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 inclusion of embedded sensors in mobile phones, and the explosion of their usage in people's daily lives provide users with the ability to collectively sense the world. The collected sensing data from such a mobile phone enabled social network can be mined for users' behaviors and their social communities, and to support a broad range of applications including mobile healthcare systems. However, such mobile healthcare systems built upon social networks are vulnerable to clone attacks, in which the adversary replicates the legitimate nodes and distributes the clones throughout the network to undermine the successful application deployment. Existing clone attack mitigation approaches either only focus on the prevention techniques or can only work in static or well-connected networks, and hence are not applicable to our targeted mobile healthcare systems. In this paper, we propose a social closeness based method in a mobile healthcare disease control system to detect any clone attacks that may be launched to disrupt the normal operations of the system. Our social closeness based method exploits the social relationships among users for clone attack detection. Specifically, we define a new metric called community betweenness, which considers mobile users' community information. We find that the value of this metric changes significantly under the clone attack, which is suitable to be used for clone attack detection. We derive both analytical and training based approaches to determine the threshold setting of the community betweenness for robust clone attack detection. Extensive trace-driven simulation studies reveal that our social closeness based method can detect clone attacks with high detection ratio and low false positive rate.**

I. INTRODUCTION

Mobile phones have become increasingly popular and play significant roles in our daily lives. In particular, with the rapid deployment of sensing technology in mobile devices, the collected sensing data can be mined for understanding human behaviors. For example, we can use mobile devices equipped with Bluetooth technology to discover the encounter events between people such that their social relationships can be derived and analyzed. Information about users' social relationships can assist in the development of applications in various domains including healthcare applications. For instance, the discovered social relationships can be used to extract social communities [1], [2], which reflect either close relationships or similar behavioral patterns among people. Such extracted social community information provides new opportunities for epidemiology research and facilitate the control of disease spread in the healthcare domain [3], [4]. Furthermore, information about social communities extracted from human contact traces can also be used in other mobile healthcare systems to determine how socially active senior citizens are.

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. The existence of the cloned devices may not affect the network performance, however, they aim to have significant impact on pervasive mobile healthcare systems. For example, in the mobile healthcare systems presented for facilitating epidemiology research and disease propagation control [3], [4] which utilize the social community information, the adversary can attract more vaccine allocation to certain geographical areas by deploying many cloned devices, and thus undermine the regular operations of social community-based mobile healthcare systems. Therefore, detecting the presence of clone attacks is important to support the wide deployment of emerging mobile healthcare systems.

Nevertheless, detecting clone attacks is not an easy task. The cloned devices have access to all the security information of the compromised device and hence they can pass all security checks without being detected. Existing techniques for mitigating malicious attacks in sensor networks mostly focused on attack prevention through key distribution schemes [5]. These prevention-oriented schemes are ineffective [6] in preventing the launching of clone attacks since cloned devices have access to the legitimate information and can thus fool other legitimate devices without being detected. Recently, new techniques utilizing the neighbors information [7], [8] have been proposed for detecting clone attacks in wireless sensor networks. However, they can be applied either only to static networks or for online social networks, making them less suitable for mobile healthcare systems.

In this work, we take the view point of exploiting users' social relationships extracted from human contact-based traces for detecting clone attacks in mobile healthcare systems. Our design goal is to enable clone attack detection without relying on other hardware such as

GPS, which may drain a phone's battery quickly and their readings can also be easily hacked [9]. The basic idea is that a user often belongs to one or two static communities during a certain time of period, and the community membership often remains unchanged during that time period. However, the presence of cloned users cause that victim user to belong to multiple distinct social communities simultaneously, contradicting that user's regular social behaviors. Thus, we propose a social closeness based approach that can detect clone attacks in a mobile healthcare system even with colluded cloned users. To the best of knowledge, our work is the first that utilizes social closeness to detect the clone attack for the emerging mobile healthcare systems.

In particular, our attack detection scheme develops a new metric called *community betweenness* for distinguishing between legitimate and clone users. This metric, computed based on mobile users' social community information, changes significantly during clone attacks. Three variants are developed to compute this community betweenness value, namely *contact-frequency based*, *contact-duration based*, and *shortest-path based*. To make the detection more robust, we present an analytical-based approach for determining the threshold setting of the community betweenness value for a homogeneous mobile social network such that our detection scheme can achieve a desirable detection rate with high confidence. We then describe a training-based method for determining such a threshold for heterogeneous mobile social networks where the sizes of the communities vary widely.

