Saving Stochastic Bandits from Poisoning Attacks via Limited Data Verification

Anshuka Rangi [1], **Long Tran-Thanh** [2], **Haifeng Xu** [3], **Massimo Franceschetti** [1],
[1] University of California San Diego
[2] University of Warwick, UK
[3] University of Virginia, USA
arangi@ucsd.edu, long.tran-thanh@warwick.ac.uk, hx4ad@virginia.edu, massimo@ece.ucsd.edu

Abstract

This paper studies bandit algorithms under data poisoning attacks in a bounded reward setting. We consider a strong attacker model in which the attacker can observe both the selected actions and their corresponding rewards, and can contaminate the rewards with additive noise. We show that *any* bandit algorithm with regret $O(\log T)$ can be forced to suffer a regret $\Omega(T)$ with an expected amount of contamination $O(\log T)$. This amount of contamination is also necessary, as we prove that there exists an $O(\log T)$ regret bandit algorithm, specifically the classical UCB, that requires $\Omega(\log T)$ amount of contamination to suffer regret $\Omega(T)$. To combat such poisoning attacks, our second main contribution is to propose verification based mechanisms, which use limited *verification* to access a limited number of uncontaminated rewards. In particular, for the case of unlimited verifications, we show that with $O(\log T)$ expected number of verifications, a simple modified version of the Explore-then-Commit type bandit algorithm can restore the order optimal $O(\log T)$ regret *irrespective of the amount of contamination* used by the attacker. We also provide a UCB-like verification scheme, called Secure-UCB, that also enjoys full recovery from any attacks, also with $O(\log T)$ expected number of verifications. To derive a matching lower bound on the number of verifications, we also prove that for any order-optimal bandit algorithm, this number of verifications $O(\log T)$ is necessary to recover the order-optimal regret. On the other hand, when the number of verifications is bounded above by a budget B, we propose a novel algorithm, Secure-BARBAR, which provably achieves $\tilde{O}(\min\{C, T/\sqrt{B}\})$ regret with high probability against weak attackers (i.e., attackers who have to place the contamination *before* seeing the actual pulls of the bandit algorithm), where C is the total amount of contamination by the attacker, which breaks the known $\Omega(C)$ lower bound of the non-verified setting if C is large.

1 Introduction

Multi Armed Bandits (MAB) algorithms are often used in web services (Agarwal et al. 2016; Li et al. 2010), sensor networks (Tran-Thanh, Rogers, and Jennings 2012), medical trials (Badanidiyuru, Kleinberg, and Slivkins 2018), and crowdsourcing systems (Rangi and Franceschetti 2018). The distributed nature of these applications makes these algorithms prone to third party attacks. For example, in web services decision making critically depends on reward collection, and this is prone to attacks that can impact observations and monitoring, delay or temper rewards, produce link failures, and generally modify or delete information through hijacking of communication links (Agarwal et al. 2016) (Cardenas, Amin, and Sastry 2008). Making these systems secure requires an understanding of the regime where the systems can be attacked, as well as designing ways to mitigate these attacks. In this paper, we study both of these aspects in a stochastic MAB setting.

We consider a data poisoning attack, also referred as man in the middle (MITM) attack. In this attack, there are three agents: the environment, the learner (MAB algorithm), and the attacker. At each discrete time-step t, the learner selects an action i_t among K choices, the environment then generates a reward $r_t(i_t) \in [0, 1]$ corresponding to the selected action, and attempts to communicate it to the learner. However, an adversary intercepts $r_t(i_t)$ and can contaminate it by adding noise $\epsilon_t(i_t) \in [-r_t(i_t), 1 - r_t(i_t)]$. It follows that the learner observes the contaminated reward $r_t^o(i_t) = r_t(i_t) + \epsilon_t(i_t)$, and $r_t^o(i_t) \in [0, 1]$. Hence, the adversary acts as a "man in the middle" between the learner and the environment. We present an upper bound on both the amount of contamination, which is the total amount of additive noise injected by the attacker, and the number of attacks, which is the number of times the adversary contaminates the observations, sufficient to ensure that the regret of the algorithm is $\Omega(T)$, where T is the total time of interaction between the learner and the environment. Additionally, we establish that this upper bound is order-optimal by providing a lower bound on the number of attacks and the amount of contamination.

A typical way to protect a distributed system from a MITM attack is to employ a secure channel between the learner and the environment (Asokan, Niemi, and Nyberg 2003; Sieka and Kshemkalyani 2007; Callegati, Cerroni, and Ramilli 2009). These secure channels ensure the CIA triad: confidentiality, integrity, and availability (Ghadeer 2018; Doddapaneni et al. 2017; Goyal and Mathew 2019). Various ways to establish these channels have been explored in the literature (Asokan, Niemi, and Nyberg 2003; Sieka and Kshemkalyani 2007; Haselsteiner and Breitfuß 2006; Callegati, Cerroni, and Ramilli 2009). An alternative way to provide security is by auditing, namely perform data verification (Karlof and Wagner 2003). The idea of data verification

or using trusted information is also embraced in the learning literature where small number of observations are verified (Charikar, Steinhardt, and Valiant 2017; Bishop, Tran-Thanh, and Gerding 2020). Establishing a secure channel or an effective auditing method or getting trusted information is generally costly (Sieka and Kshemkalyani 2007). Hence, it is crucial to design algorithms that achieve security, namely the performance of the algorithm is unaltered (or minimally altered) in presence of attack, while limiting the usage of these additional resources.

