Distributed Stochastic Bandits with Corrupted and Defective Input Commands

Meng-Che Chang and Matthieu R. Bloch
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332
Email: {mchang301,matthieu}@gatech.edu

***Abstract*—We analyze a distributed stochastic bandit model in which an agent controls multiple independent stochastic bandit machines. At each time step, the agent selects several machines for parallel exploitation but the arm pulled by each machine may differ from the command received either randomly (defective command) or adversarially (corrupted command). Machines that faithfully execute commands are called honest. We study situations in which the number of honest machines is either known or unknown and define appropriate notion of regrets. With at least one honest machine and a known number of honest bandits, we provide a simple algorithm that achieves $\tilde{O}(n^{1/2})$ regrets when commands are corrupted. Lower bounds on regret established by drawing connections to the problem of "low probability of detection," show the near optimality of the regret achieved by the algorithms.**

I. Introduction

In bandit problems, a player pulls an arm on a bandit machine at each time instant and obtains a corresponding reward. A standard objective for the player is to minimize his regret [1], defined as the difference between his rewards and those of the best arm pull strategy, over a fixed time horizon. The player then faces a trade-off between exploiting the most profitable arm identified from past rewards or exploring new arms. Standard algorithms, such as Upper Confidence Bound (UCB) [2] or Active Arm Elimination (AAE) [3], achieve optimal performance by only devoting a small fraction of the time to exploration but are vulnerable to corruptions. [4] and [5] investigate the effect of *reward* corruption and develop regret upper bounds that depend on the number of reward corruptions. Unlike [4], [5], we investigate the problem of *command* corruption, i.e., the arms pulled differ from those determined by the commands. Defective commands are those that change randomly, while corrupted commands are those that change adversarially.

Sub-linear regret is not guaranteed with defective or corrupted commands: an adversary may alter all commands to play sub-optimal arms. For the bandit problem to remain meaningful, we therefore assume that multiple distributed bandit machines are present. At each time step, an agent selects a subset of bandit machines to obtain multiple copies of a reward. Unlike [6], [7], which allow exchanges of information between distributed bandit machines, the commands are here determined locally by each bandit machine. With at least one honest machine in the environment, i.e., one without corrupted or defective commands, we shall show that the agent is able to distinguish corrupted/defective machines from honest ones, to control the number of corruptions/defections affecting the commands and achieve sub-linear regret.

Literature Review Different approaches have been proposed to analyze the performance of bandits games in the presence of adversaries. [4], [5] investigate the regret minimization problem in the stochastic bandit setting with bounded but unknown number of corruption on rewards, while [8], [9] studies the best arm identification problem in this setting. [10] explores the case of adversarial bandit, in which rewards are fully decided by an adversary, using the EXP3 algorithm. More works related to corruptions on reward can be found in [11]–[14]. Additionally, the existence of an adversary can cause privacy issues if the adversary has access to the rewards or input commands. [15], [16] address this challenge by introducing differential privacy in the stochastic bandit setting. In contrast, our model is motivated by distributed sensor networks, in which an agent probes multiple, possibly malfunctioning, sensors measuring the same physical phenomenon. The objective of the agent is to obtain the best measurements possible even in the presence of malfunctions.

The problem formulation of this work is inspired by the problem of "low probability of detection," in the communication system studied by [17]–[19]. Let n be the time duration. These works show that the fraction of non-innocent symbols transmitted needs to be at least $\frac{1}{\sqrt{n}}$ for the received outputs to have a noticeable difference from the outputs when no communication happens. In the present work, the objective of the agent in this work is to detect the existence of adversaries by observing the rewards. A similar square root law, which says that the number of corruptions/defects on commands within any time t need to be at least $\Omega(\sqrt{t})$ to have a noticeable difference in the outputs, can be obtained.

II. Problem Formulation

Distributed stochastic bandit model Consider an *agent* connected to a set $\mathcal{I}_N = \{0, ..., N-1\}$ of *bandit machines* with identical number of arms and reward distributions. Let $\mathcal{K} = \{0, ..., K-1\}$ be the set of arms. Upon pulling the kth arm on machine i, the reward is an independent realization of a Bernoulli random variable with unknown parameter μ_k. Without loss of generality we assume that $\mu_0 > \mu_1 > \cdots > \mu_{K-1}$, and we define $\Delta_k = \mu_0 - \mu_k$ as the difference of expected rewards between arm 0 and arm k. Throughout, we assume that $\Delta_k = \Omega(1)$ for all $k \neq 0$, i.e., Δ_k is not decreasing w.r.t the time horizon n for all $k \neq 0$.

