Sybil Attacks Against Mobile Users: Friends and Foes to the Rescue

Daniele Quercia
Massachusetts Institute of Technology
Cambridge, MA 02139
quercia@mit.edu

Stephen Hailes
University College London
London, WC1E 6BT, UK
s.hailes@cs.ucl.ac.uk

***Abstract*—Collaborative applications for co-located mobile users can be severely disrupted by a sybil attack to the point of being unusable. Existing decentralized defences have largely been designed for peer-to-peer networks but not for mobile networks. That is why we propose a new decentralized defence for portable devices and call it MobID. The idea is that a device manages two small networks in which it stores information about the devices it meets: its *network of friends* contains honest devices, and its *network of foes* contains suspicious devices. By reasoning on these two networks, the device is then able to determine whether an unknown individual is carrying out a sybil attack or not. We evaluate the extent to which MobID reduces the number of interactions with sybil attackers and consequently enables collaborative applications. We do so using real mobility and social network data. We also assess computational and communication costs of MobID on mobile phones.**

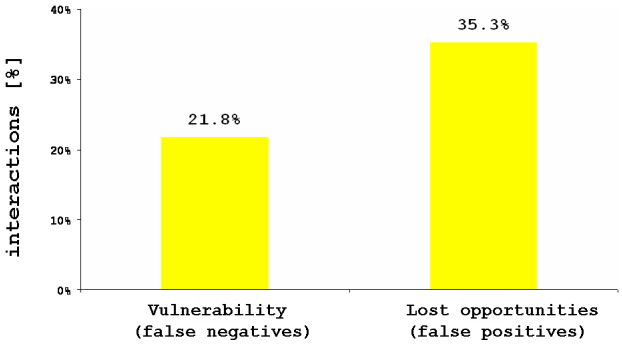


Fig. 1. Vulnerability and Lost Opportunities for Existing Defences Coping with 20% Community Members Being Attackers.

I. Introduction

Researchers have recently proposed general infrastructures with which portable devices in proximity of each other opportunistically trade various services with in a scalable and decentralized way [6], [17], [25]. Without going through any Internet server, collaborating devices are able to: synchronize their timers for playing multi-player games; run localization algorithms that increase the precision of street map software and of location-based services; and cache Web content to avoid monetary costs of cellular or wireless providers.

The problem is that collaborative applications are easily disrupted by uncooperative and malicious individuals. Those individuals profit from services without providing an adequate return and then make themselves untraceable by creating a very large number of bogus identities. In literature, those individuals are called sybil attackers or simply sybils [10].

In the next section, we will show that existing distributed defences against sybils are largely meant to work in peer-to-peer networks but not in mobile networks. We set out to fix this problem by making three main contributions:

- An effective way of identifying sybil attackers for in-range portable devices (MobID). The key idea is that each device manages two small networks in which it enlists the devices it meets: its *network of friends* contains honest devices, and its *network of foes* contains suspicious devices. By reasoning on these two networks, the device is then able to determine whether an unknown individual is carrying out a sybil attack or not. MobID guarantees that honest individuals both *reject* bogus identities and *accept* honest identities, and they do so with high probability (Section III).
- Evaluation of the robustness of MobID on real mobility and social network data (Section IV).
- Evaluation of its communication and computational overhead on mobile phones (Section IV).

II. Existing Solutions

One way of limiting the corruptive influences of sybil attackers is to stop them creating bogus identities. That is easily done by additional infrastructures that bind identities and cryptographic keys. The problem is that these infrastructures, such as admission control and public key servers, are expensive and difficult to implement in any network, let alone in a *distributed* network.

That is why researchers have recently proposed solutions for distributed networks and, more specifically, for peer-to-peer networks. Danezis *et al.* [8] made DHT lookups resilient to sybil attacks by exploiting the bootstrap tree of the DHT, where two nodes are linked if one node introduced the other into the system. The key idea is that sybil nodes will attach to the rest of the tree only at limited number of nodes. Similarly, Yu *et al.* [24] proposed a detection mechanism (called SybilGuard) that relies on social networks of peer-to-peer users. To understand how SybilGuard identifies sybils, imagine that every person exchanges keys with a limited number of well-known trusted friends. By putting together these

social networks, one observes that an attacker and its sybil identities have a limited number of friends and, consequently, are identifiable. Danezis and Mittal [9] recently showed that SybilGuard suffers from high false negatives - honest individuals are often misclassified and unfairly considered sybils. To fix this problem, Danezis and Mittal then proposed a Bayesian model that detects sybils using, again, social networks. The centralized version of this model (one that runs on a server that stores the full social network) shows low false negatives. Also, SybilGuard has been validated in peer-to-peer networks but it is not meant to work in mobile networks. That is because it requires that most devices are online, which is difficult to guarantee for mobile networks - in them, devices are portable and, as such, are often unavailable.

Fortunately, mobility has recently ceased to be a source of weakness and has been turned into a source of strength. Čapkun *et al.* [23] exploited mobility for helping co-located mobile users to exchange cryptographic material. That material may be possibly used to limit the number of identities a single user can posses. The question of how to do so was not the focus of Čapkun *et al.*'s work. More recently, Piro *et al.* [21] proposed to keep track of how identities move. By observing that sybil identities are often seen together (as opposed to honest people's identities that are free to move at will), devices are able to identify a single attacker who keeps on using the *same* bogus identities. Still, this solution allows malicious individuals to continuously create disposable identities and go unnoticed.

