_1 = (V'_1, E'_1)$, $G'_2 = (V'_2, E'_2)$, ..., $G'_W = (V'_W, E'_W)$ from non-overlapping time periods $[t'_0, t'_1]$, $[t'_1, t'_2]$,...,$[t'_{W-1}, t'_W]$ respectively will be computed. A straightforward method of detecting the clone attack is see whether there are any nodes with a community betweenness value which exceeds a pre-defined threshold T in a time period. However, a non-malicious user may also achieve high community betweenness in a particular time period. In order to improve the detection accuracy, the presence of a malicious user can be declared only a node has high community betweenness values for multiple time periods (say S) within a certain observation window that consists of L time periods.

Our detection algorithm works as follows: we consider L consecutive community betweenness values (computed from L time periods). If S out of L values exceed a pre-defined threshold T, then, we conclude that this node is suspicious of being engaged in a clone attack.

We first define a time frame which consists of L time periods and we divide the whole trace into M different non-overlapping time frames ($W = ML$) which are illustrated in Figure 3. We assume that there are S time periods in which the community betweenness are larger than the pre-defined threshold T during a time frame. Based on the extraction of time frames, our detection module searches through each time frame to determine whether S/L is larger than a threshold Ra. If the community betweenness computed in any time frame satisfies this criteria, node k will be listed as the malicious node.

V. Analytical threshold setting

The choice of the community betweenness threshold T is important for the performance of detecting clone attacks. However, the complexity and different characters of the social mobile network make it difficult to take a formal analysis on choosing a suitable T. Thus, in this section, we propose a contact model by considering users belonging to multiple communities, which is inspired by the work in [21] and the Watts and Strogatz model. However, [21] only considers users belonging to a single community and the Watts and Strogatz model did not take into consideration of the community concept. Based on the proposed contact model, we also analyze the probability distribution of community betweenness and choose a suitable threshold T in the detection under cartain confidence level $1 - p$. Simulations are then conducted to validate our threshold setting method through the synthetic traces generated by our proposed contact model.

A. Modeling of contact process

We first assume that some social communities existed between the participating nodes in mobile social network and the nodes within a community contact one another frequently. Thus, based on the small world graph model (SW)proposed in [21], we propose a new model by considering users belonging to multiple communities. The parameters in the model are illustrated as follows: N : total number of nodes in the network, K : number of nodes in one community, P : number of nodes that two communities have in common, Q : the interval between contact events, q : the probability that each node selects the peer from its community members in each contact event, $1 - q$: the probability that each node selects the peer from its non-community members in each contact event.

In our contact model, N nodes are numbered sequentially to be arranged as a ring and every K nodes are in the same social community. Considering some users may belong to multiple communities, we let each pair of neighboring communities in the ring have P nodes in common. To generate the sequence of contacts, each node selects the encountered nodes uniformly at random every Q seconds: with probability q, it selects the peer uniformly from its community members and with probability $1 - q$, it selects the peer uniformly at random from its non-community members.

An example of the contact model is illustrated in Figure 4: 76 nodes ($N = 76$) are arranged as a ring and they have been divided into 19 communities. Each pair of neighboring communities in the ring have 3 nodes ($P = 3$) in common. Thus, nodes $\{1, ..., 7\}$, $\{5, ..., 11\}$,..., $\{73, 74, 75, 76, 1, 2, 3\}$ belong to communities (1) to (19), respectively. The nodes with the dark color in Figure 4 denote that they only belong to one social community and the nodes with light color denote that they belong to two neighboring communities.

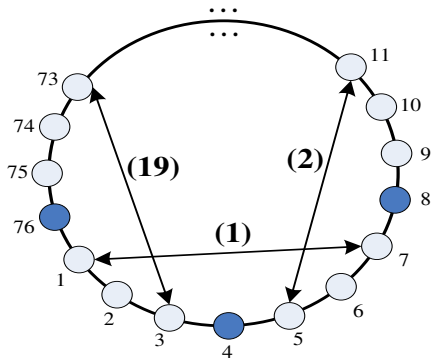


Fig. 4. An example of the social network model.

B. Probability Distribution of Community Betweenness

1) Introduction of Parameters: Based on our proposed social community model, the community betweenness of nodes which only belong to one social community is zero since nodes in their node-centric set can also only belong to one community. Thus, in this section, we focus on the nodes belonging to two communities and analyze their probability distribution of community betweenness. We consider a node k which belongs to two communities C_l and C_{l+1} (e.g., node $k = 5$ in Figure 4, $C_l = (1)$ and $C_{l+1} = (2)$) and after c contacts, its contact graph is illustrated in Figure 5.

We then divide nodes in communities C_l and C_{l+1} into five different sets $\{Y_i\}$ $(i = 1, ..., 5)$ as shown in Figure 5 and they are defined as follows:

- Y_1: the nodes only belong to community in C_l (e.g, node 4 in community (1)).
- Y_2: the nodes belong to two communities in C_l however they do not belong to C_{l+1} (e.g, node 1,2,3 in community (1)).
- Y_3: the nodes belong to both C_l and C_{l+1} except node k (e.g, node 6,7 in community (1) and (2)).
- Y_4: the nodes belong to two communities in C_{l+1} however they do not belong to C_l (e.g, node 9,10,11 in community (2)).
- Y_5: the nodes only belong to community in C_{l+1} (e.g, node 8 in community (2)).

