Data Recovery on Encrypted Databases With k-Nearest Neighbor Query Leakage

Evgenios M. Kornaropoulos
Brown University
evgenios@cs.brown.edu

Charalampos Papamanthou
University of Maryland
cpap@umd.edu

Roberto Tamassia
Brown University
rt@cs.brown.edu

***Abstract*—Recent works by Kellaris *et al.* (CCS'16) and Lacharité *et al.* (SP'18) demonstrated attacks of data recovery for encrypted databases that support rich queries such as range queries. In this paper, we develop the first data recovery attacks on encrypted databases supporting one-dimensional k-nearest neighbor (k-NN) queries, which are widely used in spatial data management. Our attacks exploit a generic k-NN query leakage profile: the attacker observes the identifiers of matched records. We consider both unordered responses, where the leakage is a set, and ordered responses, where the leakage is a k-tuple ordered by distance from the query point.**

As a first step, we perform a theoretical feasibility study on *exact reconstruction*, i.e., recovery of the exact plaintext values of the encrypted database. For ordered responses, we show that exact reconstruction *is feasible* if the attacker has additional access to some auxiliary information that is normally not available in practice. For unordered responses, we prove that exact reconstruction is *impossible* due to the infinite number of valid reconstructions. As a next step, we propose practical and more realistic *approximate reconstruction attacks* so as to recover an approximation of the plaintext values. For ordered responses, we show that after observing enough query responses, the attacker can approximate the client's encrypted database with considerable accuracy. For unordered responses we characterize the set of valid reconstructions as a convex polytope in a k-dimensional space and present a rigorous attack that reconstructs the plaintext database with *bounded approximation error*.

As multidimensional spatial data can be efficiently processed by mapping it to one dimension via Hilbert curves, we demonstrate our approximate reconstruction attacks on privacy-sensitive geolocation data. Our experiments on real-world datasets show that our attacks reconstruct the plaintext values with relative error ranging from 2.9% to 0.003%.

I. Introduction

Systems for *Searchable Encryption* (SE) [7], [9], [11], [12], [17], [24], [41], [42] allow a client to outsource an encrypted database to a server who can subsequently answer certain types of queries by operating *solely on the encrypted data*. In order to meet real-world efficiency demands, SE constructions allow, by definition, some well-defined *leakage* of information.

In the case of encrypted single-keyword search [7], [9], [11], [24], [42], this leakage reveals which file identifiers match the encrypted queried keyword—also known as *access pattern leakage*. The impact of this type of leakage had not been clear for a long time and it was only until recently that the community started to study its implications. In particular, the works of Islam *et al.* [23], Cash *et al.* [8], and recently Zhang *et al.* [47], demonstrate how an attacker can utilize access patterns to launch *query-recovery* attacks under various assumptions.

However, in the case of richer queries (e.g., range [16], [22], [38] and SQL [37], [39]), more severe *data-recovery* attacks are possible due to the expressiveness of the query. In particular, the work by Kellaris, Kollios, Nissim, and O'Neill [25] attacks SE-type systems that support range queries (e.g., [16], [21], [30]) by observing record identifiers whose plaintext values belong to the queried range. Similarly, a recent work by Lacharité, Minaud, and Paterson [28] further explores range query leakage to achieve exact and approximate reconstruction for the case of dense datasets with *orders of magnitude fewer queries* (when compared to [25]). Finally, order-preserving encryption based systems (e.g., CryptDB [39]) supporting even more expressive queries (such as SQL) have been shown to be vulnerable to data-recovery attacks [14], [20], [34] even without observing *any queries*, just by the setup leakage.

In this work, we explore the implications of another generic query leakage profile, that of k-nearest neighbor (k-NN) queries, which return the k nearest points of a database to a given query point with respect to a distance metric. A *spatial database* is engineered to model, store, and query data defined in a geometric space. There is a plethora of systems and products (e.g. Geomesa [3], PostGIS for PostgreSQL [5], and IBM's Cloudant NoSQL DB Geospatial [4]) that provide scalable solutions for handling spatial data. *Proximity queries* such as k-NN, appear in all of the above systems.

Support for *k-NN queries on encrypted databases* has drawn a lot of attention in the database community for more than a decade [15], [18], [26], [31], [32], [36], [43], [44], [45]. Several of the above designs, e.g. [26], [43], reveal as query leakage the k encrypted records returned to the client as response to a k-NN query. In this work, we analyze what a passive adversary can achieve by *only observing the set of encrypted records returned by a sequence of k-NN queries*. Our leakage-abuse attacks achieve significant accuracy of data recovery for one-dimensional k-NN queries. Also, as higher-dimensional data can be efficiently queried by mapping it to one-dimensional values (e.g., via Hilbert curves) [29], [35], [40], [46], our approach is applicable to a wider family of constructions. Our findings suggest a reevaluation of what is considered secure in the area of k-NN queries for encrypted databases.

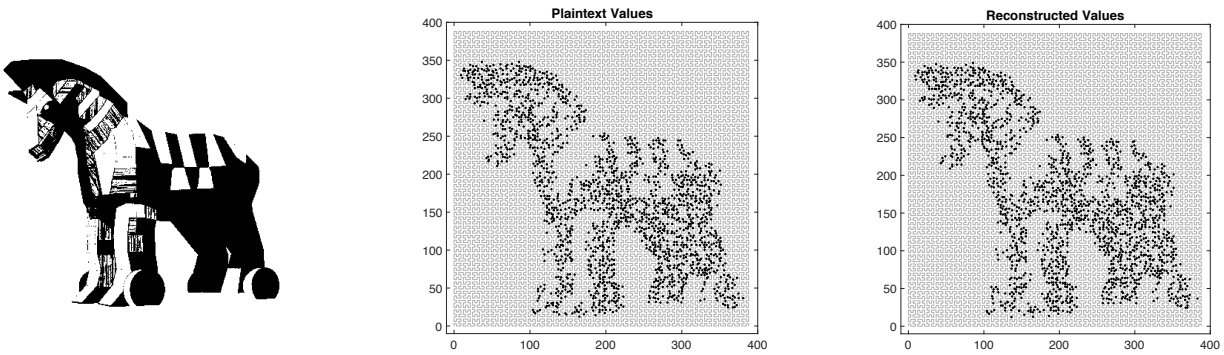

Fig. 1. On the left there is a black-and-white picture of the Trojan horse. In the middle there are $n = 1840$ sampled two-dimensional values from the original picture projected to a Hilbert curve of order 7 so as to reduce the k-NN queries to one dimension. On the right side we demonstrate the reconstruction of the plaintext values *solely based on the query leakage* under the studied assumptions. The depicted setup has relative error 0.01% both in 1D and 2D, and $k = 9$.