From this review of literature, one may well conclude that combining mobility and social networks help to defend honest users against sybils. However, in Section IV, we will see that using social networks alone is not sufficient. We will see that a real community of one hundred mobile users would suffer from having 20% of its members sybils - more than 21% of honest members' interactions (on average) would be with sybils and, partly, those members would also mistakenly refuse to interact with each other, resulting in considerable lost opportunities (Figure 1). The reason for this is that existing defences assume that social networks are necessarily fast mixing. Alas, that has not turned to be true for small social networks. It thus seems that a new way of defending mobile users against sybil attacks is needed. But what sort of defence should we use? Ideally, the defence mechanism should guarantee that only honest identities are accepted and that only bogus identities are rejected.

III. Our Proposal: MobID

We design one such mechanism and call it MobID:

What it is: MobID defends in-range portable devices against sybil attacks in a fully decentralized way. A sybil attack is one in which a malicious individual has managed to convince one or more honest people to be their friends, perhaps by social engineering. The malicious individual then introduces and controls a very large set of corrupt participants (dubbed sybils). The presence of such attackers would then make it impossible for collaborating devices: (1) to run localization algorithms that increase the precision of location-based services (as attackers would inject false information); or (2) to cache Web content to avoid monetary costs of cellular or wireless providers (as attackers would discourage any form of sharing by getting free Internet connection without providing any return).

Problem Statement: MobID guarantees that an honest individual accepts, and is accepted by, most other honest people with high probability. The end result is that honest people successfully trade services with each other.

Defences against sybils traditionally focus on excluding malicious individuals. To meet this requirement, those defences pay the price of high false negatives (excluding a considerable number of honest individuals). However, to defend against sybils, one does not have to necessarily exclude malicious individuals but may simply limit their corrupting influence by excluding the bogus identities that those individuals create. Consequently, the end goal of MobID is not to filter out malicious individuals but is to limit those individuals' influence while minimizing false negatives.

When it works: MobID works under the following assumptions (most of which happen to be research findings):

Assumption 1: People have off-line relationships (have "friends") with whom they share their identities (e.g., their public keys). Mobile users may be willing to do so because, only by sharing their identities, they would then be able to trade services with each other.

Assumption 2: People identify themselves using public keys. A user needs to exchange her key only with her friends and, consequently, there is no need for any public key infrastructure: the user exchanges her key using either Bluetooth (whenever she meets her friends) or LoKey [20] (which would simply rely on text messages for the exchange). The user is also able to revoke her key by simply stopping using it and sending her friends a new one.

Assumption 3: People do not meet at random. Past research has extensively showed that people have few regular encounters [14]. They, for example, meet their friends and their familiar strangers (i.e., people who they do not know but meet regularly, say, on the way to work or at local coffee shops). That is shown to be true not only for college students [11] (against whose movements we will run our evaluation) but also for conference attendees [5], for hundreds of thousands of mobile users [13], and for a million of subway travelers [17].

Assumption 4: Honest nodes are well-connected in social networks while sybil nodes sit in the periphery. This property is at the heart of effective defences against sybil attacks [8], [24] and has been found to hold in various types of social networks [2].

How it works: Every mobile user who runs MobID exchanges keys with her friends. MobID then places the identities of the

devices it encounters into two networks whose nodes are the encounters' identities (keys) and whose links are their strong social connections (e.g., their close friends). More specifically, MobID places honest people in a "network of friends" and suspicious people in a "network of foes". The problem is that if devices were to meet randomly, then their networks would be sparse. To see why, say that A meets C at first, and it then meets D. In its networks, A can create a link $C - D$ only if C and D are friends. If encounters were random, then the probability that two (random) people (in this case C and D) may share encounters and are friends would be very low. However, according to recent studies on human mobility [17], [14], [5], [13], encounters are not random but are biased. The bias is introduced by people who, instead of moving randomly, move in groups (e.g., they move within their communities). Consequently, in our example, C and D may share encounters and are more likely than a pair of random individuals to be friends; that is, the link $C-D$ is likely to exist. Since links are not random but preferentially exist among honest individuals, those individuals end up to be well-connected in the social network (Assumption 4). Then, the theory goes, by measuring the network centrality of a stranger, one is able to determine whether the stranger is a sybil or not [9], [24].

MobID does so on a network of friends and on a network of foes. The problem is that sybils can artificially lower their rank on the network of foes by, for example, declaring to befriend only honest people. However, if they do so, they compromise their position in the network of friends and consequently lower their rank in it. Indeed, in Section IV, we will show that camouflaging the affiliation with bogus identities has little success.

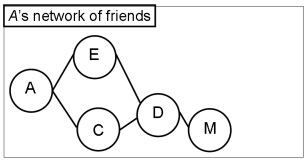
More specifically, MobID ensures that sybil attackers are detected with high probability by:

A. Recording human-established relationships.
B. & **C.** Reasoning not only on a network of friends but also on a network of foes.
D. Deciding whether to accept or reject.
E. Updating those two networks.