Motivated by these observations, we consider a *reward verification* model in which the learner can access verified (i.e. uncontaminated) rewards from the environment. This verified access can be implemented through a secure channel between the learner and the environment, or using auditing. At any round t, the learner can decide whether to access the possibly contaminated reward $r_t^o(i_t) = r_t(i_t)+\epsilon_t(i_t)$, or to access the verified reward $r_t^o(i_t) = r_t(i_t)$. Since verification is costly, the learner faces a tradeoff between its performance in terms of regret, and the number of times access to a verified reward occurs. Second, the learner needs to decide when to access a verified reward during the learning process. We design an order-optimal bandit algorithm which strategically plans the verification, and makes no assumptions on the attacker's strategy.

Against this background, we make the following contributions in this paper:

- First, in Section 3 we provide a tight characterisation about the total (expected) number of contaminations needed for a successful attack. Specifically, while it is well-known that with $O(\log T)$ expected number of contaminations, a strong attacker can successfully attack *any* bandit algorithm (see Section 3.2 for a more detailed discussion), it is not known to date whether this amount of contamination is necessary. We fill this gap by providing a matching lower bound on the amount of contamination (Theorem 1). This result is based on a novel insight of UCB's behaviour, which may be of independent interest. Specifically, we show that for arbitrary (even adversarial) reward sequences, UCB will pull every arm at least $\log(T/2)$ times for large T. Such conversativeness property of UCB guarantees its robustness against any attack strategy with $o(\log T)$ contaminations. Note that we also extend the state-of-the-art results on the sufficient condition by proposing a simpler yet optimal attack scheme, which is oblivious to the bandit algorithm's actual behaviour (Proposition 1).
- We then consider bandit algorithms with verification as a means of defense against these attacks. In our first set of investigations, we consider the case of having unlimited number of verification (Section 4.1). We first show that the minimum number of verification needed to recover from any strong attack is $\Theta(\log T)$ (Theorem 2 and Corollary 2). We then propose an Explore-Then-Commit (ETC) based method, called Secure-ETC that can achieve full recovery from any attacks with this optimal amount of verification (Observation 1). While Secure-ETC is simple, it might not stop the exploration phase before exceeding the time horizon. To avoid this situation, we also propose a UCB-like method called Secure-UCB, which also enjoys full recovery under optimal verification scheme (Theorem 3).
- Finally, we consider the case when the number of verifications is bounded above by a budget B. We first show that if the attacker has unlimited contamination budget, it is impossible to fully recover from the attack if the verification budget $B = o(T)$ (Theorem 4). However, when the attacker also has a finite contamination budget C, as typically assumed in the literature, we propose Secure-BARBAR, which achieves $\tilde{O}\left(\min\left\{C, T\log(2/\beta)/\sqrt{B}\right\}\right)$ regret against a weaker attacker (who has to place the contamination before seeing the actual pull of the bandit algorithm). It remains an intriguing open question whether there exists efficient but limited verification schemes against stronger attackers.

2 Preliminaries and Problem Statement

2.1 Poisoning Attacks on Stochastic Bandits

We consider the classical stochastic bandit setting under data poisoning attacks. In this setting, a learner can choose from a set of K actions for T rounds. At each round t, the learner chooses an action $i_t \in [K]$, triggers a reward $r_t(i_t) \in [0,1]$ and observes a possibly corrupted (and thus altered) reward $r_t^o(i_t) \in [0,1]$ corresponding to the chosen action. The reward $r_t(i)$ of action i is sampled independently from a fixed unknown distribution of action i. Let μ_i denote the expected reward of action i and $i^* = \text{argmax}_{i\in[K]}\mu_i$.* Also, let $\Delta(i) = \mu_{i^*} - \mu_i$ denote the difference between the expected reward of actions i^* and i. Finally, we assume that $\{\mu_i\}_{i\in[K]}$ are unknown to both the *learner* and the *attacker*.

The reward $r_t^o(i_t)$ observed by the learner and the true reward $r_t(i_t)$ satisfy the following relation

$$r_t^o(i_t) = r_t(i_t) + \epsilon_t(i_t), \tag{1}$$

where the contamination $\epsilon_t(i_t)$ added by the attacker can be a function of $\{i_n\}_{n=1}^t$ and $\{r_n(i_n)\}_{n=1}^t$. Additionally, since $r_t^o(i_t) \in [0,1]$, we have that $\epsilon_t(i_t) \in [-r_t(i_t), 1 - r_t(i_t)]$. If $\epsilon_t(i_t) \neq 0$, then the round t is said to be *under attack*. Hence, the *number of attacks* is $\sum_{t=1}^T \mathbf{1}(\epsilon_t(i_t) \neq 0)$ and the *amount of contamination* is $\sum_{t=1}^T |\epsilon_t(i_t)|$.

The regret $R^{\mathcal{A}}(T)$ of a learning algorithm $\mathcal{A}$ is the difference between the total expected true reward from the best fixed action and the total expected *true* reward over T rounds, namely

$$R^{\mathcal{A}}(T) = T\mu_{i^*} - \mathbb{E}[\sum_{t=1}^T r_t(i_t)], \tag{2}$$

The objective of the learner is to minimize the regret $R^{\mathcal{A}}(T)$. In contrast, the objective of the attacker is to increase the

*For convenience, we assume i^* is unique though all our conclusions hold when there are multiple optimal actions.

regret to at least $\Omega(T)$. As a convention, we say the attack is "successful" only when it leads to $\Omega(T)$ regret (Jun et al. 2018; Liu and Shroff 2019). The first question we address is the following.

