01 Jan 2023

Unifying Threats Against Information Integrity In Participatory Crowd Sensing

Shameek Bhattacharjee

Sajal K. Das
Missouri University of Science and Technology, sdas@mst.edu

Recommended Citation

S. Bhattacharjee and S. K. Das, "Unifying Threats Against Information Integrity In Participatory Crowd Sensing," *IEEE Pervasive Computing*, Institute of Electrical and Electronics Engineers; Computer Society, Jan 2023.
The definitive version is available at https://doi.org/10.1109/MPRV.2023.3296271

FEATURE ARTICLE

Unifying Threats Against Information Integrity in Participatory Crowd Sensing

Shameek Bhattacharjee, *Western Michigan University, Kalamazoo, MI, 49008, USA*

Sajal K. Das, *Missouri University of Science and Technology, Rolla, MO, 65409, USA*

This article proposes a unified threat landscape for participatory crowd sensing (P-CS) systems. Specifically, it focuses on attacks from organized malicious actors that may use the knowledge of P-CS platform's operations and exploit algorithmic weaknesses in AI-based methods of event trust, user reputation, decision-making, or recommendation models deployed to preserve information integrity in P-CS. We emphasize on intent driven malicious behaviors by advanced adversaries and how attacks are crafted to achieve those attack impacts. Three directions of the threat model are introduced, such as attack goals, types, and strategies. We expand on how various strategies are linked with different attack types and goals, underscoring formal definition, their relevance, and impact on the P-CS platform.

With the growing penetration of smart hand-held devices and smartphone apps, various forms of crowd sensing (CS) applications have emerged. In CS applications, human users are involved in providing reports or sensed data that improve civic well-being via pervasive smart services. The goal of the CS application is to identify the correct event based on the reports/data and disburse incentives to those users helping in event identification. The incentive disbursement is critical in keeping the churn under control in such commercial applications.

TYPES OF CS PLATFORMS

The CS paradigm is classified into two subdomains—Opportunistic and Participatory—as described below.

Opportunistic or Passive CS (O-CS): In O-CS, users agree to the usage of their personal devices as a sensor. The O-CS app submits data automatically "without" explicit human involvement. In this scenario, the report is an analog signal and thus similar to sensor networks. Therefore, many research works involving O-CS setting borrow methods from statistics (e.g., maximum likelihood estimates) and statistical machine learning (e.g., expectation maximization algorithms) for computing truthful aggregate value of a sensed quantity for situational event inference. Each participant's report is compared with the output of truth discovery to assign and update users' long-term reputation score.

Participatory CS (P-CS): The P-CS subdomain, in contrast, requires explicit human involvement, where some users (called "reporters") manually contributes observations in the form of reports, or any piece of information that is not an analog signal. In such scenario, the approaches used in O-CS for finding truthfulness of events or assigning user reputation do not always apply. The "Participatory sensing" is analogous to Social Media (where the users offer voluntary posts on public groups and pages); hence many works use the broader term of *social sensing*. Nonetheless, the following differences exist with pure social media: 1) a dedicated crowd reporting app (e.g., Google's Waze App,[a] Yelp) is used instead of a social media app, and 2) one usually cannot share/forward other's reports but can only provide a feedback/reaction. Thus, lessons learnt from P-CS vulnerabilities can partially help systematize social sensing vulnerabilities as well.

1536-1268 © 2023 IEEE
Digital Object Identifier 10.1109/MPRV.2023.3296271

[a] [Online]. Available: www.waze.com

INFORMATION INTEGRITY CHALLENGES IN P-CS

While the incentives attached to the contribution of reports encourage participation, it also motivates rogue reports from selfish users. Furthermore, orchestrated false reports may cause incorrect events to be published in P-CS, thus having civilian and economic impacts, which motivates organized malicious adversaries. Nonetheless, one critical challenge in CS applications is event trustworthiness or truthfulness. Furthermore, determining which participants are honest or dishonest via a reputation scoring model is another typical challenge. In the literature, artificial intelligence enabled computational trust and reputation models have been proposed to solve both challenges. However, these models have weaknesses in the design principles and P-CS operation design loopholes, which keep the door open for organized malicious intent to harm the P-CS platform's integrity.

WHY A FORMAL P-CS THREAT LANDSCAPE?

In the O-CS domain, the threat model is similar to those in cyber-physical systems and sensor networks, and does not require much leap of faith. Hence, we do not discuss O-CS in this work. However, our analysis of existing security literature in the P-CS domain revealed a lack of unified discussion on strong and elaborate threat models specific to P-CS. Those threats arise from the complex cyber-physical-human couplings, and design weaknesses in trust, reputation, and decision-making models in P-CS. Thus, an important motivation of this article is to consolidate various possible targeted threats, specifically relevant to P-CS. We aim to provide a guide for future designers wishing to build secure and robust-by-design P-CS platforms.