We show the effectiveness of our detection method using the disease control mobile healthcare system proposed in [3], [4]. We evaluated our detection scheme via simulations using the SWIM trace and the MIT reality mining trace [10], [11]. The results showed that our scheme is highly effective for detecting various clone attacks in this mobile healthcare system. Our technique can also be applied to other mobile healthcare systems that utilize human contact-based traces.

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 the disease propagation control framework (which we focus on as a case study) in the mobile healthcare system and the attack model in Section III. We next present our social community based detection scheme in Section IV with the community betweenness metric. We describe our analytical-based and training-based approaches for robust detection in Section V. In Section VI, we validate the feasibility of our proposed detection scheme using the trace-driven approach. Finally, we conclude our work in Section VII.

II. Related Work

To thwart clone attacks, using tamper-resistant hardware for mobile nodes appears effective. However, approaches with tamper-resistant hardware may involve high cost and are hard to be applied to already deployed mobile nodes. Furthermore, an attacker may still be possible to bypass the tamper-resistant hardware to extract secret keys from captured nodes given enough time and computation ability even though the tamper-resistant hardware makes it much harder to accomplish this [12]. Thus, it is desirable to seek low cost detection methods that do not require any hardware changes to cope with clone attacks.

Along this direction, several software-based clone attack detection schemes have been proposed for sensor networks [7], [13]. The basic idea of these schemes is to let each node report its neighboring information and attempt to find conflicting reports. However, these existing methods can only be applied to static networks and cannot be used in networks with mobility. In [14], when a node announces its location, each of its neighbors sends a copy of the location claim to a set of randomly selected witness nodes. Any conflicting location claims will provide the evidence for clone attack detection. However, the location information of nodes is needed constantly for this scheme to work but such information may not always be available. In our situation, the nodes may be sparsely connected in the mobile social network, it can be hard to find enough nodes as witnesses in the clone attack detection.

Schemes based on having neighboring nodes vote for checking the sanity of a given node based on their local observations have been developed [15], [16]. However, these schemes are not capable of detecting clones if cloned nodes move only in close proximity with the victim node, and may also fail when multiple clone nodes in close proximity collude. Our work is different in that we aim to detect the presence of the clone attack by using the extracted social community information enabled by the widely used mobile phones rather than adding additional overhead on mobile users. Our approach can smartly detect clone attacks when multiple users collude to trick the system. Our work is the first to address the clone attack problem in mobile healthcare systems which exploits social community information to make either control decisions (e.g. for effective vaccine distribution and reduce disease propagation rate) or diagnosis (e.g. whether seniors are depressed or socially active).

III. Model Assumptions

In this section, we first describe the system model of a mobile healthcare system that is used in this work. We then present a high level overview of dynamic community extraction method utilized in our mobile healthcare system and describe our threat model.

A. System Model

In this work, we employ the mobile healthcare system that provides effective vaccine distribution by utilizing the social community information extracted from mobile phone traces [4] and we refer the social community as subsets of users within which user to user connections are dense, but between which connections are less dense [2]. The mobile healthcare system can be deployed for a geographical region, e.g. a town or a city. Instead of the traditional 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.

In our mobile healthcare system, each person has a unique user identifier (e.g., phone number) and owns a mobile device equipped with Bluetooth technology. Each user then registers with the system by using his/her identifier and mobile device's Bluetooth ID. Encounter traces collected by Bluetooth technology are sent to the system server by each user and the server can derive encounter events between users according to the registration information in the system. The *dynamic community extraction* scheme is applied and the extracted social community information is then used in the decision process for targeted vaccine distribution. When a new disease is discovered [17], those people within the same communities as a sick person have higher risks of being infected, and thus should receive disease alert or vaccination messages sent by the server.

B. Preliminaries of Dynamic Community Extraction

In general, people may belong to different social communities during different time periods. A social community may appear several times during a day or week. Instead of directly extracting communities from the contact graphs constructed by the contact events, the basic idea of the dynamic community extraction method we are using is to first extract the communities for each non-overlapping time period and then merge those communities with high similarity [4].

In particular, a contact graph $G = (V, E)$ consists of a vertex set V and an edge set E. Each vertex denotes a person, while each edge weight denotes how frequent two persons meet during a time period. Each contact graph is a representation of people's relationships where their relationship strength is derived based on the frequency of their encounters. The dynamic community extraction scheme constructs R contact graphs $G_i = \{(V_i, E_i), i = 1, ..., R\}$ from R non-overlapping periods. Then, it extracts multiple communities from each contact graph G_i using the hierarchical clustering algorithm [1], and the modularity metric Q [18]. The community set extracted from each time period is denoted as AS_i. Assume that each AS_i has k_i communities as follows:

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

Furthermore, we compare each community in AS_i with all the communities discovered in AS_{i+1} to see if a community in AS_i can be merged or removed based on one of the following conditions:

- It is part of a bigger community in AS_{i+1} and hence can be removed.
- It can be merged with one community in AS_{i+1} using the community merge operation for two communities A_i^j and A_{i+1}^l with an adjustable community threshold τ_c:

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

- It is a superset of a community A_{i+1}^j in AS_{i+1}, then A_{i+1}^j is removed from set AS_{i+1} .