Over a fixed time horizon n, the agent and the bandit machines operates at each time $t \in [1;n]$ as follows. The agent sends *selection signals* to a subset $\mathcal{S}_t \subset \mathcal{I}_N$ of machines computed according to a policy ψ_t. For every $i \in \mathcal{S}_t$, the ith machine uses a local policy $\pi_t^{(i)}$ to determines a *command* $A_t^{(i)}$ indicating which arm to pull. A reward $X_t^{(i)}$ is then obtained and fed back to the agent along with the command $A_t^{(i)}$. The policy $\pi_t^{(i)}$ is allowed to depend on past commands $\{A_\ell^{(i)}\}_{\ell<t \,\&\, \mathcal{S}_\ell \ni i}$, rewards $\{X_t^{(i)}\}_{\ell<t \,\&\, \mathcal{S}_\ell \ni i}$ while the policy ψ_t is allowed to depend on past received rewards and commands from all machines. Unknown to the agent, a subset of machines, identified by indices in a subset $\mathcal{I}_m \subset \mathcal{I}_N$ with $|\mathcal{I}_m| = m$, is malfunctioning. Specifically, if $A_t^{(i)} \in \mathcal{K}$ is the arm pull command of the ith machine at the time t, the actual arm pulled is $A_t^{(i)} + W_t^{(i)}$, where for all $a, b \in \mathcal{K}$ the addition is defined as $(a+b) \mod K$. The machine i is called *honest* if $W_t^{(i)} = 0$ for all t; it is called *defective* if $W_t^{(i)} \sim P_W^{(i)}$ for all t and $P_W^{(i)}(0) < 1$; and it is called *corrupted* if $W_t^{(i)} \sim \phi_t^{(i)} \triangleq \mathbb{P}_{W_t^{(i)}|\mathbf{X}_{t-1}^{(i)}\mathbf{A}_t^{(i)}}$, i.e., the attack is determined by a policy $\phi_t^{(i)}$ that depends on past observations and commands. Note that $\mathbf{X}_{t-1}^{(i)}$ is defined as the vector that contains all the past rewards in the machine i before the time t, i.e., $\mathbf{X}_{t-1}^{(i)} \triangleq \{X_\ell^{(i)}\}_{\ell\in[1;t-1] \,\&\, \mathcal{S}_\ell \ni i}$, and $\mathbf{A}_t^{(i)} \triangleq \{A_\ell^{(i)}\}_{\ell\in[1;t] \,\&\, \mathcal{S}_\ell \ni i}$ is defined similarly.

Regret for known number of honest machines. With a known number of honest machines $h = N - m$, the agent should select h machines and we define the expected regret under the policies $\{\psi_t\}_{t\in[1;n]}$ and $\{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}$ as

$$\begin{aligned} R(\mathcal{I}_h, \{\psi_t\}_{t\in[1;n]}, \{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}) \\ \triangleq nh\mu_0 - \sum_{t=1}^{n}\sum_{i=0}^{N-1} \mathbb{E}\left\{X_t^{(i)} 1(i \in \mathcal{S}_t)\right\}. \qquad (1) \end{aligned}$$

Note that (1) depends on the indices of honest machines, so that a strategy might perform well for a certain hypothesis $\mathcal{I}_h$ but poorly for another. For instance, a strategy that always chooses the first h machines performs well if the honest machines are in $\mathcal{I}_h = \{0, ..., h-1\}$ but poorly otherwise since a corrupted/defective machine is then always selected. A good strategy should perform well regardless of $\mathcal{I}_h$. Therefore, we define the *universal* expected regret under the policies $\{\psi_t\}_{t\in[1;n]}$ and $\{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}$ as

$$\begin{aligned} R_{\text{Universal}}(\{\psi_t\}_{t\in[1;n]}, \{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}) \\ \triangleq \max_{\mathcal{I}_h \in \mathcal{H}_h} R(\mathcal{I}_h, \{\psi_t\}_{t\in[1;n]}, \{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}) \qquad (2) \end{aligned}$$

where $\mathcal{H}_h$ is the collection of all sets $\mathcal{I}_h \subset \mathcal{I}_N$ with size h.

Objective. The objective of the proposed algorithm is to devise policies $\{\psi_t\}_{t\in[1;n]}$ and $\{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}$ that minimize the regret defined in (2).

Notation. To streamline the presentation, we introduce the following notation. We define $\mathbf{W}^{(i)} = [W_\ell^{(i)}]_{\{\ell: i \in \mathcal{S}_\ell\}}$ as the sequence of noises perturbing the command sequence of machine i when the machine i is selected. Additionally, $\{W_{k\ell}^{(i)}\}_{\ell\in\mathbb{N}}$ be the series of noises that are added to the kth arm of the ith machine. The number of commands for pulling arm k issued by machine i up to time t is denoted as $T_k^{(i)}(t)$, i.e., $T_k^{(i)}(t) = \sum_{\{\ell\in[1;t]: \mathcal{S}_\ell \ni i\}} 1(A_t^{(i)} = k)$. The tilde Landau notation, i.e., $\tilde{O}, \tilde{o}, \tilde{\Omega}, \tilde{\omega}$, and $\tilde{\Theta}$, are defined in the same way as conventional Landau notations but ignoring logarithmic factors. For simplicity, we assume $\Delta_k = \Omega(1)$ for all $k \neq 0$, so the dependency on Δ_k are also hidden in the expression of Landau notations. $\Delta_{\min} = \min_{k\neq 0} \Delta_k$. All the Landau notation mentioned in this paper are asymptotic relative to the time horizon n.

Discussion on the definition of regrets. There is always a certain cost related to choosing a specific arm. When the value of h is known, $nh\mu_0$ is the maximal possible expected rewards obtainable without the influence of corruptions, i.e., rewards are generated from the arms indicated by the agent's commands. One can obtain expected rewards more than $nh\mu_0$ by choosing more than h machines, but it would make the "costs" related to choosing arms not as efficient due to potential corruptions from adversary. Therefore, the regret defined in (2) measures the difference between the maximal expected reward obtainable without the influence of corruptions, i.e. $nh\mu_0$, and the expected reward obtained by a specific strategy that chooses h machines.

