The impact of unlinkability on adversarial community detection: effects and countermeasures

Shishir Nagaraja

University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801, USA
sn275@illinois.edu

Abstract. We consider the threat model of a mobile-adversary drawn from contemporary computer security literature, and explore the dynamics of community detection and hiding in this setting. Using a real-world social network, we examine the extent of network topology information an adversary is required to gather in order to accurately ascertain community membership information. We show that selective surveillance strategies can improve the adversary's efficiency over random wiretapping. We then consider possible privacy preserving defenses; using anonymous communications helps, but not much; however, the use of counter-surveillance techniques can significantly reduce the adversary's ability to learn community membership. Our analysis shows that even when using anonymous communications an adversary placing a selectively chosen 8% of the nodes of this network under surveillance (using key-logger probes) can de-anonymize the community membership of as much as 50% of the network. Uncovering all community information with targeted selection requires probing as much as 75% of the network. Finally, we show that a privacy conscious community can substantially disrupt community detection using only local knowledge even while facing up to the asymmetry of a completely knowledgeable mobile-adversary.

1 Introduction

Anonymous communications are useful in building resistance against a global passive adversary who can subject the targets to traffic analysis. In the context of communication channels, anonymity is described in terms of the channel properties of *unlinkability* and *unobservability*, with many schemes as well as deployed systems [8, 10] focusing their efforts on providing the former property.

While anonymous communications plays an important role in ensuring traffic analysis resistance properties of a communication channel, ensuring user anonymity requires much more work. For instance, traffic data collected by compromising a user's personal computer necessarily impacts the privacy of others in the user's social network. If a small fraction of end-user computers are compromised then how does this impact user anonymity? This is the main question we attempt to answer in this paper.

It is well known that the practical risk to user privacy increases with the aggregation of personal data. One such instance is that of modern email service providers with huge storage allowances and accessible user interfaces attracting a large number of users. This results in the aggregation of a large amount of social network information within the administrative power of a very small number of people running the service. An attacker who has partial or complete knowledge of the social network can cause significant damage to user privacy.

Analyzing large amounts of social traffic data such as a large corpus of emails is a highly time consuming task. However, if the attacker can also accurately determine community membership then he can massively reduce his work load by reducing only clustering traffic flows one community at a time. Indeed the adversary's capability in detecting community membership brings him significantly closer to significant privacy invasion than the mere discovery of nodes and inter-node relations (substantiated further in Section 3). Thus the combined use of text analysis methods along with accurate community detection algorithms constitutes an important threat to user privacy. As we shall see, this threat is only slightly mitigated by the use of current anonymous communications technology. To what extent is the risk diminished and what can users do to defend themselves? We develop a graph-theoretic framework to analyse a mobile adversary and apply it to a real-world social network dataset to find out.

The threat model of a mobile adversary is further justified by the increasing popularity of *social-malware attacks* [27]. In their report, Nagaraja and Anderson describe a case of malware-based electronic surveillance of a political organization. By exploiting the social network connecting members within the victim organization, the mobile adversary moved from member to member and managed to copy off entire hard-drives worth of information from most individuals. Subsequently, similar attacks have been reported by hundreds of organizations and individuals in the popular press and in private communications.

The results of this paper only apply to adversarial community detection on **social networks** alone and not to similar sounding applications in very different contexts such as anomaly or misuse detection.

2 Community detection

The problem of splitting a network into a number of sub-communities is not a new one. The first algorithm for graph partitioning was proposed by Kernighan and Lin [21]. A detailed survey of partitioning algorithms in computer science can be found in [12]. The problem of community detection has also been studied in the context of many graph-theoretic clustering algorithms. In its simplest form, a community may be considered as a group of nodes which are densely connected by edges. For example a variety of node clustering algorithms for graphs with the use of shingling techniques, matrix co-clustering techniques, and tile determination in matrices [16, 17, 5] can be used for community detection in graphs. A related problem is that of local triangle counting [2], which can be leveraged to determine an idea of the unit dense structures (triangles) in the underlying graph. The problem is also related to that of finding dense cliques or dense regions in the underlying graph [1, 31, 38]. These techniques are designed for generic graphs rather than the specific case of social networks. The problem of community detection [7, 24, 35, 22, 36] in social networks has also been widely studied because of the increasing importance of social networking applications. A survey of a number of important algorithms for community detection is provided in [36]. A note on the important statistical properties of web communities is discussed in [24].

3 Motivation and context

In this paper, we shall study how the topology of the social network of users affects the amount of effort on the attacker's part to uniquely identify the community association of each user, using graph topology information alone. The effectiveness of community detection attacks depends heavily on the topology of the underlying network. If the attacker is not able to detect communities and associate each user with a particular community, then the social network topology is said to be resistant to community detection attacks under the given threat model. Among other things, the threat model specifies how much information is available to the attacker.

The attacker might also employ additional traffic-flow attacks such as capturing and directly clustering network data flows instead of working with the social network topology, we do not consider such attacks here. *Traffic flow analysis* [33, 18] can be used to cluster flows and ultimately classify users into communities using information related to the protocol or mechanism in use. Similar clustering (attack) methods [11, 39, 6, 25] can be applied to human communication in order to de-anonymize the community membership of a social network. However such clustering methods do not scale very well. The reason is simple: the effort expended by an adversary depends on the amount of information processed by the attacker per pair of communicating users Alice and Bob. The community detection attacks we consider here, only use one bit of information: does Alice communicate with Bob, or not? These attack algorithms can be readily extended to also consider the magnitude of communication between Alice and Bob. Such algorithms have two advantages: (a) they are more scalable than traffic flow attack algorithms whilst requiring lesser storage and lesser processing power, and (b) they are robust to variances traffic flow information.