A. Our Contributions

We study what a *passive* and *persistent* adversary can achieve by observing the query leakage that only reveals which k encrypted records are retrieved for a private k-NN query on a database with n one-dimensional values. We study the case of **unordered responses** where the adversary observes the set of k retrieved records as well as the case of **ordered responses** where the adversary observes the k-tuple of retrieved records ordered in ascending order with respect to the private query point. We assume that the private query points are generated uniformly at random. Our **exact reconstruction results** are:

- *Ordered Responses.* We show that an adversary with auxiliary information can achieve exact reconstruction in time $O(k\,n \log n)$. This auxiliary information is rather unrealistic, e.g. the lengths of the Voronoi diagram which is a conceptual partition of the space based on DB, but our goal is to study the *feasibility* of exact reconstruction.
- *Unordered Responses.* We prove that even a computationally unbounded adversary can not achieve exact reconstruction for this generic k-NN leakage. Our impossibility proof shows that there exist an infinite number of DB reconstructions that the observed query leakage can potentially come from, thus it is infeasible for an adversary to deterministically output client's DB.

Even though from the adversarial point of view the above results do not look promising (i.e. unrealistic auxiliary information and impossibility), there is still hope. For our main results we show the following **approximate reconstruction results**:

- *Ordered Responses.* We show an attack where the adversary has *no access to auxiliary information* but still approximately reconstructs with failure probability δ the plaintext values with relative error ϵ_R in time $O(k\,n \log n + \frac{1}{\epsilon_R^2}(k^2\,n + \log \frac{1}{\delta}))$. In the heart of this technique is an estimator that *approximates* the previously-handed auxiliary information. The recovered values are at most $\pm\epsilon_R$ afar from the client's DB values with probability at least $1 - \delta$, where ϵ_R, δ are tunable.
- *Unordered Responses.* In the main result of our work we study the geometric structure of infinite reconstructions, what we call *feasible region*. Armed with insights about the geometry of this feasible region, we present a novel approximation approach that outputs a reconstructed DB with a upper-bounded worst-case reconstruction accuracy. Interestingly, the bound is a function of a characteristic quantity of the feasible region, what we call *diameter of the feasible region*, and in the evaluation section we examine the interplay between the diameter and the accuracy of the reconstruction.

Evaluation of Approximate Reconstructions. Since mapping higher-dimensional data to one dimension is a standard approach for both unencrypted [29], [35], [40], [46] and encrypted DB [26] we conduct experiments on a publicly available dataset of geolocation trace of the German politician Malte Spitz. The two-dimensional data is mapped down to one dimension, via the so-called Hilbert curves [33], where the discussed k-NN query is simulated for different values of k. All the experiments for both ordered and unordered took only a few seconds, achieved reconstruction error from 2.9% to as low as 0.003% and required the observation of thousands to hundreds of millions of queries depending on the distribution of the values. Interestingly, we used *orders of magnitude less number of queries than our theoretical analysis*.

In Figure 1 we demonstrate the accuracy of our reconstruction in a larger dataset, about 2000 data points. The original picture of the Trojan horse on the left was sampled to create the two-dimensional set of black plaintext values depicted in the middle. By projecting the 2D points to the Hilbert curve we created a 1D dataset where the k-NN query leakage was simulated (see Section V for details). After mounting our attack on unordered responses we "folded" the recovered one-dimensional dataset back to 2D to showcase its accuracy, depicted on the right. With relative error 0.01% both in 1D and 2D this visual example demonstrates the dangers of poorly understood leakage profiles.

II. Preliminaries

Database and its Organization. A database is a collection DB of n records. Let $\alpha, \beta \in \mathbb{R}$. We consider records with one-dimensional values in the continuous range $[\alpha, \beta]$ on which one-

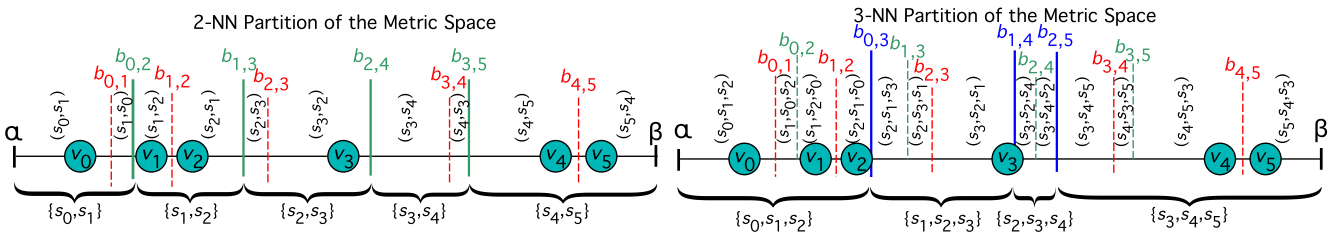

Fig. 2. The partition of $[\alpha, \beta]$ in Voronoi segments of ordered and unordered responses for $k = 2$ (left) and $k = 3$ (right). The curly brackets on the bottom indicate the *unordered responses* that correspond to each Voronoi segments. The vertically written k-tuples indicate the *ordered responses*. The term $b_{i,j}$ denotes the bisector between v_i and v_j which is also the Voronoi endpoint that separates the corresponding neighboring Voronoi segments.

dimensional k-nearest neighbor (k-NN) queries are performed. Thus, each record has two fields: (1) a unique identifier, id_i; and (2) a value $\mathbf{val}(id_i) \in [\alpha, \beta]$, for $i = 0, \cdots, n-1$. We denote with $S = (s_0, \ldots, s_{n-1})$ the sequence of record ids sorted in increasing order with respect to their values. Also, we write $v_i = \mathbf{val}(s_i)$. We denote with $\mathsf{pos}(id_i)$ the position of record id_i in sequence S. Finally we assume that a database responds to a k-NN for a fixed k decided at setup-time. For the sake of simplicity of the analysis, we assume that the mapping from records to values is injective, that is, there is a single record in the database associated with a value.

High-Order Voronoi Diagrams in One-Dimension. Given two values v_i and v_j of database DB, the *bisector* $b_{i,j}$ of v_i and v_j is the point $(v_i + v_j)/2$. For a value v_i of DB, the locus of points of $[\alpha, \beta]$ for which v_i is the nearest neighbor among the values of DB is called the *Voronoi segment* of v_i, denoted $V(v_i)$. The endpoints of $V(v_i)$ are $b_{i-1,i}$ and $b_{i,i+1}$, where we conventionally define $b_{-1,0} = \alpha$ and $b_{n-1,n} = \beta$. The *Voronoi diagram* $V(DB)$ is the partition of range $[\alpha, \beta]$ into regions associated with the Voronoi segments of DB.

