Detecting Mobile Crowdsensing Sybil Attackers via Presence Verification

Cihan Eryonucu
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
eryonucu@kth.se

Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

ABSTRACT

Mobile crowdsensing (MCS) relies on smart, portable devices to conveniently collect sensory data from our surroundings. MCS-based apps, e.g., Google Maps, are already well-integrated into our everyday lives. However, Sybil-based attacks, with an attacker creating many fake identities and the illusion of numerous contributors to influence MCS-based functionality, pose a significant threat. MCS systems need security, including mechanisms to vet incoming users and prevent the introduction of Sybil nodes. Intuitively, each incoming contributor can be verified to be an actual device near other devices by other newcomers and contributors already part of the MCS system. We propose a novel cooperative MCS user presence verification protocol based on this idea, also ensuring users are physically present in locations relevant to the MCS tasks. The protocol leverages a commodity component, Bluetooth, with each user broadcasting to prove their presence to users listening and recording Received Signal Strength Indicator (RSSI) values in multiple randomized rounds. The presence verification is done by a simple server tasked with registering users and orchestrating the protocol based on the collected information. The protocol identifies a broadcast signal on behalf of multiple users, indicating a potential Sybil behavior. We conduct extensive simulations to evaluate the performance of the proposed method, demonstrating its ability to find Sybils with high accuracy even when Sybils are nearly the majority in the protocol session.

CCS CONCEPTS

• **Security and privacy** → **Security protocols**; *Privacy-preserving protocols.*

KEYWORDS

Mobile crowdsensing, Sybil Attacks, Presence Verification

ACM Reference Format:
Cihan Eryonucu and Panos Papadimitratos. 2024. Detecting Mobile Crowdsensing Sybil Attackers via Presence Verification. In *Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy (CPSIoTSec '24), October 14–18, 2024, Salt Lake City, UT, USA.*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3690134.3694826

CPSIoTSec '24, October 14–18, 2024, Salt Lake City, UT, USA.

ACM ISBN 979-8-4007-1244-9/24/10.
https://doi.org/10.1145/3690134.3694826

1 INTRODUCTION

Mobile devices with improving computational and sensory capabilities pave the way for the emergence of a powerful paradigm: mobile crowdsensing (MCS). MCS users collect data on their environments, capturing dynamically changing information. Generally, the higher the participation, the more and better data the MCS has, and the better their analysis is. However, openness to register and participate also makes the system susceptible to Sybil-based attacks [4]: an attacker creates multiple fake identities to have a disproportionate influence in the process.

The effectiveness of Sybil attacks usually relies on the cheap creation of many accounts. This is even more effective in MCS because each participant's contributions are usually weighted equally. Each Sybil (fake) user is equivalent to a genuine one, plus the Sybil users can be placed in any location/area where the attacker wishes to affect MCS data. The Sybil-based attack can be mounted by emulating many user devices artificially placed in a specific area for a location-based service (LBS) app to falsely inflate place popularity [5] or alter road conditions [18]. Under large-scale Sybil-based attacks, the MCS system cannot provide truthful information on sensed phenomena.

Ideally, MCS systems should only accept actual users with an actual device in an actual location; thus, validating the physical presence of the users is intuitive for Sybil-free MCS. Several proposals exist to detect Sybil nodes (we use users or *nodes* interchangeably) in Wireless Sensor Networks (WSNs): verifying users using ranging-based techniques [13], employing trust-anchor nodes [9] as trusted verifiers, monitoring network traffic [16], and creating proximity graphs based on user presence [17]. Although these methods improve security and Sybil-resilience, they are not compliant with MCS and real-world scenarios. For example, MCS users are mobile, unlike static WSN nodes. Ranging techniques are susceptible to noise and obstacles, thus harder to use with precise measurements. Trust anchors can be hard to recruit/establish/deploy, especially in a large area. Finally, as MCS thrives with high participation, schemes should not discourage participation and have restrictive requirements, e.g., devices with special hardware (e.g., UWB [13] for ranging) or forcing users to solve CAPTCHA [1]. In this work, we define a stronger Sybil-based attacker with realistic assumptions for both MCS systems and real-world scenarios.