Community detection algorithms are not an end to themselves and they must be used in conjunction with communication traffic flow analysis and/or text classification algorithms for successfully de-anonymizing community membership in a social network. For a standard reference on inductive learning algorithms see Dumais et.al [11].

Applying such algorithms to social network communications is a two stage process. In the first stage, the attacker constructs a (per edge) vector of features from traffic data (say email messages) for every pair of communicating users. In the second stage, he applies a clustering algorithm over the edge vectors. A popular method from the datamining community is the agglomerative hierarchical clustering [39] which runs in $O(N^{2\log(N)}$ time and can be applied to cluster edges into communities in a bottom-up manner, where N is the number of vertices in the social network. An alternate approach is the use of stochastic inference techniques [25, 6] that provide extensions to handle classification of document networks and various other features, however these have even higher computational complexity.

While these algorithms have higher computational complexity, they are of much interest in confirming that the communities identified by the membership detection algorithms are actually interesting to the attacker. Instead of running these flow analysis algorithms on the entire communication traffic data, he simply analyzes (the much smaller amount of) traffic flow information corresponding to the identified communities. By reducing the size of the input traffic data, the attacker can not only scale-up traffic flow analysis but also reduce the amount of input noise. This opens up the problem space to sensitive flow-analysis algorithms that might otherwise be unusable in the presence of noisy data.

4 Analytical framework for hidden communities

4.1 Modularity

We consider social networks comprising of people and relations. The social network is represented by a graph $G(V, E)$, where people are represented as nodes, while relationships between people are represented as edges. Sets V and E are the set of all nodes and and the set of all edges, respectively. Each edge is associated with an integer weight which is an indicator of the quantity of information exchanged between the two end-points.

In this paper, we will study the problem of adversarial community detection in large-scale networks. As discussed earlier, we would like to determine naturally forming communities in the network. We note that such properties are naturally satisfied by utilizing the concept of *modularity* of vertex sets which previously been used with some success. Before discussing the definition of communities, we will first define some notations and definitions, and explain the concept of modularity in an intuitive way.

Modularity is a notion of community structure where communities are not defined by dense clusters of vertices connected by a small number of edges (*small cuts*). Rather, communities are defined by vertex sets that have either less than *expected* number of edges across each other. Informally, a module is a subgraph whose nodes are more likely to be connected to one another than to the nodes outside the subgraph.

We assume that the *entire network* from which the communities are defined is denoted by $G = (V, E)$, where V is the universal set of nodes, and E is the set of edges defined on V. For ease in explanation, we assume that edges are undirected, although the technique can also be generalized to the directed case. Modularity is defined on a vertex set with respect to *only the subgraph induced by a particular set of nodes.* Therefore, let us first define the concept of an induced edges and induced subgraph for a vertex subset.

Definition 1 (Induced Edges). *Let $G = (V, E)$ be a given graph with node set V and undirected edge set E. Let $S \subseteq V$ be a subset of vertices from G. Then, the induced edge set $L(S, E)$ for the vertex set V is defined as all the edges $R \subseteq E$, such that both ends of any edge in R lie in S in the original graph G. The set R is denoted by $L(S, E)$.*

Thus, the induced edge set uses only the edges for which both edges lie within a given vertex subset. All other edges are ignored. The induced edges can be immediately used to define the induced graph $I(G, V)$.

Definition 2 (Induced Graph). *Let $G = (V, E)$ be a given graph with node set V and undirected edge set E. Let $S \subseteq V$ be a subset of vertices from G. Then the induced graph $I(G, S)$ for the vertex set S is defined as the subgraph including only the vertex set S and all induced edges on this vertex set, which have both ends within S. Thus, the induced graph $I(G, S)$ essentially corresponds to the graph $(S, L(S, E))$ with vertex set S and induced edge set $L(S, E)$.*

A given network might have several embedded subgraphs with a range of connectivity characteristics. In general, we would like to determine the embedded networks, which have high level of information flow, but whose edges are a result of local emergent processes rather than defined by a globally agreed blueprint to achieve such a flow. Therefore, we define the concept of *modularity* of a set of nodes S, with respect to the *induced graph* for vertex set V.

Using previously defined terms, consider graph $G = (V, E)$ where V is the universal set of nodes, and E is the set of edges defined on V, consider the induced graph $I(G, S)$ for a vertex set S. Let d_i^G be the sum of edges incident on vertex i in graph G. Given G, we define a graph G_{random} with the same number of vertices, but where every possible edge is created with probability $d_i d_j/2|V|$. That is, the endpoints are randomly selected.

Definition 3 ((S, V)-modularity). *. The (S, V)-modularity of a set of vertices $S \subseteq V$ in G is the difference between the number of edges whose endpoints lie entirely in S computed over* **induced graph** *$I(G, S)$ and the* expected *number of edges whose endpoints lie in S computed over induced graph $I(G_{random}, V)$. Therefore, if $a(S)$ be the sum of degrees of vertices S in the induced graph $I(G, S)$, and $r(S)$ be the expected sum of degrees of vertices S in induced graph $I(G_{random}, S)$ then the (S, V)-modularity is defined as follows:*

$$Q(S, V) = \frac{a(S) - r(S)}{\sum_{i \in S} d_i^G} \tag{1}$$

We note that a high value of $Q(S, V)$ implies that most of the edges are used in *mixing* information within S rather than between S and $V - S$. A low value of $Q(S, V)$ implies that S is a rather poor choice for a community of nodes. Formally, we can now define the problem of determining all the vertex subsets which are relatively sparse and have modularity above a certain user-defined threshold. We define the (α, β)-hidden community as follows:

Definition 4 ((α, β)-hidden community). *Consider a graph $G = (V, E)$. We define the (α, β)-hidden community as a subset S of vertices, which satisfies the following properties:*

- *The set S is a subset of the universal set of vertices V.*
- *The total number of edges in the induced graph $I(G, S)$ is at most $\alpha \cdot |V|$.*
- *The induced graph $I(G, S)$ has modularity at least β. In other words, we have $Q(S, V) \geq \beta$.*

We note that the above definition is focussed on determining an *edge structure* in the community which is focussed on high amount of information flow, in the presence of edge formation based on social communication; i.e. no DHTs or random-graph topologies. Clearly, the level of information flow implies the presence of a community, but the relatively sparse presence of edges (compared to highly dense graphs) ensures that such a community is hidden to methods which work purely with techniques such as the clustering coefficient. The definition above can then be used to create a problem definition on determining hidden communities with respect to the parameters α and β.