SCOPE OF THREAT MODEL

There exists a lot of research in securing the P-CS domain that deals with traditional well-known attacks common to any networked system, such as Sybil attacks, privacy attacks, unauthorized access, etc. Our goal is to add and formalize a targeted threat model of information integrity specific to P-CS. Therefore, we do not discuss commonly reported threats that do not directly relate to algorithmic weaknesses of trust, reputation scoring, decision models, or procedural loopholes in P-CS operations. Additionally, our threat model focuses on attacks that originate from organized malicious intent rather than individual selfish intent.

ARTICLE CONTRIBUTIONS

Our novel contributions are as follows:

- We propose a threat landscape spanning three main directions: 1) attack goals, 2) attack types, and 3) attack strategies. To achieve an attack goal, the attacker may need one or more attack types that depend on the stage of P-CS information integrity being targeted. Furthermore, depending on the intended impact and the adversary's level of prior knowledge, the attack types can be launched using one or more attack strategies that belong to a certain attack type to attain an attack goal.
- To specify the attacker's intent, we propose five possible attack goals: 1) induce false events, 2) suppress true events, 3) alter event types, 4) poison user reputation model, and 5) steal the event publishing model.
- We propose three attack types: 1) sensory manipulation targeting weakness in the reporting stage, 2) feedback weaponizing attack strategies targeting weakness in the rating feedback stage, and 3) belief manipulation attacks targeting weakness in the decision-making phase.
- For each attack type, we propose multiple attack strategies, their relevance, and impact.
- We highlight how our proposed strategies are linked to different goals and what types of attack strategies require more research.

UNDERSTANDING THE INFORMATION INTEGRITY PIPELINE IN P-CS

This section describes typical design stages of P-CS platform and gives examples of CS Apps. such as Waze and Yelp to illustrate how the design features are seen in real-life apps. This will enable readers to relate to the threat landscape that has a more generic treatment.

Stages of P-CS Operation

As explained below, a typical P-CS platform consists of three operational stages—Reporting or contributions, decision-making, and feedback monitoring.

Reporting: This stage involves voluntary contributions from the crowd indicating a particular event or response to task.

Decision-Making: The reports are collected by a P-CS server and a recommendation is made based on an event decision-making model running on the P-CS

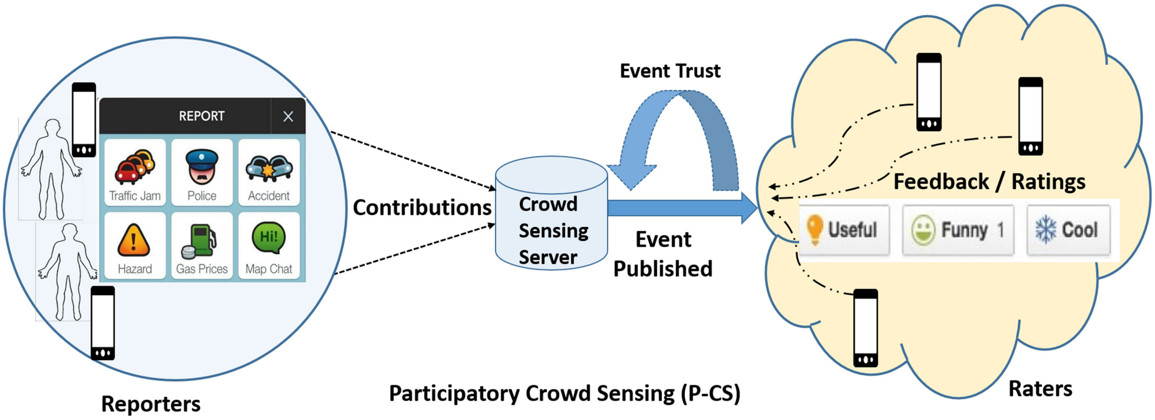

FIGURE 1. Abstraction of participatory crowd sensing (P-CS).

server that decides how to process various reports into a publishing a recommendation or event.

Feedback Monitoring: The published events can be rated based on the perceived usefulness (e.g., yes/no) by other users, called raters, with respect to that event.

