Action-Manipulation Attacks Against Stochastic Bandits: Attacks and Defense

Guanlin Liu and Lifeng Lai, *Senior Member, IEEE*

***Abstract*—Due to the broad range of applications of stochastic multi-armed bandit model, understanding the effects of adversarial attacks and designing bandit algorithms robust to attacks are essential for the safe applications of this model. In this paper, we introduce a new class of attacks named action-manipulation attacks. In this class of attacks, an adversary can change the action signal selected by the user. We show that without knowledge of mean rewards of arms, our proposed attack can manipulate Upper Confidence Bound (UCB) algorithm, a widely used bandit algorithm, into pulling a target arm very frequently by spending only logarithmic cost. To defend against this class of attacks, we introduce a novel algorithm that is robust to action-manipulation attacks when an upper bound for the total attack cost is given. We prove that our algorithm has a pseudo-regret upper bounded by $\mathcal{O}(\max\{\log T, A\})$ with a high probability, where T is the total number of rounds and A is the upper bound of the total attack cost.**

***Index Terms*—Stochastic bandits, action-manipulation attack, UCB.**

I. Introduction

In order to develop trustworthy machine learning systems, understanding adversarial attacks on learning systems and correspondingly building robust defense mechanisms have attracted significant recent research interests [2]–[9]. In this paper, we focus on multiple armed bandits (MABs), a simple but very powerful framework of online learning that makes decisions over time under uncertainty. MABs problems have been widely investigated in machine learning and signal processing [10]–[16], and has many applicants in a variety of scenarios such as displaying advertisements [17], articles recommendation [18], cognitive radios [19], [20] and search engines [21], to name a few. In the modern industry-scale applications of MABs models, action decisions, reward signal collection, and policy iterations are normally implemented in a distributed network. When data packets containing the reward signals and action decisions etc are transmitted through the network, an attacker can intercept and modify these data packets to implement adversarial attacks.

Of particular relevance to our work is a line of interesting recent work on online reward-manipulation attacks on stochastic MABs [22]–[24]. In the reward-manipulation attacks, there is an adversary who can change the reward signal from the environment, and hence the reward signal received by the user is not the true reward signal from the environment. In particular, [22] proposes an interesting attack strategy that can force a user, who runs either ϵ-Greedy or Upper Confidence Bound (UCB) algorithm, to select a target arm while only spending effort that grows in logarithmic order. [23] proposes an optimization based framework for offline reward-manipulation attacks. Furthermore, it studies a form of online attack strategy that is effective in attacking any bandit algorithm that has a regret scaling in logarithm order, without knowing what particular algorithm the user is using. [25] considers an attack model where an adversary attacks with a certain probability at each round but its attack value can be arbitrary and unbounded. The paper proposes algorithms that are robust to these types of attacks. [24] considers how to defend against reward-manipulation attacks, a complementary problem to [22], [23]. In particular, [24] introduces a bandit algorithm that is robust to reward-manipulation attacks under certain attack cost, by using a multi-layer approach. [26] introduces another model of adversary setting where each arm is able to manipulate its own reward and seeks to maximize its own expected number of pull count. Under this setting, [26] analyzes the robustness of Thompson Sampling, UCB, and ϵ-greedy under attacks, and proves that all three algorithms achieve a regret upper bound that increases over rounds in a logarithmic order or increases with attack cost in a linear order. This line of reward-manipulation attack has also recently been investigated for contextual bandits in [27], which develops an attack algorithm that can force the bandit algorithm to pull a target arm for a target contextual vector by slightly manipulating rewards in the data.

In this paper, we introduce a new class of attacks on MABs named action-manipulation attack. In the action-manipulation attack, an attacker, sitting between the environment and the user, can change the action selected by the user to another action. The user will then receive a reward from the environment corresponding to the action chosen by the attacker. Compared with the reward-manipulation attacks discussed above, the action-manipulation attack is more difficult to carry out. In particular, as the action-manipulation attack only changes the action, it can impact but does not have direct control of the reward signal, because the reward signal will be a random variable drawn from a distribution depending on the action chosen by the attacker. This is in contrast to reward-manipulation attacks where an attacker has direct control and can change the reward signal to any value.

G. Liu and L. Lai are with Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616. Email: {glnliu,lflai}@ucdavis.edu. The work of G. Liu and L. Lai was supported by National Science Foundation under Grants CCF-1717943, ECCS-1711468, CNS-1824553 and CCF-1908258. This paper has been presented in part in the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing [1].

In order to demonstrate the significant security threat of action-manipulation attacks to stochastic bandits, we propose an action-manipulation attack strategy against the widely used UCB algorithm. We choose to attack the UCB algorithm as it is widely used in practice and has been extensively studied in the literature. The proposed attack strategy aims to force the user to frequently pull a target arm chosen by the attacker. We assume that the attacker does not know the true mean reward of each arm. The assumption that the attacker does not know the mean rewards of arms is necessary for the design of attack strategies, as otherwise the attacker can perform the attack trivially. To see this, with the knowledge of the mean rewards, the attacker knows which arm has the worst mean reward and can perform the following oracle attack: when the user pulls a non-target arm, the attacker changes the arm to the worst arm. This oracle attack makes all non-target arms have expected rewards less than that of the target arm, if the target arm selected by the attacker is not the worst arm. In addition, under this attack, all sublinear-regret bandit algorithms will pull the target arm $\mathcal{O}(T)$ times. However, the oracle attack is not practical. The goal of our work is to develop an attack strategy that has similar performance of the oracle attack strategy without requiring the knowledge of the true mean rewards. When the user pulls a non-target arm, the attacker could decide to attack by changing the action to the possible worst arm. As the attacker does not know the true value of arms, our attack scheme relies on lower confidence bounds (LCB) of the value of each arm in making attack decisions. Correspondingly, we name our attack scheme as LCB attack strategy. Our analysis shows that, if the target arm selected by the attacker is not the worst arm, the LCB attack strategy can successfully manipulate the user to select the target arm almost all the time with an only logarithmic cost. In particular, LCB attack strategy can force the user to pull the target arm $T-\mathcal{O}(\log(T))$ times over T rounds, with the total attack cost being only $\mathcal{O}(\log(T))$. On the other hand, we also show that, if the target arm is the worst arm and the attacker can only incur logarithmic costs, no attack algorithm can force the user to pull the worst arm more than $T-\mathcal{O}(T^{\alpha})$ times. In addition, we study an oracle attack to illustrate the challenges arise for the case where the target arm is the worst arm.

Motivated by the analysis of the action-manipulation attacks and the significant security threat to MABs, we then design a bandit algorithm which can defend against the action-manipulation attacks and still is able to achieve a small regret. The main idea of the proposed algorithm is to bound the maximum amount of offset, in terms of user's estimate of the mean rewards, that can be introduced by the action-manipulation attacks. We then use this estimate of maximum offset to properly modify the UCB algorithm and build specially designed high-probability upper bounds of the mean rewards so as to decide which arm to pull. We name our bandit algorithm as maximum offset upper confidence bound (MOUCB). In particular, our algorithm first pulls every arm a certain of times and then pulls the arm whose modified upper confidence bound is the largest. Furthermore, we prove that MOUCB bandit algorithm has a pseudo-regret upper bounded by $\mathcal{O}(\max\{\log T, A\})$, where T is the total number of rounds and A is an upper bound for the total attack cost. In particular, if A scales as $\log(T)$, MOUCB archives a logarithm pseudo-regret which is same as the regret of UCB algorithm.

Compared with our conference paper [1], this journal paper provides several new contributions: 1) In [1], only attacks are considered. In this journal paper, we also consider how to defend against attacks. In particular, we design a new bandit algorithm MOUCB that is robust to action-manipulation attacks and analyze its regret; 2) We provide detailed discussions of the reason why LCB strategy fails when the target arm is the worst; 3) We introduce a class of oracle attacks on UCB algorithm when the target arm is the worst and analyze the bound of target arm pull count and attack cost; 4) We conduct more comprehensive numerical simulations to illustrate the results obtained in our study.

The remainder of the paper is organized as follows. In Section II, we describe the model. In Section III, we describe the LCB attack strategy and analyze its accumulative attack cost. In Section IV, we propose MOUCB and analyze its regret. In Section V, we provide numerical examples to validate the theoretic analysis. Finally, we offer several concluding remarks in Section VI. The proofs are collected in Appendix.

II. Model

In this section, we introduce our model. We consider the standard multi-armed stochastic bandit problems setting. The environment consists of K arms, with each arm corresponds to a fixed but unknown reward distribution. The bandit algorithm, which is also called "user" in this paper, proceeds in discrete time $t = 1, 2, \ldots, T$, in which T is the total number of rounds. At each round t, the user pulls an arm (or action) $I_t \in \{1, \ldots, K\}$ and receives a random reward r_t drawn from the reward distribution of arm I_t. Denote $\tau_i(t) := \{s : s \leq t, I_s = i\}$ as the set of rounds up to t where the user chooses arm i, $N_i(t) := |\tau_i(t)|$ as the number of rounds that arm i was pulled by the user up to time t and

$$\hat{\mu}_i(t) := N_i(t)^{-1} \sum_{s \in \tau_i(t)} r_s \tag{1}$$

as the empirical mean reward of arm i. The pseudo-regret $\bar{R}(T)$ is defined as

$$\bar{R}(T) = T \max_{\max_{i\in[K]} \mu_i} - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]. \tag{2}$$

The goal of the user is to minimize $\bar{R}(T)$.

In this paper, we introduce a novel adversary setting, in which the attacker sits between the user and the environment. The attacker can monitor the actions of the user and the reward signals from the environment. Furthermore, the attacker can introduce action-manipulation attacks on stochastic bandits. In particular, at each round t, after the user chooses an arm I_t, the attacker can manipulate the user's action by changing I_t to another $I_t^0 \in \{1, \ldots, K\}$. If the attacker decides not to attack,

$I_t^0 = I_t$. Then the environment generates a random reward r_t from the reward distribution of post-attack arm I_t^0. The user and the attacker receive reward r_t from the environment.

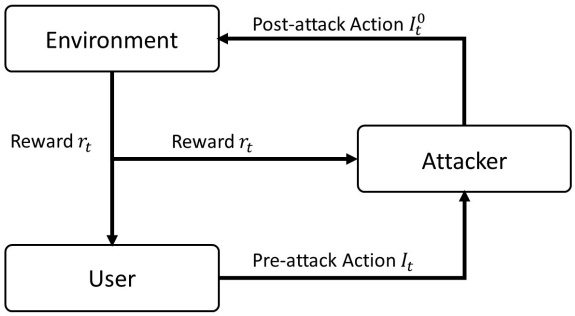

Fig. 1. Action-manipulation attack model

Without loss of generality and for notation convenience, we assume arm K is the "attack target" arm or the target arm. The attacker's goal is to manipulate the user into pulling the target arm very frequently but by making attacks as rarely as possible. Define the set of rounds when the attacker decides to attack as $\mathcal{C} := \{t : t \leq T, I_t^0 \neq I_t\}$. The cumulative attack cost is the total number of rounds where the attacker decides to attack, i.e., $|\mathcal{C}|$.