We further define some node sets based on $\{Y_i\}$ $(i = 1, ..., 5)$ in the contact graph shown in Figure 5:

- S_i $(i = 1, ..., 5)$: the nodes which node k has contacted for at least one time in Y_i. In addition, the nodes in S_i also form the node-centric set $R(k)$ of node k.
- U_j $(j = 1, 2)$: the nodes in S_j which have never contacted with any nodes in S_3, S_4 nor S_5 after c contacts. Thus, the shortest paths between each node in U_j $(j = 1, 2)$ and set $W_{l+1} = C_{l+1} \cap R(k) = \bigcup_{i=3}^{5} S_i$ will go through node k.
- U_j $(j = 4, 5)$: the nodes in S_j which have never contacted with any nodes in S_1, S_2 nor S_3 after c contacts. Similarly, the shortest paths between each node in U_j $(j = 4, 5)$ and set $W_l = C_l \cap R(k) = \bigcup_{i=1}^{3} S_i$ will go through node k.

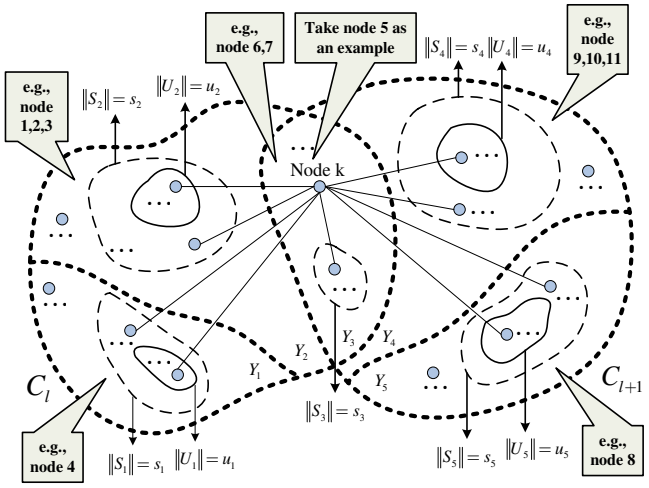


Fig. 5. Illustration of betweenness computing

We use the $\|*\|$ to denote the number of nodes in a set and assume that there are s_i and u_j nodes in set S_i and U_j respectively as shown in Figure 5.

2) Probability Distribution of Contact Frequency Based Community Betweenness: From the definitions of the community betweenness defined in Section IV and the contact graph shown in Figure 5, in this part, we try to analyze the probability distribution of the contact frequency based community betweenness for node k.

As the contact graph of node k shown in Figure 5, the contact frequency based community betweenness of node k is:

$$bet(k) = \left(\sum_{j=1}^{2} \|U_j\|\right)\left(\sum_{i=3}^{5} \|S_i\|\right) + \left(\sum_{j=4}^{5} \|U_j\|\right)\left(\sum_{i=1}^{3} \|S_i\|\right) \quad (7)$$

In equation 7, the first part $\left(\sum_{j=1}^{2} \|U_j\|\right)\left(\sum_{i=3}^{5} \|S_i\|\right)$ denotes the total betweenness link weights between node k and the nodes in U_j $(j = 1, 2)$; Similarly, the second part is the total link weights between node k and the nodes in U_j $(j = 4, 5)$. We also note that the link weights between node k and other nodes in $R(k) = \bigcup_{i=1}^{5} S_i$ are 0.

Before we compute the probability distribution of the betweenness, we first derive some contact probabilities, based on which we can compute the probability distributions. In the following analysis, we use "community nodes" of a node (e.g., node k) denote the nodes which are in the same community with node k, while the "non-community nodes" of a node (e.g., node k) denote the nodes which are not in the same community with node k. We note that for a node which only belongs to one community, there are $K-1$ community nodes and $N-K$ non-community nodes, while for a node which belongs to two communities, there are $2K - P - 1$ community nodes and $N - (2K - P)$ non-community nodes.

Thus, the probability that a node which only belongs to one community (e.g, nodes in Y_i $(i = 1, 5)$ in Figure 5) contacts with a specific community node for at least one time after c contacts is:

$$p_{sc} = 1 - (1 - p_s)^c \tag{8}$$

Where $p_s = \frac{1}{K-1}q$ and p_s is the probability that this node contacts with a community node in each contact event. Similarly, the probabilities that this node contacts with a non-community node after c contact events is:

$$p'_{sc} = 1 - (1 - p'_s)^c \tag{9}$$

Where $p'_s = \frac{1}{N-K}(1 - q)$ and p'_s is the probability that this node contacts with a non-community node in each contact event.

The probability that a node which belongs to two communities ((e.g, nodes in Y_i $(i = 2, 3, 4)$ in Figure 5)) contacts with a specific community node for at least one time after c contact events is:

$$p_{dc} = 1 - (1 - p_d)^c \tag{10}$$

Where $p_d = \frac{1}{2K-P-1}q$ and p_d is the probability of contacting with a community node in each contact event. Similarly, the probability that this node contacts with a non-community node after c contact events is:

$$p'_{dc} = 1 - (1 - p'_d)^c \tag{11}$$

Where $p'_d = \frac{1}{N-(2K-P)}(1 - q)$ and p'_d is the probability of contacting with a non-community node in each contact event.