The notions of Voronoi segment and Voronoi diagram can be *extended* to sets and tuples of values in DB. Given a *set* H of k values, we define Voronoi segment $V_k(H)$ as the locus of points for which the k-nearest values of every query that lands in this segment comprise set H. If H is a *tuple* of k values, we define the Voronoi segment $V_k(H)$ as the locus of points whose k-nearest values sorted from closest to furthest comprise the tuple H. Thus, for a query that lands in the a locus $V_k(H)$ the server returns the corresponding identifiers of the values of H. We define the *Voronoi diagram of order* k of DB, denoted with $V_k(DB)$, as the collection of all nonempty Voronoi segments $V_k(H)$, for all k-sized subsets (or tuples) H of values in DB. Finally, we denote with R the set of all possible responses for k-NN queries on DB.

k-NN Responses. We consider two variants of k-NN queries. If the returned response is a *set*, then we have an *unordered response*, denoted with r, which does not differentiate between closeness among the values of r to the query point. In case the response is a *k-tuple* where the order of the components indicates the closeness to the query point (from closest to furthest), then we have an *ordered response*. In this work, both type of responses are denoted with r and the exact meaning is either explicitly stated or can be inferred from the context. Figure 2 illustrates Voronoi segments for ordered and unordered responses on a database. In our work we consider k that takes values from the following range: $2 \leq k \leq \lceil \frac{n}{2} \rceil$. In case $k = 1$ it is not possible to reconstruct the order of the record identifiers due to absence of overlap in the responses. In case $k \geq \lceil \frac{n}{2} \rceil + 1$ there is at least one pair of records that appears in *all possible responses*, thus order reconstruction is not possible.

We denote with $\mathsf{Len}(r)$ the length of the Voronoi segment $V_k(r)$ associated with response r. For the case of unordered responses the set of Voronoi endpoints of $V_k(DB)$ is $\{b_{0,k}, b_{1,k+1}, \ldots, b_{n-k-1,n-1}\}$. The above set of bisectors is also denoted as B_k because each bisector refers to values that are k-positions apart wrt the ordering of S. For the case of ordered responses, the set of Voronoi endpoints of $V_k(DB)$ consists of the union of the sets of bisectors $B_1, B_2, \ldots, B_k$.

In Sections III-B through III-D and IV-A we study attacks on ordered responses and in Sections III-E and IV-B through IV-D we study attacks on unordered responses.

Adversarial Model. In our work, we assume that the adversary is *passive* and *persistent*, that is, the adversary sees all the communication between the client and the server. The goal of the adversary is to reconstruct the plaintext value of each record of the encrypted database by just observing the encrypted identifiers returned as responses to k-NN queries. If the attacker recovers the exact values, then the attack is called *exact reconstruction*. If the attacker recovers an approximation of the values, then the attack is called *approximate reconstruction* and in our work is accompanied by rigorous approximation guarantees. Our adversary does not have the power to issue queries or inject data and has no prior knowledge about the distribution of the data.

Leakage Profile Under Attack. To design generic attacks that are applicable to a family of present solutions, e.g. [26], [43], for k-NN queries, we consider a leakage profile that is typical in this line of work. Given a fixed k the *only* information that our adversary sees is the *query leakage* $\mathcal{L}_Q(DB)$ which is either the set (unordered) or the k-tuple (ordered) of the deterministically encrypted identifiers that are retrieved for an issued query. For simplicity in the rest of the work we refer to the deterministically encrypted identifiers as 'records'. The only setup leakage $\mathcal{L}_S(DB)$ that we assume is the *number* of encrypted records, n. We note here that leaking the encrypted record ids returned as responses to queries is

a standard approach in the vast majority of encrypted search constructions [7], [9], [11], [16], [22], [24], [26], [38], [42], [43] and to the best of our knowledge, it can only be avoided with heavier cryptographic primitives such as ORAM [19] and response-hiding STE [10], which negatively affect the running time and storage of the overall construction. From this leakage profile the attacker can detect which ids correspond to the two extreme values but it is not possible to differentiate between the ids of the first and the last value. Thus, all our reconstructions, similarly to [25], [28], are *correct up to reflection*.

Assumptions for Our Attacks. For our attacks, we have three assumptions:

A1 The queries observed are generated uniformly at random.
A2 The database is static, no data is updated after the setup.
A3 The boundaries α and β of the values are known.

Assumption A1, uniform query generation, appears in other leakage-abuse attacks [25], [28] and is crucial for our proposed estimation techniques. An application where assumption A2 holds is the historical geo-location trace of a user for a fixed time period, similar to the dataset in our evaluation.

Access to Auxiliary Information. In Section III, we show that an attacker who has additional knowledge can achieve *exact reconstruction*. In particular, for the results of Section III, the adversary is given the following *auxiliary information*, Aux:

- The set of *all possible ordered (resp. unordered) responses* to k-NN queries on DB, denoted with R.
- The *exact length of the Voronoi segment* for every response in R, with is modeled by oracle access to function $\mathsf{Len}(r)$ for a response r in R.

Note that set R has size $k(n-(k+1)/2)+1$ for ordered responses and $n-k+1$ for unordered responses. One might say that knowledge of the above auxiliary information by the attacker is too much to assume. Indeed, the results of Section III are primarily of *theoretical interest*. Nevertheless, they provide a sufficient condition that makes exact reconstruction feasible. Also, the attack of Section III can be modified to achieve approximate reconstruction without access to the auxiliary information. Indeed, as we show in Section IV the auxiliary information *can be approximated* by an attacker who observes a sufficently large number of queries. In particular, the attacker can (1) analyze the probability of the event of observing all the possible responses and (2) rigorously estimate the lengths of the Voronoi segments from the frequency of each response.

For the omitted proofs see the full version in [27].

III. Exact Reconstruction

In this section, we consider *exact* reconstruction attacks for k-NN queries on a one-dimensional encrypted database DB. An exact reconstruction attack is one that always and correctly retrieves the values of the underlying encrypted database by just accessing the leakage. We assume that the attacker has access to the auxiliary information, which we recall consists of the set R (all possible responses to k-NN queries) and oracle access to the function $\mathsf{Len}(r)$ that returns the length of the Voronoi segment associated with a response r in R. The auxiliary information subsumes Assumption A1, which is not necessary for the results in this section. However, we still rely on Assumptions A2 (static database), and A3 (knowledge of the range $[\alpha,\beta]$ of database values).

First, in Section III-A, we present an algorithm that reconstructs the order of the records by value given the set of all the possible responses, R, which is part of the auxiliary information. This algorithm only needs unordered responses. Next, we study the complete exact reconstruction attack for two cases: (i) ordered responses, for which we present an exact reconstruction attack (Sections III-B through III-D); (ii) unordered responses for which we show that exact reconstruction is impossible under this leakage profile (Section III-E). The following two theorems summarize the findings of this section.

Theorem 1. *Let DB be an encrypted database consisting of n records with values in the range $[\alpha,\beta]$. Assume the adversary is given the set R of all possible ordered responses to k-NN queries and oracle access to the length $\mathsf{Len}(r)$ of the Voronoi segment of each response r in R. Algorithm* AttackOrdered *achieves exact reconstruction of the values of DB, up to reflection, in $O(k\,n\log n)$ time.*

Theorem 2. *Let DB be an encrypted database with n records, and let $k\geq 2$. Given only the leakage of unordered responses to k-NN queries, it is impossible for any attacker (even computationally unbounded) to achieve exact reconstruction.*

A. Reconstructing the Order of Records

Consider a database that consists of three points x,y,z and where the set of possible unordered responses to 2-NN queries is $R=\{\{x,z\},\{y,x\}\}$. Clearly, the only possible order is $z<x<y$ (up to reflection) since x appears in both responses, i.e. overlaps, and thus x is the intermediate value. Our algorithm ReconstructOrder is a generalization of the above idea.