At the end of such operation, the two sets AS_i and AS_{i+1} are unioned to form a new $AS_{i+1}^{'}$, which will be merged with AS_{i+2} in the next round. The merging process iterates through R time windows to yield the final M social communities: $AS = \{A_j, j = 1, ..., M\}$.

C. Attack Model

The mobile healthcare systems are vulnerable to clone attacks [3], [4], [17]. An adversary can capture mobile devices equipped with Bluetooth capability within the network and deploy unlimited number of clone devices with duplicated Bluetooth IDs. Since these replicas have legitimate IDs, the users carrying such devices can participate in the mobile social network and thus significantly affect the effectiveness of these mobile healthcare systems.

Particularly, the adversary can distribute devices with cloned Bluetooth IDs to a group of users from the same geographical area who share similar interests with him. These users can collude with the adversary to launch a clone attack so as to gain some benefits. For example, when there is an outbreak of a new infectious disease with limited vaccine supply, some pharmacy companies or clinics may be interested in having more vaccine shots allocated to their geographical area e.g. county or town. They can distribute devices with cloned IDs to different colluded users. A registered user with cloned IDs may end up belonging to many social communities. If he is sick, users who are in the same social community with him will have higher chances of being selected to receive vaccinations. If an adversarial company replicate many clone IDs which belong to legitimate registered users, the number of vaccines allocated for this geographical area will increase significantly. This attack strategy is especially harmful when there is only a limited supply of expensive vaccine shots.

IV. Clone Attack Detection

In this section, we first describe our new metric of community betweenness. We show how to calculate the community betweenness values by providing three methods. We then present our clone attack detection criteria based on the community betweenness metric.

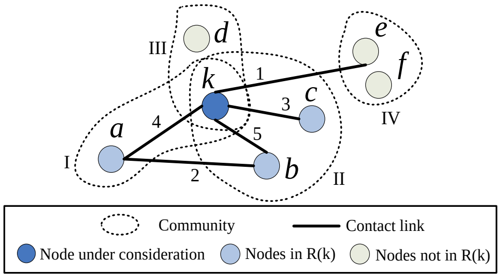


Fig. 1. An example of computing community betweenness.

A. Overview

Our clone attack detection method is based on the observation that users of mobile devices typically belong to a number of stable communities. By stable, we mean the communities that a user belongs to do not change rapidly after that user's community profile has been constructed based on observations for a sufficient long time period. However, when a clone attack is launched, the number of communities a user with cloned device ID belongs to may increase significantly within a short time period, and hence such observation can be used for clone attack detection.

To capture this observation in a quantitative way, we propose the concept of *community betweenness* inspired by the idea that the betweenness of a node in a network graph is defined as the number of shortest paths between pairs of other vertices which go through that node. [2] has used betweenness as a measure of the centrality and influence of a node in social or biological networks. Thus, in order to find out which nodes in a social network are always "between" different social communities for clone attack detection, we generalize the betweenness concept to community betweenness. The community betweenness of a node is defined as the number of shortest paths between pairs of communities that go through that node in the contact graph. The basic idea is that if there are users who share the same cloned ID in a social network and these cloned nodes belong to different communities, then almost all shortest paths between these communities in the contact graph may go through this cloned ID. Thus, this particular cloned ID will have a higher community betweenness value comparing to normal situations. More formal definition of community betweenness is given in Section IV.B.

B. Definitions

We next define some basic concepts for computation of the community betweenness: neighbor set, the shortest path between a node and a node set, the distance between a node and a node set, and the node-centric set in the contact graph.

We consider a mobile network in which each node is assigned with a unique identifier. We assume that these nodes represent mobile devices having sensing capabilities, e.g., Bluetooth discovery that allows encounter events between owners of devices to be recorded as a contact-based trace. As in Section III, we construct R contact graphs $G_i = \{(V_i, E_i), i = 1, ..., R\}$ for the R non-overlapping time periods. In each contact graph, each node represents a user (identified by a unique identifier), and edges that connect the nodes represent some encounter events between these two users. The weight of each edge represents the cumulative contact frequency of these encounters in a time period. Such a contact graph is a geometric representation of the relationships between users.

