Ensuring Honest Data Collection against Collusive CSDF Attack with Binary-minmaxs Clustering Analysis in Mobile Crowd Sensing

Jingyu Feng, Tao Li, Yujia Zhai, Shaoqing Lv and Feng Zhao

Abstract—Mobile crowd sensing (MCS) is considered as a powerful paradigm which takes advantage of the pervasive sensor-embedded smartphones to collect data. However, MCS assumes all workers always are trusted, and thus offering opportunities for malicious workers to conduct the crowd sensing data falsification (CSDF) attack. To suppress such threat, recent efforts have been made to trust mechanism. Currently, some malicious workers can collude with each other to form a collusive clique, and thus not only increasing the power of CSDF attack but also avoiding the detection of trust mechanism. To ensure honest data collection in MCS, we must fight against such collusive CSDF attack. Noting that the duality of sensing data, we propose a defense scheme called BMCA from the design idea of binary-minmaxs clustering analysis to suppress collusive CSDF attack. In the BMCA scheme, the logic AND operation corresponding to the type of “1” and “0” historical sensing data is used to measure the similarity between any two workers. Based on this, we find the feature that collusive CSDF attackers usually hold high trust value and a low variance in their similarity vector. To detect collusive CSDF attack, the min and max variance analysis is introduced to design a new binary-minmaxs clustering algorithm. Moreover, the BMCA scheme can perfect trust evaluation to prevent the trust value growth of collusive CSDF attackers. Simulation results show that the BMCA scheme can enhance the accuracy of trust evaluation, and thus successfully reducing the power of collusive CSDF attack against data collection in MCS.

Index Terms—Mobile crowd sensing, Trust mechanism, Clustering Analysis, Collusive attack.

I. INTRODUCTION

With the rapid development of mobile communication and wireless sensing, a large number of sensors are integrated into smart mobile terminals, which motivates mobile crowd sensing (MCS) as an emerging paradigm to collaboratively collect sensing data and extract knowledge in smart cities [1, 2]. MCS leverages the inherent mobility of mobile users (called workers), the sensors embedded in mobile phones and the existing communication infrastructures (Wi-Fi, 4G/5G networks) to collect and transfer urban sensing data [3]. Nowadays, MCS has been widely used in the intelligent traffic detection [4], environment detection [5], smart healthcare [6], social network [7], etc. These applications open the door for new innovative research and will significantly revolutionize our daily lives [8].

Although with the various MCS-enabled innovative applications, the new sensing paradigm also encounters new challenges as “humans” act as sensors [9]. Actually, the success of MCS requires the participation from a large number of workers [10]. Due to the nature of openness and mobility, MCS gives almost all mobile users the chance to participate in MCS activities as workers. However, the workers in MCS maybe unreliable and vary in terms of ability, honesty, dependability, loyalty and so on [11]. It may offer opportunities for malicious workers to corrupt the data collection by launching CSDF attack. Such CSDF attack pattern can be launched by two ways: individual or collusive. Compared with collusive attack, individual attack is less harmful and can be suppressed. In the collusive pattern, the malicious workers who collude with each other to form a collusive clique can increase the power of CSDF attack and fake the sensing data intentionally. If there are the adequate malicious workers, a collusive clique can lead to a wrong decision in the data aggregation.

Accordingly, trust mechanism is recognized as an important part of the MCS platform in which the trust value of mobile workers is evaluated and used to suppress CSDF attack. With trust mechanism, trusted workers can not only perform honest behaviors in MCS, but also fulfill the requirements of a certain task with high quality. The malicious workers with low trust value will be rejected to in the MCS task assignment. Currently, various trust mechanism studies have been presented [12-15]. They evaluate whether a worker is trusted or not by his historical sensing behaviors and give the low weights to less trusted workers or even delete their sensing data when making a final decision in the data aggregation. Nevertheless, collusive CSDF attackers can improve their trust value with the help of each other, except for increasing the attack power. Therefore, they may avoid the detection of trust mechanism to corrupt the data collection in MCS.