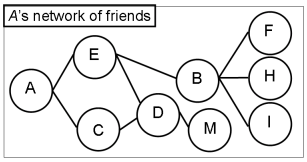
A. Recording Human-established Trust Relations

Say that A has to decide whether to accept or reject B. To do so, A needs B's list of friends. So, to begin with, B sends its list of friends to A. To prevent B from lying, B's list needs to be of special form, one in which B's friends certify their relations using their private keys (known only to them). Each friend F does so by concatenating its identifier (public key) and B's identifier (e.g., $PK_F||PK_B$) and by then signing the result with its private key (e.g., $S_F(PK_F||PK_B)$). If B's friends are F, H, and I, then B's list takes this form:

$$\begin{array}{rcl} PK_F & , & S_F(PK_F||PK_B) \\ PK_H & , & S_H(PK_H||PK_B) \\ PK_I & , & S_I(PK_I||PK_B) \end{array}$$



(a)



(b)

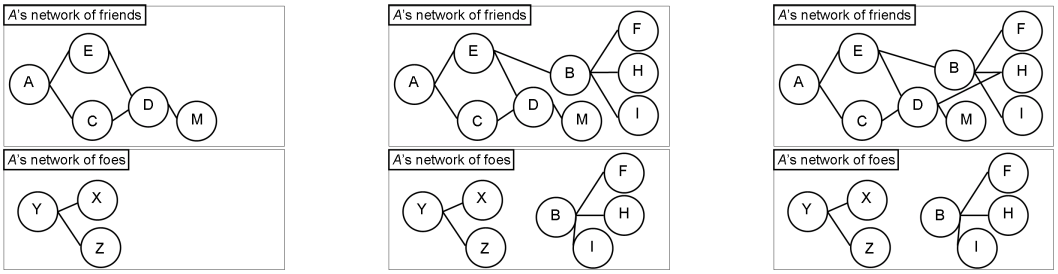
Fig. 2. **(a)** Network of Friends; **(b)** Network of Friends Updated.

B. Reasoning on a Network of Friends

Upon B's list of friends (which could be obfuscated or made anonymous as we shall discuss in Section V), A then decides whether to accept or reject B in three steps:

Step 1. *A incorporates B's list into its network of friends.* A updates its network of friends whose nodes are the identities of A's friends, A's encounters, and encounters' friends, and whose links represent the strong social connections among those identities. A connection exists between pair of individuals who trust *each other* not to launch a sybil attack (e.g., two friends). So links are bidirectional - they exist only between pair identities who trust each other. More concretely, consider that E and C are both friends of A, that A meets D, and that D claims to befriend E, C, and M. This situation produces the network in Figure 2(a). Then, if B's friends are F, H, and I, then A's updated network is that in Figure 2(b).

Step 2. *A ranks B on its network.* B's rank reflects B's importance in the network. The more central B's role in the network, the higher its rank. One common way of measuring centrality is to measure the network betweenness of B. Vertices that lie on many shortest paths between other vertices have higher betweenness that those that do not. However, such a definition assumes that information flows along shortest (ideal) paths in a network. In reality, information wanders around more randomly. That is why researchers have been introduced random-walk betweenness and have found that it performs best for several types of network (e.g., networks of Florentine families [19], of co-authorship [16], of sexual contacts [7]). Importantly, to measure the betweenness of B, A starts its random walks from itself, and it does so to break symmetry. If that would not be the case, then B could boost its rank by having its bogus identities (e.g., F, H, and I in Figure 2(b)) mirror the honest topology (e.g., that of A, E, C, D, and M in the same figure).



(a) (b) (c)

Fig. 3. **(a)** Networks; **(b)** Updated Networks; **(c)** Realistic Updated Networks.

Definition: The random-walk betweenness of B with prior A is equal to the number of times a random walk starting at A and ending at any node X passes through B, averaged over all X.

To compute B's rank and normalize it within $[0, 1]$, we divide B's betweenness by the maximum of possible paths (i.e., by $(\frac{1}{2} \cdot n \cdot (n-1))$), where n is the total number of nodes in the corresponding network.

> **Step 3.** *Depending on B's rank, A decides whether to accept or reject B.* The higher B's rank, the likelier B is honest. Since sybils do not have many real friends (they are not connected to many central nodes), they sit in the periphery of the network and are rarely traversed by a random walk (they rank poorly).

That, at least, is what would happen in a large network of friends. However, portable devices store only tiny portions of the network, and B can easily boost its rank in a tiny network. To see how, take again the network in Figure 2(b). B may have fooled E into believing that they are friend, and it may have then fabricated the public keys F, H and I, and pretended that those keys belong to multiple individuals when, in reality, are sybils under its control. In a large network, if limited, B's list of bogus identities does not have any impact [9], [24]. By contrast, in a small network, B's rank is boosted to the point of mistakenly accepting B. To fix this problem, we next let A reason not only on a network of friends but also on a network of suspicious individuals (which we call foes).

C. Reasoning Also on a Network of Foes

In this case, to decide whether to accept or reject B, A carries three steps again:

> **Step 1.** *A incorporates B's list of friends in both of its networks.* If, for example, A's two networks are those in Figure 3(a) and B's friends are F, H, and I, then A's updated networks are those in Figure 3(b).
> **Step 2.** *A ranks B on its two networks.* Again, the rank is the number of times B is traversed by a random walk between A and any other node. The only difference is that now A produces two ranks: B's rank on the network of friends (which we call $GoodRank$) and B's rank on the network of foes (which we call $BadRank$).
> **Step 3.** *Depending on both of B's ranks, A decides whether to accept or reject B.* We will see next that A takes this decision in two different ways - A either compares the two ranks in a linear way ($GoodRank > l \cdot BadRank$) or clusters them with the ranks of previously encountered nodes (and, upon that comparison, it then decides whether to accept or reject B).