Remark 1 (Why not forcing the algorithm to learn h)**.** *Defining the regret properly when the number of honest machines is unknown is challenging. Specifically, as pointed out in Section IV, the corrupted/defective machines might be impossible to detect, which happens for instance when the number of corruptions/defect is low. Therefore, we fix the number of chosen machines in* (1) *to make the regret well-defined and study the impact of corrupted/defective commands in this scenario.*

Remark 2 (Relation to covert communication)**.** *In covert communication, the number of transmitted messages should obey the square-root law in order to make the warden unaware of the transmission. Similarly, if the number of corruptions introduced by the adversaries does not follow the square-root law, the agent is able to detect them and avoid selecting those corrupted machines.*

Due to the page limit, we only show the results related to corruption machines in this paper. Results on defective machines will be available in the our full paper.

III. Regret Upper Bounds

A. Known Corruption

We first modify the UCB algorithm to obtain an algorithm that *tolerates* a certain number of corruptions on commands. Specifically, we say that an algorithm tolerates a certain number of corruptions if the number of sub-optimal commands generated by the algorithm has the same order as the number of corruptions. The formal definition is given below

Algorithm 1: Corrupt-Tolerant-UCB (CT-UCB)

```
Input: $i, \lambda, \alpha, n, t$
Define $c_1 \triangleq \left(2+\sqrt{\lambda}\right)^2$
Let
    $\Lambda_k^{(i)}(t-1) = \begin{cases} \infty & \text{if } T_k^{(i)}(t-1) = 0. \\ \hat{\mu}_k^{(i)}(t-1) + \sqrt{\frac{c_1 n^\alpha \log n}{T_k^{(i)}(t-1)}} & \text{otherwise.} \end{cases}$
The arm chosen is $A_t^{(i)} = \arg\max_{k \in \mathcal{A}} \Lambda_k^{(i)}(t-1)$.
Output: $A_t^{(i)} = \text{CT-UCB}(i, \lambda, \alpha, n, t)$.
```

Definition 1 (Tolerance)**.** *Let n be the time duration. A multi-arm bandit algorithm $\{\pi_t^{(i)}\}_{t\in[1;n]}$, which determines the commands on the machine i, tolerates $C(n)$ corruptions if*

$$\mathbb{E}[T_k^{(i)}(n)] = O(C(n)) \tag{3}$$

for all $k \in \mathcal{K} \setminus \{0\}$ and for all $\mathbf{W}^{(i)}$ that satisfy the inequality $||\mathbf{W}^{(i)}||_0 \le C(n)$ for all sufficiently large n.

The algorithm, named Corruption-Tolerant-UCB, is shown in Algorithm 1. The idea of the algorithm is to enlarge the confidence region so that the true mean is within the upper confidence bound with high probability, even if the arms pulled are perturbed by the corruptions. Specifically, let $\hat{\mu}_k^{(i)}(t) \triangleq \frac{1}{T_k^{(i)}(t)} \sum_{\ell=1}^{t} X_\ell^{(i)} 1(A_\ell^{(i)} = k)$ be the *observed* empirical mean of the output distribution of the arm $k \in \mathcal{K}$ obtained from observing the rewards from the machine i. Note that $\hat{\mu}_k^{(i)}(t)$ is not the true empirical mean of the arm k because X_ℓ might be generated from different arms. Then, the upper confidence bound $\Lambda_k^{(i)}(t)$ is defined as

$$\Lambda_k^{(i)}(t-1) = \begin{cases} \infty & \text{if } T_k^{(i)}(t-1) = 0. \\ \hat{\mu}_k^{(i)}(t-1) + \sqrt{\frac{c_1 n^\alpha \log n}{T_k^{(i)}(t-1)}} & \text{otherwise,} \end{cases}$$

where the value of α and c_1 depends on how tolerant we want the algorithm to be. When the machine determines the commands by using the Corruption-Tolerant-UCB algorithm, the expected number of commands for pulling each arm $k \in \mathcal{K} \setminus \{0\}$ is given in Theorem 2.

Theorem 2. *Assume $I_n^{(i)} \triangleq 1\left(||\mathbf{W}^{(i)}||_0 \le \lambda n^\alpha \log n\right)$ converges to 1 almost surely for all $i \in \mathcal{I}_N$ when $n \to \infty$ for some $\lambda > 0$ and $0 \le \alpha < 1$, i.e.,*

$$\mathbb{P}\left(\lim_{n\to\infty} \frac{||\mathbf{W}^{(i)}||_0}{\lambda n^\alpha \log n} \le 1\right) = 1. \tag{4}$$

Assume the value of λ and α is known. Then, the average number of commands for pulling arm $k \in \mathcal{K} \setminus \{0\}$ for the machine i is

$$\mathbb{E}[T_k^{(i)}(n)] = O\left(n^\alpha \log n\right) \tag{5}$$

regardless of the policy $\{\psi_t\}_{t\in[1;n]}$ when the machine determines the commands by the CT-UCB algorithm defined in Algorithm 1.