In this paper, we assume that the reward distribution of arm i follows σ^2-sub-Gaussian distributions with mean μ_i. Denote the true reward vector as $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_K]$. Neither the user nor the attacker knows $\boldsymbol{\mu}$, but σ^2 is known to both the user and the attacker. We note that the assumption that the attacker does not know $\boldsymbol{\mu}$ is only necessary for Section III, in which we design attack strategies. We do not use this assumption in Section IV where we design defense strategies. Define the difference of mean value of arm i and j as $\Delta_{i,j} = \mu_i - \mu_j$. Furthermore, we refer to the best arm as $i_O = \arg\max_i \mu_i$ and the worst arm as $i_W = \arg\min_i \mu_i$.

In Section III, the assumption that the attacker does not know $\boldsymbol{\mu}$ is important. If the attacker knows these values, the attacker can adopt a trivial oracle attack scheme: whenever the user pulls a non-target arm I_t, the attacker changes I_t to the worst arm i_W. Assuming that the target arm is not the worst, it is easy to show that, if the user uses a bandit algorithm that has a regret upper bounded of $\mathcal{O}(\log(T))$ when there is no attack, the oracle attack scheme can force the user to pull the target arm $T - \mathcal{O}(\log(T))$ times, using a cumulative cost $|\mathcal{C}| = \mathcal{O}(\log(T))$. However, the oracle attack scheme is not practical when the true reward vector is unknown. In this paper, we will first design an effective attack scheme, which does not assume the knowledge of true reward vector and nearly matches the performance of the oracle attack scheme, to attack the UCB algorithm. We will then design a new bandit algorithm that is robust against the action-manipulation attack.

The action-manipulation attack considered here is different from reward-manipulation attacks introduced by interesting recent work [22], [23], where the attacker can change the reward signal from the environment. In the setting considered in [22], [23], the attacker can change the reward signal r_t from the environment to an arbitrary value chosen by the attacker. Correspondingly, the cumulative attack cost in [22], [23] is defined to be the sum of the absolute value of the changes on the reward. Compared with the reward-manipulation attacks discussed above, the action-manipulation attack is more difficult to carry out. In particular, as the action-manipulation attack only changes the action, it can impact but does not have direct control of the reward signal, which will be a random variable drawn from a distribution depending on the action chosen by the attacker. This is in contrast to reward-manipulation attacks where an attacker can change the reward to any value.

III. Attack on UCB and Cost Analysis

In this section, we use UCB algorithm as an example to illustrate the effects of action-manipulation attack. We will introduce LCB attack strategy on the UCB bandit algorithm and analyze the cost.

A. Attack strategy

UCB algorithm [28] is one of the most popular bandit algorithm. In UCB algorithm, the user initially pulls each of the K arms once in the first K rounds. After that, the user chooses arms according to

$$I_t = \arg\max_i \left\{ \hat{\mu}_i(t-1) + 3\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \right\}. \tag{3}$$

Under the action-manipulation attack, as the user does not know that r_t is generated from arm I_t^0 instead of I_t, the empirical mean $\hat{\mu}_i(t)$ computed using (1) is not a proper estimate of the true mean reward of arm i anymore. On the other hand, the attack is able to obtain a good estimate of μ_i by

$$\hat{\mu}_i^0(t) := N_i^0(t)^{-1} \sum_{s \in \tau_i^0(t)} r_s, \tag{4}$$

where $\tau_i^0(t) := \{s : s \leq t, I_s^0 = i\}$ is the set of rounds up to t when the attacker changes an arm to arm i, and $N_i^0(t) = |\tau_i^0(t)|$ is the number of pulls of post-attack arm i up to round t. This information gap provides a chance for attack. In this section, we assume that the target arm is not the worst arm, i.e., $\mu_K > \mu_{i_W}$. We will discuss the case where the target arm is the worst arm in Section III-C.

The proposed attack strategy works as follows. In the first K rounds, the attacker does not attack. After that, at round t, if the user chooses a non-target arm I_t, the attacker changes it to arm I_t^0 that has the smallest lower confidence bound (LCB):

$$I_t^0 = \arg\min_i \left\{ \hat{\mu}_i^0(t-1) - \mathbf{CB}\left(N_i^0(t-1), \delta\right) \right\}, \tag{5}$$

where

$$\mathbf{CB}(N, \delta) = \sqrt{\frac{2\sigma^2}{N} \log \frac{\pi^2 K N^2}{3\delta}}. \tag{6}$$

Here $\delta \in (0,1)$ is a parameter that is related to the probability statements in the analytical results presented in Section III-B. We call our scheme as LCB attack strategy. Note that the form of (6) is slightly different from typical form used in UCB algorithms. We choose to use this form for the simplicity of proofs. If at round t the user chooses the target arm, the attacker does not attack. Thus the cumulative attack cost of our LCB attack scheme is equal to the total of times when the non-target arms are selected by the user. The algorithm is summarized in Algorithm 1.

Algorithm 1: LCB attack strategy on UCB algorithm

Input:
The user's bandit algorithm namely UCB algorithm, target arm K

for $t = 1, 2, \dots$ **do**
  The user chooses arm I_t to pull according to UCB algorithm (3).
  if $I_t = K$ **then**
    The attacker does not attack, and $I_t^0 = I_t$.
  else
    The attacker attacks and changes arm I_t to I_t^0 chosen according to (5).
  end if
  The environment generates reward r_t from arm I_t^0.
  The attacker and the user receive r_t.
end for

Here, we highlight the main idea why LCB attack strategy works. As discussed in Section II, if the attacker knows which arm is the worst, the attacker can simply change the action to the worst arm when the user pulls the non-target arm. The main idea of the attack scheme is to estimate the mean of each arm, and change the non-target arm to the arm whose lower confidence bound is the smallest. Effectively, this will almost always change the non-target arm to the worst arm. More formally, for $i \neq K$, we will show that this attack strategy will ensure that $\hat{\mu}_i$ computed using (1) by the user converges to μ_{i_W}. On the other hand, as the attacker does not attack when the user selects K, $\hat{\mu}_K$ computed by the user will still converge to the true mean μ_K with N_K increasing. Because the assumption that the target arm is not the worst, which implies that $\mu_K > \mu_{i_W}$, $\hat{\mu}_i$ could be smaller than $\hat{\mu}_K$. Then the user will rarely pull the non-target arms as $\hat{\mu}_i$ is smaller than $\hat{\mu}_K$. Hence, the attack cost would also be small. The rigorous analysis of the cost will be provided in Section III-B.

B. Cost analysis

To analyze the cost of the proposed scheme, we need to track $\hat{\mu}_i^0(t)$, the estimate obtained by the attacker using (4), and $\hat{\mu}_i(t)$, the estimate obtained by the user using (1).

The analysis of $\hat{\mu}_i^0(t)$ is relatively simple, as the attacker knows which arm is truly pulled and hence $\hat{\mu}_i^0(t)$ is the true estimate of the mean of arm i. Define event

$$\mathcal{E}_1 := \{\forall i, \forall t > K : |\hat{\mu}_i^0(t) - \mu_i| < \mathbf{CB}(N_i^0(t), \delta)\}. \quad (7)$$

Roughly speaking, event $\mathcal{E}_1$ is the event that the empirical mean computed by the attacker using (4) is close to the true mean. The following lemma, proved in [22], shows that the attacker can accurately estimate the average reward to each arm.

Lemma 1. *(Lemma 1 in [22]) For $\delta \in (0,1)$, $\mathbb{P}(\mathcal{E}_1) > 1 - \delta$.*

The analysis of $\hat{\mu}_i(t)$ computed by the user is more complicated. When the user pulls arm i, because of the action-manipulation attacks, the random rewards may be drawn from different reward distributions. Define $\tau_{i,j}(t) := \{s : s \leq t, I_s = i \text{ and } I_s^0 = j\}$ as the set of rounds up to t when the user chooses arm i and the attacker changes it to arm j. Lemma 2 shows a high-probability confidence bounds of $\hat{\mu}_{i,j}(t) := N_{i,j}(t)^{-1} \sum_{s \in \tau_{i,j}(t)} r_s$, the empirical mean rewards of a part of arm i whose post-attack arm is j, where $N_{i,j}(t) := |\tau_{i,j}(t)|$. Define event

$$\mathcal{E}_2 := \left\{\forall i, \forall j, \forall t > K : |\hat{\mu}_{i,j}(t) - \mu_j| < \mathbf{CB}\left(N_{i,j}(t), \frac{\delta}{K}\right)\right\}. \quad (8)$$

Lemma 2. *For $\delta \in (0,1)$, $\mathbb{P}(\mathcal{E}_2) > 1 - \delta$.*

Proof. Please refer to Appendix A. □

Although r_s in (1), used to calculate $\hat{\mu}_i(t)$, may be drawn from different reward distributions, we can build a high-probability bound of $\hat{\mu}_i(t)$ with the help of Lemma 2.

Lemma 3. *Under event $\mathcal{E}_2$, for all arm i and all $t > K$, we have*

$$\left|\hat{\mu}_i(t) - \frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \mu_{I_s^0}\right| < \boldsymbol{CB}\left(\frac{N_i(t)}{K}, \frac{\delta}{K}\right), \quad (9)$$

Proof. Please refer to Appendix B. □

Under events $\mathcal{E}_1$ and $\mathcal{E}_2$, we can build a connection between $\hat{\mu}_i(t)$ and μ_{i_W}. In the proposed LCB attack strategy, the attacker explores and exploits the worst arm by a lower confidence bound method. Thus, when the user pulls a non-target arm, the attacker changes it to the worst arm at most of rounds, which means that for all $i \neq K$, $\hat{\mu}_i(t)$ will converge to μ_{i_W} as $N_i(t)$ increases. Lemma 4 shows the relationship between $\hat{\mu}_i(t)$ and μ_{i_W}.

Lemma 4. *Under events $\mathcal{E}_1$ and $\mathcal{E}_2$, using LCB attack strategy 1, we have*

$$\begin{aligned}\hat{\mu}_i(t) \leq \mu_{i_W} &+ \frac{1}{N_i(t)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K t^2}{3\delta} \\ &+ \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}}, \forall i, t.\end{aligned} \quad (10)$$

Proof. Please refer to Appendix C. □

Lemma 4 shows an upper bound of the empirical mean reward of pre-attack arm i, for all arm $i \neq K$. Our main result is the following upper bound on the attack cost $|\mathcal{C}|$.