Real-Life Example of P-CS: Figure 1 illustrates an abstraction of a P-CS application for vehicular event CS, such as Google Waze app, where the reporters submit location tagged “reports” by clicking one out of the following events—road closure, jam, accident, weather hazard, police presence, gas station pricing, etc. The P-CS server decides whether and how long to publish this event on the Waze app; there is also an option for consumers to rate the perceived usefulness of the events published. A similar abstraction exists in social sensing apps, such as Yelp,[b] where reports are submitted in the form of a review on a business. Each report is visible separately on the app that can be rated by other users. For example, Yelp allows three feedbacks to each post/comment while Waze allows two feedbacks. The reports and feedbacks are combined to form an opinion on the business, and Yelp sorts them to recommend a business.

Different User Roles

The users in a CS paradigm can be classified into various roles, such as reporters, raters, and passive consumers. From the perspective of an event or entity which needs reporting, the users that contribute information on that event are *reporters* with respect to that event (or entity). A subset of the remaining user base, known as raters, can give feedback on the usefulness of the published event. The user base which neither reports nor rates a given event is a passive consumer with respect to that event.

Across different events, however, a user of a P-CS app can act as a reporter, rater, or passive consumer based on their roles with respect to that event. It is assumed that the system does not allow the same user to rate its own report. If the attacker recruits a user or hacks apps to work for his attack goals, then a malicious user can perform all three roles with respect to an event in P-CS.

Unified View of Information Integrity Pipeline

Regardless of the actual application, the architecture of assuring information integrity has the following overarching abstraction.

Upon launching a new P-CS, the initial stages are known as the cold start phase, where the user reputations are not known. Usually, in the cold start phase, the events are published by the decision-making model based on the contextual correlations among reports (e.g., event type, time, location, threshold number of reports) in an area.[12]

In many practical systems as well as novel research,[2] the P-CS implements a mechanism known as *feedback monitoring* that asks the crowd to rate or give a feedback on their perception of how truthful an event is. The data acquired as part of the feedback monitoring are used to verify, in retrospect, the event’s truthfulness or trustworthiness. The event’s veracity is indicative of the honesty levels (reputation) of ’those users who submitted the reports corresponding to this event. Intuitively, if the event truthfulness is high, the reputation of users reporting highly truthful event, gets their reputation increased, and vice versa.

[b] [Online]. Available: yelp.com

Once a reliable user reputation base is established, the P-CS enters the steady-state phase of operation. In this phase, the decision-making model takes into account three major factors: 1) prior reputation of users submitting a report; 2) contextual probability of that event occurring, and 3) contextual correlations and quantities deciding whether or not to publish an event. For in-depth discussions on this unified view, refer to Restuccia et al.[5] and Bhattacharjee et al.[2] Note that in the steady-state phase, the P-CS still keeps the feedback/rating mechanism since new users join and old users may become inactive.

Models for Information Integrity in P-CS

Following the aforementioned pipeline, there exist four types of modeling tools aiming to preserve information integrity in P-CS. A detailed survey of these models can be found in Restuccia et al.[5]

1) *Event Truthfulness Models:* These are AI-based models that use feedbacks received against each event, and assign an event truthfulness score. Depending on how many options are available in the rating mechanism, well-known methods include Beta Reputation Model,[8] Josang's Belief Model,[7] and our recently proposed QnQ model,[2] which improves upon these models.
2) *User Reputation Models:* These AI-based models assign an aggregate reputation to the users (entities) based on a history of interactions that use the perceived truthfulness per event contributed by that user. Popular methods include variations of Dempster–Shafer Belief,[11] Dirichlet reputation systems,[9] and the recently proposed QnQ model[2] improving upon these models.
3) *Aggregation-Based Decision-Making:* These models use weighted prior reputation aggregation,[2,13] prior knowledge of event occurrences,[2] and optionally current similarities in reports; and then combine them to calculate a trust level of an event and then decide whether or not to publish the event. This is only feasible in the steady-state phase.
4) *Similarity-Based Decision-Making:* These models use only contextual similarities and correlation in the reports to decide whether to publish an event or not.[12]

THREAT LANDSCAPE OVERVIEW

The threat landscape consists of key features characterizing various aspects of the P-CS threat model. The features include attacker intent, goals, types, and strategies.

Categories of Attacker Intent

The following types of intent can undermine the information integrity of P-CS platforms.

- *Honest Errors*—These users report and rate honestly, although there may be occasional errors in their reporting.
- *Malicious*—These users provide misleading reports whenever they choose to report or rate, because their only gain is to inflict maximum operational damage.
- *Selfish*—These users provide false report or rating but only when there is an individual benefit (e.g., incentives) in return for their dishonest act. This threat model is put forward by considering malicious intent.