In this paper, we analyze the characteristics of collusive CSDF attack and propose a defense scheme called BMCA from the design idea of binary-minmaxs clustering analysis to suppress such attack. The main contributions of this paper are as following:

- Abstract the MCS behaviors of each worker as a binary variable “1” or “0”, in which “1” denotes the false behavior of a worker in a task by comparing their sensing data with the final result of the data aggregation. Noting that collusive CSDF attackers often launch “1” behaviors together in a certain task time and behave high similarity among themselves, the logic AND operation is introduced to measure the similarity, which can result in less complex and lightweight in mathematical computation.
Workers, SP EUs

SP EUs

Fig. 1. System architecture of MCS.

In the MCS platform, the SP could be an organizer that assigns a task for crowd sensing. An EU could be an individual or organization that lacks an ability to perform a certain data collection task. Workers are the mobile users who participate in MCS and perform the assigned tasks.

The life-cycle of MCS can be divided into four stages [3]: task creation, task assignment, data collection and data aggregation, which are briefly described as follows:

- **Task creation**: When EUs request services, the SP creates a sensing task to be given to workers with the corresponding mobile applications.
- **Task assignment**: The SP selects workers and assigns them with the specific sensing task.
- **Data collection**: Once the workers have received the assigned task, they can complete it and send their sensing data to the SP.
- **Data aggregation**: The SP aggregates the received sensing data to determine the final decision about the current task, and sends it to EUs.

### B. Trust Mechanism

Trust mechanism is increasing influence on many application scenarios, including e-commerce [18], P2P file-sharing [19], cooperative spectrum sensing [20], online social communities [21], etc.

Trust mechanism also plays significant roles in MCS, such as 1) select trusted workers in the task assignment, 2) filter out false sensing data reported by attackers in the data collection, 3) prevent malicious workers from participating into MCS, and 4) assist the SP’s rapid decision-making reliably in the data aggregation.

Representative trust mechanism schemes in MCS are as follows. To evaluate the trust value of workers, the reference [12] comprehensively considered a number of properties that affect the honest sensing data, such as link reliability, service quality, and region heat. In [13], a ranking-based is proposed scheme that introduces trust and worker ability into the e-evaluation of worker trust value. In [14], a dynamic-trust-based recruitment framework (DTRF) is proposed for MCS, in which real-time direct trust and lightweight feedback aggregation trust are combined to select the well-suited workers. In [15], a dynamical credibility assessment of privacy-preserving [22] strategy is designed, in which the sensing data are dynamically split into slices and the number of slices is based on the trust of encountered workers. Specially, worker trust is assessed in two dimensions including the quality of contribution trust and social trust, which indicates how likely a worker can fulfill its functionality and how trustworthy the relationship between a worker and other workers will be, respectively.

In trust mechanism, one of the most popular design is based on beta function. It counts the number of false and honest behaviors a worker has conducted in the data collection, and then evaluates the trust value with beta probability density function denoted by \( \text{Beta}(\alpha, \beta) \) [23].

\[
\text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1 - \theta)^{\beta-1}
\]
where $\theta$ is the probability of MCS behaviors, $0 \leq \theta \leq 1$, $\alpha > 0$, $\beta > 0$.

Noting that the sensing data from workers can be abstracted as a binary variable (“1” or “0”), it is easy for workers to behave two types of sensing behaviors: false or honest. Based on the binary variable, a basic trust mechanism scheme called Baseline can be described, in which the trust value of each worker can be evaluated by two indexes: the number of false sensing ($fal_i$) and the number of honest sensing ($hon_i$).