As registration is a prerequisite for contributing MCS-ed data, the attacker tries to register multiple accounts to exert maximum influence. These accounts do not come from real physical devices, as they are neither effective to manage nor cheap to produce in large numbers. Instead, attackers can automate the process of Sybil generation, e.g., with mobile device emulator software or scripts to

mimic client behavior. We assume fake Sybils are orchestrated by at least one legitimate user, termed the master Sybil (node), who is the motivated party behind the attack and the eventual controller of the infiltrating Sybils.

To thwart such adversaries, we propose a crowdsensing-based user verification scheme to validate contributing users' presence on-site and, in the meantime, limit incoming Sybil accounts to the system. In essence, each new contributor's presence can be verified as a genuine device by other similar devices nearby. One can employ the mobile MCS user base, already present on-site, for this (newly joining) user presence detection simply by utilizing smartphones with mainstream sensing capabilities. Users can *sense* other participants in their proximity by broadcasting/listening to Bluetooth low-energy (BLE) signals and measuring Received Signal Strength Indicator (RSSI) values over multiple rounds, thereby vouching for their relative presence during the protocol session. When the system has enough information about the relative presence of the users, it can decide whether they are Sybil or genuine nodes. Finally, potential collaborators of the labeled Sybils are identified, and their influence is removed. Our protocol significantly raises the bar for adversaries while maintaining a seamless experience for benign users. Furthermore, it only introduces mild user exposure, only confirming the participation of a node pseudonym.

In the remainder of the paper, Section 2 describes the RSSI-based method we utilize in this work. Section 3 describes the system and adversary models. Section 4 defines the presence verification framework. The results and evaluations are presented in Section 5, verifying empirically the effectiveness of the proposed protocol. Finally, Section 6 concludes with future research directions.

2 BACKGROUND

Received Signal Strength Indicator (RSSI) measures the power level received from a transmitted radio signal. RSSI depends mainly on the distance from the transmitter, physical obstacles, interference from other devices, and the transmitter power output. We can represent a signal value as (linear form) $\text{RSSI} = P_{\text{transmit}} \cdot K/d^{\alpha}$, where P_{transmit} is the transmitted power, d is the transmitter-receiver distance, α represents the path-loss exponent (depends on the environment e.g., open space, crowded environment,), and K is a system-specific constant capturing antenna gains and interference losses. Since the constants α and K depend on the environment and the hardware used, they usually do not change over short periods (e.g., 10 minutes). As nearby users are typically in the same environment, α will be the same for all of them - unless there are significant differences in the local environment (e.g., one user is behind a wall), causing α to vary slightly.

We use an RSSI ratio-based method [3] as a black box to detect nodes broadcasting signals from the same physical position. Listeners can determine whether two devices are broadcasting from the same location using their ratio of RSSI values because this ratio depends solely on the node distances. We illustrate this in Figure 1. L_1, L_2 are two listening nodes. B_1 and B_2 are two Sybil nodes broadcasting signals from the same position, and B_3 is a benign broadcaster. RSSI values recorded by L_1 and L_2 for nodes B_1/B_2 will be $m_1 = P_1 \cdot K/d_1^{\alpha}$ and $m_2 = P_1 \cdot K/d_3^{\alpha}$, respectively. The ratio for B_1 can be calculated as $\frac{m_1}{m_2} = \frac{P_1 \cdot K}{d_1^{\alpha}} / \frac{P_1 \cdot K}{d_3^{\alpha}} = (\frac{d_3}{d_1})^{\alpha}$. Since

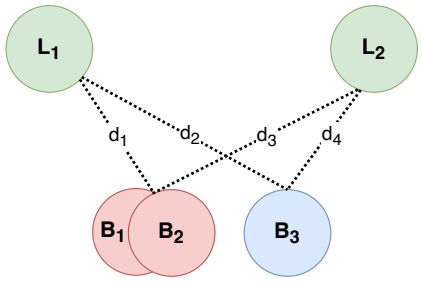

Figure 1: Example setting for RSSI ratio-based detection, with two listening nodes L_1, L_2 and three broadcasting nodes B_1, B_2, and B_3.

B_2 broadcasts from the same location, its ratio calculated by the same set of listeners is the same, i.e., $(\frac{d_3}{d_1})^{\alpha}$. For a user in a different location, B_3, the same listeners compute the different ratio $(\frac{d_4}{d_2})^{\alpha}$. In this scenario, listeners can reliably infer that B_1 and B_2 broadcast from the same location and B_3 broadcasts from a different location without knowing the actual locations.