From the analysis above, in Figure 5, because node k belongs to two communities, the probability that node k contacts with a specific community node from Y_i $(i = 1, ..., 5)$ for at least one time after c contacts should be p_{dc}. Thus, the probability that $\|S_i\| = s_i$ $(i = 1, ..., 5)$ is:

$$P\left(\|S_i\| = s_i\right) = \left(\begin{array}{c} \|Y_i\| \\ s_i \end{array} \right) (1 - p_{dc})^{\|Y_i\| - s_i} \left(p_{dc}\right)^{s_i} \tag{12}$$

Now we consider the nodes in S_i $(i = 1, 2, 4, 5)$ and analyze their contact process. The probability of a specific node in S_1 did not contact with any nodes in $W_{l+1} = \bigcup_{i=3}^{5} S_i$ after c contacts is:

$$p_1 = (1 - (p_s s_3 + p'_s s_4 + p'_s s_5))^c \tag{13}$$

The probability of a node in S_2 did not contact with any nodes in $W_{l+1} = \bigcup_{i=3}^{5} S_i$ after c contacts is:

$$p_2 = (1 - (p_d s_3 + p'_d s_4 + p'_d s_5))^c \tag{14}$$

Similarly, the probabilities of a specific node in S_4 or S_5 did not contact with any nodes in $W_l = \bigcup_{i=1}^{3} S_i$ after c contacts are:

$$p_4 = (1 - (p_d s_3 + p'_d s_1 + p'_d s_2))^c \tag{15}$$

$$p_5 = (1 - (p_s s_3 + p'_s s_1 + p'_s s_2))^c \tag{16}$$

Thus, from the analysis above we can also get the probability that $\|U_j\| = u_j$ $(j = 1, 2, 4, 5)$ is:

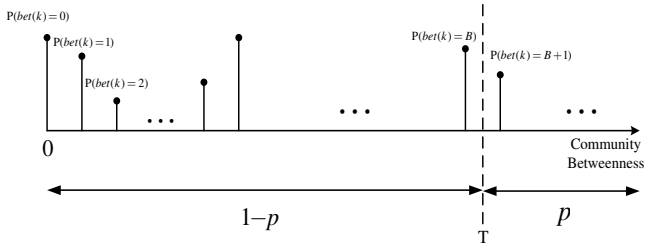


Fig. 6. Illustration of threshold setting

$$P\left(\|U_j\| = u_j | \, \|S_j\| = s_j\right) = \left(\begin{array}{c} s_j \\ u_j \end{array} \right) (1 - p_j)^{s_j - u_j} \left(p_j\right)^{u_j} \tag{17}$$

As a result, the probability distribution of node k's contact frequency based community betweenness is:

$$\begin{aligned} & P\left(bet(k) = B\right) \\ & = \sum_{\substack{\forall s_i, u_j: \\ bet(k)=B}} \left(\prod_{i=1}^{5} P\left(\|S_i\| = s_i\right) \prod_{\substack{j=1 \\ j \neq 3}}^{5} P\left(\|U_j\| = u_j | \, \|S_j\| = s_j\right) \right) \end{aligned} \tag{18}$$

The s_i $(i = 1, ..., 5)$ and u_j $(j = 1, 2, 4, 5)$ satisfy that: $0 \leq s_i \leq \|Y_i\|$ and $0 \leq u_j \leq s_j$, respectively. The betweenness value B can be derived from equation 7 as:

$$\begin{aligned} & bet(k) = \\ & \left(\sum_{j=1}^{2} \|U_j\| \right) \left(\sum_{i=3}^{5} \|S_i\| \right) + \left(\sum_{j=4}^{5} \|U_j\| \right) \left(\sum_{i=1}^{3} \|S_i\| \right) \\ & = (u_1 + u_2)(s_3 + s_4 + s_5) + (u_4 + u_5)(s_1 + s_2 + s_3) \\ & = B \end{aligned} \tag{19}$$

3) Threshold Setting and Evaluation: In this part, based on the probability distribution derived above, we first introduce our threshold setting method under certain confidence level $1 - p$. Then, we validate our proposed clone attack detection schemes by using the traces generated by our contact process model.

Threshold Setting: Provided the knowledge about the probability distribution we can determine the detectoin betweenness threshold T for each time period under a given confidence level $1 - p$. As Figure 6 illustrates, the threshold T in each time period will be chosen as:

$$\Pr(bet(k) \leq T) = \sum_{B \leq T} P(bet(k) = B) = F(T) = 1 - p \tag{20}$$

The $F(*)$ is the cumulative distribution function. It means that the probability that the community betweenness is less than T would be $1 - p$.

Evaluation: We use the following metrics to evaluate the effectiveness of our detection scheme: (a) detection ratio: it is defined as the percentage of clone nodes that are detected by the detection scheme; (b) false positive rate: it is the percentage of non-cloned nodes that are mistakenly detected.

	Number of Clone Nodes			
Scenarios	**1 node**	**2 nodes**	**3 nodes**	**4 nodes**
CFB+same	0.87	0.9	1	1
CFB+diff	1	1	1	1
CDB+same	0.8	0.83	1	1
CDB+diff	0.87	0.93	1	1
SPB+same	0.83	0.91	1	1
SPB+diff	1	1	1	1

TABLE I
DETECTION RATIO UNDER DIFFERENT SCENARIOS BY USING THE SYNTHETIC TRACE.