Take the $i$-th worker ($W_i$) as an example, $fal_i$ and $hon_i$ denote the number of false sensing data and honest sensing data reported by $W_i$. In the Baseline, the trust value of $W_i$ can be evaluated as

$$T_i = Beta(fal_i + 1, hon_i + 1)$$  \hspace{1cm} (2)

Consider the condition $T(x) = (x-1)!$ when $x$ is an integer [24]. Thus, the expectation value of the beta function is $E[Beta(\alpha, \beta)] = \alpha / (\alpha + \beta)$. In this case, $T_i$ can be further evaluated as

$$T_i = \frac{1 + hon_i}{2 + fal_i + hon_i}$$  \hspace{1cm} (3)

Let $\delta$ denotes the threshold of trust value. For $T_i \geq \delta$, $W_i$ will be identified as an attacker, and vice versa. In order to guarantee the performance of MCS, $\delta$ should satisfy two requirements: 1) $\delta$ should be a rational value between 0 and 1 as $T_i \in [0,1]$, 2) the value of $\delta$ can be adjusted to suppress malicious responses generated by malicious workers who report false sensing data.

However, how to suppress collusive CSDF attack has not been considered in current trust mechanism for MCS. In this paper, we argue that securing MCS with only trust mechanism is not enough. Malicious workers can attain high trust value with the help of each other. Specially, the malicious workers with high trust value can launch CSDF attack together in a collusive manner to corrupt the data collection.

III. COLLUSIVE CSDF ATTACK OVERVIEW

Due to the nature of openness and mobility, it is very easy for malicious workers to launch CSDF attack by reporting false sensing data, resulting in a wrong final decision of in the SP.

At first, malicious workers launch CSDF attack individually and respectively. The individual CSDF attack and defense have appeared in some existing works. In [25], the authors analyze the problem of aggregating noisy labels from crowd workers to infer the underlying true labels of binary tasks. To address this problem, they also design a reputation-based worker filtering algorithm that uses a combination of disagreement-based penalties and optimal semi-matchings to identify adversarial workers. In [26], the authors develop an optimal attack framework in which the attacker can not only maximize his attack utility but also disguise the introduced malicious workers as normal ones such that they cannot be detected easily. The strategy derived from the proposed optimal attack framework makes the malicious workers behave “smarter”. If there is little hope to achieve the attack goal on some objects, they will tend to agree with the normal workers on those objects to gain higher weights, and in turn, can exert stronger impact on other objects [26]. However, it is difficult for a malicious worker to improve his trust value fast all by himself through this occasional reliable behavior. Overall, the power of individual CSDF attack is finite and can be suppressed by trust mechanism easily.

To avoid the detection of trust mechanism, some malicious workers attempt to collude with each other and submit false sensing data together at the same time. This attack pattern can be called collusive CSDF attack. Generally, the collusive attack pattern can increase the power of malicious workers, which can be launched through three methods.

In the first method, a malicious worker can acquire multiple IDs to fake sensing data through the sybil attack [27]. There are many schemes to defense against the Sybil attack [28, 29]. Specially, if each IP is restricted to acquire an ID, this attack pattern can be addressed easily. In the second method, a malicious worker can control multiple computers by embedding trojan viruses. This attack pattern can suppressed by using a good antivirus software. In the third method, multiple malicious workers collaborate together to fake sensing data. In this attack pattern, each malicious worker only has an ID. Currently, this attack pattern is used as a popular collusive attack, especially in MCS to fake sensing data. Although this collusive attack pattern is considered in [30], how to improve the trust value of malicious workers is not discussed. Except for increasing the attack power, collusive CSDF attack can also be launched to avoid the detection of trust mechanism.

In this paper, we further analyze the characteristics of MCS and the collusive CSDF attack demand, and thus finding three types of threats that can be achieved by collusive CSDF attackers, including increase the attack power, improve trust value and disturb the data aggregation.

- **Increase the attack power:** As we know, “more hands make light work” and “more people, more powerful”. Inspired by this, malicious workers can conspire with each other to form a collusive clique to increase the attack power, and thus faking sensing data intentionally to corrupt the data collection.