In particular, Algorithm 1 initially finds the identifiers for the largest and smallest values—this is easy since these are the only ones appearing in a *single* k-NN response. Then we construct the order sequence S by finding the response r that overlaps with the $k-1$ most-recently discovered entries of S, denoted in the algorithm as seq. The single remaining identifier is the one that finally extends the discovered S. See Algorithm 1 for the detailed pseudocode.

Theorem 3. *Given the set R of all possible unordered responses to k-NN queries on an encrypted database DB with n records, Algorithm* ReconstructOrder *computes the order of the records of DB with respect to their values, up to reflection, in time $O(k^2\,n)$.*

Prior Work on Order Reconstruction. The work of Lacharité *et al.* [28] also uses order reconstruction as a step for their attack on range queries leakage. In particular, the "sorting step" proposed in [28] can be directly applied to the case of k-NN queries[1]. But just this step in [28] takes $O(k\,n^3)$ time whereas our algorithm takes $O(k^2\,n)$ time *overall*.

[1]Specifically, Lines 9-15 of Algorithm 2 in [28] iteratively build a set of responses that covers the entire set of records except a single record.

Algorithm 1: ReconstructOrder

Input: Set R of unordered responses
Output: Sequence of ordered records $(s_0, \ldots, s_{n-1})$

Let Responses$[j]$ be the set of responses containing identifier j;
Let id', id'' be the identifiers that are part of *only one* response in R;
Set $s_0 \leftarrow id'$ and $s_{n-1} \leftarrow id''$;
for all $p_i \in$ Responses$[s_0] - \{s_0\}$ **do**
    ind $\leftarrow |$Responses$[p_i]|$;
    $s_{\text{ind}} \leftarrow p_i$;
end
while $k - 1 \leq$ ind $< n - 2$ **do**
    seq $\leftarrow \{s_{\text{ind}-k+1}, \ldots, s_{\text{ind}}\}$;
    Find response r from Responses$[s_{\text{ind}}]$ s.t. $|r \cap \text{seq}| = k - 1$;
    $s_{\text{ind}+1} \leftarrow r -$ seq;
    ind$\leftarrow$ ind $+1$;
end
return $(s_0, \ldots, s_{n-1})$

B. Overview of the Attack For Ordered Responses

Our proposed attack reconstructs the Voronoi diagram of the database values as an intermediate step. This task consists of finding the order of the Voronoi segments and finding the location of the Voronoi endpoints that separate the segments. As we will see, this is enough for total reconstruction. Our attack consists of five steps, which are illustrated in Figure 3.

Step-1: Reconstruct Order of Records and Relabel. We find the order of the records *with respect to their corresponding (unknown) values* by executing Algorithm ReconstructOrder, presented in Section III-A. This algorithm takes as input unordered responses, thus ignoring the order of the ids in the response tuples. The output of this step is the n-tuple of ids of DB sorted by value, denoted $S = (s_0, \ldots, s_{n-1})$.

Step-2: Find Left-to-Right Geometric Order of Voronoi Segments. We sort lexicographically the response tuples of R using the order S from the previous step. As shown in Lemma 1, the resulting sorted sequence of responses yields the left-to-right geometric order of the Voronoi segments.

Step-3: Find Bisectors Between Voronoi Segments. By definition, except for α and β, each endpoint of a Voronoi segment is a bisector of two values from DB. In the previous step, we discovered the *neighboring relation* between Voronoi segments, in this step, we further discover *which bisector corresponds to which Voronoi segment endpoint*. Towards this goal, we use Lemma 2, which shows that by comparing the ordered responses of two neighboring Voronoi segments, we can infer which bisector separates them.

Step-4: Use Voronoi Segments' Length to Find the Location of Bisectors. Starting from α, we use the left-to-right order of the Voronoi segments, and "expand" each segment by its length so as to find the exact location of each bisector.

Step-5: Use Bisector Equations to Reconstruct Encrypted Values. At this point, we have reconstructed the exact Voronoi diagram. In the final step of the attack, we take advantage of the fact that bisectors impose *constraints* on the location of the associated values. Specifically, by the definition of the bisector, the following equality holds $b_{i,j} = (v_i + v_j)/2$. Notice that as long as $k \geq 2$ then the bisectors $B_1 = \{b_{0,1}, b_{1,2}, \ldots, b_{n-2,n-1}\}$ and $B_2 = \{b_{0,2}, b_{1,3}, \ldots, b_{n-3,n-1}\}$ appear as Voronoi endpoints (see Preliminaries). Additionally, the locations of these bisectors are known from the previous steps. Therefore, by forming a system of $|B_1| + |B_2| = 2n - 3$ linear equations with the n unknowns $v_0, \ldots, v_{n-1}$, the adversary reconstructs the encrypted values. Standard algorithms for solving such a system take $O(n^c)$ time, where $c \approx 3$. In Section III-D, we prove that there is a unique solution to this system and by taking advantage of the structure of the equations, we derive a *significantly faster* reconstruction in $O(n)$ time.

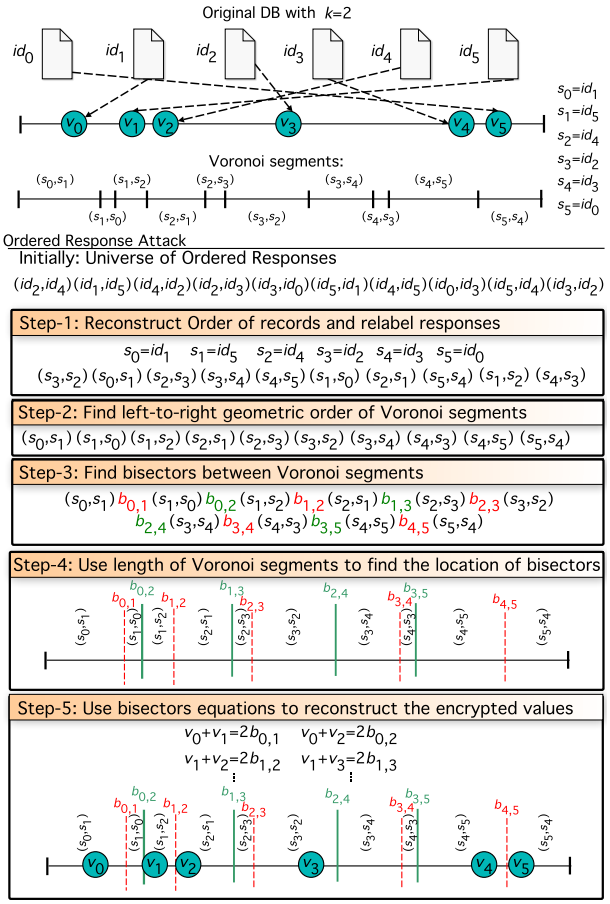

Fig. 3. An overview of the attack based on *ordered responses* where $k = 2$.