Question 1: *Is there a* tight characterization *of the amount of contamination and the number of attacks leading to a regret of* $\Omega(T)$ *in stochastic bandits?*

2.2 Remedy via Limited Reward Verification

It is well known that no stochastic bandit algorithm can be resilient to data poisoning attacks if the attacker has sufficiently large amount of contamination (Liu and Shroff 2019). Therefore, to guarantee sub-linear regret when the attacker has an unbounded amount of contamination it is necessary for the bandit algorithm to exploit additional (and possibly costly) resources. We consider one of the most natural resource — *verified rewards*. Namely, we assume that at any round t, the learner can choose to access the true, uncontaminated reward of the selected action i_t, namely, when *round* t *is verified* we have $r_t^o(i_t) = r_t(i_t)$. This process of accessing true rewards is referred to as *verification*. If the learner performs verification at each round, then it is clear that the regret of any bandit algorithm is unaltered in the presence of attacker. Unfortunately, this is unrealistic because verification is costly in practice. Therefore, the learner has to carefully balance the regret and the number of verifications. This naturally leads to the second question that we aim to answer in this paper:

Question 2: *Is there a* tight characterization *of the number of verifications needed by the learner to guarantee the optimal* $O(\log T)$ *regret for* any *poisoning attack?*

Finally, we consider the case of limited amount of contamination from the attacker and limited number of verifications from the bandit algorithm. In the direction of studying this trade-off between contamination and verification, the third question that we aim to answer in this paper is:

Question 3: *Can we improve upon the* $\Omega(C)$ *regret lower-bound if the attacker's contamination budget is at most* C*, and the number of verifications that can be used by a bandit algorithm is also bounded above by a budget* B*.*

In this paper we answer the three questions above.

3 Tight Characterization for the Cost of Poisoning Attack

In this section we show that if an attack can successfully induce $\Theta(T)$ linear regret for any bandit algorithm, both its expected number of attacks and the expected amount of contamination must be $\Theta(\log(T))$. In other words, there exists a "robust" stochastic bandit algorithm that cannot be successfully attacked by any attacker with only $o(\log T)$ expected amount of contamination, and we show the celebrated UCB algorithm satisfies this property. The key technical challenge in proving the above result is to show the sublinear regret of UCB against *arbitrary* poisoning attack using at most $o(\log T)$ amount of contamination. In order to prove this strong result, we discover a novel "convervativeness" property of the UCB algorithm which may be of independent interest and has already found application in completely different tasks (Shi et al. 2021). To complement and also to match the above lower bounds of any successful attack, we design a data poisoning attack that can indeed use $O(\log T)$ expected number of attacks to induce $\Omega(T)$ regret for any order-optimal bandit algorithm, namely any algorithm which has $O(\log T)$-regret in the absence of attack. Since $r_t^o(i_t) \in [0, 1]$, this implies that the attack would require at most $O(\log T)$ expected amount of contamination.

3.1 Lower Bound on the Contaminations

We show that there exists an order-optimal bandit algorithm — in fact, the classical UCB algorithm — which cannot be attacked with $o(\log T)$ amount of contamination by *any* poisoning attack strategy. This implies that if an attacking strategy is required to be successful for all order-optimal bandit algorithms, then the amount of contamination needed is at least $\Omega(\log T)$. Since the amount of contamination is bounded above by the number of attacks, this also implies that any attacker requires at least $\Omega(\log T)$ number of attacks to be successful. While adversarial attacks to bandits have been extensively studied recently, to our knowledge such a lower bound on the attack strategy is novel and not known before; previous results have mostly studied the upper bound, i.e, how much contaminations are need for successful attacks (Jun et al. 2018; Liu and Shroff 2019).

Here we briefly describe the well-known UCB algorithm (Auer, Cesa-Bianchi, and Fischer 2002), and defer its details to Algorithm 2 in Appendix A. At each round $t \le K$, UCB selects an action in round robin manner. At each round $t > K$, the selected action i_t has the maximum *upper confidence bound*, namely

$$i_t = \text{argmax}_{i\in[K]}\left(\hat{\mu}_{t-1}(i) + \sqrt{\frac{8\log t}{N_{t-1}(i)}}\right), \tag{3}$$

where $N_t(i) = \sum_{n=1}^{t} \mathbf{1}(i_n = i)$ is the number of rounds action i is selected until (and including) round t, and

$$\hat{\mu}_t(i) = \frac{\sum_{n=1}^{t} r_n^o(i_n)\mathbf{1}(i_n = i)}{N_t(i)}, \tag{4}$$

is the empirical mean of action i until round t. Note that the algorithm uses the *observed* rewards.

The following Theorem 1 establishes that the UCB algorithm will have sublinear regret $o(T)$ under any poisoning attack if the amount of contamination is $o(\log T)$. The proof of Theorem 1 crucially hinges on the following "conservativeness" property about the UCB algorithm, which may be of independent interest.[†]

Lemma 1 (Conservativeness of UCB)**.** *Let* t_0 *be such that* $t_0/(\log(t_0))^2 \ge 36K^2$*. Then for all* $t \ge t_0$ *and any sequence of rewards* $\{r_n^o(i)\}_{i\in[K],n\le t}$ *in* $[0, 1]$ *(can even be adversarial), UCB will select every action at least* $\log(t/2)$ *times up until round* t*.*

[†]Indeed, Lemma 1 has been applied in (Shi et al. 2021) to the task of incentivized exploration in order to show that a *principal* can get sufficient feedback from every arm even if the *agent* who pulls arms has completely different preferences from the principal.

Lemma 1 is inherently due to the design of the UCB algorithm. Its proof does *not* rely on the rewards being stochastic, and it holds deterministically — i.e., at any time $t \geq t_0$, UCB will pull each action at least $\log(t/2)$ times. This lemma leads to the following theorem.