3 SYSTEM AND ADVERSARY MODEL

We choose Bluetooth as a means to verify the presence of the users via measuring RSSI values. Bluetooth has an effective range of up to 100 meters, typically achievable in environments with no interference. The practical range can decrease to 20-30 meters indoors due to obstacles. However, as MCS deals with sensing, we assume the protocol is executed outdoors, notably in urban areas, with a practical Bluetooth range of 75-100 meters due to fewer obstructions [15]. Furthermore, Bluetooth has mechanisms (e.g., random advertisement intervals and delays) to minimize collisions with many devices broadcasting at the same time and can reliably handle up to 200 devices [12]. Our protocol is not limited to Bluetooth; it can utilize any signal that allows for the calculation of RSSI values.

The system divides the sensing task area into smaller sub-regions where users should be in sensor proximity to each other within their sub-regions. This division can be defined in multiple ways depending on the system needs. For simplicity, we assume discrete boundaries for sub-regions predefined by the system and published beforehand, allowing users to easily identify the sub-regions they reside in. Registration is conducted within each sub-region. If there are large numbers of users in a sub-region, the system breaks them apart randomly into different protocol sessions. This has both performance and security benefits that will become clear below.

3.1 System Entities

Users: Individuals with mobile sensing devices contribute data for various MCS tasks. They engage with the system by registering, enrolling in tasks, and submitting data. They have mainstream wireless transceivers (e.g., Bluetooth or WiFi), which, in the context of this work, are used essentially as 'sensors' to verify other users' presence. The presence verification protocol is run when there is a sufficient number of users, both newly arriving and already registered, in a sub-region; the former needs to have their presence

verified, and the latter assists future fellow users in the proximity to be verified.

Infrastructure: MCS system entities and components, including software, tools, and resources, that facilitate user registration, task enrollment, data collection, and user remuneration. We assume a generic MCS security and privacy infrastructure [6, 8, 11, 14] to deploy our system. Infrastructure acts as the mediator between task initiators (TIs), data consumers, and users, data producers. The coordinator of the presence verification protocol is part of the registration manager (RM); for simplicity, here, we discuss the RM and not the rest of the MCS infrastructure.

Registration Manager (RM): It oversees the registration of new users to the MCS infrastructure. More importantly, in the context of this paper, the RM coordinates the presence verification protocol, issuing temporary credentials for users and collecting RSSI measurements to distinguish genuine users from Sybils. As RM monitors the newly arriving users, it knows if a region has enough to initiate the protocol based on the system status.

3.2 Adversary Model

Legitimate nodes (genuine devices) are honest but curious; they adhere to the protocol while seeking additional information about others. In contrast, Sybil nodes are malicious and may disregard the protocol specification to maximize their infiltration chances. We focus on adversaries that want to register with their physical devices and a number of emulated devices as users. We consider Sybils and their orchestrator, the master Sybil, separately. Sybils are the emulated devices in control of the attacker, they are virtualized in a computer/server, i.e., not physically in the protocol region. The master Sybil is physically present, and it broadcasts/listens to signals on behalf of emulated Sybils. The stronger the attacker, the larger the number of physically present devices they have as different 'access points' to broadcast signals. Master Sybils on-site, with genuine users in proximity, in principle, should be able to register, given they are not colluding with a Sybil device (and thus not detected as such).

We do not consider jamming as this would affect everyone on-site, and detection of jammers is out of the scope of this work. We assume that the nodes do not change their transmission power during a broadcasting session. Broadcasting consecutive signals with varying power is not permitted by the Bluetooth specification: the adjustment of the power level applies to all device transmissions universally because BLE firmware optimizes for power consumption and communication efficiency, not for complex operations. One could change the transmission power back and forth to create an illusion of simultaneous submissions with different levels. This would require modification on both Bluetooth firmware and potentially hardware, which is not trivial for standard consumer-grade BLE devices. We discuss attacks next more specifically, notably including aspects of the proposed scheme (to be presented in Sec. 4) and thus solution-specific adversarial behavior.

Forgery Attacks: The master Sybil broadcasts on behalf of the Sybils, making benign nearby users wrongly perceive this as a physically present device. The master Sybil also acts as a listener for all its Sybils, measuring and uploading RSSI values for them. There is no limit to the number of Sybil devices an attacker could impersonate, unlike assumptions in [17].