The SWIM trace [10] which simulate 3 days' human mobility in conference and university campus environments shares the similar statistical properties of real traces. To accurately approximate nodes' performance in SWIM trace, we set similar parameters in our contact model as the SWIM trace which is also shown in Figure 4: we also generate our trace which last for 3 days with 76 participants. The 76 participants are divided into 19 communities and each community has 7 nodes ($K = 7$). Two neighboring communities in the ring have 3 nodes in common ($P = 3$) and every 600 seconds ($Q = 600$), each node selects the peer from its community members with the probability 0.9 ($q = 0.9$).

In the simulation, we further divide the trace into 72 non-overlapping time periods and each of the time period is set as 1 hour. The confidence level is set as 0.95 with $p = 0.05$ for finding the threshold T in each time period. The length L of the time frame is set as $L = 12$ and ratio threshold Ra is set as $Ra = 0.5$.

The simulation results of detection ratio under different simulation scenarios are illustrated in Table I. The scenario column in Table I presents the different scenarios in our simulation. The "CFB", "CDB" and "SPB" respectively stand for the contact frequency based, contact duration based and shortest path based community betweenness we use in the detection schemes. The "same" or "diff" denote that the clone nodes are chosen from the same community or different communities, respectively. In the number of clone nodes column, we studied the detection ratio as a function of the number of clone nodes under all the scenarios. From Table I, the key observation is that our detection scheme consistently achieves a high detection ratio for each number of clone nodes. When comparing the results from each pair of "same" or "diff" rows, we also found that the higher detection ratio is achieved when the clone nodes are from different communities. Moreover, the detection ratio also increases when the number of clone nodes increases. This observation is also inline with our analysis before: clone nodes from different communities or larger number of clone nodes can bring more shortest paths which go through the node we considered. Turning to examine the false positive rate when detecting clone nodes, we found that the false positive rate of our scheme is zero for all the scenarios. These observations are very encouraging since our proposed scheme can detect most of the clone nodes with 0 false positive by using our threshold setting method. Overall, Our findings indicate that our threshold setting is suitable and our proposed method is feasible and effective in detecting clone nodes.

VI. TRAINING BASED THRESHOLD SETTING

The analysis in previous section is done for homogeneous mobile social networks where the sizes of different communities and the number of overlapped nodes are the same. However, in real life, we often have heterogeneous social networks where the community sizes and the number of overlapping nodes between different communities vary significantly. Thus, in this section we resort to using a training based method to choose an appropriate threshold for our clone attack detection scheme. By a training based method, we meant a method where one can analyze historical contact traces to determine typical community betweeenness values for nodes with similar encounter rates and determine appropriate detection thresholds for such nodes.

In the rest of this section, we first describe how we classify nodes in a mobile social network into different sets based on their encounter frequencies. Then, we describe our training based threshold setting scheme. Next, we present simulation results for different clone attack scenarios. The results show that our proposed social closeness based clone attack detection scheme achieves high detection rate with zero false positive rate.

A. Training based Threshold Setting

Before we describe the training based threshold setting method, we first discuss how we classify the nodes in a mobile social network into different sets based on their encounter frequencies. We assume that T_t hours of historical encounter-based traces are available for training purposes. We analyze the encounter frequencies of all the nodes in a mobile social network and divide these nodes into 3 sets as follows: (i) nodes that have the lowest 30% of encounter rates are classified as **non-active** nodes, (ii) nodes that have the highest 30% of encounter rates are classified as **active nodes**, and (iii) the remaining 40% nodes are considered **regular nodes**. If a node is from an active set, on the average, that node belongs to a community with an average size of 24, 7 of these community members are also active nodes while 10 of them are regular nodes. While the average community size of a regular ndoe is 13, 4 of which are active nodes and 6 of which are regular nodes.

Next, we divide the T_t hours of training data into W_t time periods and compute the average community betweenness values over these W_t time periods for each node in three node sets specified above. Suppose there are respectively L_a, L_r and L_n nodes in the active, regular and non-active sets, thus, the average community betweenness values for these three sets of nodes can be

represented as: $B_a = \{b_a^1, ..., b_a^{L_a}\}$, $B_r = \{b_r^1, ..., b_r^{L_r}\}$ and $B_n = \{b_n^1, ..., b_n^{L_n}\}$, respectively. Next, we determine the averages and standard deviations of these L_a, L_r, and L_n values, denoted as $AvgB_a$, $AvgB_r$, $AvgB_n$ and $2\sigma_a$, $2\sigma_r$,$2\sigma_n$ respectively. We then set the detection thresholds to be $AvgB_a + 2\sigma_a$, $AvgB_r + 2\sigma_r$, $AvgB_n + 2\sigma_n$ for these 3 classes respectively.