- **Improve trust value:** By collusion, collusive CSDF attackers can improve their trust value quickly. For instance, one of collusive CSDF attackers can disguise as an EU and preselect a data aggregation decision. Then, this attacker would tell it to his conspirers in advance as well as send a Query message to the SP. Their trust value can be improved quickly if their sensing data are consistent as the SP’s decision in the data aggregation.

- **Manipulate the data aggregation:** With high trust value, collusive CSDF attackers can avoid the detection of trust mechanism, and thus reporting their false sensing data to manipulate the data aggregation successfully and mislead the SP to make a wrong decision.

To maintain the opportunity to manipulate the data aggregation, collusive CSDF attackers are extremely sensitive to their trust value. As shown in Fig.2, they begin collusive CSDF attack procedure under the constraint...
where $N_c$ is the number of collusive CSDF attackers and $W_i$ is one of them. $||T_i\leq\delta + \lambda||$ denotes the number of collusive CSDF attackers under the case $T_i\leq\delta + \lambda$. Here, $\lambda (0\leq\lambda < 1-\delta)$ is the initial trust warning line for collusive CSDF attackers. It is too late to improve trust value when $||T_i\leq\delta|| \leq \frac{N_c}{2}$. In this situation, the majority of collusive CSDF attackers may be marked as malicious. This attack pattern continues until $||T_i > \delta|| = N_c$. Then, collusive CSDF attackers can increase their attack power to corrupt the data collection and manipulate the data aggregation successfully.

**IV. PROPOSED DEFENSE SCHEME**

In this section, we propose a defense scheme called BMCA from the design idea of binary-minmaxs clustering analysis to suppress collusive CSDF attack in MCS. Moreover, the BMCA scheme can be used to perfect trust evaluation.

**A. Design Idea**

To design the defense scheme of collusive CSDF attack, we analyze its attack threats, and thus finding three kinds of general features as follows.

- **Binary data**: Workers generally report honest or false sensing data in the data collection. Thus, the sensing behaviors of workers in the MCS environment can be abstracted as the binary data with the type of “1” or “0” sensing data.

- **Action together**: Collusive CSDF attackers always report false sensing data together no matter which threats they would launch.

- **High trust value**: Collusive CSDF attackers often have high trust value. With the help of each other, they can improve their trust value to avoid the detection trust mechanism.

Considering the “Binary data” and “Action together” of general features, we introduce the design idea of binary-minmaxs clustering analysis based on the logic AND operation to construct the defense scheme called BMCA to suppress collusive CSDF attack. In addition, the “Action together” of general feature can make collusive CSDF attackers behave a low variance in their similarity vector. In this basis, we analyze the “High trust value” of general feature to design the algorithm of detecting collusive CSDF attackers.

**B. Binary Behaviors Analysis**

It is worth noting that the sensing data from each worker can be abstracted as a binary variable: false or honest. Based on this, the SP can abstract the MCS behaviors of workers as: “1” or “0”. Specifically, the false behavior of a worker in a task by comparing their sensing data with the final decision of the SP can be abstracted as “1”, whereas the honest behavior of a worker can be abstracted as “0”.

The “1-0” database is designed as an extensible database, whose size corresponds to the task times of MCS actions. After each MCS action, the SP should add a row in the “1-0” database to record the abstracted binary behaviors related on the sensing data from workers. When the current MCS action is numbered as the $k$-th task time, the size of the “1-0” database is $k$. In the “1-0” database, $n$ is the total number of workers in the MCS platform.

Take the $i$-th worker ($W_i$) as an example, $W_i(b_i)_k$ is recorded as $W_i(1)_k$ when $W_i$ reported honest sensing data, $W_i(b_i)_k \rightarrow W_i(0)_k$ when reported honest sensing data and $W_i(b_i)_k \rightarrow W_i(-)_k$ when reported nothing at the $k$-th task time.