Replay Attacks: The adversary receives signals from other nodes and simply re-broadcasts them. This can lead listeners to detect multiple signals with the same identity originating from what appear to be different sources. The user should drop the late-arriving signal[1], but the attacker hopes its signal arrives earlier or is the only one arriving due to radio environment factors, i.e., interference. We do not consider relays at the physical layer, e.g., a setting with antennas and amplifiers, as such an adversary is highly sophisticated (delays in the order of nanoseconds), expensive, and hard to detect.

Falsification Attacks: Sybil nodes report and upload bogus RSSI values instead of the real measured values to mislead the system. For example, a master Sybil node might upload identical RSSI values for all genuine broadcasters, making those users appear to be in the same location, thus increasing the chances that the server mistakenly misclassifies them as Sybils. Likewise, the master Sybil can submit random RSSI values for its Sybils, making them appear in distinct places. In other words, the adversary can try to frame genuine users as Sybil while exonerating Sybils and having them accepted as genuine users.

4 PRESENCE VERIFICATION AND SYBIL DETECTION

We outline the protocol, aligned with the illustration in Figure 2:

Step 0 **Initialization:** Users begin with an initial registration to the MCS authority (notably, the RM), notifying about their sub-region, sending each their certificate signing request (CSR), obtaining each, in response, a temporary short-term credential (pseudonym) from the RM. Each user is assigned a pseudonymous user ID label only to be used for the registration session. The pseudonym uniquely associates the user with the sub-region, and the user label for other participants can authenticate participating users on-site. Users need to repeat this step if they change their sub-region before being admitted as registered users.

Step 1 **Coordination Step:** The RM assesses the overall system status and counts the number of registered users in sub-regions. When a sub-region has N_{thr} nodes, it can then start a protocol session; the optimal (parameter) depends on the adversarial presence. The RM determines session-specific details, the required number of rounds, $T = 2 * log(N)$, where N is the number of nodes. It also sets round start and end times, $t_s^{(i)}$ and $t_e^{(i)}$, for nodes to synchronize their broadcast/listen times. Users are randomly assigned as broadcasters (blue nodes) and listeners (green nodes), ensuring that each user receives Bluetooth signals from other users during the session. Protocol settings, times, and the list of participating nodes are then communicated to all users before the presence exchange step starts.

Step 2 **Presence Exchange Step:** Users start each round simultaneously at the specified time, as communicated by the RM.

[1]We assume the direct path between two users adheres to the Triangle Inequality Theorem, where the distance from A to B to C is always greater than or equal to the direct distance from A to C. Even if B is between A and C, by the time a Sybil node (B) records and re-transmits the signal, the original signal would arrive at the destination.

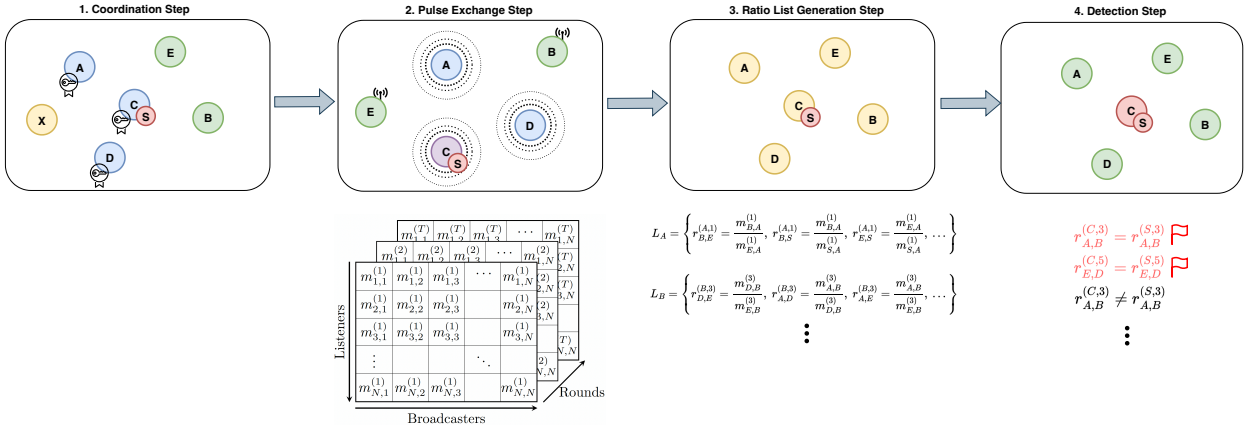

Figure 2: System Procedure. C **is the master Sybil, impersonating a set of Sybil devices** S**.**