Theorem 1. *For all $0 < \epsilon < 1$ and $\alpha > 0$ such that $0 < \epsilon\alpha \leq 1/2$, and for all $T > \max\{(t_0)^{\frac{1}{1-\alpha\epsilon}}, \exp(4^\alpha)\}$, if the total* amount *of contamination by the attacker is $\sum_{n=1}^{T} |\epsilon_n(i_n)| \leq (\log T)^{1-\epsilon}$, then there exists a constant c_1 such that the expected regret of UCB algorithm is*

$$R^{UCB}(T) \leq c_1\big(T^{1-\alpha\epsilon} \max_i \Delta(i) + \sum_{i \neq i^*} \log T/\Delta(i)\big), \quad (5)$$

which implies regret $R^{UCB}(T)$ is $o(T)$.

The constant α in Theorem 1 is an adjustable *parameter* to control the tradeoff between the scale of time horizon T ($T \geq \max\{(t_0)^{\frac{1}{1-\alpha\epsilon}}, \exp(4^\alpha)\}$) and the dominating term ($T^{1-\alpha\epsilon} \max_i \Delta(i)$) in the regret. If ϵ is small, then the larger α leads to a smaller regret, however T should be sufficiently large in order for us to see such a regret.

The upper bound on the expected regret in Theorem 1 holds if the total amount of contamination is at most $(\log T)^{1-\epsilon}$. Furthermore, if the total number of attacks is at most $(\log T)^{1-\epsilon}$, then using $|\epsilon_t(i_t)| \leq 1$, we have that $\sum_{n=1}^{T} |\epsilon_n(i_n)| \leq (\log T)^{1-\epsilon}$. Hence, Theorem 1 also establishes that if the total number of attacks is $o(\log T)$, then the expected regret of UCB is $o(T)$. Thus, the attacker requires at least $\Omega(\log T)$ amount of contamination (or number of attacks) to ensure its success.

The lower bound on the amount of contamination in Theorem 1 cannot be directly compared with the upper bound in Proposition 1 since the former assumes that the amount of contamination is bounded above by $o(\log T)$ *almost surely*, while the latter is a bound on the *expected* amount of contamination. Instead, we consider the following corollary, which can be easily derived from Theorem 1 using Markov's inequality, and establishes the lower bound on the expected amount of contamination necessary for a successful attack.

Corollary 1. *For all $\epsilon \in (0, 1)$ and T such that the conditions in Theorem 1 are satisfied, if the expected amount of contamination by the attacker is at most $(\log T)^{1-\epsilon}$, in other words $o(\log T)$, then the regret of UCB is $o(T)$.*

3.2 Matching Upper Bound on Contamination

We now show that there indeed exists attacks that can succeed with $O(\log(T))$ attacks. Consider an attacker who tries to ensure any action $i_A \in [K]$ to be selected by the bandit algorithm at least $\Omega(T)$ times in expectation. This thus implies that the expected regret of the bandit algorithm is $\Omega(T)$ if $i_A \neq i^*$. We consider the following simple attack, that pulls the observed reward down to 0 whenever the target suboptimal action i_A is not selected. Namely,

$$r_t^o(i_t) = \begin{cases} r_t(i_t) & \text{if } i_t = i_A, \\ 0 & \text{if } i_t \neq i_A. \end{cases} \quad (6)$$

Equivalently, the attacker adds $\epsilon_t(i_t) = -r_t(i_t)\mathbf{1}(i_t \neq i_A)$ to the true reward $r_t(i_t)$. Unlike the attacks in (Jun et al. 2018; Liu and Shroff 2019), the attack in (6) is oblivious to rewards, since it overwrites all the rewards observation by zero. The following proposition establishes an upper bound on the expected number of attacks sufficient to be successful.

Proposition 1. *For any stochastic bandit algorithm $\mathcal{A}$ with expected regret in the* absence *of attack given by*

$$R^{\mathcal{A}}(T) = O\Big(\sum_{i \neq i^*} \frac{\log^\alpha(T)}{(\Delta(i))^\beta}\Big), \quad (7)$$

where $\alpha \geq 1$ and $\beta \geq 1$; and for any target action $i_A \in [K]$; if an attacker follows strategy (6)*, then it will use an expected number of attacks*

$$\mathbb{E}[\sum_{t=1}^{T} \mathbf{1}(\epsilon_t(i_t) \neq 0)]] = O\bigg((K-1)\log^\alpha(T)/\mu_{i_A}^{\beta+1}\bigg), \quad (8)$$

an expected amount of contamination

$$\mathbb{E}[\sum_{t=1}^{T} |\epsilon_t(i_t)|] = O\bigg((K-1)\log^\alpha(T)/\mu_{i_A}^{\beta+1}\bigg), \quad (9)$$

and it will force $\mathcal{A}$ to select the action i_A at least $\Omega(T)$ times in expectation, namely $\mathbb{E}[\sum_{t=1}^{T} \mathbf{1}(i_t = i_A)] = \Omega(T)$.

Proposition 1 provides a relationship between the regret of the algorithm without attack and the number of attacks (or amount of contamination) sufficient to ensure that the target action i_A is selected $\Omega(T)$ times, which also implies $R^{\mathcal{A}}(T) = \Omega(T)$ if $i_A \neq i^*$. Another important consequence of the proposition is that for an order optimal algorithm such as UCB, we have that $\alpha = 1$ and $\beta = 1$ in (7). Thus, the expected number of attacks and the expected amount of contamination are $O(\log T)$.