Theorem 2 implies that Algorithm 1 is $\lambda n^\alpha \log n$ tolerating, and the expected number of commands for pulling the arm $k \neq 0$ is upper bounded by $O(n^\alpha \log n)$. One can immediately conclude that the universal expected regret is upper bounded by $O(n^\alpha \log n)$ for any policy $\psi_{t\in[1;n]}$ and $\{\pi_t^{(i)}\}_{i\in\mathcal{I}_N, t\in[1;n]}$ characterized by Algorithm 1.

Corollary 3. *If $I_n^{(i)} \triangleq 1\left(||\mathbf{W}^{(i)}||_0 \le \lambda n^\alpha \log n\right)$ converges to 1 almost surely for all $i \in \mathcal{I}_N$ when $n \to \infty$ for some $\lambda > 0$ and $0 \le \alpha < 1$, then*

$$R_{\mathit{Universal}}(\{\psi_t\}_{i\in\mathcal{I}_N, t\in[1;n]}, \{\pi_t^{(i)}\}_{t\in[1;n]}) \le O(n^\alpha \log n)$$

for any ψ_t when for all $t \in [1;n]$ and $i \in \mathcal{S}_t$, the policy $\pi_t^{(i)}$ is characterized by CT-UCB$(i, \lambda, \alpha, n, t)$.

Proof. It follows directly from Theorem 2 and the upper bound $||\mathbf{W}^{(i)}||_0 \le \lambda n^\alpha \log n$. □

B. Unknown Corruption and Known Number of Corrupted Machines

The CT-UCB algorithm and Theorem 2 in the previous section require knowledge of λ and α. When the attack is adversarial, knowing the value of λ and α is not possible without properly designing the selection rule $\{\psi_t\}_{t\in[1;n]}$. Moreover, the adversarial attacks may completely change the performance of each arm, i.e., the adversary can control the actual arm pulled $A_t^{(i)} + W_t^{(i)}$ corresponding to the command $A_t^{(i)}$. To control the number of sub-optimal commands in this case, we first extend Theorem 2 in Corollary 4. Let $N_k^{(i)}(n) = \sum_{t\in[1;n], \mathcal{S}_t \ni i} 1(A_t^{(i)} = k, A_t^{(i)} + W_t^{(i)} \neq 0)$ be the number of pulls on sub-optimal arms when the command is pulling the arm k, and let $L^*(n) = \frac{n^\alpha (\log n)^2}{\Delta_{\text{min}}^2}$. Define the event

$$\mathcal{B}_{k,n}^{(i)} \triangleq \left\{ \frac{1}{L^*(n)} \sum_{\ell=1}^{L^*(n)} 1(W_{k\ell}^{(i)} + k \neq 0) \ge \frac{\sqrt{c_1} + \sqrt{2}}{\sqrt{\log n}} \right. \\ \left. \text{and } T_k^{(i)}(n) \ge L^*(n) \right\} \bigcup \left\{ T_k^{(i)}(n) < L^*(n) \right\},$$

where $c_1 = (2+\sqrt{\lambda})^2$. Then, we have the following corollary.

Corollary 4. *Fix any $i \in \mathcal{I}_N$. Assume there exists a sequence of non-empty sets of indices $\{\mathcal{K}_i^*(n)\}_{n\in\mathbb{N}}$ depending on i such that $1\left(N_{k_i^*}^{(i)}(n) \le \lambda n^\alpha \log n \text{ for all } k_i^* \in \mathcal{K}_i^*(n)\right)$ converges to 1 almost surely when $n \to \infty$ for some $\lambda > 0$ and $0 \le \alpha < 1$. If event*

$$\left(\cap_{k\in\mathcal{K}\setminus\{\mathcal{K}_i^*(n)\}} \mathcal{B}_{k,n}^{(i)}\right)$$

happens for all but finitely many n, then the average number of commands for pulling arm $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$ for machine i is

$$\mathbb{E}[T_k^{(i)}(n)] = O\left(n^\alpha (\log n)^2\right) \tag{6}$$

when the machine determines the commands by the UCB algorithm defined in Algorithm 1 with the knowledge of α and λ.

Corollary 4 extends Theorem 2 with the following modified conditions.

- We do not need an assumption on $||\mathbf{W}^{(i)}||_0$. Instead, we only require the existence of some arm $k_i^* \in \mathcal{K}_i^*(n)$ that behaves like an optimal arm, i.e., $N_{k_i^*(n)}^{(i)} \leq \lambda n^\alpha \log n$ when n is sufficiently large.
- We require that the number of pulls on sub-optimal arms corresponding to commands of pulling any arm $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$ is greater than $L^*(n)\frac{\sqrt{c_1}+\sqrt{2}}{\sqrt{\log n}}$ if $T_k^{(i)}(n) \geq L^*(n)$.