By utilizing the final dynamic communities $AS = \{A_j, j = 1, ..., M\}$ extracted in Section III, we define the following concepts for a node k in a contact graph G_j:

Definition 1. Neighbor set $N(k)$: The collection of node k's 1-hop neighbors in a contact graph is defined as its neighbor set $N(k)$.

Definition 2. Shortest path between nodes: If node k and node d are connected within the contact graph, the shortest path $P_s(k, d)$ between them is defined as the path with the smallest hop length from node k to node d, otherwise, $P_s(k, d) = \infty$.

Definition 3. Shortest path between a node and a node set: Let $\|*\|$ be the number of nodes in a set and $\{d_1, ..., d_{\|D\|}\}$ be the nodes in a set D, the shortest path $P_s(k, D)$ from a node k to node set D is defined as:

$$P_s\left(k, D\right) = \mathop{\arg\min}_{P_s(k, d_l)} \left|P_s\left(k, d_l\right)\right|, d_l \in \left\{d_1, ..., d_{\|D\|}\right\} \tag{3}$$

Definition 4. Distance between a node and a node set: The distance between node k and node set D is defined as the hop length of the shortest path between k and D:

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

Definition 5. Node-centric set: Suppose node k belongs to M_k different social communities: $\{A_i \,| k \in A_i, i = 1, ..., M_k\}$. The node-centric set $R(k)$ of node k is defined as: $R(k) = N(k) \cap (\bigcup_{i=1}^{M_k} A_i)$. In other words, node k's node-centric set includes its neighboring nodes which belong 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 shortest paths going through k between each node in R(k) and any community which this node does not belong to in R(k). To compute the community betweenness, we need to define the weight of the betweenness link between k and a node n in $R(k)$. This link weight considers the number of shortest paths going through k between node n and other communities in $R(k)$ that node n does not belong to. Then, the community betweenness of node k is the sum of all the betweenness link weights between k and each node in $R(k)$. Typically, whether two persons

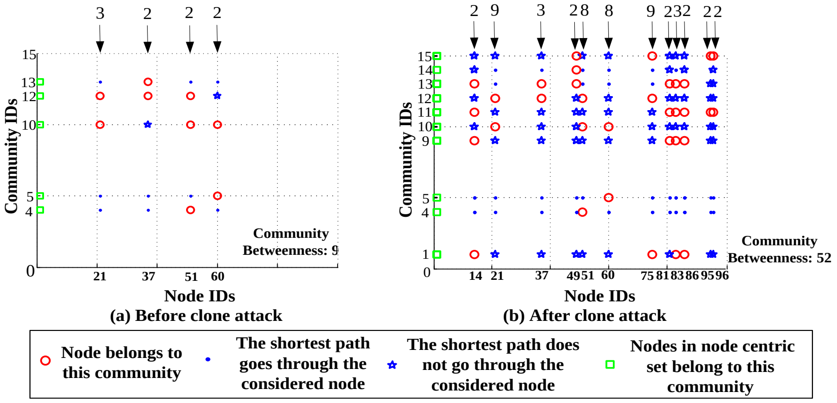


Fig. 2. The illustration of calculating community betweenness of node 7 in MIT Reality trace using Contact Frequency Based (CFB) method.

meet or not determines how likely one person can spread a disease to another. But sometimes, the larger the number of encounters between two persons, the higher the chance of one person spreading a disease to another. Thus, we define two variants on how the community betweenness value is computed: (a) Contact Frequency Based and (b) Contact Duration Based. In addition, we further define another variant (c) Shortest Path Based, where the community betweenness value is defined as the number of shortest paths going through k between any node n in $R(k)$ and nodes in other communities which node n does not belong to in $R(k)$.

2) Variants of Betweenness Computation: The main difference of the three variations is: (a) the Contact Frequency Based betweenness utilizes the size of the intersection set of $R(k)$ and each community in its link weight calculation; (b) the Contact Duration Based betweenness relies on both the size of intersection set and the cumulative contact times between nodes during the link weight calculation; and (c) the Shortest Path Based betweenness considers the number of shortest paths between nodes from different communities in $R(k)$ that go through node k for its link weight computation. Details of each variant are given below. Initially, the link weight between node k and each node in $R(k)$ is set to 0.