A small criticism to the attack strategy (6) might be that it pulls down the reward "too much". This turns out to be fixable. In Appendix C, we prove that a different type of attack that pulls the reward of any action $i \neq i_A$ down by an *estimated* gap $\Delta = 2\max\{\mu_i - \mu_{i_A}, 0\}$ (similar to the ACE algorithm in (Ma et al. 2018)) will also succeed. However, the number of attacks now will be inversely proportional to $\min_{i \neq i_A} |\mu_i - \mu_{i_A}|^{\beta+1}$, while not $\mu_{i_A}^{\beta+1}$ as in Proposition 1.

4 Verification based Algorithms

In this section we explore the idea of using verifications to rescue our bandit model from reward contaminations. In particular, we first investigate the case when the amount of verification is not limited, and therefore our main goal is to minimize the number of verifications (along with aiming to restore the order-optimal logarithmic regret bound). We then discuss the case when the number of verifications is bounded above by a budget B (typically of $o(T)$).

4.1 Saving Bandits with Unlimited Verifications

In this setting we assume that the number of verifications is not bounded above, and therefore, our goal is to minimize the number of verifications that is required to restore the logarithmic regret bound. To do so, we first show that any

successful verification based algorithm (i.e., they can restore the logarithmic regret) would require $\Omega(\log T)$ verifications. In particular, the following theorem establishes that for all consistent learning algorithm[‡] $\mathcal{A}$ and sufficiently large T, if the algorithm $\mathcal{A}$ uses $O((\log T)^{1-\alpha})$ verifications with $0 < \alpha < 1$, then the expected regret is $\Omega((\log T)^{\beta})$ with $\beta > 1$ in the MAB setting with verification.

Theorem 2. *Let $KL(i_1, i_2)$ denote the KL divergence between the distributions of actions i_1 and i_2. For all $0 < \alpha < 1$, $1 < \beta$ and all consistent learning algorithm $\mathcal{A}$, there exists a time t^* and an attacking strategy such that for all $T \geq 2t^*$ satisfying $(\log T)^{1-\alpha} + \beta \log(4 \log T) \leq \log T$, if the total number of verifications N_T^s until round T is*

$$N_T^s < (\log T)^{1-\alpha} / \min_{i_1, i_2 \in [K]} KL(i_1, i_2), \tag{10}$$

then the expected regret of $\mathcal{A}$ is at least $\Omega((\log T)^{\beta})$.

Theorem 2 establishes that $\Omega(\log T)$ verifications are necessary to obtain $O(\log T)$ regret. Here, we assume that the number of verifications is bounded above *almost surely*. Nevertheless, if instead the *expected* number of verifications is bounded, we shall obtain the following similar bound.

Corollary 2. *For all $0 < \alpha < 1$, $1 < \beta$, all consistent learning algorithm $\mathcal{A}$ and sufficiently large T such that the requirements in Theorem 2 are satisfied, there exists an attacking strategy such that if the* expected number of verifications *N_T^s until round T is $\mathbb{E}[N_T^s] < (\log T)^{1-\alpha} / \min_{i_1, i_2 \in [K]} KL(i_1, i_2)$, then the expected regret of $\mathcal{A}$ is at least $\Omega((\log T)^{\beta})$.*

We now move to design an algorithm that matches this optimal number of verifications. Our algorithm is based on the following simple idea: Contamination is only effective when the contaminated reward is used for estimating the mean reward value of the arms, and therefore, influencing the learnt order of the arms. As such, any algorithm that do not need these estimates for most of the time would not suffer much from the contamination if the remaining pulls (when the observed rewards are used for mean estimation) is properly secured via verification. This idea naturally lends us to the explore-then-commit (ETC) type of bandit algorithms (Garivier, Lattimore, and Kaufmann 2016), where in the first phase, the algorithm aims to learn the optimal arm by solving a best arm identification (BAI) problem (exploration phase), and in the second (commit) phase, it just repeatedly pulls the learnt best arm (Kaufmann, Cappé, and Garivier 2016). It is clear that if the first phase is fully secured (i.e., every single pull within that phase is verified), then we can learn the best arm with high probability, and thus, can ignore the contaminations within the second phase. The choice of the BAI algorithm for the exploration phase is important though. In particular, any BAI with fixed pulling budget would not work here, as they cannot guarantee logarithmic regret bounds (Garivier, Lattimore, and Kaufmann 2016). On the other hand, BAI with fixed confidence will suffice. In particular, we state the following:

Observation 1. *Any ETC algorithm, where the exploration phase uses BAI with fixed confidence $\delta = \frac{1}{T}$ and every single pull in that phase is verified, enjoys an expected regret bound of $O\Big(\sum_{i \neq i^*} \log T / \Delta_i\Big)$. In addition, the expected number of verifications is bounded above by $O\Big(\sum_{i \neq i^*} \log T / \Delta_i^2\Big)$.*

We refer to the ETC algorithm enhanced with verification described in the above observation as Secure-ETC. The proof of Observation 1 is simple and hence omitted from the main paper. Note that this result, alongside with Theorem 2, show that Secure-ETC uses order-optimal number of verification, and enjoys an order-optimal expected regret, irrespective of the attacker's strategy.

The main drawback of Secure-ETC algorithms is that there is positive probability that the algorithm may keep exploring until the end time T. While such small probability event turns out to not be an issue regarding its expected regret, one might prefer another type of algorithm which properly mix the exploration and exploitation. For such interested readers, we propose another algorithm, named Secure-UCB (for Secure Upper Confidence Bound), which integrates verification into the classical UCB algorithm, and also enjoys similar order-optimal regret bounds and order-optimal expected number of verifications. Due to space limitations, we defer both the detailed description of Secure-UCB and its theoretical analysis to the appendix (see Appendix J for more details). However, for the sake of completeness, we state the following theorem below.

