Mobile Encounter-based Social Sybil Control

Martin Martinez*, Arvin Hekmati†, Bhaskar Krishnamachari*†, and Seokgu Yun
*Department of Electrical and Computer Engineering
†Department of Computer Science
University of Southern California
Los Angeles, California, USA
Email: {mart698,hekmati,bkrishna}@usc.edu, phantom@sovereignwallet.network

***Abstract*—We present a novel "Proof of Social Contact" approach to Sybil control that utilizes the analysis of digitally signed information about digitally signed pairwise encounters between mobile devices that are logged in a distributed ledger. To illustrate the approach, we show examples of analysis using binary classification techniques under two different adversary detection models, and evaluate them using a real-world mobile device encounter trace. We discuss a number of open problems and future directions that could be pursued by researchers in the field to realize and improve such a system and build on top of it.**

***Index Terms*—Sybil Attack, Blockchain, Social Contacts, Anomaly Detection**

I. INTRODUCTION

In many distributed and decentralized systems, a single malicious entity that presents multiple fake identities can gain an unfair advantage in resource allocation or being able to subvert the safety and security of the system. This is the well known problem of Sybil attack [1]. In permissionless blockchain protocols, controlling Sybil attacks is an essential ingredient, to maintain integrity of the consensus process.

Proof of Work [2], adopted in Bitcoin from the earlier Hashcash work [3] is the most well-known mechanism for Sybil control in such systems. However, by definition, it requires a significant amount of computation resulting in significant power consumption (Bitcoin's proof of work was shown to consume as much power as the entire country of Ireland [4]). More recently other techniques such as Proof of Stake [5], Proof of Elapsed Time [6] and Proof of Location [7] have also been developed.

Aiming to bring Blockchain and distributed ledger technologies to mobile devices in a scalable manner, our contribution in this work is to propose and explore a different approach to Sybil control that leverages the social behavior of mobile device owners and the ability of mobile devices to send beacons and messages to other nearby devices via low power Bluetooth communications. The approach we describe in this paper could be referred to as "Proof of Social Contacts."

The crux of our approach is to have mobile nodes send beacons to each other with digitally signed ID's, and log the ID's of the nodes that they encounter to a ledger along with time-stamps of the encounter. Now, despite the digital signatures, in a permission-less system, an attacker could try to associate multiple ID's with a device. To counter these attacks, we propose to use binary classification algorithms to identify suspicious ID's. Given a sufficiently strong algorithm, Sybil attacks will be detected with high probability and the corresponding ID's can be placed on a blacklist.

As an alternative to, or rather, going beyond the simple binary classification approach that we describe and evaluate in this paper, one could also have a more sophisticated machine learning algorithm such as a deep learning network that is trained to output a real-valued "credit score" for each node - with a high score indicating a lower likelihood that the node represents a Sybil (fake) identity. Such a score may be used as the basis of a decentralized credit rating mechanism and used in the context of a protocol to confer rights to take certain actions such right to send/receive transactions, right to validate, or right to participate in governance mechanisms, etc.

We describe the general approach and present some preliminary steps towards designing the system, illustrating it through two simple adversary detection models. We show the performance of two binary detection schemes for defending against the corresponding attacks over a set of real world mobility traces. These schemes could be implemented via the use of an off-chain oracle, possibly controlled via an off-chain governance/voting mechanism, or if the decentralized computation capabilities of the underlying blockchain platform allows, possibly on-chain as well. Now, in the real world, security is often an arms race between increasingly sophisticated attacks and correspondingly effective defenses. Keeping this in mind, we also discuss how the system proposed in this paper could be enhanced by an evolutionary decentralized approach to learning that allows new detection models to be proposed and adopted by users over time as they are found to be effective.

One motivation for this work is the way in which traditional banks collect and analyze customer's credit card usage behavior and have measures in place to detect and address credit card fraud — suspicious behavior is identified as arising from anomalous interactions, e.g. someone trying to purchase large volumes of goods at particular locations that are unrepresentative of their typical purchases or location patterns. Similarly, we believe that for most users, their pattern of movement and the locations that they are present in, will serve to create a baseline of "normal" social contacts that can help to flag unusual patterns as potential evidence of malicious behavior (Sybil attack).

The rest of the paper is organized as follows - in section II, we discuss some of the relevant prior work; in section III we discuss how encounter information is logged into a ledger; in section IV we discuss adversarial models; in section V and VI we present our simulation methodology and results; we discuss related issues and future directions in section VII, and wrap up with concluding comments in VIII.