C. Ordering Voronoi Segments and Computing Bisectors

To complete the attack the attacker must order the Voronoi segments and compute the locations of the bisectors separating them. As a reminder, the ordering of the underlying identifiers is derived from Step-1 of the attack.

Lemma 1. *For a database DB with n records, let S be the sequence of identifiers sorted by increasing value. Let R be the universe of all ordered responses for k-NN queries on DB, where each response is a k-tuple of ids of DB. We have that the left-to-right geometric order of the Voronoi segments of the values of DB is given by the lexicographic order of the tuples of R with respect to the ordering of identifiers given by S.*

Finally, two neighboring Voronoi segments are separated by a Voronoi endpoint which is a bisector between two values. The next lemma explains how an attacker can infer *which bisector* separates two neighboring Voronoi segments.

Lemma 2. *Let r_{left} and r_{right} be ordered responses to k-NN queries associated with consecutive Voronoi segments. We have that k-tuples r_{left} and r_{right} differ in either:*

- *the last position, k, where the bisector that separates their segments refers to the values of the record $r_{left}(k)$ and the record $r_{right}(k)$; or*
- *two consecutive positions, l and $l+1$, where the bisector that separates their segments refers to the values of records $r_{left}(l)$ and $r_{left}(l+1)$.*

D. From Exact Bisectors to Full Reconstruction

Given the length of each Voronoi segment, given via Aux that the attacker has access to in this section, it is easy to compute the exact location of each bisector. In particular, starting from point α, we use the left-to-right order of the Voronoi segments, and "expand" each Voronoi segment by its length. Since we found which bisector separates which Voronoi segments, we can compute the exact location of every bisector. In the final step of the attack the adversary utilizes the exact locations of bisectors $B_1 = \{b_{0,1}, b_{1,2}, \ldots, b_{n-2,n-1}\}$ $B_2 = \{b_{0,2}, b_{1,3}, \ldots, b_{n-3,n-1}\}$ so as to reconstruct the *exact values* of DB. We use the relation between the bisector and the corresponding values in order to form linear equations where the unknowns are the encrypted values of DB. Specifically, from the locations of the bisectors of set B_1 we can formulate a set of $n-1$ equations, whereas from the locations of the bisectors of set B_2 we can formulate a set of $n-2$ equations. The above two sets of equations are labeled as $L_1 \ldots, L_{2n-3}$ and are depicted in the following:

$$\begin{array}{ll} L_1 : v_0 + v_1 = 2b_{0,1} & L_n : v_0 + v_2 = 2b_{0,2} \\ L_2 : v_1 + v_2 = 2b_{1,2} & L_{n+1} : v_1 + v_3 = 2b_{1,3} \\ \vdots & \vdots \\ L_{n-1} : v_{n-2} + v_{n-1} = 2b_{n-2,n-1} & L_{2n-3} : v_{n-3} + v_{n-1} = 2b_{n-3,n-1} \end{array}$$

Lemma 3. *The above linear system has a unique solution.*

Notice that each equation of the derived augmented matrix (see the proof of Lemma 3 in the Appendix) gives an expression of the corresponding value in terms of three bisectors. For example, $v_0 = b_{0,2} - b_{1,2} + b_{0,1}$ and $v_1 = b_{1,3} - b_{2,3} + b_{1,2}$ etc. As a result in AttackOrdered we don't have to solve the system of linear equations derived by the set of bisectors B_1 and B_2 which would take $O(n^c)$ time, where $c \approx 3$. Instead we use directly the derived formulas to fully reconstruct all values, which requires $O(n)$ time, as it is captured in Lines 22-25 of AttackOrdered. In terms of time complexity, Step 1 takes $O(nk^2)$, Step-2 takes[2] $O(kn\log(n))$, Step-3 & 4 take $O(k^2 n)$, and Step-5 takes $O(n)$ time.

[2]Since the total number of ordered responses is $k(n-(k+1)/2)+1$ the sorting step of that many items takes $O(kn\log(n))$

Algorithm 2: AttackOrdered

Input: Auxiliary information Aux=(R, Len), where R corresponds to the ordered responses, and $\mathsf{Len} : r \to \mathbb{R}$ is the length function where $r \in R$.
Output: Reconstructed encrypted values $v_0, \ldots, v_{n-1}$

1 $R_{set} \leftarrow$Transform each k-tuple of R to a set of size k; // Step-1
2 $(s_0, \ldots, s_{n-1}) \leftarrow \mathsf{ReconstructOrder}(R_{set})$;
3 Create an empty array VoronoiOrder ; // Step-2
4 Iterate through all $r \in R$ and add each k-tuple $(\mathsf{pos}(r(1)), \ldots, \mathsf{pos}(r(k)))$ in VoronoiOrder;
5 VoronoiOrder$\leftarrow$ Sort(VoronoiOrder,'ascending');
6 left$\leftarrow$ VoronoiOrder[1] ; // Step-3 & Step-4
7 $current\_r \leftarrow (s_{\mathsf{left}(1)}, s_{\mathsf{left}(2)}, \ldots, s_{\mathsf{left}(k)})$;
8 $covered\_area \leftarrow \alpha + \mathsf{Len}(current\_r)$;
9 **for *all*** $2 \le i \le |VoronoiOrder|$ **do**
10 left$\leftarrow$ VoronoiOrder[$i-1$], right$\leftarrow$ VoronoiOrder[i];
11 **if** *k-tuples left and right differ in only one position* **then**
12 $j \leftarrow \mathsf{left}(k)$;
13 $b_{j,j+k} \leftarrow covered\_area$;
14 **else**
15 Let x be the smallest position left and right differ;
16 $j \leftarrow \mathsf{left}(\mathsf{x})$, $j' \leftarrow \mathsf{left}(\mathsf{x}+1)$;
17 $b_{j,j'} \leftarrow covered\_area$;
18 **end**
19 $current\_r \leftarrow (s_{\mathsf{right}(1)}, s_{\mathsf{right}(2)}, \ldots, s_{\mathsf{right}(k)})$;
20 $covered\_area \leftarrow covered\_area + \mathsf{Len}(current\_r)$;
21 **end**
22 $v_0 \leftarrow b_{0,2} - b_{1,2} + b_{0,1}$, $v_1 \leftarrow b_{1,3} - b_{2,3} + b_{1,2}$; // Step-5
23 **for *all*** $2 \le i \le n-1$ **do**
24 $v_i \leftarrow b_{i-2,i} - b_{i-2,i-1} + b_{i-1,i}$;
25 **end**
26 **return** $v_0, \ldots, v_{n-1}$

E. Exact Reconstruction Impossibility for Unordered Responses

We sketch here the proof of the impossibility of exact reconstruction for the case of *unordered responses* (Theorem 2). We show that for any fixed $k \ge 2$, there exist arbitrarily many distinct databases with same unordered-responses query leakage, and thus the leakage is not enough to distinguish among them. From the leakage, we can derive the Voronoi diagram and thus the location of all the bisectors. In our proof (see the Appendix) we demonstrate how to "displace" a carefully chosen subset of values so as to create arbitrarily many distinct databases, one for every possible displacement value, while maintaining the location of the bisectors.