Theorem 1. *With probability at least $1-2\delta$, when $T \geq \left(\frac{\pi^2 K}{3\delta}\right)^{\frac{2}{5}}$, using LCB attack strategy specified in Algorithm 1, the attacker can manipulate the user into pulling the target arm in at least $T-|\mathcal{C}|$ rounds, with an attack cost*

$$|\mathcal{C}| \leq \frac{K-1}{4\Delta_{K,i_W}^2}\left(3\sigma\sqrt{\log T}+\sqrt{2\sigma^2 K\log\frac{\pi^2T^2}{3\delta}}\right. \\ +\left(\left(3\sigma\sqrt{\log T}+\sqrt{2\sigma^2 K\log\frac{\pi^2T^2}{3\delta}}\right)^2\right. \\ \left.\left.+4\Delta_{K,i_W}\sum_{j\neq i_W}\frac{8\sigma^2}{\Delta_{j,i_W}}\log\frac{\pi^2KT^2}{3\delta}\right)^{\frac{1}{2}}\right)^2. \tag{11}$$

Proof. Please refer to Appendix D. □

The expression of the cost bound in Theorem 1 is complicated. The following corollary provides a simpler bound that is more explicit and interpretable.

Corollary 1. *Under the same assumptions in Theorem 1, the total attack cost $|\mathcal{C}|$ of Algorithm 1 is upper bounded by*

$$\mathcal{O}\left(\frac{K\sigma^2\log T}{\Delta_{K,i_W}^2}\left(K+\sum_{j\neq i_W}\frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}+\sqrt{K\sum_{j\neq i_W}\frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}}\right)\right), \tag{12}$$

and the total number of target arm pulls is $T-|\mathcal{C}|$.

From Corollary 1, we can see that the attack cost scales as $\log T$. Two important constants $\frac{\sigma}{\Delta_{K,i_W}}$ and $\sum_{j\neq i_W}\frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}$ have impact on the prelog factor. In Section V, we provide some numerical examples to illustrate the effects of these two constants on the attack cost.

In the above analysis, the attacker has only one target arm and aims to force the user to pull it. We can extend our algorithm to the scenario where there is a set of target arms and the attacker aims to manipulate the user into pulling any one of them very frequently. For this case, we need an assumption that the worst arm is not in the target set. When the user pulls a target arm, the adversary does not attack. When the user pulls a non-target arm, the LCB attack strategy can change it to the worst arm at most of rounds. In this way, the estimate of any non-target arm could be smaller than the estimate of any target arm. As the result, the user will rarely pull the non-target arms and pull arms in the target set very frequently. The attack cost also scales as $\log(T)$.

C. Attacks fail when the target arm is the worst arm

One weakness of our LCB attack strategy is that the attack target arm is necessarily a non-worst arm. In the LCB attack strategy, the attacker can not force the user to pull the worst arm very frequently by spending only logarithmic cost. The main reason is that, when the target arm is the worst, the average reward of each arm is larger or equal to that of the target arm. As the result, our attack scheme is not able to ensure that the target arm has a higher expected reward than the user's estimate of the rewards of other arms. In fact, the following theorem shows that all action-manipulation attack can not manipulate the UCB algorithm into pulling the worst arm more than $T-\mathcal{O}(\log(T))$ by spending only logarithmic cost.

Theorem 2. *Let $\delta < \frac{1}{2}$. Suppose the attack cost is limited by $\mathcal{O}(\log(T))$, there is no attack that can force the UCB algorithm to pick the worst arm more than $T-\mathcal{O}(T^\alpha)$ times with probability at least $1-\delta$, in which $\alpha \leq 1$.*

Proof. Please refer to Appendix E. □

This theorem shows a contrast between the case where the target arm is not the worst arm and the case where the target arm is the worst arm. If the target arm is not the worst arm, our scheme is able to force the user to pick the target arm $T-\mathcal{O}(\log(T))$ times with only logarithmic cost. On the other hand, if the target arm is the worst, Theorem 2 shows that there is no attack strategy that can force the user to pick the worst arm more than $T-\mathcal{O}(T^\alpha)$ times while incurring only logarithmic cost.

In the proof of Theorem 2, we do not use the assumption on whether the attacker knows the true underlying mean vector or not. Hence this theorem is also valid even when the attacker knows the true underlying mean vector and can carry out an oracle attack. To further illustrate the challenges arise for the case where the target arm is the worst arm, we now study the oracle attack for this case. Even though the attacker knows the true underlying mean vector, it is difficult for him to carry out the attack. The main reason is that, since the target arm is the worst arm, in order to make this arm appears to be better to the user, the attacker now needs to attack even when the user pulls the target arm, i.e., to change it to the best arm. Hence the attack has two parts: 1) when the user pulls a non-target arm, the attacker changes the arm to the worst arm; 2) when the user pulls the target arm, the attacker changes the arm to the best arm sometimes. We set the number of rounds that the attacker change the target arm to the best arm as C_K. So the attack cost has two parts: the number of rounds where the user pulls a non-target arm and C_K. The following proposition analyze the cost of this oracle attack.

Proposition 1. *With probability at least $1-\delta$, when $T > \left(\frac{\pi^2K^2}{12\delta}\right)^4$, given the number of rounds that the attacker change the target arm to the best arm as C_K, the oracle attack can manipulate the user into pulling the target arm that is the worst arm in at most*

$$T-\min\left(\frac{\frac{1}{4}(K-1)\sigma^2T^2\log\frac{T}{K}}{\left(KC_K\Delta_{i_O,K}+6\sigma\sqrt{KT\log\frac{T}{K}}\right)^2},\frac{T(K-1)}{K}\right) \tag{13}$$

rounds, with an attack cost $|\mathcal{C}|$ at least

$$C_K + \min\left(\frac{\frac{1}{4}(K-1)\sigma^2 T^2 \log\frac{T}{K}}{\left(KC_K\Delta_{i_O,K} + 6\sigma\sqrt{KT\log\frac{T}{K}}\right)^2}, \frac{T(K-1)}{K}\right). \tag{14}$$

Proof. Please refer to Appendix F. □

Compared with the performance of LCB attacks for the cases when the target arm is the worst arm, the oracle attack for the case when the target arm is the worst arm requires significantly more attack cost to achieve the similar performance. According to Proposition 1, in order to manipulate the user into pulling the target arm in $T - O(\log T)$ rounds, the C_K should scale as T. The attack is extremely ineffective, as now the attack cost scales with T. Furthermore, from (14), to minimize the cost, we need to set

$$C_K = \frac{1}{K\Delta_{i_O,K}}\left(\left(\frac{1}{2}K(K-1)\Delta_{i_O,K}\sigma^2T^2\log\frac{T}{K}\right)^{\frac{1}{3}} - 6\sigma\sqrt{KT\log\frac{T}{K}}\right) \tag{15}$$

which scales as $\Omega\left(T^{\frac{2}{3}}(\log T)^{\frac{2}{3}}\right)$. Hence, for the case where the target arm is the worst arm, the minimal attack cost of the oracle attack is large. There is no effective attacks when the target arm is the worst arm.

IV. Robust algorithm and regret analysis

The results in Section III expose a significant security threat of the action-manipulation attacks on MABs. Under only $\mathcal{O}(\log(T))$ times of attacks carried out using the proposed LCB strategy, the UCB algorithm will almost always pull the target arm selected by the attacker. Although there are some defense algorithms [24] and universal best arm identification schemes [29] for stochastic or adversarial bandit, they do not apply to the action-manipulation attack setting. This motivates us to design a new bandit algorithm that is robust against action-manipulation attacks. In this section, we propose such a robust bandit algorithm and analyze its regret.

A. Robust Bandit algorithm

In this section, we assume that a valid upper bound A for the cumulative attack cost $|\mathcal{C}|$ is known for the user, although the user does not have to know the exact cumulative attack cost $|\mathcal{C}|$. A does not need to be constant, it can scale with T. In other words, for a given A, our proposed algorithm is robust to all action-manipulation attacks with a cumulative attack cost $|\mathcal{C}| < A$. This assumption is reasonable, as if the cost is unbounded, it will not be possible to design a robust scheme.

We first introduce some notation. Denote $\mathbf{N}(t-1) := (N_1(t-1), \ldots, N_K(t-1))$ as the vector counting how many times each action has been taken by the user, and $\hat{\boldsymbol{\mu}}(t-1) = (\hat{\mu}_1(t-1), \ldots, \hat{\mu}_K(t-1))$ as the vector of the sample means computed by the user. The proposed algorithm is a modified UCB method by taking the maximum possible mean estimate offset due to attack into consideration. We name our scheme as maximum offset UCB (MOUCB).

The proposed MOUCB works as follows. In the first $2AK$ rounds, MOUCB algorithm pulls each arm $2A$ times. After that, at round t, the user chooses an arm I_t by a modified UCB method:

$$I_t = \arg\max_a \{\hat{\mu}_a(t-1) + \beta(N_a(t-1)) + \gamma(\hat{\boldsymbol{\mu}}(t-1), \mathbf{N}(t-1))\}, \tag{16}$$

where

$$\gamma(\hat{\boldsymbol{\mu}}(t-1), \mathbf{N}(t-1)) = \frac{2A}{N_a(t-1)} \max_{i,j}\{\hat{\mu}_i(t-1) - \hat{\mu}_j(t-1) + \beta(N_i(t-1)) + \beta(N_j(t-1))\},$$

and

$$\beta(N) = \mathbf{CB}\left(\frac{N}{K}, \frac{\delta}{K}\right) = \sqrt{\frac{2\sigma^2 K}{N}\log\frac{\pi^2 N^2}{3\delta}}. \tag{17}$$

The algorithm is summarized in Algorithm 2.

