A Missing Details from Background section

Tree based aggregation scheme. Assume for simplicity that $T = 2^\alpha$ for some positive integer α. Let Tree be a complete binary tree with its leaf nodes being $f_1, \cdots, f_T$. Each internal node x in Tree stores the sum of all the leaf nodes in the tree rooted at x. First notice that one can compute any v_t using at most $\log T$ nodes of Tree. Second, notice that for any two neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$, at most $\log T$ nodes in Tree gets modified. So, if we flatten the complete tree as a vector then for any neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$ one can easily show that $\|\mathsf{Tree}(\mathcal{D}) - \mathsf{Tree}(\mathcal{D}')\|_1 \leq \log T$. Using the Laplace mechanism, it follows that adding $N \sim \mathsf{Lap}(1/\epsilon)$ (sampled i.i.d.) to each node of $\mathsf{Tree}(\mathcal{D})$ ensures $(\epsilon \log T)$-differential privacy for the complete tree. Therefore, to ensure ϵ-differential privacy for the complete tree, one needs to sample $N \sim \mathsf{Lap}(\log T/\epsilon)$. Since, the noise used in this scheme is exponential in nature, a high-probability guarantee is also immediate.

It is easy to show that if one uses the noisy tree to reconstruct v_t's, then the standard deviation of the noise in each v_t is $O\left(\sqrt{\log T}\frac{\log T}{\epsilon}\right) = O\left(\frac{\log^{1.5} T}{\epsilon}\right)$. Since all the noise random variables are exponentially distributed one can easily get tight concentration too. It is trivial to extend the above scheme to the case where $f_t \in \mathbb{R}^p$ and $\|f_t\|_2 \leq 1$ for all $t \in [T]$, using the Gamma mechanism.

B Proofs for Private UCB algorithm

B.1 Proof for Lemma 5

Proof. For the ease of notation, let $X_a(t)$ be the true total reward for arm a until time t. As argued in Section 2.1, $\mathsf{Noise}_a(t) = r_a(t) - X_a(t)$ is a sum of at most $\log T$ Laplace distributed random variables $\mathsf{Lap}(\frac{k \log T}{\epsilon})$. By the tail property of Laplace distribution, we know that for a given random variable $w \sim \mathsf{Lap}(\lambda)$, with probability $1 - \alpha$, $|w| \leq \lambda \log(1/\alpha)$. So, with probability at least $(1 - \alpha/(\log T))^{\log T} \leq 1 - \alpha$, $|\mathsf{Noise}_a(t)| \leq \frac{k \log^2 T \log((\log T)/\alpha)}{\epsilon}$. Taking the union bound over all k-arms and all time steps T and setting $\alpha = \gamma/(kT)$, we have w.p. $\geq 1 - \gamma$, for all $a \in \mathcal{C}$ and for all $t \in [T]$, $|\mathsf{Noise}_a(t)| \leq \frac{k \log^2 T \log((kT \log T)/\gamma)}{\epsilon}$. This completes the proof. □

C Proofs for Private Thompson algorithm

C.1 Proof for Lemma 9

Proof. For the brevity of notation let us divide the gap estimation phase into batches $b_1, b_2, \cdots$, where each batch corresponds to one complete execution of Line 4 in Algorithm 2. Let m_i be the number of pulls of each arm a_i in batch b_i. In each batch b_i, by the tail property of Laplace distribution it follows that with probability at least $1 - \frac{T^{-5}}{4 \log_2(1/\Delta)}$, we have $|\hat{\mu}_a - \tilde{\mu}_a| \leq \frac{2k \log_2(8T^5 \log_2(1/\Delta))}{\epsilon m_i}$ for each arm $a \in \mathcal{C}$. Here $\tilde{\mu}$ refers to the estimated mean prior to the addition of Laplace noise (see Line 4 of Algorithm 2 for the definition). Also by the use of Chernoff-Hoeffding's bound, we have with probability at least $1 - \frac{T^{-5}}{4 \log_2(1/\Delta)}$, and for $T > 8 \log_2 \frac{1}{\Delta}$, we have $|\mu_a - \tilde{\mu}_a| \leq \sqrt{\frac{\log_2\left(8T^5 \log_2(1/\Delta)\right)}{m_i}}$ for each arm $a \in \mathcal{C}$. We can thus conclude that with probability at least $1 - \frac{T^{-5}}{\log_2(1/\Delta)}$, by applying triangle inequality $|\hat{\mu}_a - \mu_a| \leq \frac{12k}{\epsilon}\frac{\log_2 T}{m_i} + \sqrt{\frac{6 \log_2 T}{m_i}}$ for each of the arms $a \in \mathcal{C}$. If $\hat{\Delta}_i$ be the value of $\hat{\Delta}$ (in Algorithm 2) during the execution of batch b_i, then with the choice of $m_i = \frac{192k \log_2 T}{\epsilon \Delta^2}$ as in the algorithm, ensures that $|\hat{\mu}_a - \mu_a| \leq \frac{\hat{\Delta}_i}{4}$. Also it is easy to see that using union bound we get, $\Pr(|\hat{\mu}_{a_{(1)}} - \mu_{a_1}| \geq \frac{\hat{\Delta}_i}{4} \cup |\hat{\mu}_{a_{(2)}} - \mu_{a_2}| \geq \frac{\hat{\Delta}_i}{4}) \leq \frac{kT^{-5}}{\log_2 \frac{1}{\Delta}}$. Hence for $T > k$ with probability at least $1 - \frac{T^{-4}}{\log_2 \frac{1}{\Delta}}$, $|\hat{\mu}_{a_{(1)}} - \mu_{a_1}| \leq \frac{\hat{\Delta}_i}{4}$ and $|\hat{\mu}_{a_{(2)}} - \mu_{a_2}| \leq \frac{\hat{\Delta}_i}{4}$

