Topology-Theoretic Approach To Address Attribute Linkage Attacks In Differential Privacy

Jincheng Wang, Zhuohua Li, John C.S. Lui
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
{jcwang, zhli, cslui}@cse.cuhk.edu.hk

Mingshen Sun
Department of Computer Science and Engineering
Baidu Security
City, Country
mingshen.sun@gmail.com

APPENDIX

In the appendix, we list the proof for theorems and lemmas which were proposed in previous sections.

A. Proof for Sequential Sequential Composition Theorem

Proof: For any sequence $\mathcal{M}_{[k]}(D)$ of outcomes $D_i \in Range(\mathcal{M}_i)$, for $i \in [1, k]$, the probability of output $\mathcal{M}_{[k]}(D)$ is

$$\Pr[\mathcal{M}_{[k]}(D)] = \prod_{i=1}^{k} \Pr[\mathcal{M}_i(D) = D_i].$$

By applying the definition of APL-Free ϵ-DP, and for any two neighboring datasets D and D', we have

$$\begin{aligned}\prod_{i=1}^{k} \Pr[\mathcal{M}_i(D) = D_i] &\leq (\prod_{i=1}^{k} \Pr[\mathcal{M}_i(D') = D_i] \times \exp(\epsilon_i \times |D - D'|))\\ &= \prod_{i=1}^{k} \Pr[\mathcal{M}_i(D') = D_i] \times \exp(\sum_{i=1}^{k} \epsilon_i)\\ &= \prod_{i=1}^{k} \Pr[\mathcal{M}_i(D') = D_i] \times \epsilon\end{aligned}$$

Further, according to the definition of APL-Free ϵ-DP, for any outcome D_i, there will be no APLs. Proof completes. ∎

B. Proof for Parallel Sequential Composition Theorem

Proof: Consider two neighboring datasets D and D', let $D_i = D \cap D_i$ and $D'_i = D' \cap D_i$, for $i \in [1, k]$. Then for any sequence $\mathcal{M}_{[k]}(D)$ of outcomes $D_i^{out} \in Range(\mathcal{M}_i)$, the probability of output $\mathcal{M}_{[k]}(D)$ is

$$\Pr[\mathcal{M}_{[k]}(D)] = \prod_{i=1}^{k} \Pr[\mathcal{M}_i(D_i) = D_i^{out}].$$

By applying the definition of APL-Free ϵ-DP, we have

$$\begin{aligned}\prod_{i=1}^{k} \Pr[\mathcal{M}_i(D) = D_i] &\leq (\prod_{i=1}^{k} \Pr[\mathcal{M}_i(D'_i) = D_i^{out'}] \times \exp(\epsilon_i \times |D_i - D'_i|))\\ &\leq \prod_{i=1}^{k} \Pr[\mathcal{M}_i(D'_i) = D_i^{out'}] \times \exp(\max_i \epsilon_i)\end{aligned}$$

Further, according to the definition of APL-Free ϵ-DP, for any outcome D_i, there will be no APLs. Proof completes. ∎

C. Proof for APLKiller satisfying APL-Free ϵ-DP

Proof: To prove that APLKiller satisifes APL-Free ϵ-DP, we need to prove that it satisfies those two constraints in Definition **??**. Because our handling of boundary itemsets is inspired by the topology-theoretic approach, which has been proved that there will be no APLs in the generated dataset, the first constraint is satisfied.