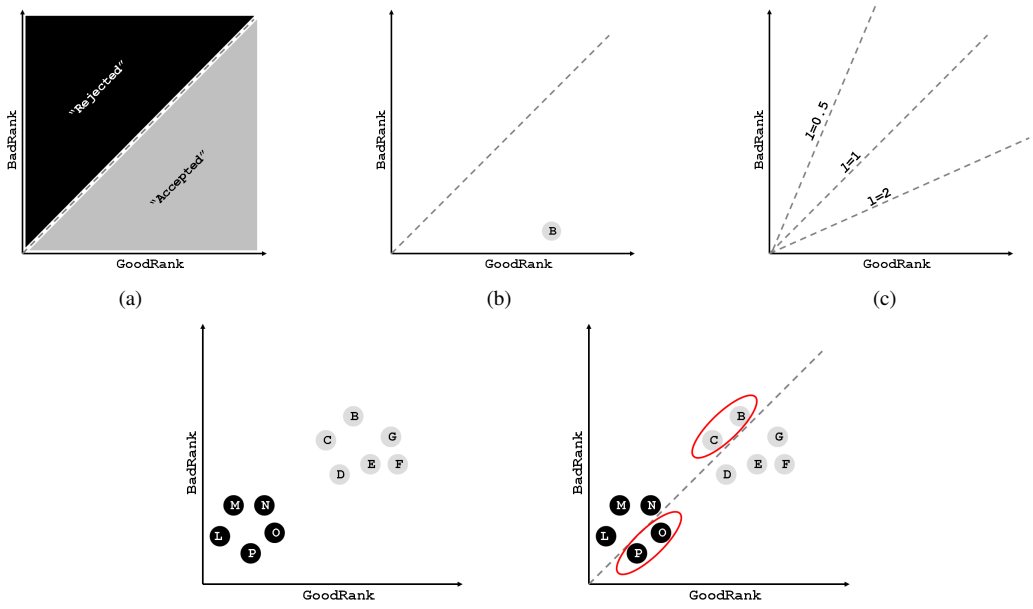
Again, to make the two rankings comparable, we normalize them to lie between 0 and 1, that is, we divide them by the maximum number of possible paths (i.e., $(\frac{1}{2} \cdot n \cdot (n-1))$, where n is the number of nodes in the corresponding network).

D. Deciding Whether to Accept or Reject

Comparing Ranks Linearly. The simplest way to compare B's ranks is to see whether $GoodRank > l \cdot BadRank$. If that is the case, then A accepts B; otherwise, it rejects B. For example, if $l = 1$, the dividing line is straight and defines two areas (Figure 4(a)): one above the line in which nodes are rejected, and the other below the line in which nodes are accepted. If B is below the line (which means $GoodRank > BadRank$), then B is accepted (Figure 4(b)).

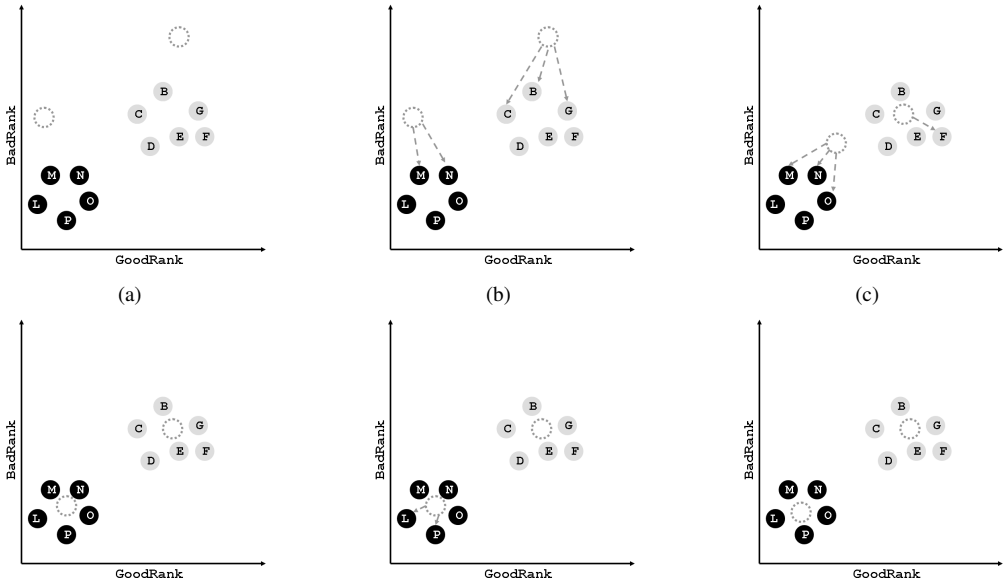
However, linear comparison poses two problems: (1) one needs to arbitrarily set l; and (2) deciding who is sybil is not always clear cut. To understand the latter point, consider Figure 4(d): the dark circles correspond to five sybils and the light circles to six honest individuals. The problem is that, by using linear comparison, one misclassifies two sybils and two honest individuals (circled in Figure 4(e)).