Theorem 3. *For all T such that $T \geq c_2 \log T / \min_{i \neq i^*} \Delta^2(i)$, Secure-UCB performs $O(\log T)$ number of verification in expectation, and the expected regret of the algorithm is $O(\log T)$ irrespective of the attacker's strategy. Namely,*

$$\sum_{i \in [K]} \mathbb{E}[N_T^s(i)] \leq c_3 \Big(\sum_{i \neq i^*} \log T / \Delta^2(i) \Big), \tag{11}$$

$$R(T) \leq c_4 \Big(\sum_{i \neq i^*} \log T / \Delta(i) \Big), \tag{12}$$

where $N_T^s(i)$ is the total number of verifications for arm i until round T and c_2, c_3 and c_4 are numerical constants (concrete values can be found in the appendix).

It is worth noting that due to the sequential nature of UCB, designing a UCB-like algorithm with verification is far from trivial and therefore its technical analysis is significantly more involved.

4.2 Saving Bandits with Limited Verifications

While unlimited verification can completely restore the original regret bounds, we will show next that this is unfortunately not the case if the number of verification are bounded. In particular, we state this negative result.

Theorem 4. *Consider an attacker with unlimited contamination budget. For any T, $K \geq 2$ and $N_T^s \geq K$, if the total number of verifications performed until round T is at*

[‡]A learning algorithm is consistent (Kaufmann, Cappé, and Garivier 2016) if for all t, the action i_{t+1} (a random variable) is measurable given the history $\mathcal{F}_t = \sigma(i_1, r_1^o(i_1), i_2, r_2^o(i_2) \ldots, i_t, r_t^o(i_t))$.

Algorithm 1: Secure-BARBAR

1: **Input**: confidences $\beta, \delta \in (0,1)$, time horizon T, verification budget B
2: Set $n_i^B = \left\lfloor B/K \right\rfloor$, $T_0 = B$, $\Delta_i^0 = 1$ for all $i \in [K]$, and $\lambda = 1024 \ln(\frac{8K}{\delta} \log_2 T)$
3: **for** epochs $m = 1, 2, \ldots$ **do**
4: Set $n_i^m = \lambda(\Delta_i^{m-1})^{-2}$ for all $i \in [K]$, $N_m = \sum_{i=1}^K n_i^m$, and $T_m = T_{m-1} + N_m$
5: **for** $t = T_{m-1}$ **to** T_m **do**
6: choose arm i with probability n_i^m / N_m and pull it
7: if $n_i^B > 0$ then *verify the pull* (i.e., *observe the true reward*), and reduce n_i^B by 1
8: **end for**
9: Let S_i^m be the total *observed* rewards from pulls of arm i within epoch m (including both verified and unverified ones)
10: **If** *all* the pulls of arm i were verified in epoch m **then** $r_i^m = S_i^m / n_i^m$
11: **Else if** $S_i^m / n_i^m \geq \mu_i^B$ **then** $r_i^m = \min\left\{S_i^m/n_i^m, \mu_i^B + \frac{\Delta_i^{m-1}}{16} + \sqrt{\frac{\ln 2/\beta}{2n_B}}\right\}$
12: **Else** $r_i^m = \max\left\{S_i^m/n_i^m, \mu_i^B - \frac{\Delta_i^{m-1}}{16} - \sqrt{\frac{\ln 2/\beta}{2n_B}}\right\}$
13: Set $r_*^m = \max_i\{r_i^m - \Delta_i^{m-1}/16\}$, $\Delta_i^m = \max\{2^{-m}, r_*^m - r_i^m\}$
14: **end for**

most N_T^s, then there exists a distribution over the assignment of rewards such that the expected gap-independent *regret of any learning algorithm is at least*

$$R(T) \geq cT\sqrt{K/N_T^s}. \tag{13}$$

where c is a numerical constant. In addition, for any T, $K \geq 2$, and $N_T^s \geq K$, there exists a distribution over the assignment of rewards such that the expected cost, defined as the sum of expected regret and the number of verifications, of any learning algorithm is at least $\Omega(T^{2/3})$.

We remark that the goal of Theorem 4 is to demonstrate that, unlike the unlimited verification case in subsection 4.1, here it is impossible to fully recover from the attack — in the sense of achieving order optimal regret bounds as in the original bandit setting without attacks — if $B \in o(T)$, and this thus motivates our following study (Theorem 5) of developing regret bounds that scale with the budget B. For this purpose it suffices to have a gap-independent lower bound as in Theorem 4. Nevertheless, we acknowledge that an interesting research question is to see whether one can achieve a gap-dependent lower bound though this is out of the scope of our current paper and is an independent open question.

Now, this impossibility result relies on the assumption that the attacker has an unlimited contamination budget (or amount of contamination). One might ask what would happen if the attacker is also limited by a contamination budget C (as typically assumed in the relevant literature (Gupta, Koren, and Talwar 2019; Bogunovic et al. 2020; Lykouris, Mirrokni, and Paes Leme 2018)).