Problem 1 (Hidden Community Detection). Consider a network $G = (V, E)$. We wish to determine *all the vertex subsets* $\mathcal{S} = \{S_1 \ldots S_r\}$ which satisfy the following properties:

- Each vertex subset S_i is an (α, β)-hidden community with respect to the graph G with $\alpha = 0.25$ and $\beta = .10$.
- Each vertex set S_i is *maximal* with respect to $\mathcal{S}$. In other words, there is no other vertex set $T \in \mathcal{S}$, such that $T \supset S_i$.

The principle of maximality is useful in reducing the size of the output, and ensuring that redundant subsets are not unnecessarily reported.

The choice of β is based on empirical observation that most detected communities have $\beta > .25$ (a complete graph has a modularity of 1).

Finally, we note that the above model of hidden communities is **not designed to take covert networks into account**, but rather members of the general public whose privacy is easily compromised due to the network externalities of electronic surveillance programs [9]. We seek to understand and address the latter category of risks.

4.2 Threat model – mobile passive adversary

Our threat model is based on the mobile adversary model first proposed by Ostrovsky and Yung [30]. The attacker is a malicious global passive adversary whose goal is to detect (α, β)-hidden communities in the network. Our model is inspired by an ISP level adversary that wishes to detect hidden communities. Our study deals with two scenarios, the case of the fully knowledgeable adversary and the partially knowledgeable adversary.

Adversary with full knowledge: Since the attacker is global, he is aware of the existence of people (vertices) in the social network graph. When communicating parties make no attempt at anonymizing communication, the attacker is also aware of the social relations (interconnecting edges) between them.

Adversary with partial knowledge: If anonymous channels [32, 8, 10] guaranteeing unlinkability [1] are used, then the attacker is only aware of all the vertices of the graph but does not have any information about the edges. To successfully detect hidden communities the attacker must uncover as much information about edge relationships as possible. He does so by placing **probes** on vertices of the graph. This might be achieved by a strategically placed keylogger on a victim's computing device as in the case of the Tibetan attacks [27], for example. Further, the attacker has finite probing capability and cannot simultaneously probe everyone. In any time interval t, the attacker can only probe a fraction of people on the social network to uncover topology information.

Additionally, our attacker is **mobile**, which means he can remove a probe from one vertex and place it on a different vertex of the attacker's choice at time $t' > t$. The number of probes ϵ is finite, hence the attacker can only compromise a fraction of vertices at any point in time. However as the attacker is mobile, he can progressively learn about the entire network over an extended period of time. Each time interval corresponds to a *round*.

Definition 5 (mobile ϵ-attacker:). *Consider a graph $G = (U, A)$ and $0 < \epsilon \leq 1$, the ϵ-attacker is a global passive adversary who can* **probe***(observe all communications originating from) a set of vertices $f_t \subseteq U$ upper bounded at $|f_t| = \epsilon|U|$ at round $t > 0$.*

[1] When nodes of a network communicate via anonymous channels that offer unlinkability, an attacker monitoring network traffic cannot identify communication endpoints but knows traffic volume information. When unobservable channels are used the attacker cannot distinguish between communicating and non-communicating users.

In other words an attacker is allowed to place probes over a constant fraction of vertices, and also, move the probes from vertex to vertex at the beginning of a new round.

Definition 6 **((ϵ, t)-view:).** *Consider graph $G = (U, A)$ and an ϵ-attacker. The (ϵ, t)-view is defined as the graph $I(G, V)$ induced by vertex set V, where $V = \{f_0 \cup f_1 \cdots \cup f_t\}$ is the set of all vertices probed by the attacker by round t.*

Finally, an attack involves the use of two types of strategies. In every round, a *surveillance strategy* drives probe placement while a *community detection strategy* processes the information gathered by the attacker. At each round t, the surveillance strategy (is an algorithm) that accepts an (ϵ, t)-graph and outputs the set of vertices T that shall be probed in round $t + 1$. Similarly, the community detection strategy (a different algorithm) accepts an (ϵ, t)-graph and outputs a set of (α, β)-hidden communities. Essentially, the attacker builds a graph using all the edge and vertex knowledge gained in previous rounds and then analyzes it using a community detection algorithm to discover any hidden communities.

4.3 Measuring anonymity

Measuring anonymity in the context of this paper, is the measurement of the efficiency of community detection – the fraction of hidden-community detected

As described in previous sections, the attacker combines various surveillance and community detection strategies to discover the community structure of a social network. Privacy sensitive users might wish to keep their community membership anonymous. However network topology inherently contains information about community memberships, and a mobile ϵ-adversary can gain access to this information. On the other hand privacy conscious members of a social network might wish not to be identified as belonging to a certain community or club even as they participate in it.

There are several ways in which one can express the anonymity a system provides. The notion of anonymity within Crowds [34] is close to the notion of anonymity we consider here – instead of being identified as a member of a specific community, the anonymity seeking user would wish to be identified as being part of a significantly larger community of users. However instead of a qualitative metric we measure anonymity quantitatively.