Half of the users broadcast signals while the other half listen. After each round, users upload the recorded RSSI values for each broadcaster they sensed and wait for the next round. The RM inserts the RSSI values from each round progressively into the presence table. We define the RSSI presence table M as an $N \times N \times T$ matrix where each element $m_{i,j}^{(t)}$ represents the RSSI value that the listener node i measured from the broadcaster node j on round t. In other words, rows and columns correspond to listeners and broadcasters, respectively, whereas the time index represents the round number presence detected. Note that $m_{i,j}^{(t)}$ is not necessarily equal to $m_{j,i}^{(t)}$ or $m_{i,j}^{(t')}$ due to user mobility, varying distances, and interference. However, this is not an issue because individual RSSI values are not used directly for analysis but rather for generating the ratio lists.

Step 3 **Ratio List Generation Step:** Once the RM collects all RSSI measurements, it creates a list of ratios for each user, which is simply the ratio of every pair of RSSI recorded for that user at a given round. These ratios enable the aforementioned RSSI ratio-based detection method (Sec. 2) used to identify and group Sybil devices: devices broadcasting from the same location have similar ratio values, whereas benign broadcasts have varying ratio values. The RM generates a ratio list, L_i, for each node i in the session, using the presence table values recorded for the node. Specifically, for each node $i \in U$, the ratio list is constructed by computing the ratios of the RSSI values $m_{j,i}^{(t)}$ and $m_{k,i}^{(t)}$ recorded by other users, i.e., $j, k \neq i$ and $j \neq k$. In order to correctly find the Sybil attackers, we need ratios from the same round so the listeners receive the same signal[2]. Formally, we define the lists as follows:

$$L_i = \left\{ r_{j,k}^{(i,t)} = \frac{m_{j,i}^{(t)}}{m_{k,i}^{(t)}} \mid j,k \in U, j \neq k, j \neq i, k \neq i, \forall t \right\}$$

[2]Different rounds might have different α due to the movement of the nodes, although the difference should be marginal.

Step 4 **Elimination Step:** The system flags the identical ratios of different broadcasters, indicating they broadcast from the same location. The ratios computed for different broadcasters by a pair of listeners should not be similar, as this would mean those nodes are broadcasting from the same location [3]. Formally, the system *flags* (penalizes) all users i, i' such that $\left| r_{j,k}^{(i,t)} - r_{j,k}^{(i',t)} \right| \leq \epsilon$ and i, i', j, k are all distinct users. For every such pair, the system punishes, by adding a penalty score, pen, for each of the users' i and i'. After calculating all penalty points, users exceeding the eviction threshold, th, are deemed Sybil. As classified Sybils are expected to be malicious, the evidence they provide should be further examined. We propose an auditing process where the contributions of evicted users during the presence verification are re-evaluated, as their submitted RSSI readings might be bogus or indicate collusion, i.e., falsification attacks. For example, users whose presence is *not* reported by the attackers are exonerated by removing the flag (implying a benign user), while those detected are penalized (implying a collaborator). The system removes flags from ratios where one of the listeners is the evicted user to counter the Sybils' malicious behavior. Furthermore, it flags the ratios where the presence is detected by the evicted Sybils. We show the effectiveness of both exoneration and penalty mechanisms in our experiments. After the auditing, the protocol halts if there are no users above the threshold. Otherwise, auditing continues, eliminating nodes above th and removing their potential malicious effect from the protocol.

4.1 Analysis

Threshold Selection: We heuristically set the elimination threshold value to $C_2^{N/4} \times N_{rounds}$, for honest nodes to be safe even in the worst-case scenarios, i.e., 50% of the nodes are Sybils. In each round, on average, half of the Sybil nodes are chosen as listeners, making the expected number of Sybils 25% of the whole user set selected as listeners. If these Sybil nodes conduct falsification attacks (wrongly giving evidence users broadcast from the same location) against

honest nodes, they are expected to be flagged by combinations of Sybil listeners, i.e., $C_2^{N/4}$. We reach the final threshold formula by multiplying this by the total number of rounds.