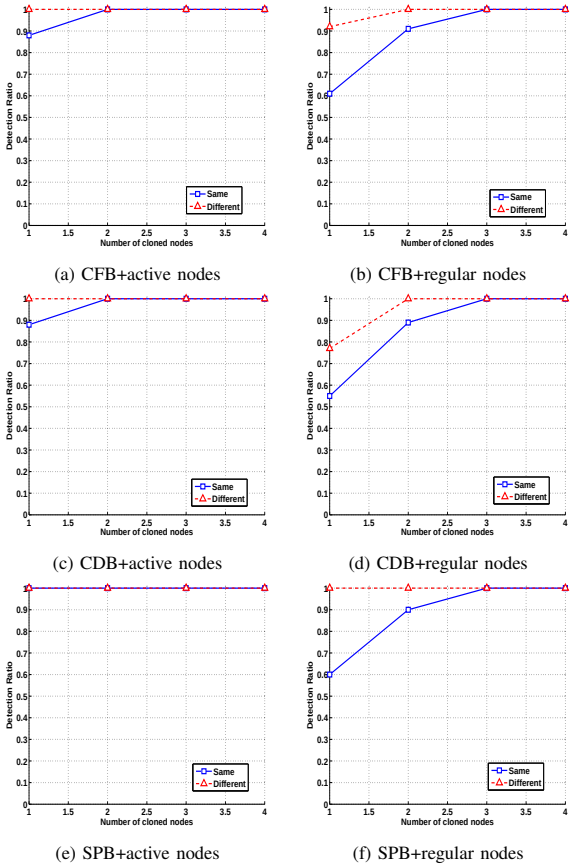
B. Simulation Methodology

To evaluate the effectiveness of our proposed scheme, we conducted simulations by using two human contact-based traces, namely the MIT reality [9] and SWIM traces [10]. The MIT traces which lasted for 20 days were collected from smart phones equipped with bluetooth devices carried by 97 participants in an university environment. The SWIM traces were generated from the SWIM mobility generator to mimic 76 participants' 3 days' human mobility traces in conference and university campus settings. Each trace contains information about the IDs of the Bluetooth devices which are within the transmission range of each other, and the starting and ending times of their encounters.

In the simulation, we divide the MIT trace into 60 non-overlapping time periods with each time period being 8 hours. Similarly, we divide the SWIM trace into 72 time periods with each time period being 1 hour. In both traces, we choose each time frame to contain $L = 12$ time periods, and set the ratio threshold Ra discussed in Section 4.4 to be 0.5. Furthermore, we use the first time frame in the MIT or SWIM trace to be our training data to determine the community betweenness thresholds, and the rest of the MIT or SWIM trace as the testing data to evaluate our clone detection approach. We conduct extensive simulations on these two sets of traces by varying the number of clone nodes chosen from the same node set or different node sets with the victim node from 1 to 4. Since nodes from non-active set seldom contact with other nodes, we only consider the nodes from either the active or regular sets. For each scenario, we randomly choose the qualified nodes and repeat the simulation 50 times. The simulation results for each scenario are the average results of 50 experimental runs. The detection ratio and false positive rate are used as metrics to evaluate the effectiveness of our detection schemes.

C. Effectiveness of our proposed social closeness method

1) Results from SWIM trace: In the first set of simulations, we evaluate the effectiveness of our proposed clone attack detection method in terms of the observed detection ratio and false positive rate at the end of our simulation using the whole SWIM trace. Figure 7 (a) and (b) depicted the results with both the victim and clone nodes belonging to the same node set(can be either active or regular) using the contact frequency based



(a) CFB+active nodes (b) CFB+regular nodes

(c) CDB+active nodes (d) CDB+regular nodes

(e) SPB+active nodes (f) SPB+regular nodes

Fig. 7. SWIM trace: Detection ratio under different number of clone nodes when the victim, and clone nodes are all from the same active or regular set

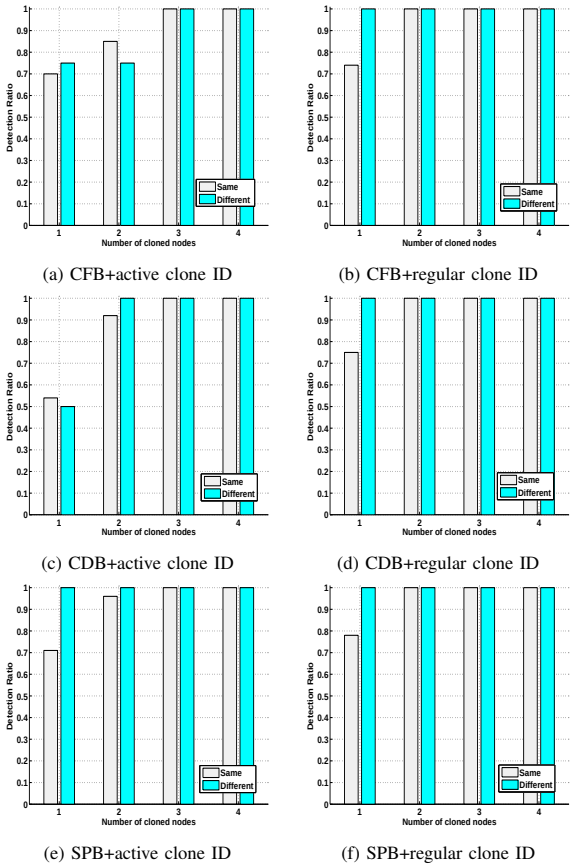
(CFB) betweenness variant in our clone attack detection scheme. Note that victim and clone nodes may either be in the same community or different communities. In Figure 7, the "active set" and "regular set" denote both victim and clone nodes are from active set and regular set, respectively. The "Same" and "Different" denote the victim and clone nodes are chosen from the same or different communities. While Figure 7 (c) and (d) and Figure 7 (e) and (f) presented the results using the contact duration based (CDB) betweenness and shortest path based (SPB) betweenness variants, respectively. We observed that the detection ratio improves as the number of cloned nodes increases. This is to be expected: more clone nodes increase the number of shortest paths which go through the victim node, which increases its observed community betweenness value.