Categories of Attack Goals

For an active adversary model, the attack impacts determine the attack types. Broadly, we identify three types of adversarial goals motivated by their potential (civilian or economic) impacts:

Induce False Event

The objective of the adversary is to induce the CS server into believing the existence of a "false event" or "false information," when in reality no such event had occurred. As a result, the CS will be tempted to take "unwarranted actions" that negatively affect the operations and having an immediate civilian impact. For example, inducing a false traffic congestion event at a key intersection, may create traffic blocks in other parts of a city due to multiple undue traffic reroutes. Finally, when combined with certain feedback manipulation attacks, such a goal can trigger undue incentive disbursement to malicious users instead of honest reporters.

Suppress True Event

The adversary's objective is to make the CS server "fail in the timely discovery of a true event or information," by giving an impression of the absence of an event. As an impact, the CS will "refrain from taking a legitimate intervening action" that lead to negative civilian consequences, creating an impact on the CS platform's passive consumers. Furthermore, the incentive mechanism will not reward the participants who reported truthfully, which will thus discourage honest reporters to participate in the future.

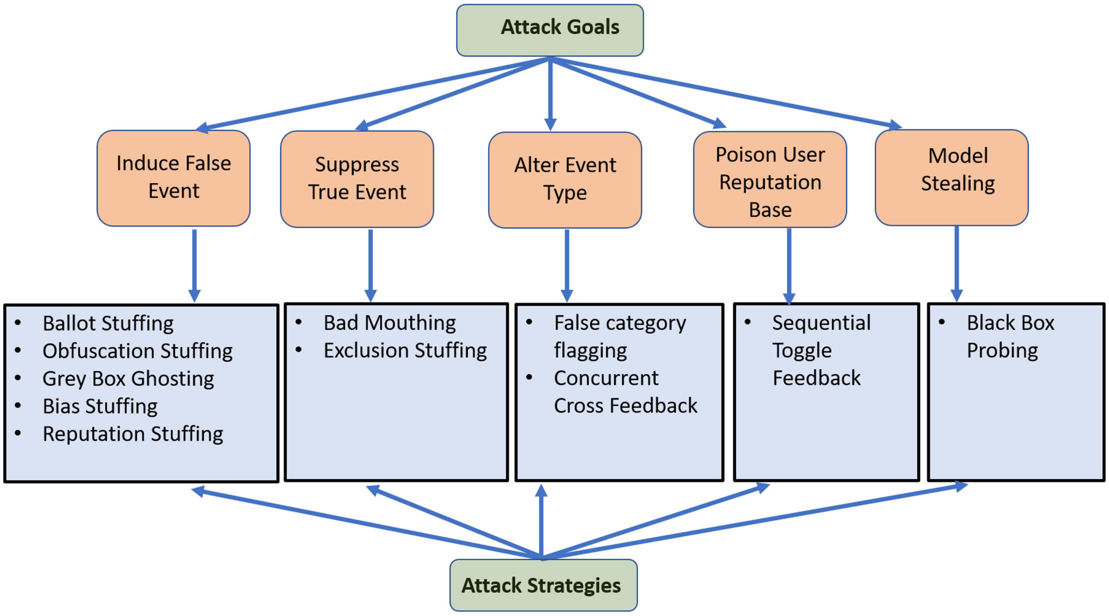

FIGURE 2. Unified attack landscape in P-CS systems.

Alter Event Type

The objective is to make the CS arrive at the wrong event type, even if an event did happen. In this case, the adversaries induce an "incorrect action" to exacerbate the consequences of the event that did happen. For example, let there be a congestion in a certain part of the city, but the malicious reporters falsely report low gas prices in the same area. By combining strategies, such as false category flagging and concurrent cross feedback strategies (discussed later), the CS server can be triggered into having sufficient confidence on publishing the event of low gas prices. This may cause many passive consumers of CS to reach this area of the city, worsening the civilian impact of the congestion already present.

Poison User Reputation Model

Since all methods learn the reputation of users to infer truthfulness of events, an adversary can poison the reputation learning model such that the CS server is unable to accurately classify malicious versus honest behaviors. The economic impact is that the honest reporters will be eventually discouraged from participating due to lack of incentives, leaving mostly malicious users active in the CS system. The civilian impact will be that events from the CS will no longer be perceived as reliable.

Steal Event Publishing Model

The goal of the adversary is to learn a surrogate model of how the CS server decides whether to publish an event or not (especially in the cold start phase), to improve the efficiency of attack budget allocation across a wider area of the P-CS network. Since the adversary can now use its budget more efficiently, a wider civilian and economic impact of previous attack goals can be achieved.