Auditing Process: This is a crucial part of the elimination phase because it focuses on inherent characteristics of the Sybil attacks, specifically cooperative adversarial effort on a target. When the flagged ratios exceed the threshold value, there is sufficient evidence from the majority of listener pairs over multiple rounds. We know that penalty points are assigned to a node if another node is broadcasting from the same location, indicating at least another node colluding. It is plausible to expect a falsification attack originating from those nodes; with adequate evidence from other nodes, the system should remove these nodes' effects on the system.

The first step removes flags involving the Sybil node, basically annulling any flag raised by the evidence provided by those nodes. Secondly, the system examines if such Sybil nodes consistently punish other users, as they can collude/be instructed as such by the master Sybil. To counter this, the system adds penalty points for nodes that could have been flagged by the pair of Sybils but were not. In other words, the system adds points to broadcasters who were spared from attackers' penalty points.

Disconnection and signal loss: There can be signal loss during the presence exchange phase, making the recorded RSSI value $m_{i,j}^{(t)}$ equal to Null. Ratios involving Null values are automatically given a penalty point, as not broadcasting is equivalent to not being present on-site. If a user disconnects from the protocol or, for some reason, does not broadcast any signals, it is punished constantly, resulting in eventual disqualification and eviction. Assuming there are still sufficiently many users in the region, such events do not disrupt the protocol because nodes change their roles in each round.

Limited user count and randomization: The system refrains from starting a protocol instance when the number of users is low. To reliably assess user status, the system needs sufficient evidence before determining whether a user is a Sybil. For example, if the protocol starts with at least 10 users, an attacker could register 5-7 users in a sub-region and wait for other users to join, easily outnumbering them and having their Sybils registered. Similarly, a sudden influx of users in a sub-region may indicate malicious activity. Although the terms *sudden* and *a lot of* users are vague, most systems already have information about their registration rates and user density, allowing them to make ad-hoc decisions per sub-region.

When enough users are present in a sub-region, the system should divide them into different sessions to possibly reduce the number of Sybils in each session. This is particularly beneficial if a master Sybil controls/represents a large set of Sybil nodes, basically diffusing adversarial nodes into different sessions. From a large pool of candidates and existing nodes known only to the RM, the server selects multiple subsets of such nodes. It then executes multiple instances of the protocol, increasing the likelihood of having subsets with honest majorities that overpower the Sybils. Already verified existing users is included in the protocol in a randomized manner to decrease the newly arriving user rate in a session. Although rare, a downside of the randomization is that some Sybils can be isolated, allowing master Sybil to register those accounts, e.g., the Sybil node ends up in a session alone, preventing genuine users from linking it with other Sybils.

Privacy: The proposed method does not collect any additional private information, such as identity, location, or other mobility data. Users must obtain pseudonyms from the RM, thus not using other credentials that could convey any identifying information during the protocol session. Each pseudonym[3] is used only once per the protocol instance run; if participants execute the protocol again, they use a different pseudonym and the corresponding private key. Although highly unlikely, even if two users end up in another protocol instance, their IDs would be different.

Users are required to inform about their sub-region in the initialization step without specifying an exact location. Given users do not engage with the RM after the procedure, this is the only mild privacy exposure. For existing, already verified users in the MCS system, the data aggregator simply communicates to the RM a random subset of recently reported pseudonyms from the region - information already disclosed to the MCS infrastructure. Although RSSI values could be used to estimate the distance to individual broadcasters, they do not allow precise localization. For an attacker to localize broadcasters, they would need to employ more sophisticated techniques, such as deploying multiple devices for a triangulation attack or preparing and using a database of pre-recorded RSSI values for indoor fingerprinting methods [2].

5 EVALUATION

We conduct extensive simulations of the protocol using Python to evaluate its efficiency and effectiveness. We simulate an MCS sub-region, a 70-meter diameter circle, with up to 100 users[4]. The attacker nodes in the simulations follow the malicious behavior outlined in Sec. 3. We set a small success rate, 5%, for replay attacks; they are hard to mount in practical scenarios as the replayed signal will always have latency compared to the original signal. We repeat experiments at least 100 times to obtain reliable results. We further limit the falsification attacks rate to 75% to regulate the aggressiveness of the attacker. The attacker forgoes falsification attacks for some rounds, hoping that disassociates the master for Sybil nodes.