IV. Approximate Reconstruction

We now turn our attention to attacks that **approximate** the plaintext values when there is *no guarantee* that all possible responses are observed by the adversary and the exact Voronoi segment lengths are *not available*, i.e. *no auxiliary information*. Again, we consider ordered and unordered responses. In both cases, our approximate reconstruction fails if the adversary has not observed the set of all possible responses, R. The probability of this happening (over m uniformly distributed queries) is summarized in the following lemma.

Lemma 4. *The probability that the set of responses to m uniform k-NN queries from $[\alpha, \beta]$ does not contain the set of all possible ordered (unordered) responses, R, is at most*

$$|R| e^{-\frac{m}{\beta-\alpha} \min_{r \in R} \mathsf{Len}(r)},$$

where $\mathsf{Len}(r)$ *is the length of the Voronoi segment of* r.

With reference to Lemma 4, recall that the size of the set R of all possible responses to k-NN queries on a database with n records is $|R| = k(n-(k+1)/2)+1$ for ordered responses and $|R| = n-k+1$ for for unordered responses. The attacker can verify whether all responses are observed since we know n from the setup leakage and k from the query leakage. Note that for a fixed number of queries, the smaller the length of the minimum Voronoi segment, the larger a probability of failure of the attacks. Namely, our approximate reconstruction attack fails with the probability given in Lemma 4 due to not having observed all the responses. However, for unordered responses, the attack can fail for another reason as well. In particular, as discussed in Section IV-D, the attacker picks its output based on an estimated k-dimensional polytope. Thus, if the estimated polytope is *empty*, the attack fails.

A. Ordered Responses: Estimating Voronoi Segment Lengths

Given all possible responses R have been observed with m uniformly generated queries in $[\alpha, \beta]$, our approximate reconstruction attack is a simple modification of attack AttackOrdered presented in the Section III-D. In particular, instead of assuming oracle access to function $\mathsf{Len}(r)$ at Line 20 of AttackOrdered, we estimate $\mathsf{Len}(r)$ as

$$(\beta - \alpha) \cdot \frac{m_r}{m}\,, \tag{1}$$

where m_r is the number of queries (out of m total queries) that returned r as a response. The resulting reconstruction attack achieves *approximate reconstruction (up to reflection) with rigorous guarantees*:

Theorem 4. *Let DB be an encrypted database with n records whose values are in the range $[\alpha, \beta]$. Suppose the attacker observes the responses to m k-NN queries that are uniformly generated from $[\alpha, \beta]$ (Assumption A1) and which contain all possible ordered responses, R. For any $0 < \epsilon < \beta - \alpha$ and $0 < \delta < 1$, the variation of Algorithm* AttackOrdered *which estimates Voronoi segment lengths using Equation 1 computes in $O(m + k\,n \log n)$ time a sequence of reconstructed values such that each reconstructed value differs from its original value by at most ϵ with probability at least $1 - \delta$, provided m is at least*

$$\max\left\{\frac{180(\beta-\alpha)^2(k(n(k+1)/2)+1)}{\epsilon^2}, \frac{225(\beta-\alpha)^2(\ln 3 - \ln \delta)}{\epsilon^2}\right\}.$$

B. Unordered Responses: Defining the Reconstruction Space

As we saw in Section III-E, in the case of unordered responses, there are more than one databases DB that map to the same query leakage $\mathcal{L}_Q(DB)$. Our first step in developing an attack is to study the space of potential reconstructions.

We first define the *universe* of all reconstructions. Let $\mathcal{V}^n$ be the set of n-tuples $v \in \mathcal{V}^n$ such that $v_0, \ldots, v_{n-1} \in [\alpha, \beta]$ and $v_0 < v_1 < \ldots < v_{n-1}$. A reconstruction algorithm returns an n-tuple from the set $\mathcal{V}^n$. We show how $\mathcal{V}^n$ can be partitioned based on the concept of Voronoi diagrams of order k.

Definition 1. *Let $\mathcal{V}^n$ be the set of ordered n-tuples with distinct values from the range $[\alpha, \beta]$. We define the* ***reconstruction relation*** *$\mathcal{P}_k$* ***with respect to*** *k as a subset of $\mathcal{V}^n \times \mathcal{V}^n$ such that if $(v, v') \in \mathcal{P}_k$, also denoted as $v \sim_k v'$, then $V_k(v) = V_k(v')$ where $V_k(\cdot)$ is the k-th order Voronoi diagram.*

To put it simply, if we have two n-tuples v, v' where the reconstruction relation $\mathcal{P}_k$ holds then they have the same k-th order Voronoi diagram. A corollary of the above definition is that the reconstruction relation is an *equivalence relation*.

Definition 2. *Given the reconstruction relation $\sim_k$ and $v \in \mathcal{V}^n$, we define $[v]$ the* ***reconstruction class of*** *v, as :*

$$[v] = \{v' \in \mathcal{V}^n : v \sim_k v'\}.$$

It is easy to prove that the reconstruction class of v is also an *equivalence class*. From the properties of equivalence classes we know that every two equivalence classes $[v]$ and $[v']$ are either equal or disjoint. Additionally, the set of all equivalence classes of $\mathcal{V}^n$ forms a partition of $\mathcal{V}^n$, i.e. every n-tuple $v \in \mathcal{V}^n$ belongs to one and only one equivalence class. Given an equivalence class $[v]$, a *representative* for $[v]$ is an n-tuple of $[v]$, i.e. it is a $v' \in \mathcal{V}^n$ such that $v' \sim_k v$.

One of the n-tuples of the reconstruction class is the *original database*, but given the considered leakage profile our attacker *does not know* which one. We focus on attacks that return a representative of the correct[3] reconstruction class. The success of the reconstruction attack is measured using the L_∞ distance metric, also known as Chebyshev distance, so as to capture the largest error among all reconstructed values. More formally:

$$d_{L_\infty}(v, v') = \max_{0 \le i < n}(|v_i - v'_i|).$$

The next step of our analysis is to characterize the n-tuples of a given reconstruction class.

Characterization via Offsets ξ from Bisectors. In order for two n-tuples to be in the same reconstruction class they must have the same k-th order Voronoi diagram. Therefore the Voronoi endpoints (i.e. the bisectors) must be in the same fixed location. In this approach we model the n-tuples of the reconstruction class using *only* k unknown offsets from the above fixed bisectors, as opposed to n unknowns of a naive approach. To give an intuitive explanation of our attack we proceed with an illustration of our result for $k = 2$, the general case where $k > 2$ is addressed in Section IV-D.