To achieve the above objectives, different categories attack types can be developed as discussed next. The attack types depend on which stage of the P-CS operating cycle the attacker wants to realize its goal. Each attack type can have multiple attack strategies classified under it. Note that the attack goals are complex and the attacker may require a combination of strategies to achieve them, as illustrated in Figure 2.

Categories of Attack Types

The "attack goals" can be achieved through various "attack types" depending on which stage of the P-CS operation life cycle, the attacks are launched. The categories of attack type include 1) sensory manipulation, 2) feedback weaponizing manipulation, and 3) belief manipulation.

An attack type can be realized via multiple attack strategies belonging to a particular attack type. The attack strategies formalize the implementation issues of an attack type and how they help realize a certain attack goal even in the presence of trust and reputation scoring models (e.g., the QnQ model[2]), and why the vulnerability exists.

First, we formally define each category of attack type, followed by enumerating different attack strategies under each attack type. The attack strategies under each attack type depend on the level of prior knowledge the attacker has. The level of knowledge is: 1) complete (white box attack); 2) partial (gray box), and 3) no knowledge (black box). The strategies also depend on the attack goal, and there is a goal to strategy mapping as described in Figure 2.

Sensory Manipulation

These attacks exploit weaknesses in the event reporting phase of P-CS operations. The adversary compromises (recruits) a set of malicious reporters who submit fake reports strategically. We propose three targeted attack strategies to launch sensory manipulation: *1) Gray Box Probe, 2) Black Box Probe, and 3) False Category Flagging.*

Feedback Weaponizing Manipulation

These attacks exploit weaknesses in the feedback monitoring phase that collects evidence to quantify the truthfulness of events contributed by the reporters. Formally, these attacks involve submitting a dishonest feedback by a rating user recruited/compromised by the adversary for different events. The feedback weaponizing includes specific attack strategies such as *1) targeted ballot stuffing, 2) targeted bad mouthing, 3) targeted obfuscation stuffing, 4) orchestrated sequential toggle feedback, and 5) concurrent cross feedback*.

Belief Manipulation

These attacks exploit algorithmic biases that originate from the use of "prior event likelihoods" and "prior user reputation" that act as weights in most truth discovery and decision-making schemes, post the cold start training phase. Formally, belief manipulation attack type involve strategies that exploit the dependence on learnt beliefs and in turn utilize such beliefs to craft attacks that nudge P-CS into taking wrong decisions. Analogically, they are similar to evasion in machine learning, where a sample input in the test phase is incorrectly classified by a model. The belief manipulation includes specific attack strategies: *1) reputation stuffing, 2) bias stuffing, and 3) exclusion stuffing.*

SPECIFIC ATTACK STRATEGIES IN P-CS

In this section, we put forward various possible attack strategies under each category of attack type and discuss how they achieve various attack goals given the malicious intent.

Sensory Manipulation Strategies

The false reports can be intelligently submitted in the cold start phase by the following approaches:

Grey Box Ghosting

Many research solutions use "context" similarity among reports[5,12] to compute the event trust or infer the correct event. Some methods known as "truth discovery" incorporate correlation, maximum likelihood estimate, and expectation maximization (first proposed by Wang et al.[17]) from the received reports to find the correct event. Regardless of the techniques, the common assumption is that the majority of the participant reporters are honest except some unreliable participants with isolated selfish objectives; therefore, this method works. While the above assumption may sound reasonable in theory, a common practical feature in P-CS is that "the honest participants need not report anything in the absence of an event." Hence, high correlation and similarity among false reports from an adversary is implicitly guaranteed regardless of the method used to compute such similarity or truth discovery, making this attack relevant.

The attacker submits a number of fake reports when there is no event, and ensures that all fake events agree on the event type and in the same spatial or temporal context.

Hence, methods based on correlation, similarity, truth discovery, and voting cannot prevent against such collusive sensory manipulation attacks in P-CS. Such methods can only help find the correct event type, if an event did occur. The above exploit is a grey box strategy since it requires some knowledge of the design philosophy that context similarity or correlation in reports are used to quantify truthfulness of events.

Black Box Probing

The gray box ghosting is simple in itself, but has one flaw in the sense that the adversary does not know how many fake reports are sufficient to actually trigger a fake event to be published. The adversary needs to steal the above information of the event publishing model, to make its sensory manipulation attacks (like gray box ghosting) very effective.

