Algorithm 1 TruncateDatabase

Input: database D; privacy budget ε_1;

Output: truncated database D'

1: Read D to get the Item Universe I, number of transactions n
2: $z' = $ EstimateDistribution(D,ε_1,$|I|$)
3: Let l be the smallest integer such that $\sum_{i=1}^{l} z_i \geq 0.85$
4: **for** each transaction t in D **do**
5:      add $t' = $ RandomTruncate(t,l) to D'
6: **return** D'
7: **function** ESTIMATEDISTRIBUTION(D,ε_1,n)
8:      Let $z = [z_1, z_2, z_3, ..., z_{|I|}]$, where z_i is the number of transactions with cardinality i in D
9:      $z' = z + [Lap_1, Lap_2, Lap_3, ..., Lap_{|I|}]$, where Lap_i is drawn i.i.d. from Laplace noise ($\frac{\triangle}{\varepsilon_1}$)
10:      **return** $\frac{z'}{n}$
11: **function** RANDOMTRUNCATE(t,l)
12:      $t' = $ Random Sample min($|t|$,l) item from t
13:      **return** t'

Algorithm 2 FrequentItemsAndMISTable

Input: database D'; privacy budget ε_2; sensitivity $\triangle$; portion parameter β; threshold λ

Output: Frequent 1-itemset F_1; MISTable M

1: **for** each transaction in D' **do**
2:      **for** each item X in transaction **do**
3:          $X.sup + +$
4:      **end for**
5: **end for**
6: **for** each item X in I **do**
7:      $X.sup' = X.sup + Lap(\frac{\triangle}{\varepsilon_2})$
8:      add X and $X.MIS = max\{\beta * X.sup', \lambda\}$ to M
9:      **if** $X.sup' \geq X.MIS$ **then**
10:          add X to F
11: **end for**
12: **return** F_1; M

Algorithm 3 NoisyFP-MEtree

Input: truncated database D'; frequent 1-itemset F_1; MISTable M; sensitivity $\triangle$; privacy budget ε_3;

Output: Frequent Itemsets F

1: **for** each transaction t' in D' **do**
2:      neglect items that are not in F_1
3:      sort t' by M in ascending order
4: **end for**
5: create FPME root node $r \leftarrow \varnothing$
6: **for** each t' in D' **do**
7:      **for** each item in t' **do**
8:          **if** no corresponding path in the tree **then**
9:              create a node with sup $= \text{Lap}(\frac{\triangle}{\varepsilon_3})$
10:      **end for**
11:      increase the last node's sup by 1
12: **end for**
13: Update the tree, parent node's sup $= \sum$ children's sup
14: DFS Traversal on the FPMEtree
15: **if** a node's sup $\geq$ a node's MIS **then**
16:      add the node to F
17: **return** F