Note that we do not demand $\mathcal{K}_i^*(n)$ to be $\{0\}$ because, in the presence of corruptions, any arm may behave like an optimal arm. When the first condition is satisfied, there exists some arm that behaves like an optimal arm after corruption. Moreover, the second condition ensures that the difference between empirical means of any $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$ and any $k_i^* \in \mathcal{K}_i^*(n)$ is large enough so that the algorithm has no ambiguity in identifying arms behaving like an optimal one. When the two conditions in Corollary 4 are satisfied, fix any $k_i^* \in \mathcal{K}_i^*(n)$, we define the following event

$$\tilde{\mathcal{G}}_k^{(i)} \triangleq \left\{\mu_0 < \min_{t\in[1;n]} \Lambda_{k_i^*}^{(i)}(t)\right\} \bigcap \left\{\hat{\mu}_{kL^*(n)}^{(i)} + \sqrt{\frac{c_1 n^\alpha \log n}{L^*(n)}} < \mu_0\right\},$$

where for any $L \in \mathbb{N}$, the notation $\hat{\mu}_{kL}$ is the empirical mean calculated from the L copies of rewards when the command is pulling arm k. One can observe that

$$\mathbb{P}\left(T_k^{(i)}(n) > L^*(n)\right) \leq \mathbb{P}\left(\left(\tilde{\mathcal{G}}_k^{(i)}\right)^c\right).$$

Proving Corollary 4 is then equivalent to proving that $\tilde{\mathcal{G}}_k^{(i)}$ happens with low probability for all $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$. The first condition and the second condition are used to show that $\left\{\mu_0 \geq \min_{t\in[1;n]} \Lambda_{k_i^*}^{(i)}(t)\right\}$ and $\left\{\hat{\mu}_{kL^*(n)}^{(i)} + \sqrt{\frac{c_1 n^\alpha \log n}{L^*(n)}} \geq \mu_0\right\}$ happens with low probability, respectively. If all the conditions in Corollary 4 are satisfied, we can upper bounded the regret contributed from each machine $i \in \mathcal{I}_N$ by $N_k^{(i)}(n) + \sum_{k\in\mathcal{K}\setminus\mathcal{K}_i^*(n)} T_k^{(i)}(n) \leq O(n^\alpha (\log n)^2)$. The design philosophy of our algorithm for the policy ψ_t, called Algorithm 2, is then to assume a predefined value of λ and α in the CT-UCB algorithm and exclude the corrupted machines as quickly as possible before the condition in Corollary 4 fails.

Outline of Algorithm 2: In Algorithm 2, the commands are determined by the CT-UCB with parameter $\lambda = 2$ and $\alpha = 1/2$ for all $t \in [1;n]$ and for all machines, i.e., we expect that there exists some arm k_i^* such that $N_{k_i^*}^{(i)} \leq 2n^{1/2}\log n$ for all $i \in \mathcal{I}_N$. At each time t, the algorithm selects h machines uniformly from the active set $\mathcal{A}_t$. If machine $i \in \mathcal{I}_N$ fails the first condition in Corollary 4, its maximum empirical mean, $\hat{\mu}_{\max}^{(i)}(t) \triangleq \max_{k\in\mathcal{K}} \hat{\mu}_k^{(i)}(t)$, is very likely to have a detectable difference from the mean of the arm 0, μ_0, at some time $t \in [1;n]$. Hence, we should exclude the machine i from the active set if the difference between $\hat{\mu}_{\max}^{(i)}(t)$ and μ_0 is greater than a threshold η_t. Moreover, if the second condition in Corollary 4 does not hold, there exists some arm $k \notin \mathcal{K}_i^*(n)$ such that its empirical mean at some time τ, $\hat{\mu}_k^{(i)}(\tau)$, is close to μ_0, where τ is the time such that $T_k^*(\tau) = L^*(n)$. Therefore, we should also exclude the machine i from the active set, if there exists some $k \notin \mathcal{K}_i^*(n)$ such that the difference between μ_0 and the empirical mean $\hat{\mu}_k^{(i)}(t)$ is less than a threshold ζ_t for some $t \geq \sqrt{n}\log n$. However, there are some challenges to make such comparison.

- For simplicity, we have assumed that arm 0 is the one with the highest mean. However, this information is actually unknown to the algorithm so is the value of μ_0.
- We do not know the set of indices $\mathcal{K}_i^*(n)$.
- The thresholds η_t and ζ_t need to be chosen properly. If η_t is too small, there might be a honest machine excluded from the active set. On the other hand, if η_t is too large, a corrupted machine which does not satisfy the first condition in Corollary 4 might not be excluded. A similar trade-off also happens for the choice of ζ_t.
- We need to ensure that no honest machine is excluded from the active set.

We solve the first challenge by comparing $\hat{\mu}_{\max}^{(i)}(t)$ to $\hat{\mu}_{\max}^{\max}(t) \triangleq \max_{k\in\mathcal{K}} \max_{i\in\mathcal{A}_t} \hat{\mu}_k^{(i)}(t)$ for each $i \in \mathcal{I}_N$ and $t \geq \sqrt{n}\log n$ as shown in line 11 in Algorithm 2. Owing to the fact that there is always an honest machine in our setting, the value of $\hat{\mu}_{\max}^{\max}(t)$ is not far from μ_0 if the number of arm pulled is large enough. In order to have a good estimate of the empirical means, we need to ensure that the number of commands for pulling each arm is not too small and Lemma 5, with proof omitted, provides such guarantee. Therefore, Algorithm 2 starts excluding machines when $t > \sqrt{n}\log n$.