II. Related Work

Sybil attacks can be generated for different purposes between mobile networks, auditing or reputation systems, however many of these approaches have been studied before and even corrected using diverse mechanisms as presented by Levine *et al.* [8]. Our study considers a novel approach for sybil control that uses anomaly detection based on encounter data.

The outlier/anomaly detection problem has been studied by various researchers going back several decades. For example, an experimental approach described by Grubbs [9] assumes an underlying distribution for the data such as Gaussian distribution in order to represent a statistical observation over the outliers in the model. Ramaswamy *et al.* [10] and Tan *et al.* [11] describe detection mechanisms which mainly are based on the geographic location of the data points. The main assumption with these approaches are that the regular nodes usually are close to each other as compared to the outliers which are far from them. One of the most important problems in this area is credit card fraud detection, which requires monitoring of users to effectively estimate, detect, or avoid undesirable behavior [12]. Several authors [13], [14], [15], [12] have proposed the use of artificial neural networks in order to learn the credit card fraud behavior of malicious nodes. Another related topic is anomaly detection in online social networks. Anomalies in this context are the individuals who act considerably different as compared to their peers in the network. Shrivastava *et al.* [16] observed that malicious nodes such as spammers have star-like structures which means each single anomaly node send out several messages to other innocent nodes. Akoglu *et al.* [17] showed that near-stars and near-cliques might be an indication of malicious nodes in social networks, quite relevant to the approach we take in this paper on encounter-based anomaly detection.

III. The Encounter Ledger

In the system we propose, each mobile device is meant to represent a unique digital identity (corresponding to an individual user). When two mobile devices are in direct communication range of each other, we consider that they are encountering each other. In order to store the encounter-based information among devices we define a distributed ledger with public access. The idea is to use the crowd-sourced data in this ledger platform to build trust based on the users' interactions. At this point the system will assume the following:

A node in the system will represent a mobile device in the real world, which at the same time will become the digital identity of an individual in this ecosystem.

Correct nodes can be either static or mobile.

Correct nodes will only access the ledger to report an direct interaction within some relatively small radio range (e.g. they could report just Bluetooth based interactions)

Nodes will beacon information that is digitally signed with their own private key

Correct nodes will validly timestamp the correct time for the encounter.

All of this interaction information will be stored in the encounter ledger (with the additional digital signature of the submitting node) and any node can have access to write on it.

Thus the encounter ledger is intended to provide authenticated information related to pairwise encounters among mobile devices and thus represents how different users perceived each other during the whole day.

Figure 1 illustrates a scenario where we assume that three correct node devices are interacting with each other. Initially device B has just found device A within range and will log the following information in the ledger:

Device A's digitally signed beacon containing its ID and time-stamp

The timestamp from the time device B that encountered device A (i.e. heard its beacon)

A digital signature from device B certifying the above information (A's digitally signed beacon and timestamp and B's time-stamp)

Note that in general both parties may report the encounter (checking for consistency in the reported receive time-stamps may be one way to provide additional security), but it is sufficient for either party to report the encounter as its report will contain both party's digitally signature.

Later when device C comes in range of A and B, they will add their encounters into the ledger as well (Fig. 1).

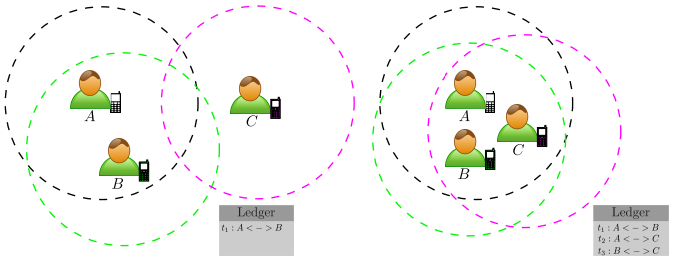

Fig. 1. Ledger

Note that in this work we are assuming that the ledger where encounter information is being logged already exists and can be used in a bootstrap manner. For example, it may be an existing public blockchain/DLT such as Ethereum or IOTA. Or it may be a Hyperledger Fabric or Tendermint-consensus based blockchain set up solely for this designated purpose with a set of permissioned validators, that nevertheless allows anyone in the world to submit transactions. In future work, we will consider how the throughput or costs associated with

this bootstrap encounter ledger may affect the security and scalability of the Sybil control mechanism.