Algorithm 2: Proposed MOUCB bandit algorithm

Input:
A valid upper bound A for the cumulative attack cost.

for $t = 1, 2, \ldots$ **do**
  if $t \le 2AK$ **then**
    The user pulls the arm whose pull count is the smallest, i.e. $I_t = \arg\min_i N_i(t-1)$.
  else
    The user chooses arm I_t to pull according (16).
  end if
  if The attacker decides to attack **then**
    The attacker attacks and changes I_t to I_t^0.
  else
    The attacker does not attack and $I_t^0 = I_t$.
  end if
  The environment generates reward r_t from arm I_t^0.
  The attacker and the user receive r_t.
end for

Compared with the original UCB algorithm in (3), the main difference is the additional term $\gamma(\hat{\boldsymbol{\mu}}(t-1), \mathbf{N}(t-1))$ in (16). We now highlight the main idea why our bandit algorithm works and the role of this additional term. In particular, in the standard multi-armed stochastic bandit problem, $\hat{\mu}_i(t)$ is a proper estimation of μ_i, the true mean reward of arm i. However, under the action-manipulation attacks, as the user does not know which arm is used to generate r_t, $\hat{\mu}_i(t)$ is not a proper estimate of the true mean reward anymore. However, we can try to find a good bound of the true mean reward. If we know Δ_{i_O,i_W}, the reward difference between the optimal arm and the worst arm, we can describe the maximum offset

of the mean rewards caused by the attack. In particular, we have

$$\mu_i - \frac{A}{N_i(t)}\Delta_{i_O,i_W} \leq \frac{1}{N_i(t)} \sum_{s\in\tau_i(t)} \mu_{I_s^0} \leq \mu_i + \frac{A}{N_i(t)}\Delta_{i_O,i_W}, \tag{18}$$

which implies

$$\mu_i \leq \frac{A}{N_i(t)}\Delta_{i_O,i_W} + \frac{1}{N_i(t)} \sum_{s\in\tau_i(t)} \mu_{I_s^0}. \tag{19}$$

In (19), the first term in the right hand side is the maximum offset that an attacker can introduce regardless of the attack strategy. The second term in the right hand side is related to the mean estimated by the user. In particular, under event $\mathcal{E}_2$, as shown in Lemma 3, we have

$$\frac{1}{N_i(t)} \sum_{s\in\tau_i(t)} \mu_{I_s^0} < \hat{\mu}_i(t) + \beta(N_i(t)). \tag{20}$$

Hence, regardless the attack strategy, we have a upper confidence bound on μ_i:

$$\mu_i \leq \hat{\mu}_i(t) + \frac{A}{N_i(t)}\Delta_{i_O,i_W} + \beta(N_i(t)). \tag{21}$$

In our case, however, Δ_{i_O,i_W} is also unknown. In our algorithm, we obtain a high-probability bound on Δ_{i_O,i_W}:

$$\Delta_{i_O,i_W} \leq 2\max_{i,j}\left\{\hat{\mu}_i - \hat{\mu}_j + \beta\left(N_i(t)\right) + \beta\left(N_j(t)\right)\right\}, \tag{22}$$

which will proved in Lemma 5 below. Now, the second term of (21) becomes $\gamma(\hat{\boldsymbol{\mu}}(t-1), \mathbf{N}(t-1))$ if we replace Δ_{i_O,i_W} with the bound (22), and we obtain our final algorithm.

The design of robust algorithms under the adversarial setup can be alternatively viewed as a MABs problem with limited number of mean changes. When the user pulls a single arm, the rewards he receives are drawn from different reward distributions with different means. The means are varying with time because of the manipulation of the attacker. The means change between only K fixed values. In our setting, if the attacker does not decide to attack, the arm chosen by the user does not change and the mean does not change. In this sense, the attack cost is the number of rounds when the mean is different from the initial value. In most rounds, each arm corresponds to a fixed but unknown reward distribution. However, in at most A rounds, the mean of each arm is varying between $K-1$ fixed values.

B. Regret analysis

Lemma 5 shows a bound of Δ_{i_O,i_W}, the maximum reward difference between any two arms, under event $\mathcal{E}_2$.

Lemma 5. *For $\delta \leq \frac{1}{3}$, $t > 2AK$ and under event $\mathcal{E}_2$, MOUCB algorithm have*

$$\begin{aligned}\Delta_{i_O,i_W} &\leq 2\max_{i,j}\left\{\hat{\mu}_i - \hat{\mu}_j + \beta\left(N_i(t)\right) + \beta\left(N_j(t)\right)\right\} \\ &\leq 2\Delta_{i_O,i_W} + 8\sqrt{\frac{\sigma^2 K}{A}\log\frac{4\pi^2A^2}{3\delta}}.\end{aligned} \tag{23}$$

Proof. Please refer to Appendix G. □

Using Lemma 5, we now bound the regret of Algorithm 2.

Theorem 3. *Let A be an upper bound on the total attack cost $|\mathcal{C}|$. For $\delta \leq \frac{1}{3}$ and $T \geq 2AK$, MOUCB algorithm has pseudo-regret $\bar{R}(T)$*

$$\begin{aligned}\bar{R}(T) \leq \sum_{a\neq i_O} \max\Bigg\{ &\frac{8\sigma^2 K}{\Delta_{i_O,a}}\log\frac{\pi^2T^2}{3\delta}, A\left(\Delta_{i_O,a}\right. \\ &\left. +2\Delta_{i_O,i_W} + 8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right)\Bigg\},\end{aligned} \tag{24}$$

with probability at least $1-\delta$.

Proof. Please refer to Appendix H. □

Theorem 3 reveals that our bandit algorithm is robust to the action-manipulation attacks. If the total attack cost is bounded by $\mathcal{O}(\log T)$, the pseudo-regret of MOUCB bandit algorithm is still bounded by $\mathcal{O}(\log T)$. This is in contrast with UCB, for which we have shown that the pseudo-regret is $\mathcal{O}(T)$ with attack cost $\mathcal{O}(\log T)$ in Section III. If the total attack cost is up to $\Omega(\log T)$, the pseudo-regret of MOUCB bandit algorithm is bounded by $\mathcal{O}(A)$, which is linear in A. Note that in the design of defense strategy, we do not assume what the attack strategy is. MOUCB can defend against both LCB attacks and oracle attacks. In fact, MOUCB is robust to all action-manipulation attacks, as long as the total attack cost is smaller than A. In a sense, A can be viewed as a parameter chosen by the user to strike a balance between performance and robustness against attacks: the larger the value A is, the larger class of attacks the user can defend against, but with the cost of a larger regret.

V. Numerical results

In this section, we provide numerical examples to illustrate the analytical results obtained. In our simulation, the bandit has 10 arms. The rewards distribution of arm i is $\mathcal{N}(\mu_i, \sigma)$. The attacker's target arm is K. We let $\delta = 0.05$. We then run the experiment for 20 trials and in each trial we run $T = 10^7$ rounds.

A. LCB attack strategy

We first illustrate the impact of the proposed LCB attack strategy on UCB algorithm.

In Figure 2, we fix $\sigma = 0.1$ and $\Delta_{K,i_W} = 0.1$ and compare the number of rounds at which the target arm is pulled with and without attack. In this experiment, the mean rewards of all arms are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.1, and 0.2 respectively. Arm K is not the worst arm, but its average reward is lower than most arms. The results are averaged over 20 trials. The attacker successfully manipulates the user into pulling the target arm very frequently.

In Figure 3, in order to study how $\frac{\sigma}{\Delta_{K,i_W}}$ affects the attack cost, we fix $\Delta_{K,i_W} = 0.1$ and set σ as 0.1, 0.3 and 0.5 respectively. The mean rewards of all arms are the same as above. From the figure, we can see that as $\frac{\sigma}{\Delta_{K,i_W}}$ increases, the attack cost increases. In addition, as predicted in our analysis,

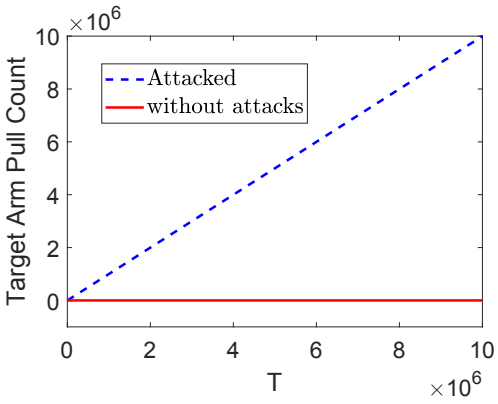

Fig. 2. Number of rounds the target arm was pulled

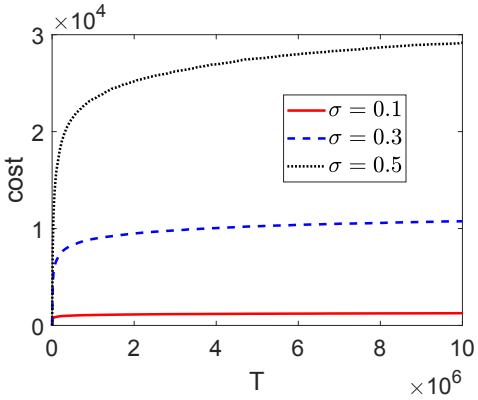

Fig. 3. Attack cost vs $\frac{\sigma}{\Delta_{K,i_W}}$

the attack cost increases with T, the total number of rounds, in a logarithmic order.

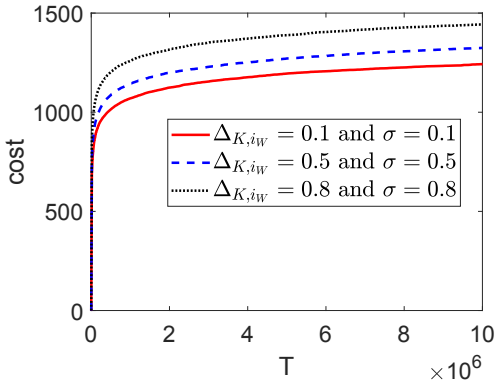

Fig. 4. Attack cost vs $\sum_{j\neq i_W} \frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}$

Figure 4 illustrates how $\sum_{j\neq i_W} \frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}$ affects the attack cost. In this experiment, we fix $\frac{\sigma}{\Delta_{K,i_W}} = 1$ and set Δ_{K,i_W} as 0.2, 0.6 and 0.9 respectively. The mean rewards of all arms are the same as above. The figure illustrates that, as $\sum_{j\neq i_W} \frac{\Delta_{K,i_W}}{\Delta_{j,i_W}}$ increases, the attack cost also increases. This is consistent with our analysis in Corollary 1.

B. MOUCB bandit algorithm

We now illustrate the effectiveness of MOUCB bandit algorithm.

In this experiment, we use the similar setting as in the simulation of the LCB attack scheme. The mean rewards of all arms are set to be 1.0, 0.8, 0.9, 0.5, 0.2, 0.3, 0.1, 0.4, 0.7, and 0.6 respectively. The total attack cost $|\mathcal{C}|$ is limited by 2000. A given valid upper bound for total attack cost is $A = 3000$. The results are averaged over 20 trials.

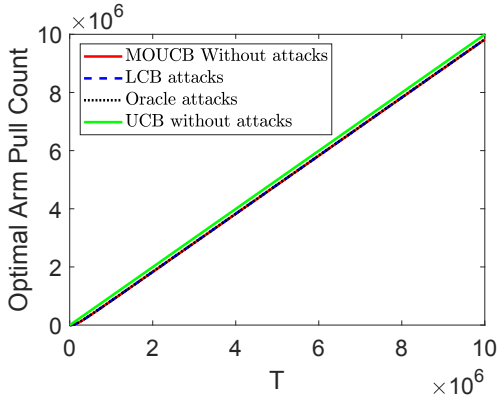

Fig. 5. Comparison of number of rounds the optimal arm was pulled

In Figure 5, we simulate MOUCB algorithm with two different attacks, and compare the numbers of rounds when the optimal arm is pulled under these attacks. The first attack is the LCB attack discussed in Section III. The second attack is the oracle attack, in which the attacker knows the true mean reward of arms and implements the oracle attacks that change any non-target arm to a worst arm (see the discussion in Section II). For comparison purposes, we also add the curve for MOUCB under no attack, and the curve for UCB under no attack. The results show that, even under the oracle attack, the proposed MOUCB bandit algorithm achieves almost the same performance as the UCB without attack.