We now turn to the investigation of this setting in more detail where contamination budget is at most C. To start with, we assume for now that the attacker can only place the contamination before seeing the actual actions of the bandit algorithm. We refer to this type of attackers as *weak* attackers, as opposed to the ones we have been dealing with in this paper (see Section 5 for a comprehensive comparison of different attacker models). We describe an algorithm that addresses this case in a provably efficient way. In particular, we introduce Secure-BARBAR (Algorithm 1), which is built on top of the BARBAR algorithm proposed by (Gupta, Koren, and Talwar 2019). The key differences are: (i) Secure-BARBAR sets up a verification budget n_i^B for each arm i and verify that arm until this budget deplets (lines $6-7$); and (ii) use these reward estimate to adjust the estimates (lines $9-13$). By doing so, we achieve the following result:

Theorem 5. *With probability at least $1-\delta-\beta$, the regret of Secure-BARBAR against any weak attackers with contamination budget C is bounded by*

$$O\Bigg(K \min\Big\{C, \frac{T \log \frac{2}{\beta} \ln(\frac{8K}{\delta} \log_2 T)}{\sqrt{B/K}}\Big\} + \sum_{i \neq i^*} \frac{\log T}{\Delta_i} \log\Big(\frac{K}{\delta} \log T\Big)\Bigg). \tag{14}$$

The regret bound is of $\tilde{O}\Big(\min\Big\{C, T\log(2/\beta)/\sqrt{B}\Big\}\Big)$, which breaks the known $\Omega(C)$ lower bound of the non-verified setting if C is large (Gupta, Koren, and Talwar 2019).

A note on efficient verification schemes against strong attackers. In the case of strong attackers, with a careful combination of the idea described in Secure-BARBAR to incorporate the verified pulls into the estimate of the average reward at each round (lines $9-12$ in Algorithm 1), and the techniques used in the proof of Theorem 1 from (Bogunovic et al. 2020) [§], we can prove the following result: With probability at least $1-\delta-\beta$, we can achieve a regret upper bound of $\tilde{O}\Big(\min\big\{C, T\log(2/\beta)/\sqrt{B}\big\}\log T\Big)$. This can be done by modifying the Robust Phase Elimination (RPE) algorithm described in (Bogunovic et al. 2020) with the verification and estimation steps from Algorithm 1. The drawback of this approach is that it only works when the contamination budget C is known in advance. Although (Bogunovic et al. 2020) have also provided a method against strong attackers with unknown contamination budget C, their method can only achieve $\tilde{O}(C^2)$ under some restrictive constraints (e.g., C has to be sufficiently small). In addition, it is not clear how to incorporate our ideas introduced for Secure-BARBAR to that approach in an efficient way (i.e., to significantly reduce the regret bound from $\tilde{O}(C^2)$). Given this, it remains future work to derive an efficient verification method against strong

[§]The key step is to replace Lemma 1 from (Bogunovic et al. 2020) with a verification aware version, using similar ideas applied in the proof of Theorem 5.

attackers with unknown contamination budget C, which can yield regret bounds better than $\tilde{O}(C^2)$.

5 Comparison of Attacker Models

This section provides a more detailed comparison between the different attacker models from the (robust bandits) literature and their corresponding performance guarantees. In particular, at each round t, a *weak attacker* has to make the contamination *before* the actual action is chosen. On the other hand, a *strong attacker* can observe both the chosen actions and the corresponding rewards before making the contamination. From the perspective of contamination budget (or the amount of contamination), it can either be bounded above surely by a threshold, or that bound only holds in expectation. We refer to the former as *deterministic budget*, while we call the latter as *expected budget*. To date, the following three attacker models have been studied: (i) weak attacker with deterministic budget; (ii) strong attacker with deterministic budget; and (iii) strong attacker with expected budget.

Weak attacker with deterministic budget. For this attacker model, (Gupta, Koren, and Talwar 2019) have proposed a robust bandit algorithm (called BARBAR) that provably achieves $O(KC + (\log T)^2)$ regret against a weak attacker with (unknown) deterministic budget C. They have also proved a matching regret lower bound of $\Omega(C)$. These results imply that in order to successfully attack BARBAR (i.e., to force a $\Omega(T)$ regret), a weak attacker with deterministic budget would need a contamination budget of $\Omega(T)$.

Strong attacker with deterministic budget. (Bogunovic et al. 2020) have shown that there is a phased elimination based bandit algorithm that achieves $O(\sqrt{T} + C \log T)$ regret if C is known to the algorithm, and $O(\sqrt{T} + C \log T + C^2)$ if C is unknown. Note that by moving from the weaker attacker model to the stronger one, we suffer an extra loss in terms of achievable regret (i.e., from $O(C)$ to $O(C^2)$) in case of unknown C. While the authors have also proved a matching regret lower bound of $\Omega(C)$ for the known budget case, they have not provided any similar results for the case of unknown budget. Nevertheless, their results show that in order to successfully attack their algorithm, an attacker of this type would need a contamination budget of $\Omega(T)$ for the case of known contamination budget, and $\Omega(\sqrt{T})$ if that budget is unknown.

Strong attacker with expected budget. Our Proposition 1 shows that this attacker can successfully attack any order-optimal algorithm with a $O(\log T)$ expected contamination budget (note that (Liu and Shroff 2019) have also proved a similar, but somewhat weaker result). We have also provided a matching lower bound on the necessary amount of expected contamination budget against UCB. It is worth noting that if the rewards are unbounded, then the attacker may use even less amount contamination (e.g., $O(\sqrt{\log T})$) to achieve a successful attack (Zuo 2020).

Saving bandit algorithms with verification. The above mentioned results also indicate that if an attacker uses a contamination budget C (either deterministic or expected), the regret that any (robust) algorithm would suffer is $\Omega(C)$. A simple implication of this is that if an attacker has a budget of $\Theta(T)$ (e.g., he can contaminate all the rewards), then no algorithm can maintain a sub-linear regret if they can only rely on the observed rewards. Secure-ETC, Secure-UCB, and Secure-BARBAR break this barrier of $\Omega(C)$ regret with verification. In particular, the former two still enjoy an order-optimal regret of $O(\log T)$ against any attacker (even when they have $\Theta(T)$ contamination budget) while only using $O(\log T)$ verifications. The latter, when playing against a weak attacker, still suffers a swift increase in the regret as C is increased. But this increase is not linear in C as in the non-verified setting.