To achieve the above, the attacker launches a black box probe strategy: *During the reporting phase, the adversary recruits (or deploys) a set of participant users and blends itself in the user population. This malicious reporter base tries different candidate numbers of false reports and false event categories, and monitors which combinations were successful in* *inducing a false event and which ones failed to induce a false event.*

Note that, since P-CS is an open paradigm, an event's presence or absence on the mobile app are visible to all the users. The absence of the fake event on the app proves that the input attack combination was invalid. Thus, the adversary can learn an input–output relationship between the candidate attack inputs and the boundary between successful and failed false events triggered in the app. This allows the adversary to learn the lowest quantity of false reports in order to induce a false event. By preventing local overprovisioning of its total attack budget, the adversary can improve its network wide spatial attack coverage or save the remaining budget for other attack types (e.g., feedback weaponizing.)

False Category Flagging

This strategy is relevant because different categories of events are possible at the same spatial/temporal or some other context; and it is equally important for the P-CS to know the type of event that happened exactly under conflicting reports.

During the reporting phase, the malicious reporter base gives false reports only if an event actually occurs, but chooses a different event category than the actual event type, to mislead the response to the event.

The response is misled since the inferred event type can potentially be altered. Most of the existing approaches to similarity, correlation, and truth discovery[5] offer some protection against this particular attack type, but these works do not explicitly differentiate between the ghost event and false category flagging. Later we show that such attack can be made effective using *concurrent cross feedback strategy* to maximize the probability of altering event type.

Feedback Weaponizing Manipulation Strategies

The feedback monitoring apparatus behaves like a voting system. It is highly sensitive to variations in legitimate participation of the user base in the rating process and the attacker's recruitment/attack budget. This tradeoff can be exploited by an intelligent adversary to circumvent the event trust models and user reputation models, such as Josang's,[7] Dempster-Shafer,[11] and our previous work on the QnQ method.[2] Below we provide details of different types of feedback weaponizing attacks in P-CS.

Orchestrated Ballot Stuffing

In ballot stuffing attacks, false events are given positive feedbacks by the adversary.[2,7] The goal is to make a false event to be inferred by the P-CS as truthful and boost reputations of malicious reporting users. While the QnQ method[2] provides a defense to mitigate such attacks, orchestrated versions of this can be successful when there is 1) sparseness of rating population (new app or spatial sparseness), and 2) lack of incentives in the honest population to rate an event.

The strategy works as follows: *When or where there is low participation in the rating process, the adversary focuses his budget in those contexts, to ensure a high proportion of fake positive ratings given to false events, even with a seemingly low attack budget. Therefore, a false event ends up with a high event truthfulness score and incorrectly appears to be true to the P-CS.*

The impact of strategy is that the false events persist on the P-CS platform as a result of the high truthfulness score. Consequently, those malicious participants who were originally involved in the orchestration of the bogus false event, improve their reputation, since their reports produced a seemingly truthful event. Such attacks in the cold start phase help the malicious participants start building an edge in terms of their reputation scores compared to the honest participants, which achieve the goal of poisoning the user reputation learning model. Finally, high biased reputation to malicious users also causes incentives to be given out to malicious participants causing economic loss.

Orchestrated Obfuscation Stuffing

In such attacks, the adversary deliberately gives false events a large number of uncertain ratings. Since in all established trust models (e.g., Josang's, Dempster Shafer, Dirichlet Reputation, QnQ), the uncertainty contributes to the trust score, malicious users creating a spike in the number of uncertain ratings will cause the false events a high truthfulness score, making this strategy relevant.

In the orchestrated form, the attack strategy is exactly the same as the orchestrated ballot stuffing, but instead of giving positive ratings, all uncertain ratings are provided to the false event.

The impact of this strategy is similar to the orchestrated ballot stuffing, but can be less obvious to the detection mechanisms like Josang's and Dempster Shafer due to null invariance as detailed in.[2,3]

Orchestrated Bad Mouthing

In bad mouthing attacks, the true events are provided with negative ratings by the adversary. The goal of adversary is to suppress or recall a true event as well as degrade the reputation scores of truthful reporting users. While the QnQ method[2] mitigates bad mouthing attacks, it suffers from the same vulnerabilities as low participation in ratings and lack of incentives attached to the ratings.

The strategy works as follows: *In contexts with low participation in the rating process, the adversary puts its budget in those contexts, to ensure a high proportion of* fake negative ratings*to the true events, even with a seemingly low attack budget. Therefore, true events end up with a low event truthfulness score and incorrectly appears as a false event to the P-CS.*

Consequently, the P-CS platform withdraws these published events, resulting in the suppression of true events. Then, the user reputation system will penalize those honest users reporting this event (since truthfulness of events is key to improving reputations). After repeating this attack multiple times, the honest participants end up with lower reputation scores having the following impacts: Honest participants with low reputation will not get a high weight during the test phase decision-making and will also get progressively lower or no incentives, thereby discouraging them and new users to participate truthfully. Thus, a P-CS will be left with a user base that consists of participants largely controlled by the adversary.