The only thing we need to prove is that APLKiller satisfies the second constraint. It can be checked that only two components of APLKiller use the original dataset D. The first component is LevelPart, which uses D_i to generate D'_i in Line **??**. The second component is adding boundary itemsets in Line **??**. Note that Q_i and D_{i-1} are generated in a deterministic way once D'_i is generated, so it can be derived that

$$\begin{aligned}\Pr[D_p] &= \Pr[D'_1 D'_2 ... D'_n Q_1 ... Q_n]\\ &= \prod_{i=n}^{1} (\Pr[\mathcal{L}(D_i) = D'_i] \cdot \Pr[D_{i-1}, Q_i | D'_i] \cdot \prod_{S_k^i \in Q_i} \Pr[|S_k^i| = N_k^i])\\ &= \prod_{i=n}^{1} (\Pr[\mathcal{L}(D_i) = D'_i] \cdot \prod_{S_k^i \in Q_i} \Pr[|S_k^i| = N_k^i]),\end{aligned}$$

where $\mathcal{L}$ represents APLKiller, $|S_k^i|$ is the true occurrence of kth boundary itemset in Q_i, and N_k^i is the noisy occurrence.

Because all D_i and S_k^i are disjoint, according to Theorem **??**, the total privacy budget ϵ can be assigned to the generation of each D'_i and S_k^i. Since D'_i is generated by LevelPart, and LevelPart consumes exactly ϵ privacy budget, it satisfies the second constraint. For each S_k^i, APLKiller directly generates the positive Laplace noise using privacy budget ϵ. Therefore, the APLKiller satisfies APL-Free ϵ-DP, and proof is completed. ∎

D. Proof for Theorem ??

Proof: In [**?**], it has been shown that if there is no pruning operation, the maximum number of partitioning operations to reach the leaf partition is $\frac{n-1}{f-1}$, which is the number of internal nodes in the taxonomy tree T rooted at u. Now given pruning operations, one can simply derive that $Par(u, l) \leq \frac{n-1}{f-1}$. It

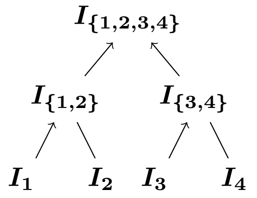

Fig. 1: Counting the number of partitioning operations needed for generating the itemset {1,3} in T. First from the level of leaf nodes, locate I_1 and I_3, then we aggregate them to nodes in the upper level. The final number of partitioning operations equals to the number of internal nodes traversed during the aggregation, which is 3.

is not difficult to show out that the number of partitioning operations equals to the number of internal nodes visited during aggregation of items from the bottom of the taxonomy tree to the top of the tree. To illustrate this fact, an example is shown in Figure 1.

Let the level of leaf nodes in T be 0. There are two cases to consider. In level $i > 0$,

1) for a l-itemset, if $l \leq \lceil \frac{n}{f^i} \rceil$, which is maximum number of nodes in level i, it will visit at most l nodes and leave $\lceil \frac{n}{f^j} \rceil - l$ number of nodes unvisited at each level $j < i$.
2) for a l-itemset, if $l \geq \lceil \frac{n}{f^i} \rceil$, it will visit at most $\lceil \frac{n}{f^j} \rceil$ nodes at each level $j \geq i$.

Consider $l \in [\frac{n}{f^{k+1}}, \frac{n}{f^k})$. For level $j \geq k+1$, all nodes can be visited. For level $j \leq k$, it will leave at least $\lceil \frac{n}{f^j} \rceil - l$ nodes unvisited. So

$$Par(u,l) \leq \frac{n-1}{f-1} - \sum_{j=1}^{k} (\lceil \frac{n}{f^j} \rceil - l),$$

and the proof is completed. ■

E. Proof for the time complexity

Proof: In each round of APLKiller, The main computational cost comes from Line **??**, where LevelPart is called to generate D'_i, given the length i.

Now let's prove the time complexity of LevelPart. **Similar with the proof for DiffPart: For each partitioning operation, the main computational cost comes from the distribution of records from a partition to its subpartitions**, and the time complexity is $O(|D_i|)$. Since the maximum number of partitioning operations, according to Theorem **??**, is bound by $\frac{n-1}{f-1}$, the time complexity of LevelPart for generating i-itemset is $O(n|D_i|)$.

Finally, for generating itemsets for all lengths, the time complexity is $O(n \sum_{i=1}^{n} |D_i|)$, which is $O(mn)$, and the proof is completed. ■