Contact Frequency Based (CFB). If $\exists A_i$, $n \notin A_i$ and $W_i = A_i \cap R(k) \neq \emptyset$ where $n \in R(k)$, we then compute the link weight L_{nk} between node k and n as follows: If there exists one shortest path going through node k from node n to set W_i ($k \in P_s(n, W_i)$), the 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. This process is repeated for all n's qualifying communities A_i to obtain the final value of the link weight L_{nk}. The community betweenness of node k is then the sum of all the link weights between k and each node in $R(k)$:

$$bet(k) = \sum_{n \in R(k)} L_{nk} \tag{5}$$

We next give an example to show how the CFB betweenness value is computed: A contact graph of node k is shown in Figure 1 with nodes $\{a, k\}$, $\{k, c, b\}$, $\{k, d\}$ and $\{e, f\}$ belong to communities I-IV respectively and the communities are shown as dotted circles. The cumulative contact frequency in a time period between each pair of neighboring nodes are also shown in Figure 1. From the definitions above we see that k's neighbors a, b and c are in node-centric set $R(k)$ because they belong to the same community as k. We consider each node in $R(k)$ and compute its link weight between node k: node a belongs to community I and its shortest path to node set $W_2 = \{b, c\}$ (i.e., the intersection of $R(k)$ and community II) is path a-b and it does not go through node k. Thus, from the definition of link weight, the link weight between a and k should be 0. Similarly, the weight link between b and k is also 0 because the shortest path between b and $W_1 = \{a\}$ (i.e., the intersection of $R(k)$ and community I) 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 W_1. By now, we have considered every node in $R(k)$ and thus the computation process stops. Thus, the contact frequency based community betweenness of k should be $0 + 0 + 1 = 1$.

Contact Duration Based (CDB). The computation difference of the CFB and CDB community betweenness lies in the link weight computation. If there exists one shortest path which goes through node k ($k \in P_s(n, W_i)$) from node n to W_i, the link weight L_{nk} will be increased by $\|W_i\| \times w_{nk}$. where w_{nk} is the cumulative contact times between node n and k in this time period.

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

Shortest Path Based (SPB). This method considers the shortest paths between pairs of nodes from different communities in $R(k)$ instead of considering the shortest paths between a node and node sets as in the former two variants. The computation of L_{nk} for each qualified community W_i is carried out as follows: if there exist m_i

($0 < m_i \leq \|W_i\|$) nodes in W_i and each of them (node g) satisfies the condition that the shortest path from node n to g goes through node k ($k \in P_s(n,g)$), then the link weight L_{nk} will be increased by m_i.

Using the same example in Figure 1, node a in community I has only one shortest path a-k-c which goes through node k to W_2 (i.e., the intersection of $R(k)$ and community II). Thus, the link weight 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 cannot find a shortest path which goes through k between b and a node in W_1 (i.e., the intersection of $R(k)$ and community I). Thus, the SFB based betweenness value for node k is 2.

3) Feasibility Study of Betweeness: Figure 2 illustrates how the CFB community betweenness value changes when a clone attack involving a particular node (node 7) happens within the MIT [11] reality trace. Due to the space limitation, we only show the results of CFB. The similar trend is observed when using CDB and SPB. In Figure 2, we show the neighbors and community information of node 7 in an eight-hour time period by the Node IDs versus Community IDs.

In the example depicted in Figure 2, the community IDs marked as green squares represent the communities which the nodes in $R(7)$ belong to. The nodes depicted in red circles represent the community IDs that a node in $R(7)$ belongs to. The blue dots and blue stars show whether or not there exists a shortest path which goes through node 7 between a node n in $R(7)$ and the intersection set of $R(k)$ and another community that node n does not belong to . For instance, in Figure 2 (a), node 37 belongs to communities 12 and 13 as shown by the red circles. It also shows that the shortest path between node 37 and the intersection set of $R(k)$ and communities 4,5 goes through node 7 while the shortest path between node 37 and the intersection set of $R(k)$ and community 10 does not go through node 7.

Thus, using the definition of the Contact-Frequency Based (CFB) community betweenness, we can compute the betweenness link weights between each node n in $R(7)$ and node 7 by adding the number of red circles in the rows which have blue dots in the corresponding column of node n. The total community betweenness can be computed by adding the link weights together. Figure 2 (a) shows that the CFB community betweenness of node 7 is 9 prior to a clone attack. After the clone attack is launched, the CFB community betweenness increases from 9 to 52 as Figure 2 (b) shows. The changes of CFB community betweenness in Figure 2 confirm the feasibility of detecting the clone attack by using community betweenness.

D. Detection Criteria

A straightforward method of detecting a clone attack is to determine whether there exists any node with a community betweenness value which exceeds a predefined threshold during a time period. However, an uncloned node may also have a high community betweenness value in a particular time period. In order to make the detection more robust, the presence of a cloned node will be declared only when a node has community betweenness values that exceed a pre-defined betweenness threshold, τ_b for multiple (i.e., S) time periods within an observation window which consists of L time periods. Our detection module searches through each consecutive observation window to declare the presence of the clone attack when the ratio S/L is larger than a (ratio) threshold τ_r.