Orchestrated Sequential Toggle Feedback

This kind of attack strategy is relevant if the adversary has a long-term objective of poisoning the user reputation learning process. The attack happens as follows: *Orchestrated bad mouthing and ballot stuffing are launched in alternating manner to different events over time. First, targeted bad mouthing will slowly discourage the honest user base to refrain from participation. Then, via targeted ballot stuffing, the user base will be simultaneously replaced with compromised participants having artificially boosted reputations.*

The impact will be a P-CS system with a seemingly high trusted base controlled by a motivated adversary, and faces little competition from honest users. This will destroy the credibility of the P-CS provider. The impact of a sequential toggle feedback attack is remarkably different compared to just ballot stuffing, bad mouthing, or obfuscation stuffing. It will create a completely poisoned reputation base, where the malicious or dishonest users have higher reputation compared to the honest users.

Concurrent Cross Feedback

This attack is relevant only when each user report is separately visible to the rater population (e.g., social media plug-ins, Yelp, Yik Yak) and each report indicates an event type. The goal is to allow the P-CS server make an error in judging the correct event type using the feedback apparatus.

Using its recruited user base, the adversary concurrently give positive feedback to the reports with incorrect event category (from malicious reporters), and negative feedback to the reports with correct event category (from honest reporters) for the same event.

The impact of strategy is that it enhances the chance of the CS server making an error in judging the correct *event type*, inducing a misguided response.

Belief Manipulation Strategies

Three aspects are typically used to take decisions in a typical P-CS in the steady-state phase: 1) prior reputation of the reporters,[5] 2) historical contextual likelihood of the event,[12] and 3) quantity of unique reports indicating an event. Typically a weighted approach is taken that is some variation of weighted reputation aggregation[13] or decision tree formulation[19] to decide whether or not to publish and report in the steady-state phase. Below we summarize the type of attacks that are possible under this category.

Reputation Stuffing

Since decision trees or weighted reputation aggregation methods give higher importance to the more reputed participants in the event publishing models, it would make sense for an adversary to recruit/compromise highly or most reputed users.

The adversary recruits/compromises a fraction of highly reputed participants and asks them to report a fake event in the same context; thus the decision-making module believes in the event.

The event accuracy will drop while the event inaccuracy will rise with the increase of the fraction of most highly reputed participants recruited for false reporting.

Bias Stuffing

A high importance is given to the prior likelihood of event (given a context) from the cold start phase, in most event publishing models.

The adversary exploits the bias toward high prior likelihood of events in decision-making and decision classification models used in the steady-state phase. Basically it spoofs a false event report from its recruited malicious user base strategically in "contexts," where that event type had a high prior likelihood of contextual occurrence in the cold start phase.

Such attacks are able to convince the decision-making module of the CS to publish a false event incorrectly. An implementation of this attack and how it negatively impacts the accuracy of events published by a classical decision tree is discussed in Bhattacharjee et al.[2] Furthermore, a prospect theory inspired decision tree approach is proposed in Bhattacharjee et al.[2] to mitigate such attacks.

Exclusion Stuffing

The quantity of reports[5] received indicating a particular event, influences event publishing decisions apart from the prior likelihood of the event type. *In exclusion stuffing, an adversary suppresses the reception of true reports from legitimate reporters via a jamming attack or distributed denial of service (DDoS) attack between the reporters and the CS server, only in contexts with a low prior likelihood of an event. Such reduction in the number of true reports (thus reducing the quantity) tilts the odds in favor of inferring that the event is not true.*

Since both the likelihood and the quantity of support is low, such events although true, do not get published and the P-CS server misses true events. Thus, the goal of true event suppression is achieved by this strategy. An implementation of this attack and how it negatively impacts the accuracy of events published by a classical decision tree is discussed in Bhattacharjee et al.[2]

CONCLUSION

This article provided a detailed threat model that targets weaknesses in shared design philosophies with a goal to improve the security and trustworthiness in participatory crowd sensing (P-CS) paradigms. We conclude that some level of cyber deception is required for feedback manipulation strategies that are targeted. Furthermore, event publishing in the cold start phase for a participatory CS application depends not only on the correlation and similarity of reports, but also on the additional design considerations that can mitigate event publishing model stealing via black box probe strategies. Sequential toggle feedback can poison the user reputation base when participation in the rating apparatus is low, and this requires further research for effective solutions. While false category flagging has been studied earlier, more research is required on concurrent cross feedback attacks, sequential toggle feedback attacks, black box probing, how event truthfulness or publish decision models are able to handle this threat.