Lemma 5. *For all $j \in \mathcal{I}_N$, $k \in \mathcal{K}$ and $t > \sqrt{n}\log n$,*

$$\mathbb{P}\left(\lim_{n\to\infty} 1\left(\frac{T_k^{(j)}(t)}{n^{1/2}\log n} > 0\right) = 1\right) = 1$$

if Algorithm 2 is applied.

The second challenge comes from the fact that we do not require $\mathcal{K}_i^*(n) = \{0\}$, and hence the set $\mathcal{K} \setminus \mathcal{K}_i^*(n)$ is unknown. We solve this by comparing $\hat{\mu}_k^{(i)}(t)$ to $\hat{\mu}_{\max}^{(i)}(t)$ for all $k \in \mathcal{K} \setminus \{A_{\max}^{(i)}(t)\}$, where $A_{\max}^{(i)}(t) \triangleq \arg\max_{k\in\mathcal{K}} \hat{\mu}_k^{(i)}(t)$ is the empirical best arm of the machine i at the time t. If any $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$ does not satisfy the second condition in Corollary 4, then the machine i must be in the active set $\mathcal{A}(\tau)$ for some $\tau \geq \sqrt{n}\log n$ such that $T_k^{(i)}(\tau - 1) = L^*(n) - 1$ and $\sum_{\ell=1}^{L^*(n)-1} 1(W_{k\ell}^{(i)} + k \neq 0) < L^*(n)\frac{\sqrt{c_1}+\sqrt{2}}{\sqrt{\log n}}$ for some $k \in \mathcal{K} \setminus \mathcal{K}_i^*(n)$. This means $\hat{\mu}_{\max}^{(i)}(\tau-1) - \hat{\mu}_k^{(i)}(\tau-1) \geq \zeta_{\tau-1}$ for all $k \in \mathcal{K} \setminus \{A_{\max}^{(i)}(\tau-1)\}$, and we show in the proof of Lemma 7 that this happens with low probability.

The thresholds defined below have a good balance between excluding corrupted machines and maintaining honest machines as shown in the proof of Lemma 6 and Lemma 7.

$$\eta_t^{(i)} \triangleq 4\left(T_{A_{\max}^{(i)}(t)}(t)\right)^{-1/2} \sqrt{\log T_{A_{\max}^{(i)}(t)}(t)} \tag{7}$$

$$\eta_t \triangleq \max_{i\in\mathcal{A}_t} \eta_t^{(i)} \tag{8}$$

$$\zeta_t \triangleq 4\left(\frac{\sqrt{c_1}+\sqrt{2}}{\sqrt{\log n}}\right). \tag{9}$$

Line 15-16 in Algorithm 2 means that the empirical best arm for any machine that remains in the active set should not change. This modification is made mainly to solve a technical issue in the proof of Lemma 7, by which results $A_{\max}^{(i)}(t)$ might not be belong to the set $\mathcal{K}_i^*(n)$ for some $t > \sqrt{n}\log n$. Finally, Lemma 6 ensures that no honest machine is excluded.

Lemma 6. *If Algorithm 2 is applied, no honest machine is excluded from the active set $\mathcal{A}_t$ for all time t when $n \to \infty$. Specifically, defining the event*

$$\mathcal{V}_n \triangleq \bigcap_{t\geq\sqrt{n}\log n}\bigcap_{i\in\mathcal{I}_h}\left(\left\{A_{\max}^{(i)}(t) = A_{\max}^{(i)}(t-1)\right\}\right.$$
$$\bigcap\left\{|\hat{\mu}_{\max}^{(i)}(t) - \hat{\mu}_{\max}^{max}(t)| < \eta_t \textit{ and } T_{A_{\max}^{(i)}(t)}(t) \geq \frac{t}{\sqrt{\log t}}\right\}$$
$$\left.\bigcap\left\{\hat{\mu}_{\max}^{(i)}(t) - \hat{\mu}_k^{(i)}(t) \geq \zeta_t \;\; \forall k \in \mathcal{K}\setminus\{A_{\max}^{(i)}(t)\right\}\right),$$

then $\mathbb{P}\left(\cap_{n=1}^{\infty}\cup_{m=n}^{\infty}\mathcal{V}_m\right) = 1$.

Lemma 7. *Fix any $i \in \mathcal{I}_N$. For any attack policy, there exists a sequence of non-empty sets of indices $\{\mathcal{K}_i^*(n)\}_{n\in\mathbb{N}}$ such that* $1\left(N_{k_i^*}^{(i)}(n) \leq 2n^{1/2}\log n \text{ for all } k_i^* \in \mathcal{K}_i^*(n)\right) \xrightarrow{a.s} 1$ *when $n \to \infty$ and $\bigcap_{k\in\mathcal{K}\setminus\{\mathcal{K}_i^*(n)\}}\mathcal{B}_{k,n}^{(i)}$ happens for infinitely many n if Algorithm 2 is applied with parameter $\alpha = 1/2$ and $\lambda = 2$.*