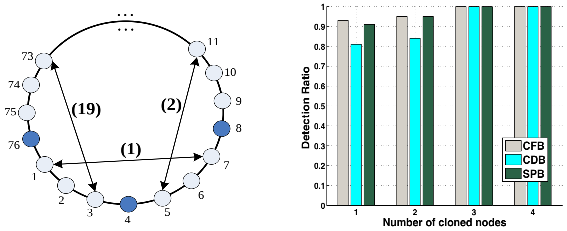
V. Robust Detection

The choice of the community betweenness threshold τ_b is critical for the performance of our clone attack detection scheme. However, the complexity and varying characteristics of the mobile social networks make it difficult to provide a generic analytical framework for determining a suitable τ_b. In this section, we show two possible approaches to determine τ_b such that our clone attack detection scheme yields robust detection results: an analytical approach for homogeneous social networks and a training based method for heterogeneous social networks.

A. Analytical Approach

1) Modeling of the Contact Process: In a homogeneous social network, the size of each community is the same and the number of nodes which two communities have in common is also fixed. We propose a new contact model by considering scenarios where nodes belong to multiple communities, inspired by the small world graph model [19] and the Watts and Strogatz model [20]. We note that the small world graph model [19] only considered users belonging to a single community and the Watts and Strogatz model did not take into consideration the community concept. In our contact model, N nodes are numbered sequentially and arranged in a ring. Every consecutive K nodes within a ring are in the same social community. Since one user may belong to multiple communities, we let each pair of neighboring communities in a ring have P common nodes. To generate the sequence of encounter events, each node selects the encountered nodes uniformly at random every Q seconds: with probability q, it selects a peer uniformly from its community members, whereas it selects a peer uniformly at random from its non-community members with probability $1-q$.

An example of the contact model is illustrated in Figure 3 (a): 76 nodes ($N = 76$) are arranged in a ring and they have been divided into 19 communities. Each community has 7 nodes ($K = 7$) and each pair of neighboring communities in the ring have 3 nodes ($P = 3$) in common. Thus, nodes $\{1, ..., 7\}$, $\{5, ..., 11\}$,...,



(a) Social network model.

(b) Simulation results.

Fig. 3. An example of the social network model and simulation results.

$\{73, 74, 75, 76, 1, 2, 3\}$ belong to communities (1) to (19), respectively. The nodes marked with the dark color in Figure 3 only belong to one social community and the nodes marked with the light color belong to two neighboring communities.

Based on our proposed contact model, we first derive the probability distribution of community betweenness $bet(k)$ under normal situations and the detailed description of our analysis may be found in the technical report [21]. Provided the knowledge about the probability distribution, we can then determine the detection threshold τ_b by using a confidence level α (e.g., $\alpha = 95\%$):

$$P(bet(k) \leq \tau_b) = \sum_{x_b \leq \tau_b} P(bet(k) = x_b) = \alpha \qquad (6)$$

For a high confidence level $\alpha = 95\%$, it is not likely that the community betweenness value will be larger than τ_b in a normal social network.

2) Evaluation: To evaluate the effectiveness of our detection threshold setting under homogeneous social networks, we generated human contact traces with parameter setting similar to the well-known SWIM trace [10], which contains 3 days of 76 persons' mobility patterns in conference and university campus environments. We let each node select a peer from its community members with the probability 0.9 ($q = 0.9$) every 600 seconds ($Q = 600$). We launch clone attacks by randomly selecting a certain number of nodes in the trace and modifying their node identifiers to be that of the victim node. We studied detection ratio, which is the percentage of clone nodes that are detected by the detection scheme, and the corresponding false positive ratio, the percentage of normal nodes that are mistakenly detected as clone nodes. Figure 3 (b) depicted the detection ratio when using CFB, CDB, and CPB for community betweenness computation under different number of clone nodes. We observed that our detection scheme is effective under the homogeneous social networks and can achieve detection ratio over 80% and the corresponding false positive ratios are zero for all the scenarios.

B. Training Based Approach

The analysis using homogeneous social networks shows the feasibility of our approach. When community sizes and the number of overlapping nodes between different communities vary significantly, we resort to a training-based approach for choosing an appropriate threshold so that we can achieve robust detection. The basic idea is that we analyze historical contact traces to determine typical community betweeenness values and then choose appropriate detection thresholds for nodes with similar encounter rates.

In particular, we first classify the nodes in a mobile social network into different sets based on their encounter frequencies. We analyze the encounter frequencies of all the nodes in a mobile social network and divide these nodes into 3 categories: (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 as *regular nodes*.