Analyzing Case $k = 2$. Our goal is to *characterize the space* of n-tuples of a reconstruction class given its 2nd order Voronoi diagram. We will demonstrate that we need to define unknowns for the location of only two values and the rest of the $n-2$ values can be expressed as function of these two. The unknown variables are $\xi = (\xi_0, \xi_1)$ and their geometric description follows. Let the first value v_0 be ξ_0 to the left of the bisector $b_{0,2}$. Since the location of $b_{0,2}$ is fixed, it follows that the value v_2 must be ξ_0 to the right of the bisector $b_{0,2}$. Using the formulated equation $v_2 = b_{0,2} + \xi_0$ we can express v_4 so as v_4 and v_2 are equidistant from the fixed location of $b_{2,4}$. Using the same reasoning let v_1 be ξ_1 to the left of the bisector $b_{1,3}$. The location of values $v_3, v_5, v_7, \ldots$ can be expressed as a function

[3]Other attack techniques might be possible if the attacker is willing to output an n-tuple from *any* (and possibly incorrect) reconstruction class.

of the offset ξ_1 and the location of the relevant bisectors. As one can easily see, by picking a value for the unknown ξ_0 we fix the location of v_0 (resp. v_1) which in turn has *a domino effect* on the location of $v_2, v_4, v_6, \ldots$ (resp. $v_3, v_5, v_7, \ldots$). Figure 4 highlights which values can be expressed as a function of the unknown offsets ξ_0, ξ_1. Specifically for $n = 10$ we have:

$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$
$$v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0$$
$$v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0$$
$$v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0$$
$$v_1 = b_{1,3} - \xi_1$$
$$v_3 = b_{1,3} + \xi_1$$
$$v_5 = 2b_{3,5} - v_3 = 2b_{3,5} - b_{1,3} - \xi_1$$
$$v_7 = 2b_{5,7} - v_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1$$
$$v_9 = 2b_{7,9} - v_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1$$

The next lemma describes the closed-form of each value as function of the appropriate bisectors and offset for any $k \geq 2$.

Lemma 5. *Let $V_k(v)$ be the Voronoi diagram of the reconstruction class $[v]$ with Voronoi endpoints $b_{0,k}, \ldots, b_{n-k-1,n-1}$. If an n-tuple v' belongs to $[v]$, there exists a k-tuple denoted as ξ where $\xi_0, \ldots, \xi_{k-1} \geq 0$ such that for all $0 \leq i \leq n-1$:*

$$v'_i = \begin{cases} b_{i,i+k} - \xi_i & ,for\ 0 \leq i < k \\ b_{i \bmod k, i \bmod k+k} + \xi_{i \bmod k} & ,for\ k \leq i < 2k \\ (-1)^{\lfloor i/k-1 \rfloor}(b_{i \bmod k,(i \bmod k)+k} + \xi_{i \bmod k}) + & ,for\ 2k \leq i \leq n-1 \\ + \sum_{2 \leq j \leq \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2b_{(i \bmod k)+(j-1)k,(i \bmod k)+jk}, & \end{cases}$$

*We call the k-tuple $\xi = (\xi_0, \ldots, \xi_{k-1})$ the **offset vector of** v'.*

After characterizing the n-tuples of $[v]$ using 2 unknowns we address the following question: *What values can ξ take so as to give an n-tuple that belongs to $[v]$?*

Ordering Constraints. We define the *ordering constraints* as the inequalities that guarantee that two consecutive values do not surpass each other, e.g. $v_0 \leq v_1$, $v_1 \leq v_2$ etc. By substituting the formulas for $v_0, \ldots, v_{n-1}$ defined in Lemma 5, we get inequalities with the unknowns ξ_0 and ξ_1. As an example, for $n = 10$ we have the following inequalities:

$v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1}$, where $c_{0,1} = (b_{1,3} - b_{0,2})$

$v_1 < v_2 \Rightarrow -\xi_0 - \xi_1 < c_{1,2}$, where $c_{1,2} = -(b_{1,3} - b_{0,2})$

$v_2 < v_3 \Rightarrow \xi_0 - \xi_1 < c_{2,3}$, where $c_{2,3} = (b_{1,3} - b_{0,2})$

$v_3 < v_4 \Rightarrow \xi_0 + \xi_1 < c_{3,4}$, where $c_{3,4} = (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$

$v_4 < v_5 \Rightarrow -\xi_0 + \xi_1 < c_{4,5}$, where $c_{4,5} = 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$

$v_5 < v_6 \Rightarrow -\xi_0 - \xi_1 < c_{5,6}$

, where $c_{5,6} = 2(b_{4,6} - b_{3,5}) - (b_{2,4} - b_{0,2}) - (b_{2,4} - b_{1,3})$

$v_6 < v_7 \Rightarrow \xi_0 - \xi_1 < c_{6,7}$

, where $c_{6,7} = 2(b_{5,7} - b_{4,6}) - 2(b_{3,5} - b_{2,4}) + (b_{1,3} - b_{0,2})$

$v_7 < v_8 \Rightarrow \xi_0 + \xi_1 < c_{7,8}$

, where $c_{7,8} = 2(b_{6,8} - b_{5,7}) - 2(b_{4,6} - b_{3,5}) + (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$

$v_8 < v_9 \Rightarrow -\xi_0 + \xi_1 < c_{8,9}$

, where $c_{8,9} = 2(b_{7,9} - b_{6,8}) - 2(b_{5,7} - b_{4,6}) + 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$

By examining the first four inequalities one can see that the first $\xi_1 \leq \xi_0 + c_{0,1}$ and the third $\xi_1 \geq \xi_0 - c_{2,3}$ concern half-planes based on *parallel lines* with positive slope. The y-intercept of the first is positive while the y-intercept of the third is negative. Similarly, the second inequality $\xi_1 \geq -\xi_0 - c_{1,2}$ and the fourth $\xi_1 \leq -\xi_0 + c_{3,4}$, concern half-planes based on parallel lines with negative slope and y-intercepts that are negative and positive, respectively. If we extend this reasoning to the rest of the constraints we see that we can partition the ordering constraints in four categories of half-planes, i.e. 1) $slope = 1$ and positive y-intercept, 2) $slope = 1$ and negative y-intercept 3) $slope = -1$ and positive y-intercept, and 4) $slope = -1$ and negative y-intercept. From each of the four categories *all but one constraint are redundant*. By omitting redundant constraints we do not change the region that satisfies the constraints. To find the *non-redundant* one can go through the y-intercepts of each category and remove the overlapping constraints, can be accomplished in time $O(n)$.

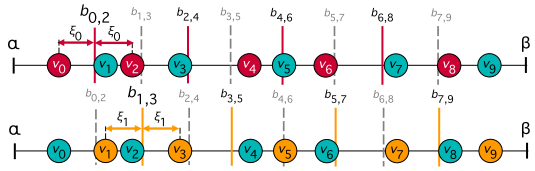

Fig. 4. Values v_0 and v_2 must be equidistant, specifically ξ_0 afar, from $b_{0,2}$. Using the derived equations we can express the locations of v_2, v_4, v_6, v_8 because the locations of bisectors $b_{0,2}, b_{2,4}, b_{4,6}, b_{6,8}$ stay fixed. Similarly, by using offset ξ_1 we can express the locations of v_3, v_5, v_7, v_9.