Algorithm 2:

```
   Input: $n, h, m$
   Initialization: $\mathcal{A}_1 = \mathcal{I}_N$
1  while $t \leq n$ do
2      if $t \leq \sqrt{n}\log n$ then
3          $\mathcal{A}_t = \mathcal{A}_1$
4          Choose $h$ machines out of $\mathcal{A}_t$ uniformly.
5          For all $i \in \mathcal{S}_t$, $A_t^{(i)} = \text{CT-UCB}(i, 2, 1/2, n, t)$.
6      else
7          $\hat{\mu}_{\max}^{\max} = \max_{k\in\mathcal{K}}\left(\max_{i\in\mathcal{A}_{t-1}}\hat{\mu}_k^{(i)}(t-1)\right)$.
8          $\forall i \in \mathcal{A}_{t-1}$, $\hat{\mu}_{\max}^{(i)}(t-1) = \max_{k\in\mathcal{K}}\hat{\mu}_k^{(i)}(t-1)$,
             $A_{\max}^{(i)}(t-1) = \arg\max_{k\in\mathcal{K}}\hat{\mu}_k^{(i)}(t-1)$
9          Let $\mathcal{J} = \emptyset$.
10         for $i \in \mathcal{A}_{t-1}$ do
11             if $|\hat{\mu}_{\max}^{(i)}(t-1) - \hat{\mu}_{\max}^{\max}(t-1)| \geq \eta_{t-1}$ or
                 $T_{A_{\max}^{(i)}(t-1)}(t-1) < \frac{t-1}{\sqrt{\log(t-1)}}$ then
12                 $\mathcal{J} = \mathcal{J} \cup \{i\}$.
13             if $\hat{\mu}_{\max}^{(i)}(t-1) - \hat{\mu}_k^{(i)}(t-1) < \zeta_{t-1}$ for some
                 $k \in \mathcal{K}\setminus\{A_{\max}^{(i)}(t-1)\}$ then
14                 $\mathcal{J} = \mathcal{J} \cup \{i\}$.
15             if $A_{\max}^{(i)}(t-1) \neq A_{\max}^{(i)}(t-2)$ then
16                 $\mathcal{J} = \mathcal{J} \cup \{i\}$.
17         $\mathcal{A}_t = \mathcal{A}_{t-1}\setminus\mathcal{J}$.
18         Choose $h$ machines out of $\mathcal{A}_t$ uniformly.
19         For all $i \in \mathcal{S}_t$, $A_t^{(i)} = \text{CT-UCB}(i, 2, 1/2, n, t)$
```

Upper Bounds of Regret for Algorithm 2

Theorem 8. *The universal expected regret defined in (2) achieved by Algorithm 2 is upper bounded by $O\left(n^{1/2}(\log n)^2\right)$ for all adversarial attack policies $\{\mathbb{P}_{W_t^{(i)}|\mathbf{X}_{t-1}^{(i)}\mathbf{A}_{t-1}^{(i)}}\}_{i\in\mathcal{I}_m,t\in[1;n]}$.*

Sketch Proof of Theorem 8: The proof of Theorem 8 follows directly from Lemma 6 and Lemma 7. Lemma 6 guarantees that no honest sensor is excluded from the active set $\mathcal{A}_t$ for all $t \in [1;n]$, while Lemma 7 says that the two conditions in Corollary 4 are satisfied for all machines with $\alpha = 1/2$ and $\lambda = 2$. The theorem is then proved by applying the result in Corollary 4.

IV. Regret Lower Bounds

Theorem 9. *There exist attack policies $\{\mathbb{P}_{W_t^{(i)}|\mathbf{X}_{t-1}^{(i)}\mathbf{A}_t^{(i)}}\}_{i\in\mathcal{I}_m,t\in[1;n]}$ such that the universal expected regret defined in (2) is at least $\tilde{\Omega}(n^{1/2})$ for any policies $\{\pi_t^{(i)}\}_{i\in\mathcal{I}_N,t\in[1;n]}$ and $\{\psi_t\}_{t=1}^n$.*

Sketch of Proof We prove Theorem 9 by contradiction. If there exists an algorithm such that the universal expected regret is $\tilde{o}(n^{1/2})$ for all attack policies, then the regret is $\tilde{O}(n^{\alpha})$ for all attack policies and for some $\alpha < 1/2$. However, we can show that there exists an attack policy such that the regret is at least $\tilde{\omega}(n^{\alpha})$ whenever $\alpha < 1/2$, and this complete the proof. To begin with, we define the attack policy $\mathbb{P}_{W_t^{(i)}|\mathbf{X}_{t-1}^{(i)}\mathbf{A}_{t-1}^{(i)}}$ as follows.

$$\mathbb{P}_{W_t^{(i)}|\mathbf{X}_{t-1}^{(i)}\mathbf{A}_{t-1}^{(i)}}(w) = \begin{cases}1 - n^{\beta-1} & \text{if } w = 0\\ \frac{n^{\beta-1}}{K-1} & \text{if } w \neq 0,\end{cases}$$

for all $i \in \mathcal{I}_m$ and $t \in [1;n]$ and for some $1/2 > \beta > \alpha > 0$. Note that by the Chernoff bound, $||\mathbf{W}_0^{(i)}||_0 = \Omega(n^{\beta})$ with probability arbitrarily close to 1 for any $i \in \mathcal{I}_m$. This implies that if the algorithm cannot exclude the machine $i \in \mathcal{I}_m$ from the active set before time n for all possible $\mathcal{I}_m$, then the expected accumulated regret is at least $\Omega(n^{\beta}) = \tilde{\omega}(n^{\alpha})$. By denoting $\mathbf{X}^t$ and $\mathbf{A}^t$ as all rewards and commands from all machines before the time t, if there exists an $(\mathcal{I}_h, \mathcal{I}'_h)$ pair and a positive sequence $\{\delta_t\}$ such that