Next, we divide the training data into different time periods and compute the average community betweenness values over these time periods for each node in three node sets specified above. Suppose there are N_a, N_r and N_n nodes respectively in the active, regular and non-active sets and b_a^i, b_r^i and b_n^i denote the average community betweenness value of each node in these three sets. Thus, the average community betweenness values over time periods for these three sets of nodes can be represented as: $B_a = \{b_a^i, i = 1, ..., N_a\}$, $B_r = \{b_r^i, i = 1, ..., N_r\}$ and $B_n = \{b_n^i, i = 1, ..., N_n\}$, respectively. Then, we determine the averages and standard deviations of community betweenness values from B_a, B_r and B_n and they are denoted as $AvgB_a$, $AvgB_r$, $AvgB_n$ and σ_a, σ_r, σ_n respectively. In our threshold setting method, we empirically set the threshold τ_b as the average value plus two standard deviations ($AvgB_a + 2\sigma_a$, $AvgB_r + 2\sigma_r$, $AvgB_n + 2\sigma_n$) for nodes from the active, regular and non-active sets respectively.

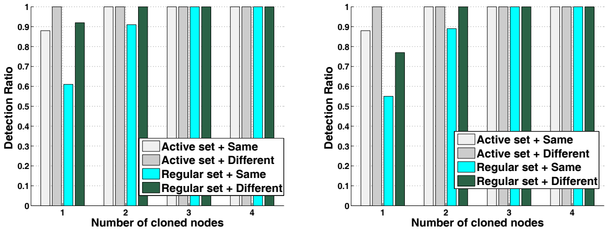
VI. Performance Evaluation

In this section, we conduct performance evaluation of our detection methods based on the training based approach when the community size and the number of overlapping node vary.

A. Simulation Methodology

We conducted simulations by using two human contact-based traces, SWIM [10] and MIT reality [11] traces. The SWIM traces were generated from the SWIM mobility generator to mimic human mobility traces for 76 participants during a 3-day period in conference and university campus settings. 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. 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.

We divide the SWIM trace into 72 time periods with each time period being 1 hour. Similarly, we divide the



(a) Contact Frequency Based (CFB)

(b) Contact Duration Based (CDB)

Fig. 4. Detection ratio under different number of clone nodes when the victim and clone nodes are from the same node set, either active or regular, from the SWIM trace.

MIT trace into 60 non-overlapping time periods with each time period being 8 hours. In both traces, we let each observation window contain $L = 12$ time periods, and set the ratio threshold $\tau_r = 0.5$. We use the first observation window of the traces as training data to determine the community betweenness thresholds, and the remaining observation windows as the testing data to validate our detection approach.

We conduct extensive simulations on these two sets of traces by varying the number of clone nodes from 1 to 4. Since nodes from the non-active set seldom contact with other nodes, we only consider the nodes from either the active or regular sets. For each testing scenario, we randomly choose the nodes from either the active or regular sets and repeat the simulation 50 times to get a statistical view of the results. As in Section V.A, we use the detection ratio and the corresponding false positive ratio to evaluate the effectiveness of our detection scheme.

B. Results

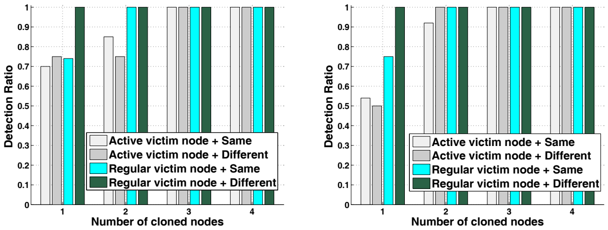
The results of using the Shortest Path Based (SPB) betweenness in our clone attack detection scheme exhibits the same trend as we observed when using the Contact Frequency Based (CFB) betweenness. We thus only present the results obtained from Contact Frequency Based (CFB) and Contact Duration Based (CDB). Furthermore, the results from the SWIM traces achieves the similar detection ratio and false positive ratio to those from the MIT Reality traces. Due to the space limitation, we only present the results using the SWIM traces in the following subsections. The detailed results may be found in our comprehensive technical report [21].

Same Node Set Study. Figure 4 presents the detection ratio versus the number of clone nodes with both the victim and clone nodes belonging to the same node set (can be either active or regular) using the Contact Frequency Based (CFB) and Contact Duration Based (CDB) betweenness in our clone attack detection scheme. In Figure 4, the "active set" and "regular set" denote both victim and clone nodes are from active set or regular set, respectively. The "Same" and "Different" denote the victim and clone nodes are chosen from the same or different communities.