Clustering Ranks. To fix that problem, one should be able to group (cluster) the circles of Figure 4(d) into two sets - "sybil set" and "honest set". The simplest and fastest clustering algorithm is *K-means* clustering. This algorithm generates k clusters and determines which circles belong to which cluster depending on the structure of the data. In our case, since $k = 2$ (we have two sets to cluster - "sybil set" and "honest set"), *K-means* clustering begins with two randomly placed centroids - circles representing the centers of the clusters (dashed empty circles of Figure 5(a)). The clustering then assigns every circle



(a) (b) (c) (d) (e)

Fig. 4. Comparing Ranks Linearly.



(a) (b) (c) (d) (e) (f)

Fig. 5. *K-means* Clustering on Ranks.

to the nearest centroid - in the case of Figure 5(b), C, B, and G are assigned to the top centroid and M and N are assigned to the bottom centroid. After the assignment, the two centroids are moved to the average location of all nodes assigned to them, and the assignments are redone (Figure 5(c)) - it now turns out that also O is close to the bottom centroid, and F to the top one - so the two centroids accordingly move (Figure 5(d)). This process repeats until the assignments stop changing. Figure 5(f) shows the final result for which L, M, N, O and P are in one cluster (sybil), and B, C, D, E, F, and G in the other (honest). A cluster is considered sybil (honest) if the majority of its circles are sybil (honest). The clustering for B is done only if B's $GoodRank$ and $BadRank$ are both defined (are different than zero). Otherwise, B is considered:

$$\begin{cases} \text{sybil}, & \text{if } BadRank > 0 \text{ and } GoodRank = 0, \\ & \text{or if } BadRank = 0 \text{ and } GoodRank = 0; \\ \text{honest}, & \text{if } GoodRank > 0 \text{ and } BadRank = 0. \end{cases}$$

Importantly, for either way of comparing ranks, by creating bogus identities, B does not gain anything. That is because if

B creates bogus identities, then it would artificially boost not only its $GoodRank$ but also its $BadRank$.

However, one may rightly say that, even if B is honest, it would be rejected if it is unknown to either A, A's friends, or A's encounters. In Section IV, we will see that the extent to which honest people are mistakenly rejected is very limited. That is because honest people tend to have social connections with other honest people. So, if B's friend H is honest, then it would likely link to at least one of the node in A's network; for example, to D. The result would be that A adds $D-H$ in its network of friends (Figure 3(c)) and, because of that, B's $GoodRank$ increases - B will be traversed by random walks more often.

Also, networks of foes help to detect colluding attackers. To see why, consider that F and X (B's and Y's sybil identities) collude and claim to befriend each other. That results into an additional link $X-F$ in the network of foes of Figure 3(b), and that link increases the probability that B and Y are traversed by random walks - that is, it increases both B's and Y's $BadRank$s.

E. Updating the Two Networks

At this point, B and its friends are still stored in A's networks, but that network needs to be updated depending on whether B has been accepted or rejected. A does so by removing B and its friends from its network of foes, if A accepts B; or from its network of friends, if A rejects B. This results into a network of friends that contains people who have been accepted (plus their friends) and into a network of foes that contains people who have been rejected (plus their friends).

This way of updating the two networks is reasonable but may fail at times. More specifically, it may:

- Enlist honest people in the network of foes, and it may do so in two occasions:
  - A sybil lies and says it befriends a set of (real) honest people. However, to be believable, the sybil needs to produce relationships that are certified using its friends' private keys. But the sybil cannot do so simply because those keys, being private, are known only to their owners.
  - Device A mistakenly rejects honest device B simply because B is unknown. In that case, B is in the network of foes and will removed only if A meets and accepts at least one of B's friends. To see why, take one of B's friends C. If A accepts C, then, A deletes C and its friends (including B) from the network of foes.
- Enlist sybils in the network of friends, and it may do so in two occasions:
  - A sybil may fool honest people into believing they are her friends. Realistically, only few people may fall victims of the sybil. Consequently, the sybil's identity would not rank as honest identities do largely because of its marginal position in the network.
  - A sybil is mistakenly accepted. That would happen only if the sybil ranks well in the network of friends. In our evaluation, we will see that this is very unlikely.

IV. Evaluation

The goal of MobID is to both reject sybils and accept honest people. To ascertain the effectiveness of MobID at meeting this goal, our evaluation ought to answer two questions:

- **Robustness:** How effectively does MobID protect against sybils? More specifically, does MobID fail to detect some sybils (does it suffer from false negatives)? What is the fraction of honest individuals (mistakenly) considered sybils (fraction of false positives)? (Section IV-A)
- **Overhead:** What time, storage, and communication overhead does MobID impose on a mobile phone? (Section IV-B)

A. Robustness

We set up simulations driven by real data (empirical observations) about how individuals move and when they interact. Then, while running our simulations, we evaluate the robustness of MobID by keeping track of:

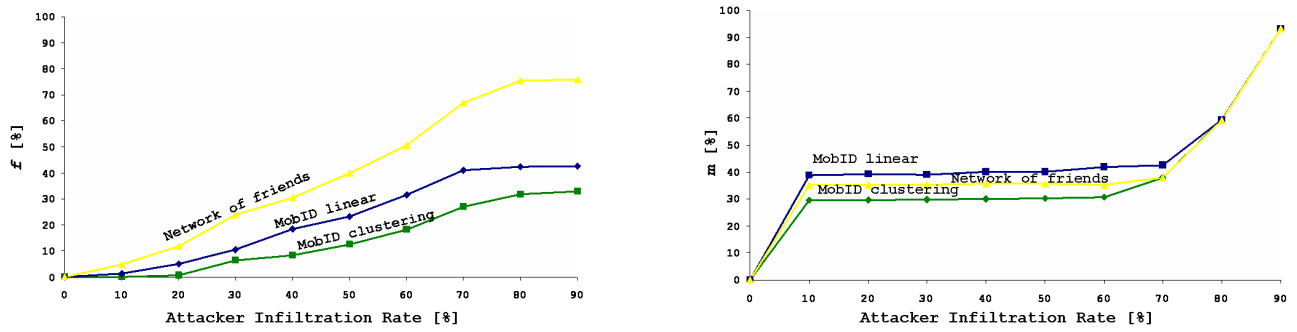
1) The fraction f of *fulfilled sybil interactions* (i.e., interactions that have been fulfilled by Sybils over those attempted);
2) The fraction m of *missed interactions* (i.e., interactions mistakenly refused over those attempted by honest people).

By doing so, we assess to what extent MobID reduces both f and m.

Simulation Setup. The setup of our simulations is based on observations about:

- *How individuals move.* We need to know how people move and their social networks. Mobility traces do not come with corresponding social networks - one usually has the mobility traces of some people and the social network of others. The only exception is the Reality Mining project at MIT [11], which offers data about how 96 people moved while carrying their mobile phones for 9 months and about who those people befriend. We use this project's mobility traces and social network (largest connected component) for our evaluation. While focusing on these mobility traces, we expect the results obtained to equally hold in other human mobility scenarios; in fact, as existing analysis demonstrates, such traces share many unifying features (e.g., node inter-contact time, formation of cliques) with other mobility traces (e.g., Cambridge and Dartmouth traces [1]).
- *When they interact.* We consider that mobile users run an application for sharing digital content (e.g., podcasts). To model *when* they interact (i.e., exchange digital content), we consider that two individuals interact if they come into range and have interest in common. We choose

[1] http://crawdad.cs.dartmouth.edu/



(a) (b)

Fig. 6. MobID's effectiveness against attackers' infiltration rates. Against infiltration rate, they are plotted: **(a)** Fraction of fulfilled sybil interactions (f). **(b)** Fraction of missed honest interactions (m).

this simple interaction model to better interpret the corresponding results. Our mobility traces tell those who come into range, so we simply need to model those who have interests in common. To do so, we model interests as categories of digital content (e.g., music genres) and distribute those categories across individuals. We do so by assigning categories at *random*. However, that does not reflect reality on two counts. The first is that some categories are more popular than others. More specifically, category popularity often follows a Zipf distribution. For example, that is true for videos [4] and music files [17]. The second count is that one may well befriend similar people since homophily (i.e., love of "similar others") has been found to play a starring role in human society. Therefore, to assign categories, we need to account for those two aspects. We can do so because the Reality Mining project not only tracked how individuals moved but also recorded their social network ("who knows whom"). We have exploited this added knowledge to *realistically* distribute interests so that: (1) friends share more categories than unknown individuals do; (2) category popularity follows a Zipf distribution. In short, any two individuals interact if they share content categories. We assign content categories to individuals in two ways: *random* and *realistic*. Below we will show the results for the realistic distribution. For the random distribution, the results are slightly better but the difference is negligible.

- *Who is sybil.* Finally, we need to determine which individuals are sybils. Similarly to Danezis *et al.*'s evaluation [8], we introduce 100 new people (sybils) who infiltrate $1\%, 20\%, 30\%, \ldots, 90\%$ of the Reality Mining social network. By infiltrating, we mean the ability of attackers to have real social connections in the social network. So "100 sybils infiltrate 20% of the network" means that "19 real individuals (20% of the network) turn to be attackers and control the remaining 81 sybil identities". Each sybil uses any of its bogus identity with equal probability.

Reducing (vulnerability) f. One would expect that the fraction of interactions that sybils fulfill (f) mainly depends on how diffusively sybils infiltrate the social network. Figure 6(a) plots f against sybils' infiltration rate for three situations:

- *Network of friends.* Attackers declare only their affiliation with the honest people they have managed to fool. They do so in the attempt to camouflage affiliation with any bogus identity. On the other hand, to defend themselves, mobile users employ only networks of friends (no network of foes). This setting shows how well existing solutions based on social networks would do in the best case.
- *MobID linear.* Attackers infiltrate the community as we have just described under the item "Who is sybil". To defend themselves, mobile users compare two rankings (one from the network of friends, and the other from the network of foes), and they do so linearly using the three coefficients $l = \{\frac{1}{2}, 1, 2\}$.
- *MobID clustering.* This situation is as the last one except for the comparison of the two rankings, which is now done by *K-means* clustering.

As one expects, Figure 6(a) shows that f increases with the attackers' infiltration rate for the three strategies. For *MobID linear*, the best linear coefficient l is 2. That is because, by increasing l, one conservatively reduces the acceptance area (the dividing line goes down) and consequently is less exposed to attackers (lower f). Overall, *MobID clustering* performs best. For example, if 20% of people in the community turn to be attackers, against *MobID clustering*, those attackers only manage to fulfill less than 1% of the interactions that would have happened if no protection had been in place. Also, the way the two remaining strategies perform suggests that it pays to reason not only on a network of friends but also on a network of foes - all the more so if one uses *K-means* clustering instead of linear comparison.