In addition, comparing the results in Figure 7 when victim and clone nodes are from different communities with the results where they are from the same community, we found that the detection ratio is higher when the nodes are from different communities. Moreover, the higher detection ratio is achieved when both the victim node and clone nodes are from the active set. This indicates that our proposed approach is more effective

when the victim and clone nodes come from different communities or from the active set: this is also inline with our expectation because clone nodes from different communities or from the active set introduce more shortest paths which go through the victim node.

Next, we let the clone nodes and the victim node come from different node sets: the victim node is chosen from the active set while the clone nodes are chosen from the regular set, and vice versa. Figure 8 (a), (c) and (e) depicted the results when victim node is chosen from active set and clone nodes are from regular sets for CFB, CDB and SPB betweenness variants respectively, and vice versa in Figure 8 (b), (d) and (f): victim node is chosen from regular set and clone nodes are from active sets. As in Figure 7, we also found that the detection ratio improves when the number of clone nodes increases or when the clone nodes are from different communities. If we compare Figure 8 (a), (c) and (e) with (b), (d) and (f)respectively, we see that the detection ratio is higher when the victim node is chosen from the regular node set and the clone nodes are chosen from the active node set. This is because the active nodes have higher number of neighboring nodes in their node-centric sets, and thus more shortest paths exist. In addition, a lower detection threshold is also used if the victim node is a regular node. Compared Figure 8 with Figure 7, we observed that the detection ratio when the victim node and clone nodes are all from active set is higher than the detection ratio when the victim node is from regular and the clone nodes are from active. This is interesting and it seems not inline with our intuition because the regular node has a lower detection threshold. This can be explained as follows: each of the active nodes always belongs to a large number of communities. If all the clone nodes are from the active set, the number of communities the victim node belongs to can increase dramatically and thus the number of shortest paths which go through the victim node will also largely increase which results in a high community betweenness value.

Finally, we studied the detection ratio versus the number of clone nodes when the victim node and clone nodes are a mixture of active and regular nodes, which means that we always guarantee that there exists at least one clone node be chosen from a different set with the victim node. The detection ratio is the average over all the possible combinations and the results are tabulated in Table II. The "active" and "regular" in Table II denote that the victim node is chosen from active or regular set. While the "Same" and "Different" denote that the victim node and clone nodes are from the same or different communities. For example, in Scenario "active+same" with the number of clone nodes is 2, the victim node is from the active set while the two clone nodes are either all from regular set or one of them from regular set and another one from active set. From the results shown in Table II, we see that the results are consistent with what



(a) CFB+active clone ID (b) CFB+regular clone ID

(c) CDB+active clone ID (d) CDB+regular clone ID

(e) SPB+active clone ID (f) SPB+regular clone ID

Fig. 8. SWIM trace: Detection ratio under different number of clone nodes when original and clone nodes are mix of acitve and regular nodes

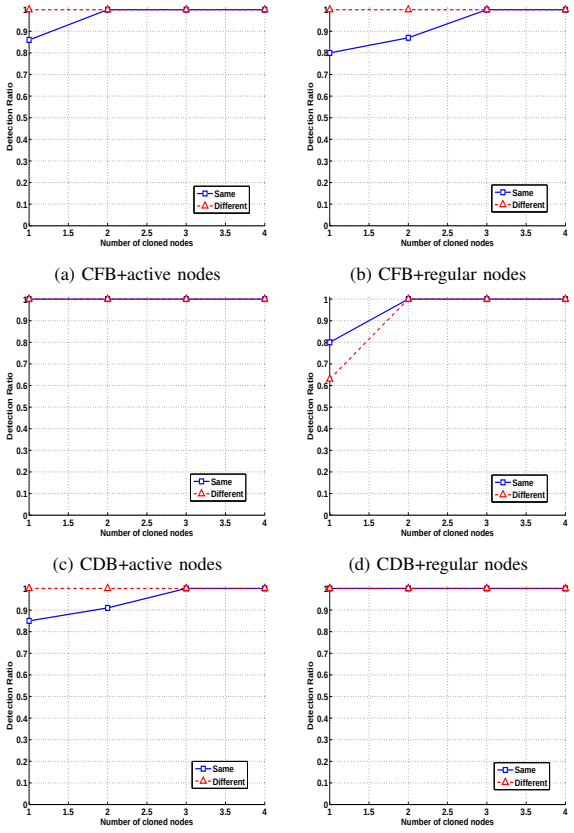
we observe from Figure 7 and Figure 8: higher detection ratio is achieved when there exist more clone nodes or when victim node and clone nodes are from different communities.

Furtermore, we observed that the false positive rate is zero for all clone attack scenarios, which is encouraging since our proposed clone attack detection scheme where the threshold setting is chosen based on historical training can detect most of the clone nodes with 0 false positive rate. Overall, these results suggest that our proposed approach is very effective in detecting the presence of clone nodes in a mobile social network.