IV. Adversary and Detection Models

We assume the Sybil attacker can make multiple identities and also connect to a random number of mobile phones through all or some of its fake identities. As we can see in figure 2, A is the attacker and has made 2 fake identities, A_1 and A_2. The fake identities make connection with random number of the adjacency list of A. Here, A_1 made connection with B and C. But, A_2 made connection with C and D.

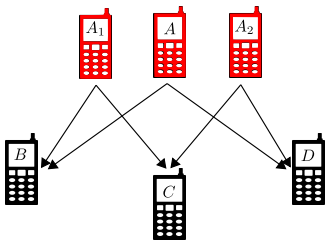

Fig. 2. Adversary Model

The adversarial model is defined with the following assumptions:

- The current model considers only one attacker as a possible adversary (to be extended in future work)
- Such attacker can generate many copies or "fake" identities that can interact with other neighboring nodes.
- The attacker cannot forge the digital signature of a correct (non-malicious) node.
- For this study, we consider that the attacker only logs encounters between any of its fake identities and correct nodes, not encounters between the fake identities (this can be relaxed in future work).
- Since we assume the neighbors are correct nodes that provide their own digitally signed time-stamps, the attacker has limited ability to manipulate the time-stamp by delaying the interaction recorded on the ledger (i.e. it could only pick a time-stamp that corresponded to or was close to the time-stamp for a real encounter).

A. Anomaly Detection by using Full Encounter Ledger

For this detection mechanism approach, we will use the information provided by the ledger which includes the identity of the devices who met and the time this took place. We will focus on utilizing the raw information provided on the ledger.

The defense method will consist of a comparison among interactions as long as these are within an arbitrary time threshold value . We claim a node is suspicious if we find on the ledger two or more records of an interaction taking place among one device and one or more different device within a timestamp difference less than . For example, in figure 2 and with the assumption that all of these interactions shown are within then our defense mechanism will consider the three nodes A , A_1 , A_2 as suspicious nodes because each will have an interaction reported in the ledger with node B , C and D, respectively. However, it is also necessary to mention that these nodes will also be consider suspicious since we do not know which of them generated the attack.

B. Anomaly Detection by only Using Mobile Encounters

In this method we are going to only use the mobile encounters information and neglect the time stamp, and the frequency of each encounter. This model is useful for the case that we only have the mobile encounter data available. The detection method that we used in this paper uses two threshold value and . We claim a node is suspicious if there are at least nodes who share at least the same neighbors. Continuing on our example in figure 2, we define = 2, and = 2. In this case, the three nodes A , A_1 , A_2 will be reported as suspicious because there are three ($\geq$ = 2) nodes who share at least two ($\geq$ = 2) adjacent nodes.

V. Simulation Setup

In this paper we use the dataset named Asturies which is provided by CRAWDAD.org. The dataset contains the encounter of mobile devices with each other at different timestamps. In order to evaluate our detection method, we assume there is going to be only one attacker who creates fake identities and it attacks the 80 percent of its neighbors at random. In order to present the results we calculate the confusion matrix parameters and show the results in Receiver Operating Characteristic (ROC) curves. In order to compute the mentioned parameters, we run the experiment by considering each mobile as attacker, one at a time, and get average over the results.

VI. Results

A. Anomaly Detection by using Full Encounter Ledger

For this approach, we vary the number of fake identities or copies that the adversary can generate to attack the network. For the results presented here, we consider an attacker with 1, 3 and 6 copies (arbitrarily chosen to show how the performance varies with increasing number of copies). Also, since we are unaware of when one of the attacker's copies might place the interaction between them and honest nodes, for these experiments we decided to have these "fake" nodes to write its interactions into the ledger after a uniformly random period of time between 0 and t_d sec.

1) t_d = 5min: When doing the ROC curve analysis, we are comparing the TPR(True Positive Rate) and FPR (False Positive Rate) which will provide more insight on how effective is our anomaly detection mechanism . As we can see from figures 3, 4, and 5, the number of copies that an attacker can use will reflect on the TPR value for a certain FPR. For example, in the case of threshold = 45 sec we seem to obtain the best TPR results in each scenario, however, the TPR value is higher when we are only against 1 copy than when we have to deal with 3 or 6 copies. At the same time we can see that the higher is the threshold value, the better results we will get in regard of our anomaly detection.