In each batch b_i, if $\left|\hat{\mu}_{a_{(1)}} - \hat{\mu}_{a_{(2)}}\right| \leq \hat{\Delta}_i$, then $|\mu_{a_1} - \mu_{a_2}| \leq 1.5\hat{\Delta}_i$. Similarly, if $\left|\hat{\mu}_{a_{(1)}} - \hat{\mu}_{a_{(2)}}\right| > \hat{\Delta}_i$, then $|\mu_{a_1} - \mu_{a_2}| > \frac{\hat{\Delta}_i}{2}$. Let $b_{i^\dagger}$ be the batch after which the gap estimation phase halts. From the above to conditions it follows that, $\frac{\Delta}{3} \leq \hat{\Delta}_{i^\dagger} \leq 2\Delta$. Since $\hat{\Delta}_i = \frac{1}{2^i}$, therefore when the algorithm halts $i^\dagger \leq \log_2(1/\Delta)$. Accounting for the failure probability $\frac{T^{-4}}{\log_2(1/\Delta)}$ in each of these $i^\dagger$ batches and taking an union bound over the batches, the bound on the estimated gap in Lemma 9 follows.

Let N_{Gap} be the total number of time steps for which the gap estimation phase lasts. We have with probability at least $1 - T^{-4}$,

$$N_{\mathsf{Gap}} \leq \frac{192k \log_2 T}{\epsilon} \sum_{i=0}^{\infty} \frac{1}{2^i}. \tag{9}$$

Bounding the geometric series in (9), we get the required bound on N_{Gap}. Finally, to obtain the bound on $\mathbb{E}[N_1]$, notice that $N_1 \leq N_{\mathsf{Gap}}$. Now using the standard trick of converting a high-probability guarantee to an expected guarantee, the proof is complete. □

D Missing Details from the Experiment Section

D.1 Experimental Setup Parameters for Section 5

All the upper confidence bound (UCB) type sampling algorithms have a common parameter, the confidence interval; same as Line 9 in Algorithm 1. For our experiment on private UCB sampling (Algorithm 1), we use a particular confidence interval, given in [8], which seems to perform the best. The confidence interval is given as, $\frac{\sqrt{r_a(t) \log t}}{n_a(t)} + \frac{\log t}{n_a(t)}$, where $r_a(t)$ and $n_a(t)$ are the reward and number of pulls for arm $a \in \mathcal{C}$ up to time t respectively. For our experiments on private Thompson sampling algorithm (see Algorithm 2) we do not implement the *gap estimation* phase and for the second phase that involves *random pullings* (see Line 10 of 2), we use a smaller value for m.

D.2 Differentially Private Contextual UCB Sampling

In this section we provide the missing details for the private contextual UCB sampling algorithm (Algorithm 3) used in our experiments section (Section 5). The confidence interval parameter in Algorithm 3 has a multiplier α in Line 7. The parameter α controls the exploration and exploitation for the algorithm, lower value of α, leads to more exploitation.

Algorithm 3 Private Contextual UCB Sampling

Input: Time horizon: T, arms: $\mathcal{C} = \{a_1, \cdots, a_k\}$, privacy parameter: ϵ, explore/exploit parameter: α, Context vector length: d.