2) Results from MIT trace: Finally, we repeat our study using the MIT traces. As in the SWIM trace, we also changed the number of clone nodes using different community betweenness variants. From the results depicted in Figure 9 using the CFB, CDB and SPB betweenness variants respectively, we observed that our proposed detection method exhibits the same trend as what we observed with the SWIM trace in Figure 7: it can achieve a higher detection ratio when the number of clone nodes become larger or when the victim and clone nodes are from different communities.

Scenarios		Number of Clone Nodes			
		1	2	3	4
CFB	active +same	0.7	0.92	1	1
	active+diff	0.75	0.87	1	1
	regular+same	0.74	0.98	1	1
	regular+diff	1	1	1	1
CDB	active+same	0.54	0.96	1	1
	active+diff	0.5	1	1	1
	regular+same	0.75	1	1	1
	regular+diff	1	1	1	1
SPB	active+same	0.71	0.98	1	1
	active+diff	1	1	1	1
	regular+same	0.78	0.98	1	1
	regular+diff	1	1	1	1

TABLE II
SWIM TRACE: DETECTION RATIO UNDER DIFFERENT NUMBER OF CLONE NODES BY AVERAGING OVER THE MIX OF ACTIVE AND REGULAR NODES.



(a) CFB+active nodes (b) CFB+regular nodes

(c) CDB+active nodes (d) CDB+regular nodes

(e) SPB+active nodes (f) SPB+regular nodes

Fig. 9. MIT trace: Detection ratio under different number of clone nodes when original and clone nodes are all from active or regular set

Next, as the simulations conducted in Figure 8, we also let the victim node, and the clone nodes be chosen from different node sets. The results which are shown in Figure 10 exhibit the same trend as we observed when using SWIM traces: Higher detection ratio is also achieved when there are more clone nodes or when the victim node and clone nodes are from regular set and active set respectively. The detection ratios when the victim node and clone nodes are a mixture of active and regular nodes are tabulated in Table III and these results are also consistent with our previous observations in Table II. These observations also show that our proposed method can achieve high detection ratio under different combinations of victim and clone nodes.

Overall, these observations suggest that our proposed detection approach is effective in detecting the presence of clone attacks under different sets of contact based traces.

We also conduct a sensitive analysis study where we vary the length of the time period for computing community betweenness value. We evaluated our proposed method using 8 different scenarios: (A)all the victim node and clone nodes are chosen from the active node category of different communities, (B) all the victim node and clone nodes are chosen from the active node category of the same communities, (C) all the victim node and clone nodes are chosen from the regular node category of different communities, (D) all the victim node and clone nodes are chosen from the regular node category of the same community, (E) the victim node and clone nodes are from active and regular set respectively of different communities, (F) the victim node and clone nodes are chosen from the active and regular categories respectively of the same community, (G) the victim node and clone nodes are chosen from active and regular set respectively of different communities, (H) the victim node and clone nodes are chosen from regular and active set respectively of different communities, (I) the victim node and clone nodes are chosen from regular and active set respectively of the same community. The detection ratios for all scenarios are tabulated in Table IV: we found that the detection ratio decreases when the time period inceases. This is because the number of shortest paths which go through a victim node k decreases when the contact graph is built using a longer period: the probability of two nodes contact increases as the length of time period increases.

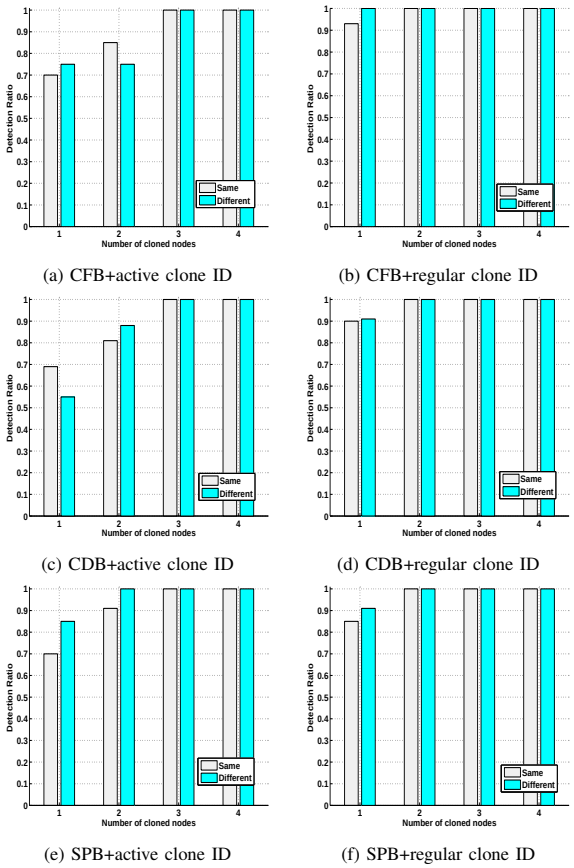
Similarly, the false positive rates for all these scenarios still remain zeros which illustrates that our proposed scheme is robust in detecting all clone nodes using our proposed training based threshold setting method. Overall, these observations suggest that our proposed approach is effective in detecting the presence of clone nodes in both SWIM and MIT traces.