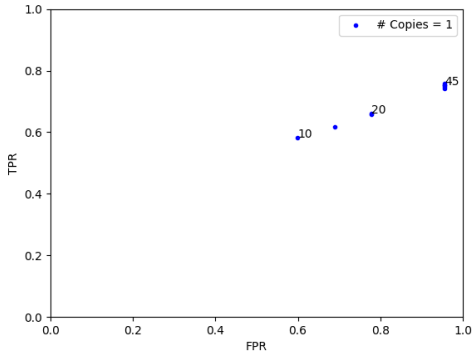

Fig. 3. Number of Copies = 1

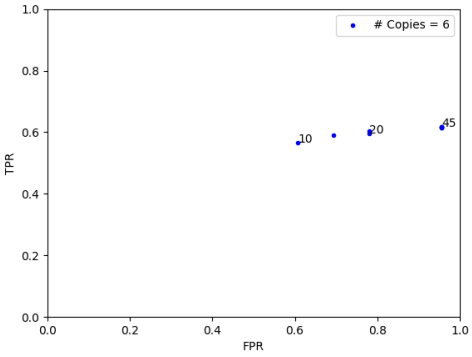

Fig. 5. Number of Copies = 6

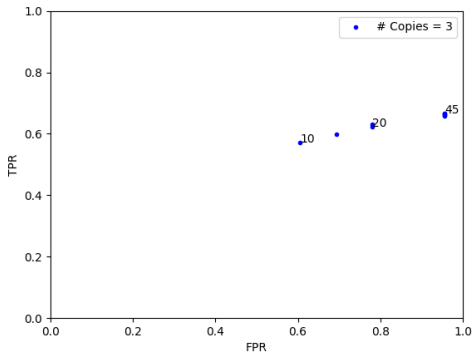

Fig. 4. Number of Copies = 3

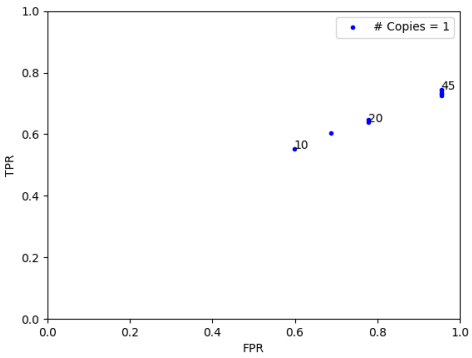

Fig. 6. Number of Copies = 1

2) t_d = 10*min:* Now in the case of figures 6, 7, and 8, we can see that the results also improve as the threshold value increases, as well as the defense mechanism is more effective for an attacker with one copy than with a larger number of copies. Additionally, when comparing the results between a 5 or 10 min random input time for *fake* interactions, we see a better overall performance for the 5 min case. This results is expected since a larger time for input will cause the mechanism to miss some attacks.

B. Anomaly Detection by only Using Mobile Encounters

In the following experiments we vary the number of fake identities that attacker have made to attack the mobile devices. We also vary the value of between 2 to 29 with step 3 (each node in the figure represent a specific value of , but, we fix the the value of .

1) = 1*:* As we can see in figures 9, 10, and 11, for the lower values of we are getting close to 1 TPR, but we also have a high FPR. This means that we are detecting all of the attacker nodes but there are also so many innocent nodes which we detect them as suspicious. By increasing the value of , both TPR, and FPR decreases. Generally, we are getting better results for the cases we have larger number of attacker copies. In fact, having more copies of the attacker makes it more impossible to have other innocent mobile devices who have the same connections as the fake nodes.

2) = 3*:* Figures 12, 13, and 14 are similar to previous ones but we change the value of from 1 to 3. According to these figures, again, for the larger value of the attacker copies we are getting better results. The important difference between this set of figures and previous ones is that, here we are never getting to 1 for TPR which means that we are missing some of the attacker nodes and announce them as innocent.

VII. Future Directions

The preliminary work we have presented here leaves open a number of interesting directions for future work. These include fleshing out more adversarial scenarios and developing algorithms to detect Sybil attacks under those scenarios. We briefly outline two other major directions below:

Richer Peer-to-Peer Reports and Decentralized Credit : While the scheme proposed in this paper only logs encounters in terms of the physical proximity of devices, more sophisticated schemes could be envisioned in which mobile nodes report additional data (such as location, financial transactions, verified claims based on decentralized identity systems on an opt-in basis etc.) from their interactions to create richer decentralized reports. As suggested in this work, machine learning algorithms could be designed to convert these reports