In the collusive CSDF attack, we have known that malicious workers report false sensing data together to corrupt data...
collection. That is, they often launch “1” behaviors together in a certain task time and behave high similarity among themselves. Therefore, the logic AND operation (&) can be introduced to measure the similarity. Take \( W_i \) and \( W_j \) as an example of any two workers, if \( W_i(b_{ij}) \rightarrow W_i(1)_k \), \( W_j(b_{ij}) \rightarrow W_j(1)_k \), \( 1 \& 1 = 1 \) under the logic AND operation. Otherwise, \( 1 \& 0 = 0 \) or \( 0 \& 1 = 0 \).

For the convenience of measuring the similarity between any two workers, the history binary behaviors of each worker should be extracted from the “1-0” database as a vector in the current MCS action. For \( W_i \), its behaviors vector can be represented as \( B_i = [W_i(b_{ij})_1, W_i(b_{ij})_2, \ldots, W_i(b_{ij})_k] \). If \( W_i(b_{ij})_1 \rightarrow W_i(0)_1 \), \( W_i(b_{ij})_2 \rightarrow W_i(2)_2 \), \( W_i(b_{ij})_k \rightarrow W_i(1)_k \), \( B_i \) can be definitely described as \([0, \ldots, 1]\).

Obviously, the redundant data such as \( W_i(-2) \) or \( W_j(-6) \) are useless to measure the similarity between \( W_i \) and \( W_j \). Since “1” is useful to analyze the collusive features of malicious workers, it can be set \( W_i(-2) = W_i(0)_2 \) and \( W_j(-6) = W_j(0)_6 \). As a consequence, both the honest behavior and nothing reported by a worker can be abstracted as “0”.

For \( W_i \) and \( W_j \), the logic AND operation between \( B_i \) and \( B_j \) can be described as

\[
B_{ij} = B_i \& B_j
\]

Then, the similarity between \( W_i \) and \( W_j \) can be measured as

\[
sim_{ij} = \frac{|\ell_{ij}| + c_{ij}}{|B_{ij}| + c_{ij}}, 1 \leq |\ell_{ij}| \leq |B_{ij}|
\]

where \(|\ell_{ij}|\) denotes the amount of “1” in \( B_{ij} \) and \(|B_{ij}| = k - 1\) is corresponding to the amount of elements in \( B_{ij} \).

Specially, \( c_{ij} \) is used to record the continuous false behaviors of malicious workers. For instance, if \( W_i \) and \( W_j \) suddenly report false sensing data simultaneously at the \( q \)-th task time, they may continue to report false sensing data at the \( (q+1) \)-th, \( \ldots, (q+c_{ij}) \)-th task time. Such continuous false behaviors can increase the similarity between \( W_i \) and \( W_j \). The continuous parameter \( c_{ij} \) can be recorded by Algorithm 1.

\begin{table}[h]
\centering
\caption{Description of the “1-0” database style}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Task times & W ID (the abstracted binary behaviors related on sensing data) \\
\hline
1 & \( W_1(b_{ij})_1 \) & \( W_2(b_{ij})_1 \) & \ldots & \( W_j(b_{ij})_1 \) & \( W_n(b_{ij})_1 \) \\
2 & \( W_1(b_{ij})_2 \) & \( W_2(b_{ij})_2 \) & \ldots & \( W_j(b_{ij})_2 \) & \( W_n(b_{ij})_2 \) \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
\hline
k & \( W_1(b_{ij})_k \) & \( W_2(b_{ij})_k \) & \ldots & \( W_j(b_{ij})_k \) & \( W_n(b_{ij})_k \) \\
\hline
\end{tabular}
\end{table}

The continuous parameter \( c_{ij} \) can make \( W_i \) and \( W_j \) behave a higher similarity due to their continuous false behaviors. It can be proved that

\[
\frac{|\ell_{ij}| + c_{ij}}{|B_{ij}| + c_{ij}} > \frac{|\ell_{ij}|}{|B_{ij}|}
\]

\textbf{Proof:}

\[
\frac{|\ell_{ij}| + c_{ij}}{|B_{ij}| + c_{ij}} = \frac{|\ell_{ij}|}{|B_{ij}|} \star (|B_{ij}| - |\ell_{ij}|) \geq 0
\]

