5 Supplementary Document

5.1 *Benchmark using Multi-Armed Bandit Approach and Computational Infeasibility Beyond Toy Problems*

Computational Infeasibility: As mentioned in the manuscript, the multi-armed bandit approach is computationally infeasible since it has an exponential dependency on the state space. For example, for the numerical experiment presented, the number of bandit arms is roughly $\sim (50)^{6\times 3} \approx 3 \times 10^{30}$ arms, which makes it infeasible for it to optimize using any of the known bandit algorithms. The objective of putting the multi armed bandit formulation was to propose a discrete optimization algorithm for estimating the thresholds which has finite-time regret bounds. Indeed such an algorithm is useful when the state and action space are small, as demonstrated below.

For this benchmarking experiment, we consider a state space $\mathcal{Y}_O = \{0 = \texttt{bad}, 1 = \texttt{good}\}$ and a two dimensional action space $\{0 = \texttt{obfuscate}, 1 = \texttt{learn}\}$. We consider a task of performing $M = 10$ updates in $N = 25$ queries. The oracle probability transition matrix is given by, $\Delta(o'|o) = [0.8, 0.2; 0.2, 0.8]$. The success probabilities are taken to b$s(1,1) = 0.2$ and $s(2,1) = 0.8$ respectively for the oracle states and 0 for an obfuscate action.

SPSA and multi armed bandit parameters: For training the SPSA we consider a step size of $\phi_k = 0.9 * (0.99)^k$ and a perturbation of $\gamma = 0.8$. Since there are 2 actions, a single threshold for a particular oracle state can represent the action space. Hence, there are 2 thresholds, making the number of bandit arms 10^2. We compute the approximate cost for any given set of parameters for both methods by interacting and performing $N_{mc} = 1000$ episodes with $N = 25$ queries each. Additionally, we benchmark the approximate cost of the estimated optimal threshold policy with the optimal non-stationary policy obtained using backward induction for one of the considered cost, since making backward induction compatible with the eavesdropper estimate requires changes in the formulation.

Cost Function	Backward Induction	Stochastic Approximate		Multi Armed Bandit	
	Cost	Thresholds	Cost	Thresholds	Costs
C_1	-	0.02, 0	0.291	0, 0	0.2
C_2	-	10,0	10.75	9, 0	11.05
C_3	-2.58	3.92, 0	-1.81	5, 0	-1.59

Table 3: Benchmarking simultaneous perturbation stochastic approximation (SPSA) with multi armed Bandit approach in approximating thresholds of the optimal stationary policy with thresholds. SPSA perfoms equally well as an multi-armed bandit approach, which is a noisy discrete optimization method. SPSA but does not suffer from the memory and time constraints of multi-armed bandits. SPSA also does not require knowledge of transition probabilities and incorporates a dynamic cost function in contrast to backward induction.

Cost Functions: We consider 3 different running cost functions C_1, C_2, C_3, described below. We use C_1 and C_2 similar to the described cost function in the main text. C_3 is a toy cost function without dependence on the eavesdropper belief, used to benchmark with backward induction.

$$\begin{aligned}
C_1(o,b,a) &= \frac{b^2}{o^2}\log\left(\frac{I_n + 1/\delta_n}{I_n + 1}\right)\mathbb{1}(a) + \frac{o^2}{b^2}\log\left(\frac{I_n}{I_n + 1}\right)(1 - \mathbb{1}(a)) \\
C_2(o,b,a) &= \frac{b^2}{o^2}\log\left(\frac{I_n + 1/\delta_n}{I_n + 1}\right)\mathbb{1}(a) + \frac{o^2}{b^2}\log\left(\frac{I_n}{I_n + 1}\right)(1 - \mathbb{1}(a)) \\
C_3(o,b,a) &= \frac{1}{b^2}B[o,a],
\end{aligned}$$

where $B = \begin{bmatrix} -1 & 2 \\ -0.1 & 0 \end{bmatrix}$ and the $B[o,u]$ is the entry in row o and column u. And consider the terminal cost fixed at $d(b) = b^2$. We summarize our results in Table 3. The thresholds are clipped in the interval $[0, M]$.