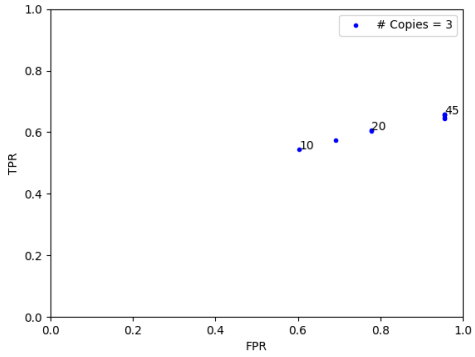

Fig. 7. Number of Copies = 3

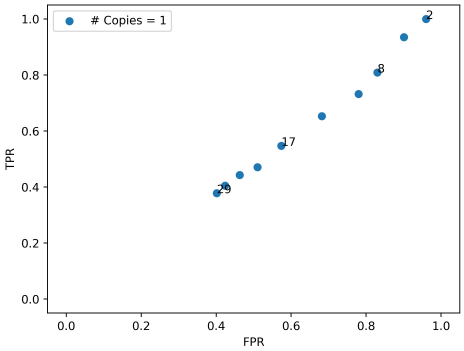

Fig. 9. Number of Copies = 1, = 1

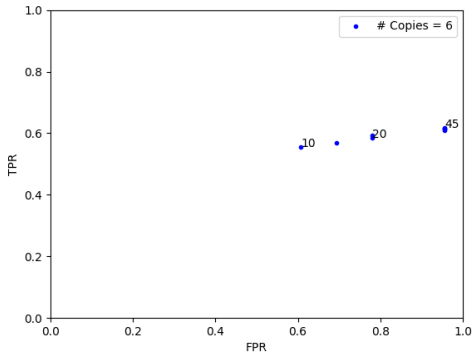

Fig. 8. Number of Copies = 6

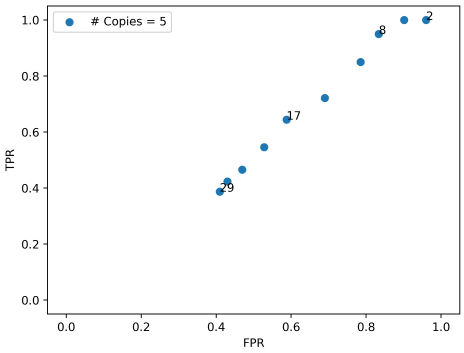

Fig. 10. Number of Copies = 5, = 1

to obtain a decentralized "credit score" that determines what potential actions the node can take in the system such as sending/receiving transactions, participating as a validator, participating in governance, etc. Individual recipients of a transaction may also use such information to speed up confirmation times by, for example, allowing a lower confirmation threshold for transactions involving individuals with a higher credit scores.

Decentralized Machine Learning: In this paper, we have shown how classification algorithms can be used to analyze ledger entries, determine and blacklist malicious (Sybil) nodes. Such an algorithm could be implemented either off-line as an oracle, possibly selected via voting using some governance mechanism, or on-chain as a smart contract or in-protocol mechanism if the platform supports significant decentralized computation. However, it is unlikely that a single algorithm will always work, in the face of increasingly sophisticated attackers that adapt to any defense. To address this, we suggest to incorporate a novel evolutionary approach to deploying novel algorithms over time. The basic idea is that individual researchers or organizations may propose new algorithms, that others get to evaluate and vote on. They are offered an incentive if their algorithm is "approved". Nodes in the system may choose from an ensemble of such algorithms to decide how they want to implement their own blacklist, or the system as a whole may pick a single "best" algorithm (perhaps changed over time) or an ensemble of algorithms to be used to evaluate Sybil attacks based on the encounter ledger. In future work, we plan to elaborate on this evolutionary decentralized machine learning approach.

VIII. CONCLUSIONS

Motivated by the large-scale availability of mobile devices and the need to broaden access to distributed ledgers to users of mobile devices, we have presented in this work a novel way to enable Sybil control based on the mutual logging of mobile device encounters. We have described a somewhat simple system that uses and analyzes information from logged encounters to determine which identities are likely to be the result of Sybil attacks. The underlying assumption behind this approach is that the social lives of most individuals form relatively predictable and routine patterns and that attackers who try to create new identities will be detected because of these patterns.

We have presented two simple examples of adversarial models and corresponding detection algorithms and evaluated them on real mobility traces. The results suggest that high true positive rates must contend with high false positive rates, so that care is needed to set detection thresholds. This is our first