Likewise, we can measure the similarity between \( W_i \) and other workers, and thus generating a similarity vector for \( W_i \), which is expressed as

\[
SIM_i = [\text{sim}_{i1}, \ldots, \text{sim}_{ij}, \ldots, \text{sim}_{in}]
\]

For all workers, their similarity vectors can compose a matrix \( SIM_{n \times n} \).

\[
SIM_{n \times n} = \begin{pmatrix}
\text{sim}_{11} & \cdots & \text{sim}_{in} \\
\vdots & \ddots & \vdots \\
\text{sim}_{n1} & \cdots & \text{sim}_{nn}
\end{pmatrix}
\]

With the similarity measure, the distance between \( W_i \) and \( W_j \) can be calculated as

\[
d_{ij} = \frac{1}{n} \sum_{p=1}^{n} (\text{sim}_{ip} - \text{sim}_{jp})
\]

It can be found that collusion CSDF attack can make \( W_i \) and \( W_j \) behave high similarity but short distance between themselves.

\section{Collusive CSDF Attackers Detection}

In the BMCA scheme, we design a binary-minmaxs clustering algorithm to differentiate collusive CSDF attackers and trusted workers. First of all, we need to select two samples as the initial minmax vectors of this algorithm by analyzing the variance of each similarity vector from workers.

Considering that both collusion CSDF attackers and trusted workers have high trust value, the variance analysis should be performed in \( SIM_{h \times h} \) from the workers with high trust value (\( \Omega \)) at the current MCS action, in which \( h \) is the number of the workers with high trust value. Obviously, \( h \leq n \). Compared with \( SIM_{n \times n} \), it can prompt the detection efficiency to perform binary-minmaxs clustering algorithm in \( SIM_{h \times h} \), since our main purpose is to detect collusion CSDF attackers who hold the characteristics of high trust value.

Since collusion attackers often launch “1” behaviors together, each of them may get a low variance in \( SIM_i \). Let \( var(\cdot) \)
denote the variance function, the initial min vector \((\mu_1)\) is corresponding to collusive attackers, which can be derived from the \(SIM_i\) with the lowest variance.

\[
\mu_1 = \min_{W_i \in \Omega} (\text{var}(SIM_i))
\]  \(10\)

Since trusted workers often report honest sensing data individually, each of them may get a high variance in \(SIM_i\). The initial max vector \((\mu_2)\) is corresponding to trusted workers, which can be derived from the \(SIM_i\) with the highest variance.

\[
\mu_2 = \max_{W_i \in \Omega} (\text{var}(SIM_i))
\]  \(11\)

With the initial minmax vectors \(\{\mu_1, \mu_2\}\), the binary-minmax clustering analysis can be designed by Algorithm 2 to detect collusive CSDF attackers.

**Algorithm 2 Binary-minmax clustering analysis**

**Input:** \(\Omega, SIM_{i,h}, \{\mu_1, \mu_2\}\).

**Output:** the set of collusive CSDF attackers \((\Psi_1)\) and trusted workers \((\Psi_2)\).

1: repeat
2: Initialize \(\Psi_1 = \Psi_2 = \emptyset\)
3: for \(i = 1, i \leq h, i + +\) do
4: Calculate the distance \(d_{is}\) between \(SIM_i\) and \(\mu_s\) \((1 \leq s \leq 2)\) with with equation (9)
5: if \(d_{i1} > d_{i2}\) then
6: \(\Psi_1 = \{W_i\} \cup \Psi_1\)
7: else
8: \(\Psi_2 = \{W_i\} \cup \Psi_2\)
9: end if
10: end for
11: Update the new min vector \(\mu_1' = \min_{W_i \in \Psi_1} (\text{var}(SIM_i))\)
12: Update the new min vector \(\mu_2' = \max_{W_i \in \Psi_2} (\text{var}(SIM_i))\)
13: for \(s = 1, s \leq 2, s + +\) do
14: if \(\mu_1' \neq \mu_s\) then
15: Update \(\mu_s' = \mu_s\)
16: else
17: Keep the current minmax vectors unchanged
18: end if
19: end for
20: until the current minmax vectors are not updated again