Results: It is evident from the table that the estimated thresholds from both methods differ by 1.08. The approximate costs are also close and differ from the cost computed using backward induction by 0.3 unit. This illustrates how the two methods perform equally well, but the SPSA method has the added advantage of being computationally tractable.

5.2 *Illustrating Stochastic Control Approach*

To better explain our stochastic control approach and complement the algorithm presented in the paper we have added Figure 1 in this document.

5.3 *Convergence Curve for Learners and Eavesdropper for optimal and greedy policies*

We have added comparision of the optimal and greedy policy for the described numerical experiment in Figure 2.

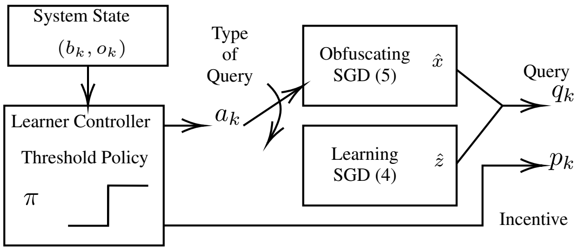

Figure 1: Extra Figure to demonstrate stochastic control for Covert Optimization: The learner uses π to dynamically switch between two stochastic gradient descents (2) and incentivize the oracle based on the queue state b_k and oracle state o_k. The query q_k is chosen as the last estimate from the selected SGD using (3). The optimal policy minimizes the cumulative cost function of the finite-horizon MDP.

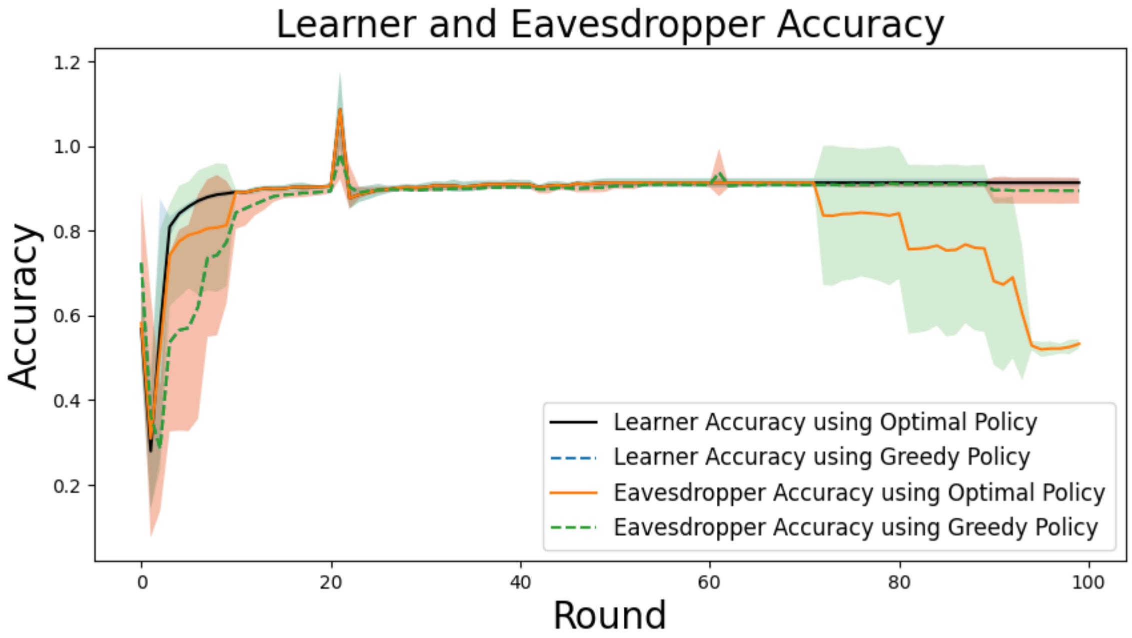

Figure 2: Convergence curve for optimal and greedy policy shows how the learner performance is almost same under both the policies but there is a significant difference in the eavesdropper performance. This illustrates the main benefit of our MDP formulation: The optimal policy exploits the stochasticity of the oracle to learn when the oracle is in a good state and obfuscate otherwise. Under the optimal policy, the accuracy of the eavesdropper is the same initially but decreases as the learner obfuscates more towards the end.