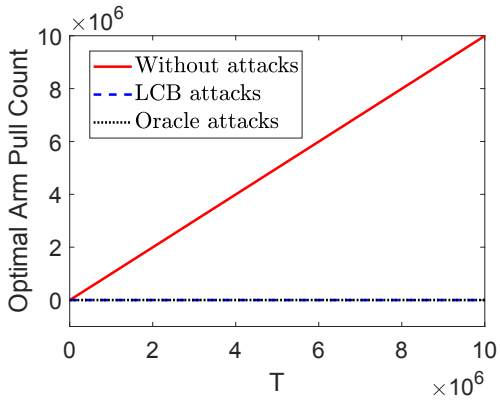

Fig. 6. Number of rounds the optimal arm was pulled using UCB algorithm

To further compare the performance of UCB and MOUCB, in Figure 6, we illustrate the performance of UCB algorithm for the three scenarios discussed above: under LCB attack, under oracle attack and under no attack. The results show that both LCB and oracle attacks can successfully manipulates the UCB algorithm into pulling a non-optimal arm very frequently, as the curves for the LCB attack and oracle attack are far away from the curve for no attack. This is in sharp contrast with the situation for MOUCB algorithm shown in Figure 5, where the all curves are almost identical.

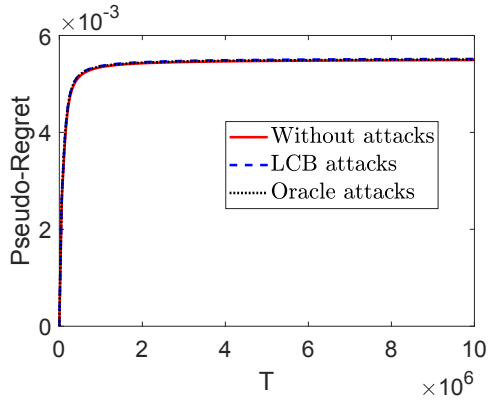

Fig. 7. Pseudo-regret of MOUCB algorithm

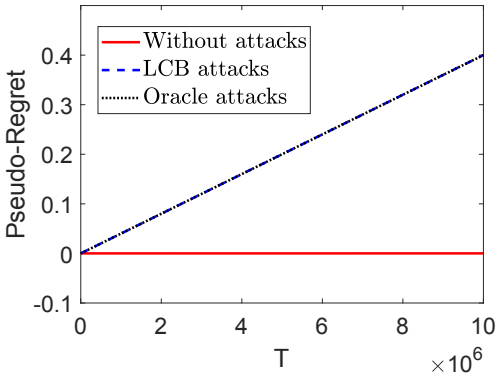

Fig. 8. Pseudo-regret of UCB algorithm

Figure 7 and Figure 8 illustrate the pseudo-regret of MOUCB bandit algorithm and UCB bandit algorithm respectively. In Figure 7, as predicted in our analysis, MOUCB algorithm archives logarithmic pseudo-regrets under both LCB attacks and the oracle attacks. Furthermore, the curves under both attacks are very close to that of the case without attacks. However, as shown in Figure 8, the pseudo-regret of UCB grows linearly under both attacks, while grows logarithmically under no attack. The figures again show that UCB is vulnerable to action-manipulation attacks while the proposed MOUCB is robust to the attacks (even for oracle attacks).

VI. Conclusion

In this paper, we have introduced a new class of attacks on stochastic bandits: action-manipulation attacks. We have analyzed the attack against the UCB algorithm and proved that the proposed LCB attack scheme can force the user to almost always pull a non-worst arm with only logarithm effort. To defend against this type of attacks, we have further designed a new bandit algorithm MOUCB that is robust to action-manipulation attacks. We have analyzed the regret of MOUCB under any attack with bounded cost, and have showed that the proposed algorithm is robust to the action-manipulation attacks.

In terms of future work, it is of interest to investigate robustness of bandit algorithms that are designed under the Bayesian setup. The additional randomness brought by the Bayesian framework may render the algorithms more resistant to attacks.

Appendix A
Proof of Lemma 2

The proof is similar with the proof of Lemma 1 that was proved in [22]. Let $\{X_j\}_{j=1}^{\infty}$ be a sequence of $i.i.d$ σ^2-sub-Gaussian random variables with mean μ. Let $\hat{\mu}^0(t) = \frac{1}{N(t)}\sum_{j=1}^{N(t)} X_j$. By Hoeffding's inequality.

$$\mathbb{P}(|\hat{\mu}^0(t) - \mu| \geq \eta) \leq 2\exp\left(-\frac{N(t)\eta^2}{2\sigma^2}\right). \tag{25}$$

In order to ensure that $\mathcal{E}_2$ holds for all arm i, all arm j and all pull counts $N = N_{i,j}(t)$, we set $\delta_{i,j,N} := \frac{6\delta}{\pi^2 K^2 N^2}$. We have

$$\begin{aligned}&\mathbb{P}\left(\exists i, \exists j, \exists N : |\hat{\mu}_{i,j}(t) - \mu_j| \geq \sqrt{\frac{2\sigma^2}{N}\log\frac{\pi^2K^2N^2}{3\delta}}\right)\\ &\leq \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{N=1}^{\infty}\delta_{i,j,N} = \delta.\end{aligned} \tag{26}$$

Appendix B
Proof of Lemma 3

According to event $\mathcal{E}_2$, we have

$$\begin{aligned}&\left|\hat{\mu}_i(t) - \frac{1}{N_i(t)}\sum_{s\in\tau_i(t)}\mu_{I_s^0}\right|\\ =&\left|\sum_{j=1}^{K}\frac{N_{i,j}(t)}{N_i(t)}\left(\hat{\mu}_{i,j}(t) - \mu_j\right)\right|\\ \leq&\sum_{j=1}^{K}\frac{N_{i,j}(t)}{N_i(t)}\left|\hat{\mu}_{i,j}(t) - \mu_j\right|\\ <&\frac{1}{N_i(t)}\sum_{j=1}^{K}\sqrt{2\sigma^2 N_{i,j}(t)\log\frac{\pi^2K^2(N_{i,j}(t))^2}{3\delta}}.\end{aligned} \tag{27}$$

Define a function $f(N) = \sqrt{2\sigma^2 N \log \frac{\pi^2 K^2 N^2}{3\delta}} : (0, +\infty) \to \mathbb{R}$, and we have

$$\begin{aligned} f''(N) =& \frac{\partial^2}{\partial N^2} \sqrt{2\sigma^2 N \log \frac{\pi^2 K^2 N^2}{3\delta}} \\ =& -\frac{\left(2\sigma^2 \log \frac{\pi^2 K^2 N^2}{3\delta}\right)^2 + 16\sigma^4}{4\left(2\sigma^2 N \log \frac{\pi^2 K^2 N^2}{3\delta}\right)^{\frac{3}{2}}} \\ <& 0, \end{aligned} \tag{28}$$

when $N \geq 1$.

Hence f is strictly concave when $N \geq 1$, and according to the property of the concave function,

$$\sum_{j=1}^{K} f(N_{i,j}(t)) < K f\left(\frac{1}{K}\sum_{j=1}^{K} N_{i,j}(t)\right) = K f\left(\frac{N_i(t)}{K}\right). \tag{29}$$

Thus,

$$\begin{aligned} &\left|\hat{\mu}_i(t) - \frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \mu_{I_s^0}\right| \\ <& \frac{1}{N_i(t)} K \sqrt{2\sigma^2 \frac{N_i(t)}{K} \log \frac{\pi^2 K^2 \left(\frac{N_i(t)}{K}\right)^2}{3\delta}} \\ =& \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}}. \end{aligned} \tag{30}$$

Appendix C
Proof of Lemma 4

The LCB attack scheme uses lower confidence bound to exploit the worst arm, so we need to prove that the attacker's pull counts of all non-worst arms should be limited at round t.

Consider the case that in round $t+1$, the user chooses a non-target arm $I_{t+1} = i \neq K$ and the attacker changes it to a non-worst arm $I_{t+1}^0 = j \neq i_W$. On one hand, under event $\mathcal{E}_1$, we have

$$\begin{aligned} &\hat{\mu}_{i_W}^0(t) - \mu_{i_W} < \mathbf{CB}(N_{i_W}^0(t), \delta), \\ \text{and} \quad &\hat{\mu}_j^0(t) - \mu_j > -\mathbf{CB}(N_j^0(t), \delta). \end{aligned} \tag{31}$$

On the other hand, according to the attack scheme, it must be the case that

$$\hat{\mu}_{i_W}^0(t) - \mathbf{CB}(N_{i_W}^0(t), \delta) > \hat{\mu}_j^0(t) - \mathbf{CB}(N_j^0(t), \delta), \tag{32}$$

which is equivalent to

$$\mathbf{CB}(N_j^0(t), \delta) > \hat{\mu}_j^0(t) - (\hat{\mu}_{i_W}^0(t) - \mathbf{CB}(N_{i_W}^0(t), \delta)). \tag{33}$$

Combining (33) with (31), we have

$$\begin{aligned} &\mathbf{CB}(N_j^0(t), \delta) > \mu_j - \mathbf{CB}(N_j^0(t), \delta) - \mu_{i_W} \\ \text{and } &\mathbf{CB}(N_j^0(t), \delta) > \frac{\Delta_{j,i_W}}{2}. \end{aligned} \tag{34}$$

Using the fact that $N_j^0(t) \leq t$ and $N_{i,j}(t) \leq N_j^0(t)$, we have

$$\begin{aligned} \frac{\Delta_{j,i_W}}{2} <& \mathbf{CB}(N_j^0(t), \delta) \\ =& \sqrt{\frac{2\sigma^2}{N_j^0(t)} \log \frac{\pi^2 K (N_j^0(t))^2}{3\delta}} \\ \leq& \sqrt{\frac{2\sigma^2}{N_j^0(t)} \log \frac{\pi^2 K t^2}{3\delta}} \\ \leq& \sqrt{\frac{2\sigma^2}{N_{i,j}(t)} \log \frac{\pi^2 K t^2}{3\delta}}, \end{aligned} \tag{35}$$

which is equivalent to

$$N_{i,j}(t) < \frac{8\sigma^2}{\Delta_{j,i_W}^2} \log \frac{\pi^2 K t^2}{3\delta}. \tag{36}$$

Hence, under event $\mathcal{E}_2$, we have

$$\begin{aligned} \hat{\mu}_i(t) <& \frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \mu_{I_s^0} + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ =& \frac{1}{N_i(t)} \sum_j \sum_{s \in \tau_{i,j}(t)} \mu_{I_s^0} + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ =& \frac{1}{N_i(t)} \sum_j N_{i,j}(t) \mu_j + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ =& \sum_j \frac{N_{i,j}(t)}{N_i(t)} (\Delta_{j,i_W} + \mu_{i_W}) + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ <& \mu_{i_W} + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ &+ \frac{1}{N_i(t)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K t^2}{3\delta}. \end{aligned} \tag{37}$$

The lemma is proved.