Therefore, the trust value of \(W_i\) at the \(k\)-th MCS action can be dynamically updated as:

\[
DT_i = \begin{cases} 
T_i - \Delta_i * \varphi_i, & T_i > \varphi_i \\
0, & \text{else} 
\end{cases}
\]  \(13\)

where \(\Delta_i = 1\) means that the special punishment factor will mitigate when \(W_i\) suddenly reports false sensing data at his \(k\)-th MCS action. Or else, \(\Delta_i = 0\).

Let \(\Phi\) is the set of workers in the current MCS action and \(\Gamma\) is the set of these workers’ trust value. Algorithm 3 can be performed to perfect trust evaluation.

**Algorithm 3 Perfect trust evaluation**

**Input:** \(\Phi, T, \Psi_1\) and \(\Psi_2\)

**Output:** the updated \(\Gamma\)

1: for each \(W_i \in \Phi\) do
2: if \(W_i \in \Psi_1\) then
3: \(fal_i = fal_i + 1\) and his sensing data are deleted
4: \(c_i + +\)
5: if \(c_i > 1\) then
6: Dynamically update the trust value of \(W_i\) with equation (13)
7: end if
8: end if
9: \(hon_i = hon_i + 1\)
10: end if
11: end for

V. SIMULATION ANALYSIS

A. Simulation Setup

We perform simulations to validate the performance of the BMCA scheme and discuss the simulation results. The simulation elements are shown in table II.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_w)</td>
<td>Number of workers</td>
<td>60</td>
</tr>
<tr>
<td>(N_c)</td>
<td>Number of cycle simulation</td>
<td>100</td>
</tr>
<tr>
<td>(P_c)</td>
<td>ROUNDS of attack</td>
<td>50</td>
</tr>
<tr>
<td>(P_a)</td>
<td>Percentage of collusive CSDF attackers</td>
<td>10~50%</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>Initial trust warning line</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The simulations are performed by cycle-based fashion. At each cycle, workers are selected randomly to execute MCS actions with each other. After several cycles, a trusted network topology is gradually generated by trust mechanism. The SP then utilizes it to execute the following MCS actions, and update the trust value on the corresponding workers. In our simulations, the behavior pattern for trusted workers is modeled to always report honest sensing data, while the behavior pattern for collusive CSDF attackers is to alternatively report false or honest sensing data together depending on whether their trust value is greater than the threshold \(\delta\).
B. Simulation Results

One of main goals of collusive CSDF attackers is to improve their trust value. As we know, an attacker such as $W_i$ can be detected by $T_i < \delta$ in trust mechanism. To get more attack opportunities, $W_i$ need to be disguised as a high-trust attacker, i.e. $T_i \geq \delta$. As shown in Fig.4, collusive CSDF attackers can form high-trust attackers with less rounds of attack. Generally speaking, occasional CSDF attackers improve their trust value by reporting honest sensing data when there is little hope to achieve the attack goal [26], whereas collusive CSDF attackers can help with each other. So, collusive CSDF attackers can become high-trust faster than occasional CSDF attackers.

To suppress collusive CSDF attack better under the defense of the BMCA scheme, it is necessary to select a rational value of $\delta$. As $T_i \in [0, 1]$, $\delta$ can be considered from the three types of optional states [low, medium, high]. Then, we can perform the simulation of suppressing malicious responses to validate the effectiveness of the BMCA scheme under the three types of optional states of $\delta$. In this simulation, 0.3, 0.5 and 0.8 denotes the low, medium and high state of $\delta$ respectively. As shown in Fig.5, the performance of the BMCA scheme at $\delta = 0.8$ is the best. Therefore, the rational value of $\delta$ should be selected as 0.8 in the simulations. We can also find that the BMCA scheme is better than Baseline in suppressing malicious responses, even though $\delta$ is selected as 0.8 for Baseline.