Reducing (lost opportunities) m. Since they are unable to distinguish between completely unknown individuals and sybils, defence strategies may mistakenly reject honest people. Now the question is to which extent they do so. By plotting the fraction of missed interactions with honest people in

Figure 6(b), we observe that, up to a 60% infiltration rate, each of the three strategies shows a flat fraction of missed interactions: *MobID clustering* approximately is flat at 30% of missed interactions, *network of friends* at 35%, and *MobID linear* at 40% (with $l = \frac{1}{2}$). Predictably, for *MobID linear*, the best coefficient is $l = \frac{1}{2}$. That is because, by decreasing l, one increases the acceptance area (the dividing line goes up), openly accepts more people, and consequently suffers little from lost opportunities (lower m). If attackers manage to diffusely infiltrate the community (more than 70% of its members), most honest people are abruptly excluded from the system. That is because their networks become extremely sparse and they are unable to identify sybils. These results are in line with research on complex systems, which shows that phase transitions tend to describe the robustness of social networks, in that, after a critical point, networks abruptly break [1]. Once again, *MobID clustering* proves to be the most effective strategy - for high attacker infiltration rate (up to 60%), it rejects 35% of the interactions, which happen to come from 17% of community members. This result improves on existing approaches. However, to avoid the social exclusion of that 17% of the community, one should integrate MobID with other mechanisms, some of which are listed in Section V under "Social Exclusion".

B. Overhead

Communication Overhead. MobID requires devices to exchange their lists of trusted friends. Each item in this list consists of an identity (public key of 1024 bytes) and a signed relationship (128 bytes). A list of t trusted friends requires to transmit $(t \cdot 1.12Kb)$ (e.g., $112Kb$ for 100 trusted friends, which is pessimistically high). In theory, using Bluetooth version 1.2, devices can transfer 434 kb/s. In practice, environmental conditions (e.g., human bodies that interfere with Bluetooth's frequencies) lower that speed. Still, at a speed as low as 112 kb/s, a mobile phone can transmit that list in one second.

Computational Overhead. MobID should perform random walks on small networks. The complexity of performing random walks depends on the size of the network and on the type of algorithm used. The running time for Newman's centrality measure [19] is $O((m + n)n)$ and that for Brandes's [3] is $O(mn)$ (where n is the number of nodes, and m is the number of links). So, if one uses the latter, the computational overhead on each device is acceptable and is limited by the fact that networks are relatively small - a device's networks contain only the people the device has encountered.

Also, in addition to random walks, each device should run *K-means* clustering (which is the fastest clustering algorithm), and it should create a list of friends, and that requires public key encryptions. For each friend, a device concatenates and encrypts a pair of identifiers. To attain a minimum level of security, America's National Institute of Standards and Technology (NIST) recently suggested RSA (for public key encryption) with a key of at least 1024 bytes. However, the use of RSA may slow down current models of mobile phones. So, we consider a second public key encryption algorithm - ECDSA [15]. We do so because, security level being equal, compared to RSA, ECDSA uses smaller keys and, consequently, signs messages faster. To encrypt, a J2ME implementation of RSA takes 4.07 seconds (on Nokia 6600) or 2.72 seconds (on Ericsson P900). As one expects, ECDSA takes much less: 0.76 seconds (on Nokia 6600) or 0.42 seconds (on Ericsson P900). This overhead is acceptable and may be significantly reduced: public-key encryptions are well-established operations, and one may consequently imagine a (near) future in which those operations will be partially or fully implemented in hardware for higher performance.

Storage Overhead. MobID stores two networks whose nodes are identities (public keys) and whose links are social relations. Identities are stored as a list of public keys and links as a connectivity matrix. If n is the maximum between the sizes of the two networks, then the connectivity matrix requires n^2 bits (e.g., $1.22Kb$ for $n = 100$), and storing n public keys requires $(n \cdot 1024)$ bytes (e.g., $100Kb$ for $n = 100$). Overall, to store two networks of 100 nodes, a device needs $103Kb$ $(2 \cdot 1.22 + 100Kb)$. This is negligible, mostly because phones come with GBs of storage nowadays.

V. Discussion

Based on the previous results, we now discuss some open questions.

Bootstrapping. To join a community, new MobID users should be introduced by current members. To do so, new users may identify the members they know by scanning their contact lists, and they may then ask those members for an introduction.

Social exclusion. For those who do not have friends, MobID and, for that matter, any sybil prevention mechanism based on social networks would not work. Indeed, such mechanisms translate into social exclusion: mobile users with no friends are excluded from the system and cannot trade services. So, for those individuals, real systems should also deploy alternative mechanisms. One such mechanism is *trust negotiation* - to gradually establish trust, strangers iteratively exchange digital credentials [12]. For example, a mobile user might receive a credential from her university that certifies that she is a student. Then, to access her mobile e-learning repository, the user could employ that credential rather than proving her identity. Another alternative mechanism is to adapt a point-based system called *Thawte*. In it, people have their identities certified by meeting one or more notaries. Those notaries check identification and assign points based on their experience. After collecting a certain number of points, people obtain certification of their identities; after a higher number of points, they can also become notaries themselves. Currently, Internet servers collect and store user points in a "Web of Trust". However, that process could be made fully decentralized by having mobile users run existing distributed algorithms for reasoning on a "Web of Trust" [22].