The adversary is said to have successfully de-anonymized the social network membership if he can accurately uncover vertex sets corresponding to one or more embedded communities. The *false negative rate* FN is defined as the fraction of (α, β)-community nodes being misclassified as being part of the larger community of vertices $V - S$.

Definition 7 (Community anonymity). *Invoking the concept of anonymity sets, the anonymity of a (α, β)-hidden community under a specific community detection strategy $\varOmega$ is defined as follows:*

$$A((\alpha, \beta), \varOmega) = \frac{V - FN * S}{|V - S|} \tag{2}$$

We shall also refer to A using the term **miss-ratio**.

5 Newman's community detection algorithm

As we have explained previously, a modularity based community detection method can be used to de-anonymize community membership in a social network by identifying vertex subsets corresponding to high modularity *scores*. Modularity based methods are fairly accurate among the scalable methods of community detection. They have been well tested and studied in a variety of social and biological networks within the complex networks community [28, 14].

Modularity reflects the extent, relative to a random network, to which edges are formed within communities rather than across them. By using modularity as a metric, we can assess the quality of any assignment of nodes to the same community (Eqn. 1). Hence, identifying community membership becomes a modularity maximization problem. Accordingly, Newman [28] proposed a community detection algorithm that optimizes the selection of S by calculating the second eigenvector over a matrix of modularity scores for each edge in $G(V, E)$. Let $N = |V|$ and we assume each vertex has $\log(N)$ edges as a rough approximation in a social network, Newman's algorithm computes scales as $O(N^2 + N\log(N))$.

Due to shortage of space, we refer the interested reader to the full description of Newman's algorithm in the original paper [28].

5.1 Alternate algorithms and approaches to community detection

Apart from Newman's modularity algorithm, we did consider *edge importance* based community structure detection approaches as well. In these approaches, the attacker iteratively removes the edges with the highest *importance*, which can be defined in different ways. Girvan and Newman [29] defined edge importance by its shortest path betweenness. The idea is that the edge with higher betweenness is typically responsible for connecting nodes from different communities. The fastest algorithm to calculate betweenness centrality is credited to Brandes [4], it has a computational complexity of $O(N^2\log(N))$.

Fortunato [13] proposed *information centrality* to measure edge importance, defined as the relative *network efficiency* [23] drop caused by the removal of an edge. The time complexity of his algorithm is $O((N\log(N))^3 \times N)$. As the time complexity of betweenness and information centrality algorithms is not acceptable for community detection in massive networks, we removed these from our evaluation.

Further, preliminary experiments we conducted showed that when communities in G are separated by a *small-cut*, Newman's community detection algorithm performed fairly well in the presence of random errors (edges and nodes randomly added/removed across the cut) in the input topology – a linear increase in random noise in the network topology results in a linear increase in number of false positives, as opposed to an exponential increase of false positive rate we observed in the case of min-cut based methods.

Conductance based techniques such as SybilInfer [37] and SybilGuard [37] are interesting approaches that can be used for community detection based on the metric of graph conductance [20]. While these approaches are good for analyzing structured tightly knit communities that are separated by small cuts (such as DHTs and Sybil networks), their applicability in our context needs further study. One possible hurdle might be that conductance of the cut separating the hidden community (see section 5.2) and the rest of the graph is 0.0998. The SybilInfer work used a cutoff threshold of 0.10 to identify a cut between a

Sybil community and the rest of the graph, while other cuts in the social network tended to have conductance in excess of 0.9.

5.2 Email communication network

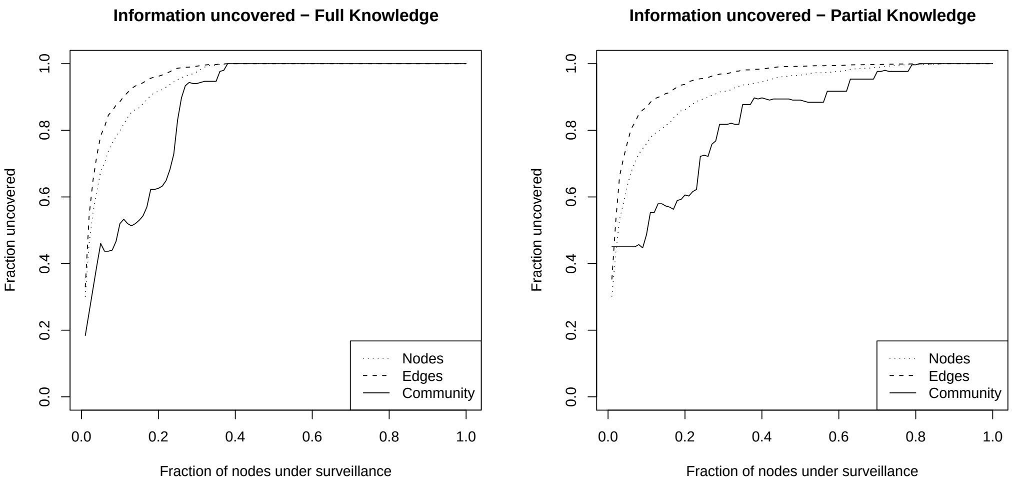
Our network dataset [19] comprises of a social network harvested from email exchanges within a mid sized university of 1700 researchers, graduate students and staff. Each email address was mapped to a person. We discarded all email messages where either the sender or the receiver email address was not a university email address. This means we have left out relationships where two persons at the university might be connected via an outsider, and this could impact our results. We disregarded unidirectional email messages which removed bulk email messages as well as most spam. We added an edge between every two nodes that had sent at least one message in each direction. The weight of the edge was set as the sum total of messages exchanged between two nodes.