VII. CONCLUSION

In this paper, we proposed a social community based method that exploits the social relationships to detect the clone attack in a mobile device enabled disease control framework. The concept of community betweenness is introduced by considering both the community and neighboring information of mobile users and the existence of a clone attack is identified by checking the community betweenness value in multiple time periods. To determine a suitable threshold for the community

		ID of Scenarios							
	Period Length	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
CFB	4 hours	1	0.81	0.9	0.66	0.71	0.65	0.91	0.71
	8 hours	1	0.86	1	0.8	0.79	0.75	1	0.93
	12 hours	0.9	0.79	0.8	0.61	0.75	0.61	0.9	0.82
CDB	4 hours	1	0.79	0.63	0.75	0.66	0.51	0.87	0.8
	8 hours	1	1	0.63	0.8	0.55	0.69	0.91	0.9
	12 hours	1	0.71	0.66	0.5	0.5	0.43	0.72	0.71
SPB	4 hours	0.8	1	0.84	0.7	0.79	0.66	0.88	0.73
	8 hours	1	0.85	1	1	0.85	0.7	0.91	0.85
	12 hours	0.82	0.75	0.79	0.74	0.75	0.68	0.82	0.78

TABLE IV
MIT TRACE: DETECTION RATIO UNDER DIFFERENT LENGTH OF TIME PERIODS.



(a) CFB+active clone ID (b) CFB+regular clone ID

(c) CDB+active clone ID (d) CDB+regular clone ID

(e) SPB+active clone ID (f) SPB+regular clone ID

Fig. 10. MIT trace: Detection ratio under different number of clone nodes when original and clone nodes are mix of acitve and regular nodes

betweenness in the detection process, in this paper, we developed two methods, analytical based and training based, to find a suitable community betweenness threshold in the clone attack detection. Through extensive simulations under different trace sets, we show that by considering both the neighboring and social community information, our proposed method can detect clone attacks efficiently with high detection ratio and low false positive rate. Our results also demonstrate the feasibility of exploiting the social community information derived from mobile devices for detecting the clone attacks.

		Number of Clone Nodes			
Scenarios		**1**	**2**	**3**	**4**
CFB	active+same	0.75	0.91	1	1
	active+diff	0.79	0.89	1	1
	regular+same	0.93	1	1	1
	regular+diff	1	1	1	1
CDB	active+same	0.69	0.9	1	1
	active+diff	0.55	0.89	1	1
	regular+same	0.9	1	1	1
	regular+diff	0.91	1	1	1
SPB	active+same	0.7	0.95	1	1
	active+diff	0.85	1	1	1
	regular+same	0.85	1	1	1
	regular+diff	0.91	1	1	1

TABLE III
MIT TRACE: DETECTION RATIO UNDER DIFFERENT NUMBER OF CLONE NODES BY AVERAGING OVER THE MIX OF ACTIVE AND REGULAR NODES.

REFERENCES

[1] J.Scott, *Social Network Analysis: A Handbook*. Sage Publication Ltd, 2000.

[2] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proceedings of the National Academy of Sciences of the United States of America*, June 2002.

[3] M. A. Kazandjieva, J. W. Lee, M. Salath, M. W. Feldman, J. H. Jones, and P. Levis, "Experiences in measuring a human contact network for epidemiology research," in *Proceedings of the ACM Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets)*, 2010.

[4] Y. Ren, J. Yang, M. C. Chuah, and Y. Chen, "Mobile phone enabled social community extraction for controlling of disease propagation in healthcare," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (IEEE MASS), concise paper*, 2011.

[5] R. R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key predistribution." *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, pp. 1246–1258, 2007.

[6] H. Choi, S. Zhu, and T. F. La Porta, "Set: Detecting node clones in sensor networks," in *Security and Privacy in Communications Networks and the Workshops, 2007.*, sept. 2007, pp. 341 –350.

[7] K. Xing, F. Liu, X. Cheng, and D. H. C. Du, "Real-time detection of clone attacks in wireless sensor networks," in *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems*.

[8] L. Jin, H. Takabi, and J. B. Joshi, "Towards active detection of identity clone attacks on online social networks," in *Proceedings of the first ACM conference on Data and application security and privacy*.

[9] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *In Personal and Ubiquitous Computing*, vol. 10, no. 4, 2005.

[10] M. V. Barbera and A. Mei, "Personal marks and community certificates: Detecting clones in mobile wireless networks of smart-phones," *CoRR*, 2011.
[11] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*.
[12] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 2005, pp. 49–63.
[13]
[14] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *26th IEEE International Conference on Computer Communications, Anchorage, Alaska, USA*, 2007, pp. 1937–1945.
[15] P. Kyasanur and N. H. Vaidya, "Detection and handling of mac layer misbehavior in wireless networks," *IEEE DSN*, 2003.
[16] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *IPSN'04*, 2004, pp. 259–268.
[17] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proc. of IEEE INFOCOM*, 2005, pp. 902–913.
[18] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Mobile sensor network resilient against node replication attacks," in *Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008, pp. 597–599.
[19] S. Huang, "Probabilistic model checking of disease spread and prevention," in *Scholarly Paper for the Degree of Masters in University of Maryland*, 2009.
[20] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," in *Proceedings of IEEE International Conference on Data Mining(ICDM)*, 2009.
[21] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know thy neighbor: towards optimal mapping of contacts to social graphs for dtn routing," in *Proceedings of the 29th conference on Information communications*.