Privacy Concerns. By exchanging their networks, users reveal people with whom they have interacted, and some users may not feel comfortable doing so for privacy concerns. Our design

partially alleviates these concerns because it uses anonymous public keys for identifying users and only friends can associate keys with real-life identities. However, profiling people based on the use of their anonymous identities is still possible. That is why recent research has been focusing on how to verify social ties while exposing minimal information about them [18].

Deployability. MobID is easily deployable largely because it does not require any infrastructure or any specialized hardware, and it taps into the well-understood concept of friendship.

Real-life friends versus virtual friends. MobID requires that users have and specify real-life connections. In virtual communities (e.g., social network websites), people do not specify only their real-life social connections but often tend to connect to hundreds of virtual identities. The question of whether MobID users will be able to differentiate between their real-life friends and virtual ones needs further research - it should be treated as a testable hypothesis rather than an established fact.

Key revocation and making new friends. Whenever a user makes new friends, she enlists them (and their public keys) in her list of friends. To revoke her key, a user simply stops using it, creates a new key, and sends her friends the new key.

Attacking MobID. The effectiveness of MobID relies on sybil attackers having a very limited number of real-life social connections. However, there are several ways an attacker might acquire connections:

- The attacker convinces honest users in the system to "be her friends" in real life. But that is difficult to do with a significant number of users. Still, *MobID clustering* proved to be resilient to a large fraction of community members who exploited their real-life connections to launch sybil attacks.
- Worryingly, if the attacker manages to convince an honest user to be her friend, then she can create bogus identities at will. However, those identities would sit "behind" the attacker and, as such, they would be rejected because they rank poorly on a network of friends.
- More worryingly, multiple attackers may collude to increase their chance of being considered honest. However, collusion not only results in increasing the colluders' $GoodRank$ but also their $BadRank$, and, consequently, the colluders will be aptly rejected.

VI. Conclusion

MobID is a protection mechanism that makes in-range portable devices resilient to sybil attackers with high probability. These attackers disrupt sharing communities and then make themselves untraceable by producing bogus identities. MobID relies on the fact that attackers may create many bogus identities but few real-life relationships. Using real mobility and social network data, we have validated that the version of MobID that uses *K-means* clustering performs best - for example, it protects against attackers who infiltrate 20% of a real mobile community without causing any disruption. MobID also scales - it entails reasonable storage, communication, and computational overhead. To further evaluate MobID, we are studying how underground passengers (of the order of 1 million) happen to be co-located and how their mobility patterns can be overlayed with synthetic social networks.

References

[1] R. Albert, H. Jeong, and A. L. Barabasi. Error and attack tolerance of complex networks. *Nature*, July 2000.

[2] S. Boyd, A. G. B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *Proc. of INFOCOM*, 2005.

[3] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 2001.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. of IMC*, 2007.

[5] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 2007.

[6] R. Chakravorty, S. Agarwal, S. Banerjee, and I. Pratt. MoB: A mobile bazaar for wide-area wireless services. In *Proc. of ACM MobiCom*.

[7] R. Chrisley, G. Pinchbeck, R. G. Bowers, D. Clancy, N. French, R. Bennett, and J. Turner. Infection in social networks: using network analysis to identify high-risk individuals. *American Journal of Epidemiology*, 2005.

[8] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. J. Anderson. Sybil-Resistant DHT Routing. In *Proc. of ESORICS*, 2005.

[9] G. Danezis and P. Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. In *Proc. of NDSS*, 2009.

[10] J. R. Douceur. The Sybil Attack. In *Proc. of IPTPS*, 2002.

[11] N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Computing*, 2006.

[12] K. B. Frikken, J. Li, and M. J. Atallah. Trust Negotiation with Hidden Credentials, Hidden Policies, and Policy Cycles. In *Proc. of NDSS*, 2006.

[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 2008.

[14] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2008.

[15] D. B. Johnson and A. J. Menezes. Elliptic Curve DSA (ECSDA): An Enhanced DSA. In *Proc. of USENIX SSYM*, 1998.

[16] X. Liu, J. Bollen, M. L. Nelson, and H. V. de Sompel. Co-authorship networks in the digital library research community. *CoRR*, abs/cs/0502056, 2005.

[17] L. McNamara, C. Mascolo, and L. Capra. Media Sharing based on Colocation Prediction in Urban Transport. In *Proc. of the ACM MobiCom*, 2008.

[18] Michael J. Freedman and Antonio Nicolosi. Efficient Private Techniques for Verifying Social Proximity. In *Proc. of IPTPS*, 2007.

[19] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 2005.

[20] A. Nicholson, I. Smith, J. Hughes, and B. Noble. Lokey: Leveraging the sms network in decentralized, end-to-end trust establishment. In *Proc. of Pervasive*, 2006.

[21] C. Piro, C. Shields, and B. N. Levine. Detecting the Sybil Attack in Ad hoc Networks. In *Proc. of SecureComm*, 2006.

[22] D. Quercia, S. Hailes, and L. Capra. Lightweight Distributed Trust Propagation. In *Proc. of IEEE ICDM*, 2007.

[23] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Security in Ad Hoc Networks. In *Proc. of MobiHoc*, 2003.

[24] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *Proc. of ACM SIGCOMM*, 2006.

[25] D. Zhu and M. W. Mutka. Promoting Cooperation Among Strangers to Access Internet Services from an Ad Hoc Network. In *Proc. of PERCOM*, 2004.