Next, we extracted the largest connected component or giant component consisting of 1133 people and 10903 relation. The data we obtained contained emails from two different departments, and this was correctly detected by the modularity community detection algorithm. We shall consider the smaller of the two partitions as the “hidden” community (in the sense that community members desire privacy) as far as our experiments are concerned, and the larger one as the “main” network into which the nodes of the hidden community will attempt to blend into. The giant component consists of two partitions: partition G_M with 831 nodes and 6807 edges shall be our main network and G_C, will be our (α, β)-hidden community of 302 nodes and 2574 edges.

6 Efficiency of community membership de-anonymization

In our model, the adversary’s goal is to accurately determine the membership of each community in the network. Our first experiment attempts to measure the efficiency of community detection by different surveillance strategies. The ϵ-mobile adversary is limited by the number of probes (ϵ), which in turn limits his rate of gathering topology information. This in turn affects the success of community de-anonymization goals, and we wish to measure how the adversary’s success varies as the fraction of the network being directly probed increases.

We wish to determine the minimum ϵ value at which the ϵ-mobile adversary can fully determine community membership. To do so, we first establish the upper bound of attacker efficiency. For this purpose we consider the ϵ-mobile adversary in the context of full knowledge, as discussed in section 4.2. This might seem like a trivial exercise since the fully knowledgeable adversary learns little by probing. However full knowledge allows the attacker to compute the optimal probing sequence. Therefore this represents the best possible performance (upper bound) an ϵ bounded adversary might possibly come up with. We plot the fraction of nodes and edges discovered by the fully-knowledgeable adversary **using the probes alone**, in figure- 1.a, to show the upper bound of attacker efficiency. We also plot the fraction of the hidden-community detected (this is simply $1 - missratio$) using the graph uncovered by probing. Later in the section we will compare these with the efficiency of the *epsilon*-mobile adversary in the case of partial knowledge in figure- 1.b.



(a)

(b)

Fig. 1: De-anonymization efficiency

Of the many possible centrality measures that could be used to generate the optimal probing sequence, the most appropriate from the perspective of information flow is the flow betweenness centrality measure devised by Freeman [15]. The betweenness centrality C_b^v of a node v is defined as the number of all pairs shortest paths that pass through v:

$$C_B^v = \sum_{x \in V} \sum_{y \neq x \in V} \frac{\sigma_{xy}(v)}{\sigma_{xy}} \tag{3}$$

Where σ_{xy} is the number of shortest paths between nodes x and y.

The upper bound of de-anonymizing community membership using this strategy is shown in figure- 1.a. After probing 8% of the nodes in the decreasing order of betweenness centrality, the adversary is aware of the existence of 76% of the nodes and 85% of the edges. This partially confirms one of the results of an earlier study by Danezis and Wittneben [9].

Interestingly, our study reveals that the community membership of only 50% of the nodes is correctly identified. The adversary then makes further progress as shown in table 1.

$\%probed = t\epsilon\|V\|$	% nodes	% edges	%Community uncovered
8	76	85	48
28	96	99	95
38	100	100	100

Table 1: Knowledge gained by attacker using optimal probing strategy

$\%probed = t\epsilon\lvert V\rvert$	% nodes	% edges	%Community uncovered
8	73	84	45
28	96	90	58
38	97	91	86

Table 2: Knowledge gained by attacker using probe placement based on traffic volumes

We now measure the adversary's accuracy in discovering community membership under the case of partial information. The ϵ-adversary with partial knowledge performs surveillance with a (non-optimal) probing sequence generated using traffic volume information. The adversary sorts the nodes in descending order of traffic volumes and places a constant fraction of nodes under surveillance in successive rounds.

Figure- 1.b, shows the fruits of the adversary's efforts. The lack of full topology knowledge particularly dents the adversary's ability to effectively spy on the network as summarized in table 2: When 8% of the nodes are spied upon, 45% of the hidden community nodes are correctly identified whilst uncovering 73% of nodes and 84% of edges. However, while 90% of the nodes and 96% of the edges are known to the adversary by putting 28% of the nodes under surveillance, the only 58% of the hidden community known.

In terms of pure numbers the adversary is able to acquire a significant amount of topology knowledge by putting 28% of the network under surveillance in both threat models. However, hidden-community discovery is much harder. In the partial case, it requires the adversary to put almost 80% of the nodes under direct surveillance to enable him to accurately localize 99% of the hidden community nodes. This is in stark contrast to the upper bound provided by the case of the fully knowledgeable adversary who only needs to place 37% of the nodes under surveillance to gain the same amount of membership information.

6.1 Discussion

Does this mean that the use of anonymous communications increases the work load of the adversary by almost 100%? No, this is true only when the adversary needs to uncover the membership information of all nodes within the network. Our results show that in both the adversary models, placing 8% of the nodes under direct surveillance compromises the community membership of almost 50% of the nodes.

Since close to 80% of the population must be monitored to detect all the communities, it means that in the short run, government surveillance budgets are more likely to cause harm to privacy than to uncover hardened cells. On the other hand, it also means the social malware will be significantly successful even if only a small fraction of the user base is infected.

We are therefore interested in the privacy preserving countermeasures for **larger user communities** rather than for **covert communities** which will be invisible anyway. Being larger they are associated with higher detection rates, and are more difficult to hide. This is the reasoning behind the choice of community sizes for our experiments in section 5.2.

We note that the adversary's efficiency in community membership assignment in the case of partial knowledge is not only markedly lower than that of full topology knowledge, but has a slower growth rate. This is not surprising given the relatively complex structural characteristics of the information the adversary is trying to uncover.