$$\begin{aligned}&\mathbb{P}(\psi_t(\mathbf{X}^{t-1},\mathbf{A}^{t-1}) = \mathcal{I}'_h|\mathcal{I}_h)\\ &\quad + \mathbb{P}(\psi_t(\mathbf{X}^{t-1},\mathbf{A}^{t-1}) = \mathcal{I}_h|\mathcal{I}'_h) \geq \delta_t\end{aligned} \tag{10}$$

for all $t \in [1;n]$, where $\delta_t > \delta > 0$ for all $t \in [1;n]$ and for some $\delta > 0$, we show in supplementary document that

$$R_{\text{Universal}}(\{\psi_t\}_{t\in[1;n]}, \{\pi_t^{(i)}\}_{i\in\mathcal{I}_N,t\in[1;n]}) \geq \frac{\delta}{2}\Omega(n^{\beta}).$$

The condition in (10) implies that the algorithm cannot distinguish between the hypothesis $\mathcal{I}_h$ and $\mathcal{I}'_h$, and hence, there is a non-zero probability that a defective machine remains in the active set. Note that $\mathbb{P}(\psi_t(\mathbf{X}^{t-1},\mathbf{A}^{t-1}) = \mathcal{I}'_h|\mathcal{I}_h) + \mathbb{P}(\psi_t(\mathbf{X}^{t},\mathbf{A}^{t}) = \mathcal{I}_h|\mathcal{I}'_h) \geq 1 - \sqrt{D(P_{\mathbf{X}^t|\mathcal{I}_h}||P_{\mathbf{X}^t|\mathcal{I}'_h})}$. Therefore, it suffices to show that the divergence $D(P_{\mathbf{X}^t|\mathcal{I}_h}||P_{\mathbf{X}^t|\mathcal{I}'_h})$ is small for all $t \in [1;n]$. Finally, we show in the supplementary document that $D(P_{\mathbf{X}^t|\mathcal{I}_h}||P_{\mathbf{X}^t|\mathcal{I}'_h}) = o(1)$ for all $t \in [1;n]$ whenever $\beta < 1/2$, which completes the proof.

References

[1] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, mar 1985.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, *Machine Learning*, vol. 47, no. 2/3, pp. 235–256, 2002.

[3] E. Even-Dar, S. Mannor, and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *Journal of Machine Learning Research*, vol. 7, no. 39, pp. 1079–1105, 2006.

[4] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 114–122.

[5] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," in *Proceedings of the Thirty-Second Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. PMLR, 25–28 Jun 2019, pp. 1562–1578.

[6] E. Hillel, Z. S. Karnin, T. Koren, R. Lempel, and O. Somekh, "Distributed exploration in multi-armed bandits," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[7] Y. Wang, J. Hu, X. Chen, and L. Wang, "Distributed bandit learning: Near-optimal regret with efficient communication," in *International Conference on Learning Representations*, 2020.

[8] Z. Zhong, W. C. Cheung, and V. Y. F. Tan, "Probabilistic sequential shrinking: A best arm identification algorithm for stochastic bandits with corruptions," *CoRR*, vol. abs/2010.07904, 2020.

[9] A. Mukherjee, A. Tajer, P.-Y. Chen, and P. Das, "Mean-based best arm identification in stochastic bandits under reward contamination," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[10] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, jan 2002.

[11] Y. Seldin and A. Slivkins, "One practical algorithm for both stochastic and adversarial bandits," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1287–1295.

[12] Y. Seldin and G. Lugosi, "An improved parametrization and analysis of the EXP3++ algorithm for stochastic and adversarial bandits," in *Proceedings of the 2017 Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, S. Kale and O. Shamir, Eds., vol. 65. PMLR, 07–10 Jul 2017, pp. 1743–1759.

[13] S. Bubeck and A. Slivkins, "The best of both worlds: Stochastic and adversarial bandits," in *Proceedings of the 25th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, S. Mannor, N. Srebro, and R. C. Williamson, Eds., vol. 23. Edinburgh, Scotland: PMLR, 25–27 Jun 2012, pp. 42.1–42.23.

[14] l. yang, M. Hajiesmaili, M. S. Talebi, J. C. S. Lui, and W. S. Wong, "Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 19 943–19 952.

[15] D. Basu, C. Dimitrakakis, and A. Tossou, "Differential privacy for multi-armed bandits: What is it and what is its cost?" 05 2019.

[16] W. Ren, X. Zhou, J. Liu, and N. Shroff, "Multi-armed bandits with local differential privacy," 07 2020.

[17] B. A. Bash, D. Goeckel, and D. Towsley, "Limits of reliable communication with low probability of detection on awgn channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1921–1930, 2013.

[18] L. Wang, G. W. Wornell, and L. Zheng, "Fundamental limits of communication with low probability of detection," vol. 62, no. 6, pp. 3493–3503, Jun. 2016.

[19] M. R. Bloch, "Covert communication over noisy channels: A resolvability perspective," *IEEE Transactions on Information Theory*, vol. 62, no. 5, pp. 2334–2354, 2016.