We start by observing the total penalty score difference between the Sybils and legitimate nodes under varying ratios of Sybils in the system. Fig. 3 shows the average penalty points (scores) of legitimate and Sybil nodes in a 30-node system with and without an auditing process. The first observation is that the average penalty of genuine nodes is significantly lower, even when 45% of the nodes are Sybils. However, as the Sybil rate increases, the benign nodes score also increases as Sybil nodes frame honest users. This is where auditing comes to enhance the Sybil detection, illustrated in opaque colors. Auditing always decreases the total genuine user scores while increasing the Sybil scores. We also show the eviction rates before and after the audit in Fig. 4. Without auditing, the system fails to capture all Sybil nodes.

We investigate the optimal number of users in a session, considering three adversarial scenarios with varying rates of Sybils in the

[3]Note: not to be confused with pseudonyms obtained after the registration is completed, e.g., in [6–8])

[4]Such a circle can support more, e.g., 1924 people, if everyone takes 2 square meters of space, it is very hard to have that many people present in an MCS system sub-region at a given time.

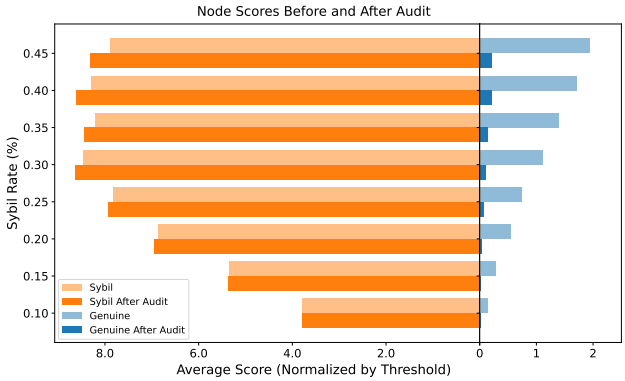

Figure 3: Average scores of nodes before and after the audit.

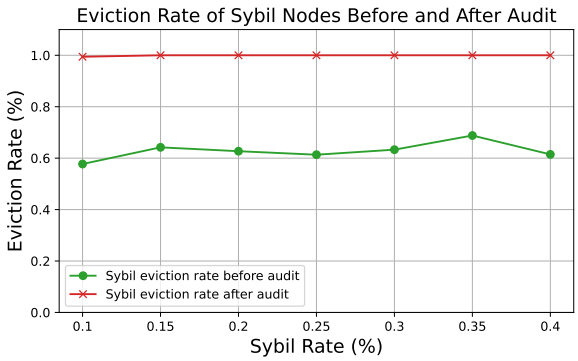

Figure 4: Eviction rates with and without the auditing process.

system, 45%, 30%, and 15%, respectively. Fig. 5 shows the average scores of genuine and Sybil nodes before the audit, with score values scaled to their respective thresholds since th depends on the user number of the session. We can see user counts do not affect the performance of the protocol for all scenarios, as all Sybil nodes are evicted from the system. However, a low Sybil rate coupled with low user counts makes Sybils harder to detect, as can be seen in the initial average low penalty points of the blue line, because the Sybil node's roles are more likely to be separated when the user set is small. In those cases, Sybil nodes do not broadcast simultaneously, which makes genuine listeners miss them. When there are enough users in the system, this problem disappears as with presence of enough users enables stable scores for Sybils.

We further experiment with falsification attacks to evaluate attacker capabilities. As the audit process focuses on potential Sybils conducting falsification attacks, it may be beneficial to limit this behavior. We again set the Sybil rate in the protocol to 45%, 30%, and 15% and have 50 users in the protocol, illustrated in Fig. 6. We see that falsification rates do not impact the average Sybil penalty points, and the node scores are well above the threshold because Sybil node scores mainly come from benign nodes.

6 CONCLUSION

The openness of MCS is a double-edged sword, and Sybil-based attacks can severely degrade the data quality. To mitigate them, we

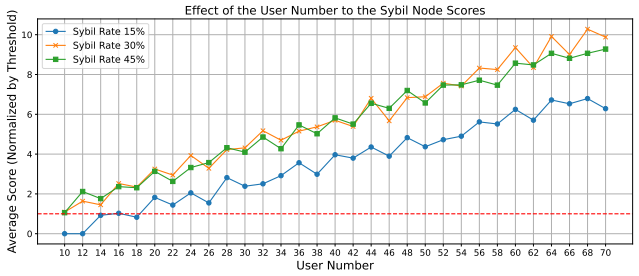

Figure 5: Average Sybil node score with varying total user numbers and rates of Sybils.