7 The efficiency of counter-detection measures

We shall now consider defense responses to the ϵ-mobile community detecting adversary. To do so, we allow the (α, β)-hidden community to rewire itself in order to disrupt community detection.

We adopt the following defense model with multiple rounds: members of the hidden community can employ one of a number of counter-detection strategies involving topological rewiring, limited by a counter-detection budget at the beginning of every round. The adversary then runs community detection algorithms to deduce membership in every round.

Several topological manipulation options are open to the (α, β)-hidden community. Since the community is defined by vertex sets we do not explore counter-detection strategies based on removing vertices from the community. We also discount edge removal as a countermeasure since that would also disrupt information flow. Therefore, we shall focus our analysis on various strategies of *edge-addition* alone. The strategy of edge-addition can be also be understood as a method of selectively adding *cover traffic* to the network. Note that depending on the context of the social network this may not be a feasible defense in all contexts.

7.1 Counter-detection strategies based on edge addition

The application of any counter-detection strategy consists of adding C additional edges (defense budget) whose end-points are chosen as follows.

Each end point is chosen according to a vertex centrality metric. We have previously discussed flow betweenness centrality, see Eqn. 3. We shall consider two more vertex centrality metrics: eigenvector centrality and degree centrality. A fourth option is to treat vertex centrality as a random value.

Degree centrality of a vertex i, d_i^G, is the sum of edges incident on vertex i in graph G.

Eigenvector centrality score [3] of a vertex corresponds to the values of the first eigenvector of the graph adjacency matrix; these scores may, in turn, be interpreted as arising from a reciprocal process in which the centrality of each vertex is proportional to the sum of the centralities of those vertices with whom he or she shares an edge.

Let $W = \{HB, HE, HD, RAND\}$ be a set of centrality measure functions. Let c be the set of hidden community nodes, and m be the set of main network nodes, corresponding to the entire vertex set of graphs G_C and G_M from Section 5.2. Each strategy involves adding an edge (i, j) with edge endpoints i and j chosen either from sequence $R_{X \in M}(c)$ the sequence of nodes in decreasing order of centrality measure X from the hidden-community vertex set, or similarly from $R_{X \in W}(m)$ the sequence of nodes drawn from the main network vertex set in decreasing order of centrality measure X. Also, each edge has one end-point in G_C and another in G_M.

For instance, $R_{HB}(c)$ defines a sequence of ordered vertices over c with decreasing of high-betweenness centrality values. Similarly, we have sequences $R_{HD}(c)$, $R_{HE}(c)$, $R_{RND}(c)$, $R_{HD}(m)$, $R_{HE}(m)$, and $R_{RND}(m)$.

An edge addition strategy is denoted as $Xx - Yy$ where $X, Y \in W$ and $x, y \in c, m$. Counter-detection strategy **HBc-HBm** involves adding an edge with an endpoint in $R_{HB}(c)$ and the other in $R_{HB}(m)$. Counter-detection strategies **HDc-HDm**, **HEc-HEm**, **RNDc-RNDm**, and hybrid measures, **RNDc-HBm**, **RNDDc-HDm**, **RNDc-HEm**, **HBc-RNDm**, **HDc-RNDm**, **HEc-RNDm** are similarly defined.

The effect of countermeasures on the anonymity of community membership is shown in figure 2. Each graph is averaged over 50 iterations. We note here that some of the graphs in figure 2 exhibit high kurtosis with the result that they appear to have abrupt rises and falls. While we could have removed those values to make the graphs smoother, without changing the conclusions we have drawn from our results, we have chosen to retain them in order to better understand the reason for high kurtosis.

7.2 Evaluating counter-detection defense measures

Let us first consider the case of the adversary with full topology knowledge and an unlimited surveillance budget, which will provide us with an upper bound in our results.

The first strategy we analyzed for hiding G_C was the naive RNDc-RNDm strategy: edge addition with random end points selected from either G_C or G_M. Figure 2 shows the privacy gains brought by this strategy, indicated by the blue line with an 'x' motif. This curve shows the resilience of the modularity detection algorithm to the presence of random errors – a linear increase in random error leads to a linear increase of injected faults. Therefore it makes for a poor defense strategy and is the worst performer among the strategies we considered; the addition of a 1000 random edges (50% of hidden community edge budget) results in a miss-ratio of only 20%.

Having learnt that the naive strategy is of little use in hiding the hidden community in a real world network, we proceeded to apply the next set of techniques, namely the purely centrality based strategies. The betweenness centrality strategy HBc-HBm (black line with circle motif) and HDc-HDm (red line with triangle motif) are average performers, with a peak miss-ratio of 50%, for additional edges of 10% of the hidden community. In addition, HDc-HDm also requires relatively larger amount of resources before delivering a miss-ratio above 30%.

Finally we look at the four hybrid strategies of edge addition, combining random node selection in one community with strategic selection in the other. Of the three strategies involving strategic selection in the main partition (requiring knowledge of popular nodes), RNDc-HBm (blue line with a diamond) and RNDc-HDm (pink line and triangle motif) perform equally well, with 78% of the hidden community nodes hidden from detection with only an additional edge budget of 1% of the hidden community.

The final three hybrid strategies involve local topology knowledge: HBc-RNDm indicated by the '*' motif and HDc-RNDm indicated by solid diamond. They work almost as well as the previous hybrid strategies on this network, however they require almost twice as many edges to offer the same level of protection. Even more interestingly, these strategies offer a high anonymity with 93% of the hidden community being wrongly classified as being part of the larger community. This indicates that strategies with local knowledge can be as efficient at countering community detection as strategies dependent on global topology information which bodes well for increased privacy.