We observed that the detection ratio increases to 100% as the number of cloned nodes increases. This is because more clone nodes increase the number of shortest paths going through the victim node, which increases its observed community betweenness value. The corresponding false positive ratio remains at zero. In addition, we found that the detection ratio is higher when the nodes are from different communities than those from the same communities. Moreover, higher detection ratio is also 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 inline with the design of our betweenness-based detection algorithm because clone nodes from different communities or from the active set result in having more shortest paths which go through the victim node.

Different Node Set Study. Next, we let the clone nodes and the victim node come from different node sets, i.e., the victim node is chosen from the active sets while the clone nodes are chosen from the regular sets, and vice versa. Figure 5 depicted the results of using CFB and CDB betweenness respectively, when victim node and clone nodes are chosen from different node sets, i.e., active or regular. The "active victim node" denotes the victim node comes from active sets and the clone nodes are from regular sets, and vice versa for "regular victim node". The "Same" and "Different" also denote the victim and clone nodes are chosen from the same or different communities.

We found that the detection ratio often improves when the number of clone nodes increases or when the clone nodes are from different communities. Comparing the results from different combinations of victim and clone nodes in Figure 5, we further observed 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 typically active nodes have larger number of neighboring nodes, and thus more shortest paths exist. Thus, with a lower detection threshold used for a regular node, the community betweenness value for such a node during a clone attack can easily exceed the threshold and hence be detected. Comparing the first two bars in Figure 5 with Figure 4 when the number of cloned nodes is 1 or 2, we found 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 the active set while the clone nodes are from the regular set. The victim node from an active set uses a higher detection threshold and thus if the cloned nodes are from regular set, the community betweenness value of the victim node may not exceed the threshold for the attack to be detected. In addition, when we compare the last two bars in Figure 5 with Figure 4 when the number of cloned nodes is 1 or 2, we



(a) Contact Frequency Based (CFB) (b) Contact Duration Based (CDB)

Fig. 5. Detection ratio under different number of clone nodes when victim and clone nodes are from different node sets (active or regular) from SWIM trace.

observe 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. It is also inline with our expectations: the cloned nodes from the active set with more neighboring nodes cause the community betweenness value of the victim node to easily exceed the threshold.

Finally, we found that the false positive ratio is zero for all clone attack scenarios, which is encouraging as these results suggest that our proposed approach is feasible and effective in detecting the presence of clone nodes in a mobile social network.

VII. Conclusion

In this paper, we introduced clone attacks in a mobile healthcare system and proposed a social community based detection method that exploits the social relationships to detect the presence of clone attacks. The concept of community betweenness is introduced by considering both the community and neighboring information of mobile users. The existence of a clone attack is identified by calculating the community betweenness value of a node in multiple time periods. To achieve robust attack detection, we developed both analytical and training based approaches to find a suitable community betweenness threshold for clone attack detection. Through extensive simulations using SWIM and MIT Reality traces, we showed that by considering social community information, our proposed method can detect clone attacks efficiently with high detection ratio and zero false positive rate. Our results demonstrated the feasibility of exploiting the social community information derived from mobile device daily traces for solving security problems (e.g., clone attacks) in mobile healthcare systems.

VIII. Acknowledgment

This work is supported in part by National Science Foundation Grants CNS1016303 and CNS1016296.

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. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, “On the requirements for successful gps spoofing attacks,” in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75–86.

[10] A. Mei and J. Stefa, “Swim: A simple model to generate small mobile worlds,” in *Proc. of IEEE INFOCOM*, 2009, pp. 2106–2113.

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

[12] A. Becher, E. Becher, Z. Benenson, and M. Dornseif, “Tampering with motes: Real-world physical attacks on wireless sensor networks,” in *Proceeding of the 3rd International Conference on Security in Pervasive Computing (SPC)*, 2006, pp. 104–118.

[13] 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*.

[14] 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.

[15] 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.

[16] H. Jun-won, W. Matthew, and D. S. K, “Fast detection of replica node attacks in mobile sensor networks using sequential analysis,” in *Proc. of IEEE INFOCOM*, 2009, pp. 1773–1781.

[17] S. Huang, “Probabilistic model checking of disease spread and prevention,” in *Scholarly Paper for the Degree of Masters in University of Maryland*, 2009.

[18] L. Tang, X. Wang, and H. Liu, “Uncovering groups via heterogeneous interaction analysis,” in *Proceedings of IEEE International Conference on Data Mining(ICDM)*, 2009.

[19] 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*.

[20] D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 2003.

[21] Y. Ren, Y. Chen, and M. C. Chuah, “Detection schemes against clone attacks in mobile social network,” Technical report, Stevens Institute of Technology, November 2011.