Appendix D
Proof of Theorem 1

By inferring from Lemma 1, we have that with probability $1 - \frac{\delta}{K}$, $\forall t > K : |\hat{\mu}_K^0(t) - \mu_K| < \mathbf{CB}(N_K^0(t), \delta)$.

Because the LCB attack scheme does not attack the target arm, we can also conclude that with probability $1 - \frac{\delta}{K}$, $\forall t > K : |\hat{\mu}_K(t) - \mu_K| < \mathbf{CB}(N_K(t), \delta)$.

The user relies on the UCB algorithm to choose arms. If at round t, the user chooses an arm $I_t = i \neq K$, which is not the target arm, we have

$$\hat{\mu}_i(t-1) + 3\sigma \sqrt{\frac{\log t}{N_i(t-1)}} > \hat{\mu}_K(t-1) + 3\sigma \sqrt{\frac{\log t}{N_K(t-1)}}, \tag{38}$$

which is equivalent to

$$3\sigma\sqrt{\frac{\log t}{N_i(t-1)}} > -\hat{\mu}_i(t-1) + \hat{\mu}_K(t-1) + 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}}. \tag{39}$$

We need to connect the estimate of arms to the true means. Under event $\mathcal{E}_1$, we have

$$\hat{\mu}_K(t) > \mu_K - \mathbf{CB}(N_K(t), \delta). \tag{40}$$

Under event $\mathcal{E}_1 \cap \mathcal{E}_2$, according to Lemma 4, we have

$$\begin{aligned}\hat{\mu}_i(t) \leq \mu_{i_W} &+ \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}} \\ &+ \frac{1}{N_i(t)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K t^2}{3\delta}.\end{aligned} \tag{41}$$

Combing the inequalities above,

$$\begin{aligned}3\sigma\sqrt{\frac{\log t}{N_i(t-1)}} > &-\mu_{i_W} - \sqrt{\frac{2\sigma^2 K}{N_i(t-1)} \log \frac{\pi^2 (N_i(t-1))^2}{3\delta}} \\ &- \frac{1}{N_i(t-1)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K (t-1)^2}{3\delta} + \\ &\mu_K - \mathbf{CB}(N_K(t-1), \delta) + 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}}.\end{aligned} \tag{42}$$

The sum of the last two terms in the RHS of (42) is equal or larger than zero. We show it by further bounding the last term as follows: when $t \geq \left(\frac{\pi^2 K}{3\delta}\right)^{\frac{2}{5}}$,

$$\begin{aligned}3\sigma\sqrt{\frac{\log t}{N_K(t-1)}} &\geq \sqrt{4\sigma^2 \frac{\log t}{N_K(t-1)} + 5\sigma^2 \frac{\log(\frac{\pi^2 K}{3\delta})^{\frac{2}{5}}}{N_K(t-1)}} \\ &\geq \sqrt{2\sigma^2 \frac{\log \frac{\pi^2 K t^2}{3\delta}}{N_K(t-1)}} \\ &\geq \sqrt{2\sigma^2 \frac{\log \frac{\pi^2 K (N_K(t-1))^2}{3\delta}}{N_K(t-1)}} \\ &= \mathbf{CB}(N_K(t-1), \delta).\end{aligned} \tag{43}$$

Now the inequality only depends on $N_i(t-1)$ and some constants:

$$\begin{aligned}3\sigma\sqrt{\frac{\log t}{N_i(t-1)}} >& \Delta_{K,i_W} - \sqrt{\frac{2\sigma^2 K}{N_i(t-1)} \log \frac{\pi^2 (N_i(t-1))^2}{3\delta}} \\ &- \frac{1}{N_i(t-1)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K (t-1)^2}{3\delta} \\ >& \Delta_{K,i_W} - \sqrt{\frac{2\sigma^2 K}{N_i(t-1)} \log \frac{\pi^2 t^2}{3\delta}} \\ &- \frac{1}{N_i(t-1)} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K t^2}{3\delta}.\end{aligned} \tag{44}$$

The last inequality is based on the fact that $N_i(t-1) < t$. By solving the inequality above, we have:

$$\begin{aligned}N_i(t-1) <& \frac{1}{4\Delta_{K,i_W}^2} \left(3\sigma\sqrt{\log t} + \sqrt{2\sigma^2 K \log \frac{\pi^2 t^2}{3\delta}} \right. \\ &+ \left(\left(3\sigma\sqrt{\log t} + \sqrt{2\sigma^2 K \log \frac{\pi^2 t^2}{3\delta}} \right)^2 \right. \\ &\left. \left. + 4\Delta_{K,i_W} \sum_{j \neq i_W} \frac{8\sigma^2}{\Delta_{j,i_W}} \log \frac{\pi^2 K t^2}{3\delta} \right)^{\frac{1}{2}} \right)^2.\end{aligned} \tag{45}$$

Since event $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs with probability at least $1 - 2\delta$, we have that (45) holds with probability at least $1 - 2\delta$. Theorem 1 follows immediately from the definition of the attack cost and (45).

Appendix E
Proof of Theorem 2

Because the target arm is the worst arm, the mean rewards of all arms are larger than or equal to that of the target arm. Thus, for any attack scheme, we have

$$\frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \mu_{I_s^0} \geq \mu_K. \tag{46}$$

If the user pulls arm K at round t, according to UCB algorithm, we have for the optimal arm $i_O \neq K$,

$$\hat{\mu}_{i_O}(t-1) + 3\sigma\sqrt{\frac{\log t}{N_{i_O}(t-1)}} < \hat{\mu}_K(t-1) + 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}}. \tag{47}$$

Under event $\mathcal{E}_2$, we have Lemma 3 and (30) holds for all arm i, which implies

$$\begin{aligned}\hat{\mu}_{i_O}(t-1) >& \frac{1}{N_{i_O}(t-1)} \sum_{s \in \tau_{i_O}(t-1)} \mu_{I_s^0} - \\ &\sqrt{\frac{2\sigma^2 K}{N_{i_O}(t-1)} \log \frac{\pi^2 (N_{i_O}(t-1))^2}{3\delta}},\end{aligned} \tag{48}$$

and

$$\hat{\mu}_K(t-1) < \frac{1}{N_K(t-1)} \sum_{s\in\tau_K(t-1)} \mu_{I_s^0} + \sqrt{\frac{2\sigma^2 K}{N_K(t-1)} \log \frac{\pi^2 (N_K(t-1))^2}{3\delta}}. \tag{49}$$

Noted that for $\delta > \frac{1}{2}$, when δ is fixed, $\mathbf{CB}\left(\frac{N}{K}, \frac{\delta}{K}\right) = \sqrt{2\sigma^2 \frac{N}{K} \log \frac{\pi^2 N^2}{3\delta}} : (0, +\infty) \to \mathbb{R}$ is monotonically decreasing in $N \geq 1$.

We aim to prove that the total number of non-target arms pull scales as T. We divide the problem into three different cases.

Firstly, if $N_i(t-1) \geq \frac{1}{16} N_K(t-1)$, Theorem 2 holds.

Secondly, if $N_{i_O}(t-1) < \frac{1}{16} N_K(t-1)$ and $N_{i_O}(t-1) < \frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}}$ hold for the optimal arm i_O, we have

$$3\sigma \sqrt{\frac{\log t}{N_K(t-1)}} < \frac{3}{4}\sigma \sqrt{\frac{\log t}{N_{i_O}(t-1)}}, \tag{50}$$

and

$$\begin{aligned} &\sqrt{\frac{2\sigma^2 K}{N_K(t-1)} \log \frac{\pi^2 (N_K(t-1))^2}{3\delta}} \\ <&\sqrt{\frac{2\sigma^2 K}{N_{i_O}(t-1)} \log \frac{\pi^2 (N_{i_O}(t-1))^2}{3\delta}} \\ <&\frac{3}{4}\sigma \sqrt{\frac{\log t}{N_{i_O}(t-1)}}. \end{aligned} \tag{51}$$

Combining the inequalities above, we find

$$\begin{aligned} \frac{1}{N_K(t-1)} \sum_{s\in\tau_K(t-1)} \mu_{I_s^0} - \mu_K &> \frac{3}{4}\sigma \sqrt{\frac{\log t}{N_{i_O}(t-1)}} \\ &> \frac{3}{4}\sigma \sqrt{\frac{\pi \log t}{\sqrt{3\delta} t^{\frac{9}{64K}}}} \\ &> \frac{3}{4}\sigma \sqrt{\frac{\pi \log t}{\sqrt{3\delta} t}}. \end{aligned} \tag{52}$$

The RHS of (52) is monotonically decreasing in $t \geq 3$, so $\frac{3}{4}\sigma \sqrt{\frac{\pi \log t}{\sqrt{3\delta} t}} > \frac{3}{4}\sigma \sqrt{\frac{\pi \log T}{\sqrt{3\delta} T}}$.

Since the attack cost is limited by $\mathcal{O}(\log T)$,

$$\frac{1}{N_K(t-1)} \sum_{s\in\tau_K(t-1)} \mu_{I_s^0} - \mu_K = \frac{\mathcal{O}(\log T)}{N_K(t-1)}, \tag{53}$$

so

$$N_K(t-1) = O\left(\sqrt{T \log T}\right), \tag{54}$$

in which Theorem 2 holds.