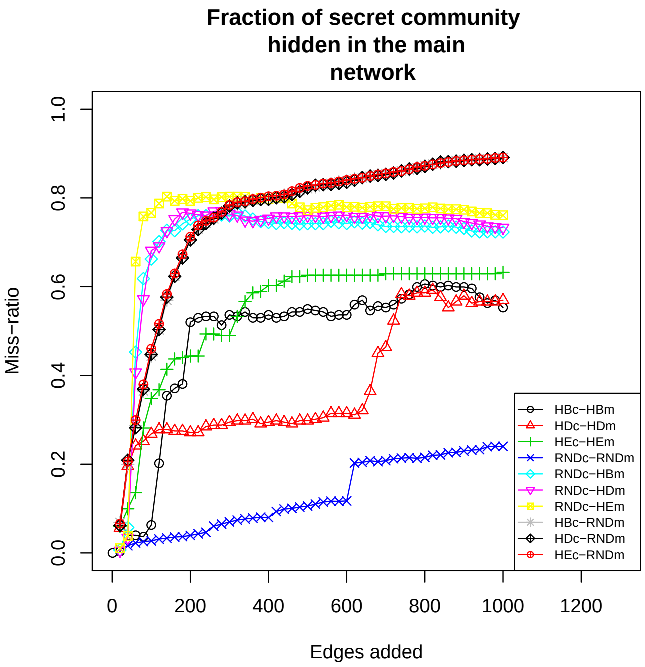
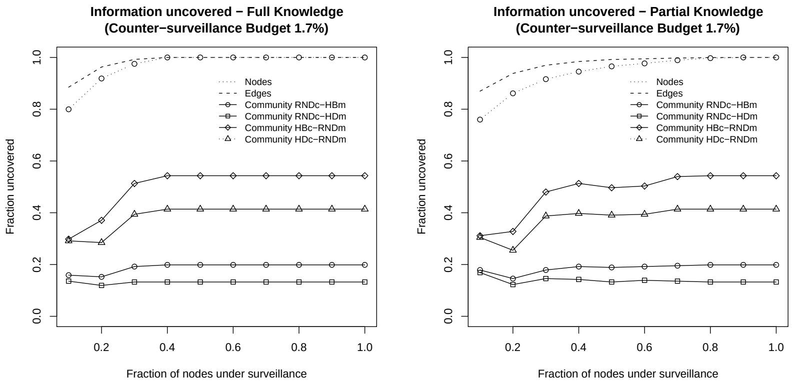


Fig. 2: Anonymity of hidden community under modularity based community detection



(a) Full knowledge

(b) Partial knowledge

Fig. 3: Effects of counter-detection measures on adversarial community detection

Our counter-detection defense measures show that, if the defenders have full topology knowledge, then at a cost of 3 additional communication links per hidden community node, the modularity of the graph can be rewired leading to increasing the miss-ratio up to 90%, in the adversary's membership calculations. In both cases, of partial and full topology knowledge, the defenders can drive up the miss-ratio to 80%, with a investment of only 0.01 edges per hidden community node or approximately 1% of existing hidden community edge resources.

Figure 3 shows the efficiency of surveillance when counter-detection tactics are deployed by hidden communities. Strikingly, while people and relations are discovered rapidly, the percentage of community membership information available to the attacker remains limited to 12% in the best performing counter-detection tactic. The RNDc-HDm strategy, where random members in the hidden community connect with hub nodes in the larger community, is attractive as a counter strategy in several ways. First, hub nodes with high degrees of connectivity are well known; and second, since hidden community members bearing responsibility for implementing the strategy do so in a decentralized manner and are easily replaceable if removed [26].

Further, we observe that the use of current anonymous communications technology by the hidden community does not change their privacy gains against community detection. As such, mere unlinkability falls short of the required level of security, and full unobservability is required to prevent the adversary's success. Since current anonymous communication networks do not provide unobservability, the best defenses lie in modifications at the fundamental level of network topology.

It is worth noting that as the size of the network increases, modularity based community detection can meet with a lower degree of success [14]. This is because as the network grows, it reaches a point where the expected number of edges between communities drops below 1. Past this point, modularity based detection does not give good results. Therefore while the small network of thousand or so nodes worked out well in our study, the results (of counter-surveillance) may not directly apply to larger social networks.

Why do these countermeasures work? So far, we have avoided all discussion of how the countermeasures work from a theoretical viewpoint. We now address this aspect. Going back to the definition of modularity in Eqn. 1, we can see that community boundaries are delineated on the basis of where there are fewer edges than expected. Edges between high centrality nodes are expected with a lower probability than those between nodes with lower centrality scores. When the number of edges connecting high-centrality nodes (or even random nodes) with other high-centrality nodes increases the number of actual edges becomes closer to the "expected" number of edges as per eqn. 1. This is the theory behind the successful countermeasures we have considered.

It is interesting to note that, in Fig. 2, community detection sometimes drops before increasing again. It appears that additional topology knowledge can sometimes be detrimental to community detection. We see similar non-linear behavior in Fig. 3, where extra defense edges can sometimes cause a decrease in community anonymity. We have no explanation for this phenomena.

8 Related work

Past work by Danezis and Wittneben [9] highlighted the privacy compromising network externalities involved in computer insecurity when police execute wiretapping warrants. Their work considered the risk of privacy invasion due to indirect surveillance concluding that the privacy of a large fraction of users would be compromised once unlinkability was broken.

We take a markedly different approach by extending the definition of user privacy to include information about community memberships in a social network. We show the close link between the use of anonymous communications and its impact on the success of a community detecting adversary. Additionally, we also consider countermeasures and empirically demonstrate their effectiveness in enhancing user privacy.