ACKNOWLEDGMENTS

This work was supported by the US National Science Foundation under Grant # SATC-2030611, Grant SATC-2030624, Grant DGE-1914771, and Grant OAC-2017289.

REFERENCES

1. R. P. Barnwal, N. Ghosh, S. K. Ghosh, and S. K. Das, “Publish or drop traffic event alerts? Quality-aware decision making in participatory sensing-based vehicular CPS,” *ACM Trans. Cyber- Phys. Syst. (Special Issue Transp. Cyber-Phys. Syst.)*, vol. 4, no. 1, pp. 1–28, Jan. 2020.
2. S. Bhattacharjee, N. Ghosh, V. Shah, and S. K. Das, “QnQ: Quality and quantity based unified approach for secure and trustworthy mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, 200–216, Jan. 2020.
3. S. Bhattacharjee, N. Ghosh, V. K. Shah, and S. K. Das, “QnQ: A reputation model for securing mobile crowdsourcing systems from incentive losses,” in *Proc. IEEE Commun. Netw. Secur.*, 2017, pp. 1–9.
4. P. Roy, S. Bhattacharjee, H. Alsheakh, and S. K. Das, “Resilience against bad mouthing attacks in mobile crowdsensing systems via cyber deception,” in *Proc. IEEE World Wireless Mobile Multimedia Netw.*2021, pp. 169–178.
5. F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, “Quality of information in mobile crowdsensing: A survey,” *ACM Trans. Sensor Netw.*, vol. 13, no. 4, pp. 1–43, 2018.
6. Z. Feng, Y. Zhu, Q. Zhang, L. Ni, and A. Vasilakos, “TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing,” in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 1231–1239.
7. A. Jøsang, “An algebra for assessing trust in certification chains,” *Netw. Distrib. Syst. Secur. Sympos.*, 1999.
8. R. Ismail and A. Jøsang, “The Beta reputation system,” in *Proc. Bled eConference*, 2002, pp. 2502–2511.
9. A. Josang and J. Haller, “Dirichlet reputation systems,” in *Proc. IEEE Conf. Availability, Rel. Secur.*, 2007, pp. 112–119.
10. D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, “On truth discovery in social sensing: A maximum likelihood estimation approach,” in *Proc. ACM Inf. Process. Sensor Netw.*, 2012, pp. 233–244.
11. B. Yu and M. P. Singh, “An evidential model of distributed reputation management,” in *Proc. ACM Int. Conf. Auton. Agents Multiagent Syst.*, 2002, pp. 294–301.

12. X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, “ARTSense: Anonymous reputation and trust in participatory sensing,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2013, pp. 2517–2525.
13. Y. Li et al., “Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 1986–1999, Aug. 2016.
14. C. Miao, Q. Li, H. Xiao, W. Jiang, M. Huai, and L. Su, “Towards data poisoning attacks in crowd sensing systems,” in *Proc. 8th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2018, pp. 111–120.
15. 2014. Accessed: Jul. 2023. [Online]. Available: https://www.androidauthority.com/residents-put-fake-reports-waze-divert-traffic-574744/
16. D. Yue Zhang, J. Badilla, Y. Zhang, and D. Wang, “Towards reliable missing truth discovery in online social media sensing applications,” in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2018, pp. 143–150.
17. D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, “On truth discovery in social sensing: A maximum likelihood estimation approach,” in *Proc. IEEE/ACM Intl. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 233–244.
18. Y. Li et al., “A survey on truth discovery,” *ACM SIGKDD Explorations Newslett.*, vol. 17, no. 2, pp. 1–16, 2016.
19. Y. Du, V. Issarny, and F. Sailhan, “User-centric context inference for mobile crowdsensing,” in *Proc. ACM IoT-Des. Implementation*, 2019, pp. 261–266.

SHAMEEK BHATTACHARJEE is an assistant professor with Western Michigan University, Kalamazoo, MI, 49008, USA. His research interests include theory of anomaly detection, artificial intelligence based security, and data science for cyber security. Bhattacharjee received his Ph.D. degree in computer engineering from the University of Central Florida. He is an IEEE and ACM professional member. He is the corresponding author of this article. Contact him at shameek.bhattacharjee@wmich.edu.

SAJAL K. DAS is a curators’ distinguished professor of computer science and Daniel St. Clair Endowed Chair with the Missouri University of Science and Technology, Rolla, MO, 65409, USA. His research interests include cyber-physical systems, cybersecurity, machine learning, pervasive and mobile computing, wireless sensor networks, and mobile crowdsensing. Das received his Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA. He is a fellow of IEEE. Contact him at sdas@mst.edu.