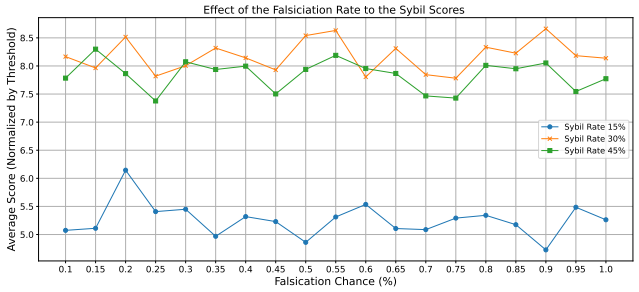

Figure 6: Effect of the falsification attack rates to average Sybil node scores with varying Sybil rates.

designed a distributed presence verification protocol. Devices that physically exist in a certain area receive low scores, while Sybil devices receive relatively high scores based on on-site collaborative evidence using Bluetooth signals. In different adversarial scenarios and deployment settings, our method is effective, and it can complement existing hardened registration processes [1, 10].

ACKNOWLEDGMENTS

We thank Ronghua Li for his contributions while completing his MSc thesis. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Jatesada Borsub and Panos Papadimitratos. 2018. Hardened Registration Process for Participatory Sensing. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 281–282.
[2] A. Chatzimichail, A. Tsanousa, G. Meditskos, S. Vrochidis, and I. Kompatsiaris. 2021. RSSI fingerprinting techniques for indoor localization datasets. In *Internet of Things, Infrastructures and Mobile Applications: 13th IMCL Conference*.
[3] M. Demirbas and Y. Song. 2006. An RSSI-based scheme for sybil attack detection in wireless sensor networks. In *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*.
[4] J. Douceur. 2002. The sybil attack. In *International workshop on peer-to-peer systems*. Springer.
[5] C. Eryonucu and P. Papadimitratos. 2022. Sybil-Based Attacks on Google Maps or How to Forge the Image of City Life. In *15th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (San Antonio, TX, USA). 12 pages.
[6] C. Eryonucu and P. Papadimitratos. 2023. Security and Privacy for Mobile Crowdsensing: Improving User Relevance and Privacy. In *European Symposium on Research in Computer Security (ESORICS)*.
[7] Stylianos Gisdakis, Thanassis Giannetsos, and Panos Papadimitratos. 2014. SPPEAR: Security and Privacy-preserving Architecture for Participatory-sensing

Applications. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (ACM WiSec)*.

[8] S. Gisdakis, T. Giannetsos, and P. Papadimitratos. 2016. Security, Privacy, and Incentive Provision for Mobile Crowd Sensing Systems. *IEEE Internet of Things Journal* (Oct. 2016).

[9] M. Jamshidi, E. Zangeneh, M. Esnaashari, and M. R. Meybodi. 2017. A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks. *Computers & Electrical Engineering* (Nov. 2017).

[10] Ronghua Li. 2022. Advanced Hardened Registration Process for Mobile Crowd Sensing. https://www.diva-portal.org/smash/get/diva2:1668730/FULLTEXT01.pdf

[11] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen. 2020. Enabling Strong Privacy Preservation and Accurate Task Allocation for Mobile Crowdsensing. *IEEE Transactions on Mobile Computing* (June 2020).

[12] M. Nikodem and M. Bawiec. 2019. Experimental evaluation of advertisement-based bluetooth low energy communication. *Sensors, MDPI* (2019).

[13] P. Sarigiannidis, E. Karapistoli, and A. A. Economides. 2015. Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information. *Expert Systems with Applications* (Nov. 2015).

[14] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. 2011. AnonySense: A system for anonymous opportunistic sensing. *Pervasive and Mobile Computing* (2011).

[15] Bluetooth SIG. 2024. Range. https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/. Accessed: 2024-06-22.

[16] A. Tangpong, G. Kesidis, H. Hsu, and A. Hurson. 2009. Robust Sybil Detection for MANETs. In *International Conference on Computer Communications and Networks*. IEEE.

[17] N. Verchok and A. Orailoğlu. 2020. Hunting Sybils in Participatory Mobile Consensus-Based Networks. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS)*.

[18] I. Šuran. [n.d.]. Fake road closure - Google Maps Community. https://support.google.com/maps/thread/137301768/fake-road-closure?hl=en [Online; accessed 11-February-202].