6 Conclusions

In this paper we introduced a reward verification model for bandits to counteract against data contamination attacks. In particular, we contributions can be grouped as follows: We first revisited the analysis of strong attacker and proved the first attack lower bound of $\Theta(\log T)$ expected number of contaminations for a successful attack. This lower bound is shown to be tight with our oblivious attack scheme, the contamination of which matches the lower bound. We then move to verification based approaches with unlimited verification, where we first provided two algorithms, Secure-ETC and Secure-UCB, which can recover any attacks with logarithmic number of verifications. We also provided a matching lower bound on the number of verifications. For the case of limited verifications, we first showed that full recovery is impossible if the attacked has unlimited contamination budget, unless the verification budget $B = \Theta(T)$. In case the attacker is also limited by a budget C, we proposed Secure-BARBAR, which achieves a regret lower than the $\Theta(C)$ regret barrier, if used against a weak attacker.

For future research, when facing a strong attacker with contamination budget C, we briefly discussed how a similar idea from Secure-BARBAR with limited verification can be used to achieve a regret bound better than $O(C \log T)$. However, this idea requires that C is known in advance. It is an open question whether for the case of unknown C we can get a similar regret bound that is better than the regret we can achieve for the non-verified case. Second, since bounding the contamination in expectation and almost surely leads to different results (see Section 5), it would be interesting to study the setting where number of verifications is bounded almost surely. Third, another interesting extension is a *partial feedback verification* model, where the learner can only request a feedback about whether the observed reward is corrupted or not but cannot see the true reward. Finally, extending our study to RL is an intriguing future direction.

References

Agarwal, A.; Bird, S.; Cozowicz, M.; Hoang, L.; Langford, J.; Lee, S.; Li, J.; Melamed, D.; Oshri, G.; Ribas, O.; et al. 2016. Making contextual decisions with low technical debt. *arXiv preprint arXiv:1606.03966*.

Asokan, N.; Niemi, V.; and Nyberg, K. 2003. Man-in-the-

middle in tunnelled authentication protocols. In *International Workshop on Security Protocols*, 28–41. Springer.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3): 235–256.

Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2018. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3): 1–55.

Besson, L.; and Kaufmann, E. 2018. What Doubling Tricks Can and Can't Do for Multi-Armed Bandits. *arXiv preprint arXiv:1803.06971*.

Bishop, N.; Tran-Thanh, L.; and Gerding, E. 2020. Optimal learning from verified training data.

Bogunovic, I.; Losalka, A.; Krause, A.; and Scarlett, J. 2020. Stochastic linear bandits robust to adversarial attacks. *arXiv preprint arXiv:2007.03285*.

Callegati, F.; Cerroni, W.; and Ramilli, M. 2009. Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy*, 7(1): 78–81.

Cardenas, A. A.; Amin, S.; and Sastry, S. 2008. Secure control: Towards survivable cyber-physical systems. In *2008 The 28th International Conference on Distributed Computing Systems Workshops*, 495–500. IEEE.

Charikar, M.; Steinhardt, J.; and Valiant, G. 2017. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 47–60.

Doddapaneni, K.; Lakkundi, R.; Rao, S.; Kulkarni, S. G.; and Bhat, B. 2017. Secure fota object for iot. In *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 154–159. IEEE.

Garivier, A.; Lattimore, T.; and Kaufmann, E. 2016. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29: 784–792.

Ghadeer, H. 2018. Cybersecurity issues in internet of things and countermeasures. In *2018 IEEE International Conference on Industrial Internet (ICII)*, 195–201. IEEE.

Goyal, S.; and Mathew, R. 2019. Security Issues in Cloud Computing. In *International conference on Computer Networks, Big data and IoT*, 363–373. Springer.

Gupta, A.; Koren, T.; and Talwar, K. 2019. Better Algorithms for Stochastic Bandits with Adversarial Corruptions. In *Conference on Learning Theory*, 1562–1578.

Haselsteiner, E.; and Breitfuß, K. 2006. Security in near field communication (NFC). In *Workshop on RFID security*, 12–14. sn.

Jun, K.-S.; Li, L.; Ma, Y.; and Zhu, J. 2018. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, 3640–3649.

Karlof, C.; and Wagner, D. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2-3): 293–315.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1): 1–42.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Liu, F.; and Shroff, N. 2019. Data Poisoning Attacks on Stochastic Bandits. In *International Conference on Machine Learning*, 4042–4050.

Lykouris, T.; Mirrokni, V.; and Paes Leme, R. 2018. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 114–122.

Ma, Y.; Jun, K.-S.; Li, L.; and Zhu, X. 2018. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, 186–204. Springer.

Rangi, A.; and Franceschetti, M. 2018. Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 1345–1352.

Shi, C.; Xu, H.; Xiong, W.; and Shen, C. 2021. (Almost) Free Incentivized Exploration from Decentralized Learning Agents. In *Advances in Neural Information Processing Systems, NeurIPS 2021*.

Sieka, B.; and Kshemkalyani, A. D. 2007. Establishing Authenticated Channels and Secure Identifiers in Ad-hoc Networks. *IJ Network Security*, 5(1): 51–61.

Tran-Thanh, L.; Rogers, A.; and Jennings, N. R. 2012. Long-term information collection with energy harvesting wireless sensors: a multi-armed bandit based approach. *Autonomous Agents and Multi-Agent Systems*, 25(2): 352–394.

Zuo, S. 2020. Near Optimal Adversarial Attack on UCB Bandits. *arXiv preprint arXiv:2008.09312*.