Thirdly, if $N_{i_O}(t-1) < \frac{1}{16} N_K(t-1)$ and $N_{i_O}(t-1) \geq \frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}}$ hold for the optimal arm i_O, we have

$$3\sigma \sqrt{\frac{\log t}{N_K(t-1)}} < 3\sigma \sqrt{\frac{\log t}{N_{i_O}(t-1)}}, \tag{55}$$

and (47) is equivalent to

$$\hat{\mu}_{i_O}(t-1) < \hat{\mu}_K(t-1). \tag{56}$$

Setting the number of attacks on the optimal arm as C_{i_O} and the number of attacks on the target arm as C_K, we have

$$\frac{1}{N_{i_O}(t-1)} \sum_{s\in\tau_{i_O}(t-1)} \mu_{I_s^0} \geq \mu_{i_O} - \frac{C_{i_O}}{N_{i_O}(t-1)} \Delta_{i_O,K}, \tag{57}$$

and

$$\frac{1}{N_K(t-1)} \sum_{s\in\tau_K(t-1)} \mu_{I_s^0} \leq \mu_K + \frac{C_K}{N_K(t-1)} \Delta_{i_O,K}, \tag{58}$$

Thus, the inequality (56) becomes

$$\begin{aligned} &\mu_{i_O} - \frac{C_{i_O}}{N_{i_O}(t-1)} \Delta_{i_O,K} - \mathbf{CB}\left(\frac{N_{iO}(t-1)}{K}, \frac{\delta}{K}\right) \\ <&\mu_K + \frac{C_K}{N_K(t-1)} \Delta_{i_O,K} + \mathbf{CB}\left(\frac{N_K(t-1)}{K}, \frac{\delta}{K}\right). \end{aligned} \tag{59}$$

Because $N_{i_O}(t-1) < \frac{1}{16} N_K(t-1) < N_K(t-1)$, we have $\mathbf{CB}\left(\frac{N_{iO}(t-1)}{K}, \frac{\delta}{K}\right) > \mathbf{CB}\left(\frac{N_K(t-1)}{K}, \frac{\delta}{K}\right)$.

From (59), we have

$$\begin{aligned} &\frac{C_K}{N_K(t-1)} \Delta_{i_O,K} \\ >&\Delta_{i_O,K} - \frac{C_{i_O}}{N_{i_O}(t-1)} \Delta_{i_O,K} - 2\mathbf{CB}\left(\frac{N_{i_O}(t-1)}{K}, \frac{\delta}{K}\right) \\ >&\Delta_{i_O,K} - \frac{C_{i_O}}{N_{i_O}(t-1)} \Delta_{i_O,K} \\ &- 2\sqrt{\frac{2\sigma^2 K}{N_{i_O}(t-1)} \log \frac{\pi^2 (N_{i_O}(t-1))^2}{3\delta}}. \end{aligned} \tag{60}$$

Here, based on $N_{i_O}(t-1) \geq \frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}}$ and the fact $t \geq N_K(t-1)$,

$$\begin{aligned} &\frac{C_K}{N_K(t-1)} \Delta_{i_O,K} \\ >&\Delta_{i_O,K} - \frac{C_{i_O}}{\frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}}} \Delta_{i_O,K} \\ &- 2\sqrt{\frac{2\sigma^2 K}{\frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}}} \log \frac{\pi^2 (\frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}})^2}{3\delta}} \\ >&\Delta_{i_O,K} - \frac{C_{i_O}}{\frac{\sqrt{3\delta}}{\pi} (N_K(t-1))^{\frac{9}{64K}}} \Delta_{i_O,K} \\ &- 2\sqrt{\frac{2\sigma^2 K}{\frac{\sqrt{3\delta}}{\pi} (N_K(t-1))^{\frac{9}{64K}}} \log \frac{\pi^2 (\frac{\sqrt{3\delta}}{\pi} t^{\frac{9}{64K}})^2}{3\delta}}. \end{aligned} \tag{61}$$

Since the attack cost is limited by $\mathcal{O}(\log T)$,

$$N_K(t-1) = O((\log T)^{\frac{64K}{9}}), \tag{62}$$

and Theorem 2 holds.

In summary, all cases show that the user pulls the non-target arm more than $\mathcal{O}(T^{\alpha})$ times, in which $\alpha \leq 1$. Since event $\mathcal{E}_2$ holds with probability at least $1-\delta$, the conclusion in the theorem holds with probability at least $1-\delta$.

Appendix F
Proof of Proposition 1

The oracle attack needs to occasionally change the action to the best arm when the user pulls the target arm. Similar to Lemma 3, under event $\mathcal{E}_2$, for arm K and all $t > K$, we have

$$\left| \hat{\mu}_K(t) - \frac{1}{N_K(t)} \sum_{s \in \tau_K(t)} \mu_{I_s^0} \right| < \mathbf{CB}\left(\frac{N_K(t)}{2}, \frac{\delta}{K} \right), \quad (63)$$

because when the user pulls the target arm, the rewards the user observes are only drawn from two distributions.

Given the number of rounds that the attacker changes the action to the best arm as C_K, we have

$$\frac{1}{N_K(t)} \sum_{s \in \tau_K(t)} \mu_{I_s^0} \leq \mu_K + \frac{C_K}{N_K(t)} \Delta_{i_O,K}, \quad (64)$$

in which the equality when $N_K(t) \geq C_K$.

The user relies on the UCB algorithm to choose arms. We denote the last round when the user chooses the target arm before round T as t. At round t, the user chooses the target arm $I_t = K$. For any non-target arm i, we have

$$\hat{\mu}_i(t-1) + 3\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \leq \hat{\mu}_K(t-1) + 3\sigma \sqrt{\frac{\log t}{N_K(t-1)}}. \quad (65)$$

We focus on the last term of the RHS of (65). When $t \geq \left(\frac{\pi^2 K^2}{12\delta} \right)^4$, we have

$$3\sigma \sqrt{\frac{\log t}{N_K(t)}} \geq \sqrt{\frac{4\sigma^2}{N_K(t)} \log \frac{\pi^2 K^2 t^2}{12\delta}} \geq \mathbf{CB}\left(\frac{N_K(t)}{2}, \frac{\delta}{K} \right). \quad (66)$$

Thus, the RHS of (65) can be further bounded as:

$$\begin{aligned} &\hat{\mu}_K(t-1) + 3\sigma \sqrt{\frac{\log t}{N_K(t-1)}} \\ \leq &\mu_K + \frac{C_K}{N_K(t-1)} \Delta_{i_O,K} + 6\sigma \sqrt{\frac{\log t}{N_K(t-1)}}. \end{aligned} \quad (67)$$

Similar to (66), when $t \geq \frac{\pi^2 K^2}{3\delta}$,

$$\frac{5}{2}\sigma \sqrt{\frac{\log t}{N_i(t)}} > \sqrt{\frac{2\sigma^2}{N_i(t)} \log \frac{\pi^2 K t^2}{3\delta}} \geq \mathbf{CB}\left(N_i(t), \frac{\delta}{K} \right). \quad (68)$$

The oracle attack changes every non-target arm to the worst arm. Using Lemma 1, we have that with probability $1 - \frac{\delta(K-1)}{K}$, $\forall t > K$ and $i \neq K : |\hat{\mu}_i(t) - \mu_K| < \mathbf{CB}(N_i(t), \delta)$.

Then, by combing (67) and (68), (65) is equivalent to:

$$\begin{aligned} &\mu_K + \frac{1}{2}\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \\ <&\mu_K + \frac{C_K}{N_K(t-1)} \Delta_{i_O,K} + 6\sigma \sqrt{\frac{\log t}{N_K(t-1)}}. \end{aligned} \quad (69)$$

If the attacker does not attack the target arm, all arms are changed to the worst arm. Thus, at round t, the expectation of the target arm pull counts would be $\frac{t}{K}$. Here, we divide the problem into two cases: $N_K(t-1) \geq \frac{T}{K}$ and $N_K(t-1) < \frac{T}{K}$.

If $N_K(t-1) \geq \frac{T}{K}$, from (69), we have

$$\begin{aligned} &\frac{1}{2}\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \\ <&\frac{KC_K}{T} \Delta_{i_O,K} + 6\sigma \sqrt{\frac{K \log t}{T}}, \end{aligned} \quad (70)$$

which is equivalent to

$$N_i(t-1) > \frac{\sigma^2 T^2 \log t}{4\left(KC_K \Delta_{i_O,K} + 6\sigma \sqrt{KT \log t} \right)^2}. \quad (71)$$

Since equation (71) is monotonically increasing in $t \geq 1$ and the fact that $t > \frac{T}{K}$,

$$N_i(t-1) > \frac{\sigma^2 T^2 \log \frac{T}{K}}{4\left(KC_K \Delta_{i_O,K} + 6\sigma \sqrt{KT \log \frac{T}{K}} \right)^2}. \quad (72)$$

If $N_K(t-1) < \frac{T}{K}$, the attack cost $|\mathcal{C}| > \frac{T(K-1)}{K} + C_K$ and $\sum_{i \neq K} N_i(t-1) > \frac{T(K-1)}{K}$.

Combining the two cases, the proof is completed.

Appendix G
Proof of Lemma 5

Note that for $\delta \leq \frac{1}{3}$, $\beta(N) = \sqrt{\frac{2\sigma^2 K}{N} \log \frac{\pi^2 N^2}{3\delta}}$ is monotonically decreasing in N, as

$$\begin{aligned} \frac{\partial}{\partial N} \beta^2(N) =& \frac{2\sigma^2 K}{N^2} \left(2 - \log \frac{\pi^2 N^2}{3\delta} \right) \\ \leq& \frac{2\sigma^2 K}{N^2} \left(2 - \log \frac{\pi^2}{3\delta} \right) < 0. \end{aligned} \quad (73)$$

We first prove the first inequality in Lemma 5. Consider the optimal arm i_O and the worst arm i_W. Define $C_i := |\{t : t \leq T, I_t^0 \neq I_t = i\}|$. In the action-manipulation setting, when $t > 2AK$, MOUCB algorithm has

$$\begin{aligned} \frac{1}{N_{i_O}(t)} \sum_{s \in \tau_{i_O}(t)} \mu_{I_s^0} &\geq \frac{N_{i_O}(t) - C_{i_O}}{N_{i_O}(t)} \mu_{i_O} + \frac{C_{i_O}}{N_{i_O}(t)} \mu_{i_W} \\ &= \mu_{i_O} - \Delta_{i_O,i_W} \frac{C_{i_O}}{N_{i_O}(t)} \\ &\geq \mu_{i_O} - \Delta_{i_O,i_W} \frac{C_{i_O}}{2A}, \end{aligned} \quad (74)$$

and

$$\begin{aligned}\frac{1}{N_{i_W}(t)}\sum_{s\in\tau_{i_W}(t)}\mu_{I_s^0} &\le \frac{N_{i_W}(t)-C_{i_W}}{N_{i_W}(t)}\mu_{i_W}+\frac{C_{i_W}}{N_{i_W}(t)}\mu_{i_O}\\ &=\mu_{i_W}+\Delta_{i_O,i_W}\frac{C_{i_W}}{N_{i_W}(t)}\\ &\le \mu_{i_W}+\Delta_{i_O,i_W}\frac{C_{i_W}}{2A}.\end{aligned} \tag{75}$$