9 Discussion and Conclusions

We have presented a model of surveillance and privacy preservation based on the detection of community structure in social networks. We have studied the network externalities of privacy compromise from a new angle, the detection of community structure and membership. In this paper, we have analyzed the interplay between detection and counter-detection strategies. We have some concrete results to present. We have shown that while structural elements of a network such as nodes and edges are easily discovered when small fractions of the network are placed under surveillance, discovering community structure information requires the adversary to invest in a significantly higher surveillance budget.

We have also shown that, regardless of whether network members communicate through an anonymous communications channel, placing 8% of the network under selective surveillance based on traffic volume is enough to compromise the community membership information of at least 45% of the nodes in the network. Our results also show that where the adversary is interested in understanding the community membership information of a far higher fraction of the nodes, the use of anonymous communication networks can increase the adversary's cost by almost 100% (80% of nodes under surveillance to uncover 99% community membership information).

Further, we have analyzed the dynamics of community hiding. First, we have shown that naive strategies of edge addition between randomly selected pairs of nodes from either partition have limited community hiding capability.

Hybrid strategies involving a combination of random and high centrality endpoints work best – edges are added between randomly chosen community nodes and a high centrality nodes in the main network allowing thus requiring only local knowledge on the part of the hidden community. Specifically, up 80% of the hidden community went undetected with a counter-detection budget of only 1% of total hidden community network edge resources. A variant strategy that associates high centrality hidden community nodes to randomly chosen nodes in the main delivers a more striking result: Up to 93% of the hidden community remained hidden with 10% additional edge resources (too expensive) while 80% could be hidden with a mere 2% additional edges (reasonable).

Our results show that membership de-anonymization attacks based on exploiting partial link knowledge as well as full link knowledge can be successfully repelled if the hidden community carries out selective topological rewiring. Counter-detection mechanisms merely require local knowledge and can bring clear privacy gains even when faced by an adversary with global knowledge.

10 Acknowledgements

The paper greatly benefited from discussions with Prateek Mittal and encouragement from George Danezis. The author would like to thank Aaron Johnson for agreeing to shepherd the paper. The author also expresses his thanks for feedback from Himanshu Khurana and the detailed comments from anonymous reviewers.

References

1. J. Abello, M. G. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN*, 2002.
2. Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *KDD*, 2008.
3. Phillip Bonacich. Power and centrality: A family of measures. *The American Journal of Sociology*, 92(5):1170–1182, March 1987.
4. Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
5. Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra Modha, and Christos Faloutsos. Fully automatic cross-associations. In *KDD*, 2004.
6. Jonathan Chang and David M. Blei. Hierarchical relational models for document networks, 2009.
7. Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks, August 2004.
8. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *IEEE Symposium on Security and Privacy*, pages 2–15, 2003.
9. George Danezis and Bettina Wittneben. The economics of mass surveillance and the questionable value of anonymous communications. In Ross Anderson, editor, *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, Cambridge, UK, June 2006.
10. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.
11. Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155, New York, NY, USA, 1998. ACM.
12. Ulrich Elsner. Graph partitioning - a survey. *MONARCH - Dokumenten- und Publikationsservice [http://archiv.tu-chemnitz.de/cgi-bin/interfaces/oai/oai2.pl] (Germany)*, 2005.
13. S. Fortunato, V. Latora, and M. Marchiori. Method to find community structures based on information centrality. *Physical Review E*, 70(5), 2004.
14. Santo Fortunato. Community detection in graphs. *arxiv eprint 0906.0612 http://arxiv.org/abs/0906.0612*, Jan 2010.
15. Linton C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1978.
16. D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB*, 2005.

17. Aristides Gionis, Heikki Mannila, and Jouni K. Seppänen. Geometric and combinatorial tiles in 0-1 data. In *PKDD*, 2004.
18. Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium (Security'08)*, 2008.
19. R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68(6):065103, Dec 2003.
20. Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.
21. B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technology J.*, 49(2):292–370, 1970.
22. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *WWW*, 1999.
23. Vito Latora and Massimo Marchiori. Economic small-world behavior in weighted networks. *The European Physical Journal B - Condensed Matter*, 32(2), 2002.
24. Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, 2008.
25. Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.
26. Shishir Nagaraja and Ross Anderson. the topology of covert conflict. In Tyler Moore, editor, *Pre-Proceedings of The Fifth Workshop on the Economics of Information Security*, June 2006.
27. Shishir Nagaraja and Ross Anderson. The snooping dragon: social-malware surveillance of the tibetan movement. Technical Report UCAM-CL-TR-746, University of Cambridge, March 2009.
28. Mark Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, June 2006.
29. Mark Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 69(2), 2004.
30. Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks (extended abstract). In *PODC '91: Proceedings of the tenth annual ACM symposium on Principles of distributed computing*, pages 51–59, New York, NY, USA, 1991. ACM.
31. J. Pei, D. Jiang, and A. Zhang. On mining cross-graph quasi-cliques. In *ACM KDD Conference*, 2005.
32. Andreas Pfitzmann and Marit Hansen. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology. Draft, July 2000.
33. Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4), 1998.
34. Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998.
35. V. Satulouri and S. Parthasarathy. Scalable graph clustering using stochastic flows: Applications to community discovery. In *KDD Conference*, 2009.
36. W. Tang and H. Liu. Graph mining applications to social network analysis. In *Managing and Mining Graph Data, Ed. Charu Aggarwal, Haixun Wang*, 2010.
37. H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *SIGCOMM*, 2006.
38. Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Out-of-core coherent closed quasi-clique mining from large dense graph databases. In *ACM Transactions on Database Systems, Vol 31(2)*, 2007.
39. Ying Zhao, George Karypis, and Usama Fayyad. Hierarchical clustering algorithms for document datasets. *Data Min. Knowl. Discov.*, 10(2):141–168, 2005.