Initialize: $A = \mathbb{I}_d$ (*Identity matrix of size-d*), $\mu = 0_d$ (*Vector of length-d with all 0 entries*), $b = 0_d$.
Create empty trees $\mathsf{Tree}_{A_{i,j}} \forall i \leq j \leq d$ and $\mathsf{Tree}_{b_i} \forall i \leq d$ with (T)-leaves. Set $\epsilon_0 \leftarrow \frac{2\epsilon}{(d^2+3d)}$.
for $t \leftarrow 1$ to T **do**
  Receive **Arm context:** $z_a(t) \forall a \in \mathcal{C}$.
  Receive $\tilde{A_{i,j}} \leftarrow$ from $\mathsf{Tree}_{A_{i,j}}$, set $\tilde{A}_{i,j} = \tilde{A}_{j,i}$ and Receive $\tilde{b_i} \leftarrow$ from Tree_{b_i}.
  if $\tilde{A}$ is positive definite **then**
    Pull arm $a^* = \arg\max_{a \in \mathcal{C}} (z_a(t)^T \tilde{A}^{-1} \tilde{b} + \alpha \sqrt{z_a(t)^T \tilde{A} z_a(t)})$, observe reward $f_t(a^*)$.
  else
    Pull arm $a^* = \arg\max_{a \in \mathcal{C}} (z_a(t)^T \tilde{b} + \alpha \sqrt{z_a(t)^T z_a(t)})$, observe reward $f_t(a^*)$.
  end if
  Insert $z_{a^*}(i) z_{a^*}(j)$ into $\mathsf{Tree}_{A_{i,j}} \forall i \leq j \leq d$ and
  $z_{a^*}(i) f_t(a^*)$ into $\mathsf{Tree}_{b_i} \forall i \leq d$, using *tree based aggregation* and privacy parameter ϵ_0.
end for

D.3 Differentially Private Contextual Thompson Sampling

In this section we provide the missing details for the private contextual Thompson sampling algorithm (Algorithm 4) used in our experiments section (Section 5). Similar to *differentially private contextual UCB sampling algorithm*, we restrict our access to the parameters which aggregate over each time stamp and use *tree based aggregation* scheme to retrieve those parameters. Additionally, since the arm set is dynamic, unlike our private Thompson algorithm (Algorithm 2), we do not run the *gap-estimation* and *random pulling* phases.

Algorithm 4 Private Contextual Thompson Sampling

Input: Time horizon: T, arms: $\mathcal{C} = \{a_1, \cdots, a_k\}$, privacy parameter: ϵ, explore/exploit parameter: α, Context vector length: d.

Initialize: $A = \mathbb{I}_d$ (*Identity matrix of size-d*), $\mu = 0_d$ (*Vector of length-d with all 0 entries*), $b = 0_d$.
Create empty trees $\mathsf{Tree}_{A_{i,j}} \forall i \leq j \leq d$ and $\mathsf{Tree}_{b_i} \forall i \leq d$ with (T)-leaves. Set $\epsilon_0 \leftarrow \frac{2\epsilon}{(d^2+3d)}$.
for $t \leftarrow 1$ to T **do**
  Receive **Arm context:** $z_a(t) \forall a \in \mathcal{C}$
  Receive $\tilde{A_{i,j}} \leftarrow$ from $\mathsf{Tree}_{A_{i,j}}$, set $\tilde{A}_{i,j} = \tilde{A}_{j,i}$ and Receive $\tilde{b_i} \leftarrow$ from Tree_{b_i}.
  if $\tilde{A}$ is positive definite **then**
    $\hat{\mu} = \tilde{A}^{-1} \tilde{b}$ and $\tilde{\mu} \sim \mathcal{N}(\hat{\mu}, \alpha \tilde{A})$.
  else
    $\hat{\mu} = \tilde{b}$ and $\tilde{\mu} \sim \mathcal{N}(\hat{\mu}, \alpha \tilde{\mathbb{I}_d})$.
  end if
  Pull arm $a^* = \arg\max_{a \in \mathcal{C}} (z_a(t)^T \hat{\mu}(t))$ and observe reward $f_t(a^*)$.
  Insert $z_{a^*}(i) z_{a^*}(j)$ into $\mathsf{Tree}_{A_{i,j}} \forall i \leq j \leq d$ and
  $z_{a^*}(i) f_t(a^*)$ into $\mathsf{Tree}_{b_i} \forall i \leq d$, using *tree based aggregation* and privacy parameter ϵ_0.
end for