Combining (74) and (75), we have

$$\begin{aligned}&\frac{1}{N_{i_O}(t)}\sum_{s\in\tau_{i_O}(t)}\mu_{I_s^0}-\frac{1}{N_{i_W}(t)}\sum_{s\in\tau_{i_W}(t)}\mu_{I_s^0}\\ \ge&\mu_{i_O}-\mu_{i_W}-\Delta_{i_O,i_W}\frac{C_{i_W}}{2A}-\Delta_{i_O,i_W}\frac{C_{i_O}}{2A}\\ \ge&\mu_{i_O}-\mu_{i_W}-\Delta_{i_O,i_W}\frac{A}{2A}\\ =&\frac{\Delta_{i_O,i_W}}{2}.\end{aligned} \tag{76}$$

From (30), we could find

$$\begin{aligned}&\frac{1}{N_{i_O}(t)}\sum_{s\in\tau_{i_O}(t)}\mu_{I_s^0}-\frac{1}{N_{i_W}(t)}\sum_{s\in\tau_{i_W}(t)}\mu_{I_s^0}\\ \le&\hat{\mu}_{i_O}(t)+\beta(N_{i_O}(t))-(\hat{\mu}_{i_W}(t)-\beta(N_{i_W}(t)))\\ \le&\max_{i,j}\left\{\hat{\mu}_i(t)+\beta(N_i(t))-(\hat{\mu}_j(t)-\beta\left(N_j(t)\right))\right\}.\end{aligned} \tag{77}$$

We now prove the second inequality in Lemma 5:

$$\begin{aligned}&\max_{i,j}\left\{\hat{\mu}_i(t)+\beta(N_i(t))-(\hat{\mu}_j(t)-\beta(N_j(t)))\right\}\\ \le&\max_{i,j}\left\{\frac{1}{N_i(t)}\sum_{s\in\tau_i(t)}\mu_{I_s^0}+2\beta(N_i(t))\right.\\ &\left.-\left(\frac{1}{N_j(t)}\sum_{s\in\tau_j(t)}\mu_{I_s^0}-2\beta(N_j(t))\right)\right\}\\ \le&\Delta_{i_O,i_W}+\max_{i,j}\left\{2\beta(N_i(t))+2\beta(N_j(t))\right\}.\end{aligned} \tag{78}$$

Recall that for $\delta\le\frac{1}{3}$, $\beta(N)=\sqrt{\frac{2\sigma^2K}{N}\log\frac{\pi^2N^2}{3\delta}}$ is monotonically decreasing in N. Therefore,

$$\max_{i,j}\left\{2\beta(N_i(t))+2\beta(N_j(t))\right\}\le 4\beta(2A). \tag{79}$$

Appendix H
Proof of Theorem 3

MOUCB algorithm first pulls each arm $2A$ times. Then for $t>2AK$ and under event $\mathcal{E}_2$, if at round $t+1$, MOUCB algorithm choose a non-optimal arm $I_{t+1}=a\neq i_O$, we have

$$\begin{aligned}&\hat{\mu}_a+\beta(N_a(t))+\\ &\frac{2A}{N_a(t)}\max_{i,j}\left\{\hat{\mu}_i-\hat{\mu}_j+\beta(N_i(t))+\beta(N_j(t))\right\}\\ \ge&\hat{\mu}_{i_O}+\beta(N_{i_O}(t))+\\ &\frac{2A}{N_{i_O}(t)}\max_{i,j}\left\{\hat{\mu}_i-\hat{\mu}_j+\beta(N_i(t))+\beta(N_j(t))\right\},\end{aligned}$$

which implies

$$\begin{aligned}&\hat{\mu}_a+\frac{A}{N_a(t)}\left(2\Delta_{i_O,i_W}+8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right)+\beta(N_a(t))\\ \ge&\hat{\mu}_{i_O}+\frac{A}{N_{i_O}(t)}\Delta_{i_O,i_W}+\beta(N_{i_O}(t)),\end{aligned}$$

according to Lemma 5.

From equation (30), we could find

$$\begin{aligned}\hat{\mu}_a&\le\frac{1}{N_a(t)}\sum_{s\in\tau_a(t)}\mu_{I_s^0}+\beta(N_a(t))\\ &\le\mu_a+\Delta_{i_O,a}\frac{C_a}{N_a(t)}+\beta(N_a(t))\\ &\le\mu_a+\Delta_{i_O,a}\frac{A}{N_a(t)}+\beta(N_a(t)),\end{aligned}$$

and

$$\begin{aligned}\hat{\mu}_{i_O}&\ge\frac{1}{N_{i_O}(t)}\sum_{s\in\tau_{i_O}(t)}\mu_{I_s^0}-\beta(N_{i_O}(t))\\ &\ge\mu_{i_O}-\Delta_{i_O,i_W}\frac{C_{i_O}}{N_{i_O}(t)}-\beta(N_{i_O}(t))\\ &\ge\mu_{i_O}-\Delta_{i_O,i_W}\frac{A}{N_{i_O}(t)}-\beta(N_{i_O}(t)).\end{aligned}$$

By combining the inequalities above, we have

$$\begin{aligned}\mu_{i_O}\le&\mu_a+\Delta_{i_O,a}\frac{A}{N_a(t)}+2\beta(N_a(t))+\\ &\frac{A}{N_a(t)}\left(2\Delta_{i_O,i_W}+8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right),\end{aligned}$$

which is equivalent to

$$\begin{aligned}\Delta_{i_o,a}\le&\Delta_{i_O,a}\frac{A}{N_a(t)}+2\sqrt{\frac{2\sigma^2K}{N_a(t)}\log\frac{\pi^2(N_a(t))^2}{3\delta}}+\\ &\frac{A}{N_a(t)}\left(2\Delta_{i_O,i_W}+8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right)\\ \le&2\sqrt{\frac{2\sigma^2K}{N_a(t)}\log\frac{\pi^2t^2}{3\delta}}+\frac{A}{N_a(t)}\left(\Delta_{i_O,a}+\right.\\ &\left.2\Delta_{i_O,i_W}+8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}N_a(t)\le\max\Bigg\{&\frac{8\sigma^2K}{\Delta_{i_O,a}^2}\log\frac{\pi^2t^2}{3\delta},\frac{A}{\Delta_{i_o,a}}\left(\Delta_{i_O,a}\right.\\ &\left.+2\Delta_{i_O,i_W}+8\sqrt{\frac{\sigma^2K}{A}\log\frac{4\pi^2A^2}{3\delta}}\right)\Bigg\}.\end{aligned} \tag{80}$$

As event $\mathcal{E}_2$ holds with probability at least $1-\delta$, (45) holds with probability at least $1-\delta$. Then Theorem 3 follows immediately from the definition of the pseudo-regret in (2) and equation (80).

REFERENCES

[1] G. Liu and L. Lai, "Action-manipulation attacks on stochastic bandits," in *Proc. of ICASSP*, Barcelona, Spain, May 2020.

[2] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[3] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[4] Y. Lin, Z. Hong, Y. Liao, M. Shih, M. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *arXiv preprint arXiv:1703.06748*, 2017.

[5] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. of AAAI*, Austin, TX, Jan. 2015, pp. 2871–2877.

[6] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[7] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?," in *Proc. of ICML*, Francis Bach and David Blei, Eds., Lille, France, July 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 1689–1698.

[8] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in *Advances in NeurIPS*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 1885–1893.

[9] S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," in *Proc. of AAAI*, Phoenix, AZ, Feb. 2016, pp. 1452–1458.

[10] H. S. Chang, J. Hu, M. C. Fu, and S. I. Marcus, "Adaptive adversarial multi-armed bandit approach to two-person zero-sum markov games," *IEEE TAC*, vol. 55, no. 2, pp. 463–468, Feb 2010.

[11] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE TSP*, vol. 63, no. 14, pp. 3700–3714, July 2015.

[12] N. M. Vural, H. Gokcesu, K. Gokcesu, and S. S. Kozat, "Minimax optimal algorithms for adversarial bandit problem with multiple plays," *IEEE TSP*, vol. 67, no. 16, pp. 4383–4398, Aug 2019.

[13] K. Liu, Q. Zhao, and B. Krishnamachari, "Dynamic multichannel access with imperfect channel state detection," *IEEE TSP*, vol. 58, no. 5, pp. 2795–2808, May 2010.

[14] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE TSP*, vol. 58, no. 11, pp. 5667–5681, Nov 2010.

[15] S. Shahrampour, M. Noshad, and V. Tarokh, "On sequential elimination algorithms for best-arm identification in multi-armed bandits," *IEEE TSP*, vol. 65, no. 16, pp. 4281–4292, Aug 2017.

[16] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE TIT*, vol. 58, no. 8, pp. 5588–5611, Aug 2012.

[17] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM TIST*, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2014.

[18] L. Li, W. Chu, J. Langford, and R. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. of WWW*, New York, NY, Apr. 2010, pp. 661–670.

[19] L. Lai, H. El Gamal, H. Jiang, and H. Vincent Poor, "Cognitive medium access: Exploration, exploitation and competition," *IEEE TMC*, vol. 10, no. 2, pp. 239–253, Feb. 2011.

[20] M. Bande and V. V. Veeravalli, "Adversarial multi-user bandits for uncoordinated spectrum access," in *Proc. IEEE ICASSP*, Brighton, United Kingdom, May 2019, pp. 4514–4518.

[21] B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan, "Cascading bandits: Learning to rank in the cascade model," in *Proc. of ICML*, Francis Bach and David Blei, Eds., Lille, France, July 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 767–776.

[22] K. Jun, L. Li, Y. Ma, and X. Zhu, "Adversarial attacks on stochastic bandits," in *Proc. of NeurIPS*, Montréal, Canada, Dec. 2018, pp. 3644–3653.

[23] F. Liu and N. Shroff, "Data poisoning attacks on stochastic bandits," in *Proc. of ICML*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, CA, June 2019, vol. 97, pp. 4042–4050.

[24] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proc. of ACM STOC*, Los Angeles, CA, June 2018, pp. 114–122.

[25] Z. Guan, K. Ji, D. Bucci, T. Hu, J. Palombo, M. Liston, and Y. Liang, "Robust stochastic bandit algorithmsunder probabilistic unbounded adversarial attack," in *Proc. AAAI*, New York City, NY, Feb. 2020.

[26] Z. Feng, D. Parkes, and H. Xu, "The intrinsic robustness of stochastic bandits to strategic manipulation," *arXiv preprint arXiv:1906.01528*, 2019.

[27] Y. Ma, K. Jun, L. Li, and X. Zhu, "Data poisoning attacks in contextual bandits," *arXiv preprint arXiv:1808.05760*, 2018.

[28] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *FTML*, vol. 5, no. 1, pp. 1–122, 2012.

[29] C. Shen, "Universal best arm identification," *IEEE TSP*, vol. 67, no. 17, pp. 4464–4478, Sep. 2019.