To analyze how collusive CSDF attack can affect the performance of trust mechanism, we firstly choose six collusive CSDF attackers randomly to observe the variation of their trust value in the Baseline and BMCA scheme. Fig.6 shows that collusive CSDF strategies make the attacker’s trust value fluctuates along with cycles. The trust value usually outweighs $\delta$ in Baseline. Fortunately, the trust value of collusive CSDF attackers can be reduced by the BMCA scheme. This is because the special punishment to sudden false MCS behaviors is introduced to dynamically updating the trust value of collusive CSDF attackers at each MCS action.

We note that collusive CSDF attackers will deviate the real trust value by forming high-trust workers, and thus causing some network trust errors ($nte$). A higher errors indicate lower accuracy in the evaluation of trust value. With $nte$, we can analyze how collusive CSDF attack affects the performance of trust mechanism from the entire network. $nte$ can be specified by:

$$nte = \frac{1}{N_w} \sum_{i=1}^{N_w} \sqrt{\frac{1}{T_i} (T'_i - T_i)^2}$$

(14)

where $T'_i$ and $T_i$ are the actual and measured trust value of $T_i$, respectively.

In the simulation of $nte$, the actual trust value of a malicious worker is randomly assigned in the interval (0, 0.5]. Without loss of generality, we employ the averaged $nte$ data of 100 cycles as the simulation results. As shown in Fig.7, the BMCA scheme can reduce $nte$ better than Baseline. Even though the percentage of collusive CSDF attackers is 50%, the $nte$ of the TFAA scheme merely achieves 0.28.

Finally, we validate the performance of BMCA in terms of attack success ratio when collusive CSDF attackers fake sensing data with high trust value. Without loss of generality, we employ the averaged attack success ratio data of 50 rounds of attack as the simulation results. At each round of attack, several workers are selected randomly to perform an MCS action from trusted workers and collusive CSDF attackers. In an MCS action, if the collusive CSDF attackers with high trust value are more than the majority of workers, they would attack successfully.

As shown in Fig.8, the attack success ratio against the Baseline scheme amplifies with the percentage of collusive CSDF attackers. This is because collusive CSDF attackers’ trust value usually outweigh $\delta$ in the Baseline scheme, so they can manipulate the decision result of the SP with false sensing data easily. Fortunately, collusive CSDF attackers are difficult to improve their trust value to outweigh $\delta$ in the BMCA scheme. Consequently, collusive CSDF attackers are impossible to manipulate the decision result of the SP, since
Fig. 6. Variation of collusive CSDF attackers’ trust value.
the collusive CSDF attackers with high trust value cannot become the majority of workers.

VI. CONCLUSION

In this paper, we analyze the threats of collusive CSDF attack and propose the BMCA scheme to suppress such attack. The BMCA scheme is conducted in two successive stages: binary behaviors analysis and collusive CSDF attackers detection, in which binary-minmaxs clustering analysis based on the logic AND operation is introduced to design the BMCA scheme due to the type of “1” or “0” historical sensing data. With the special punishment to sudden false MCS behaviors, the BMCA scheme can be used to prevent collusive CSDF attackers’ trust value from growing in the trust evaluation. Simulation results show that our BMCA scheme can enhance the accuracy of trust evaluation and suppress collusive CSDF attack success ratio effectively. In addition, we must continue to fight against the potential threats against MCS. Except for collusive CSDF attack, malicious workers may launch their attack strategies in different tasks. For instance, they behave harmless in some uninterested tasks in order to attain high trust value and always behave harmful in their interested tasks. In the future work, we need to further study a defense scheme from the design idea of distinctive trust evaluation for different tasks to suppress such threat.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China (U1836210, 61572460), the National Science Foundation of Shaanxi Province (2019JM-442), the Science Foundation of Shaanxi Provincial Education Office (17JK0703).

REFERENCES