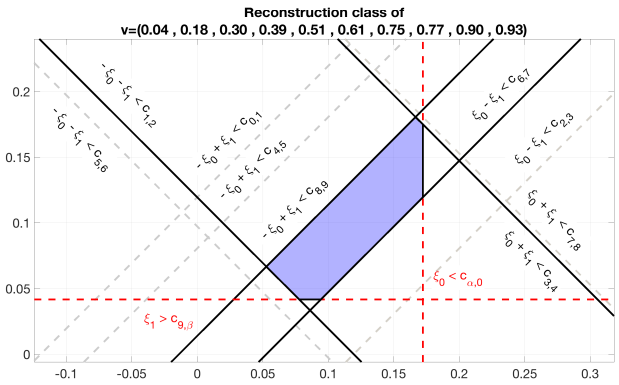

Fig. 5. The possible values of ξ_0, ξ_1 in X- and Y-axis respectively for a given $[v]$. The feasible region is colored with blue. The set of redundant ordering constraints are depicted with gray dotted-lines, and the non-redundant ordering constraints with black bold-lines. The boundary constraints are depicted with red dotted-lines.

Boundary Constraints. We need two inequalities to guarantee that the first/last value do not surpass the boundary of the range of values, i.e. $[\alpha, \beta]$. For the example where $n = 10$ we have 1) $\alpha < v_0 \Rightarrow \xi_0 < c_{\alpha,0}$ where $c_{\alpha,0} = b_{0,2} - \alpha$, and 2) $v_9 < \beta \Rightarrow \xi_1 > c_{9,\beta}$ where $c_{9,\beta} = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \beta$.

The Reconstruction Class: A Convex Polygon. The pairs of feasible offset values (ξ_0, ξ_1) is the set of values that satisfy: 1) the four non-redundant ordering constraints as well as 2) the two boundary constraints. Figure 5 gives a detailed geometric illustration of the feasible region for the running example. Depending on the values of the database the boundary constraints can be redundant. Generally, we denote the **feasible region of reconstruction class** $[v]$ as $\mathcal{F}_{[v]} = \{\xi' \in \mathbb{R}^k : A \cdot \xi' \leq c\}$, where each row of $A \cdot \xi' \leq c$ represents a constraint on ξ. Overall, we have (1) **Ordering constraints:** $n-1$ in number, (2) **Boundary constraints:** two in number, (3) **Positive-offset**

constraints: k constraints to guarantee that the offsets are positive. Therefore, A is a $(n+k+1)\times k$ matrix of coefficients for the inequalities, ξ is a column vector with k offsets, and c is the column vector with the $n+k+1$ constants (i.e. $c_{a,0}, c_{n-1,b}, c_{0,1}, c_{1,2}, \ldots, c_{n-2,n-1}$). Since we only have *linear* inequalities in $\mathcal{F}_{[v]}$, the region is a *convex polytope*.

Diameter of the Feasible Region. Given the feasible region $\mathcal{F}_{[v]}$ of the reconstruction class $[v]$, the L_∞ distance between a pair of n-tuples $v', v'' \in [v]$ of the class is:

$$d_{L_\infty}(v', v'') = \max_{0\le i\le n-1}(\{|v'_i - v''_i|\}) = \max_{0\le i\le k-1}(\{|\xi'_i - \xi''_i|\}$$
$$= d_{L_\infty}(\xi', \xi'') \le d_{L_2}(\xi', \xi'') \le diam(\mathcal{F}_{[v]}),$$

where the second equality is derived by substituting the values with the offset formulas of Lemma 5. The *polytope diameter* $diam(\cdot)$, or simply diameter, is the largest Euclidean distance between any pair of vertices of the polytope. Therefore if the attacker is able to compute $\mathcal{F}_{[v]}$ he can compute an upper-bound of the distance of any pair of n-tuples in the reconstruction class. We note here that the final output of the reconstruction attack is a representative v^* of the reconstruction class $[v]$, and that the original database can be *any* n-tuple of the reconstruction class. The last key observation of the attack is that if the attacker outputs v^* for which the offset vector is the *mean* of the offsets ξ', ξ'' of the diameter, then *all* the potential original database n-tuples are at most $diam(\mathcal{F}_{[v]})/2$ distance afar.

C. Overview of the Unordered Response Attack

We give an overview of the approximate reconstruction for $k = 2$. See Section IV-D for a generalization.

Step 1. The attacker reconstructs the order of the records with respect to their (unknown) values by using the algorithm ReconstructOrder. After relabeling the record ids using $S = (s_0, \ldots, s_{n-1})$ the attacker computes the left-to-right order of the Voronoi segments. For the case of unordered responses this step is straightforward and can be done by just "shifting" a k-length window over the sequence S, e.g. the left-to-right order is $\{s_0, \ldots, s_{k-1}\}, \{s_1, \ldots, s_k\}, \ldots, \{s_{n-k-1}, \ldots, s_{n-1}\}$. For the case of unordered responses assigning the bisectors to Voronoi endpoints is straightforward as well. The corresponding left-to-right order of the bisectors is $b_{0,k}, b_{1,k+1}, \ldots, b_{n-k-1,n-1}$. This attack differs significantly from the the Ordered Responses Attack in the next two steps.

Step 2: Estimate the Constraints of the Feasible Region. There are infinitely many value n-tuples for DB that can give a fixed k-th order Voronoi diagram. The next step of our attack characterizes the set of *all such* n*-tuples* using only k unknowns, namely the offsets $\xi = (\xi_0, \ldots, \xi_{k-1})$. We define a set of linear constraints, namely the ordering, the boundary, and the positive-offset constraints, imposed on the unknowns ξ so as to find the offsets assignments that correspond to a valid n-tuple of the reconstruction class. Each constraint imposed on ξ is a half-space and the intersection of these constraints defines the *feasible region* $\mathcal{F}_{[v]}$. Geometrically, $\mathcal{F}_{[v]}$ is a bounded convex k-dimensional polyhedron, i.e. a polytope. But since the constraints' constants *are not known to the attacker* we propose a way to estimate them. In particular the new algorithm estimates the right-hand side constant of each constraint, e.g. estimation of terms $c_{\alpha,0}, c_{0,1}, \ldots, c_{4,5}, c_{5,\beta}$ in Figure 6. The key observation is that each c term can be expressed as the *linear combination of lengths* of Voronoi segments, e.g. in Figure 6 term $c_{4,5}$ involves $\mathsf{Len}(\{s_1, s_2\})$ and $\mathsf{Len}(\{s_3, s_4\})$. Our estimator uses the frequency of each unordered response to estimate the appropriate linear combination of lengths of each constraint with rigorous probabilistic guarantees.

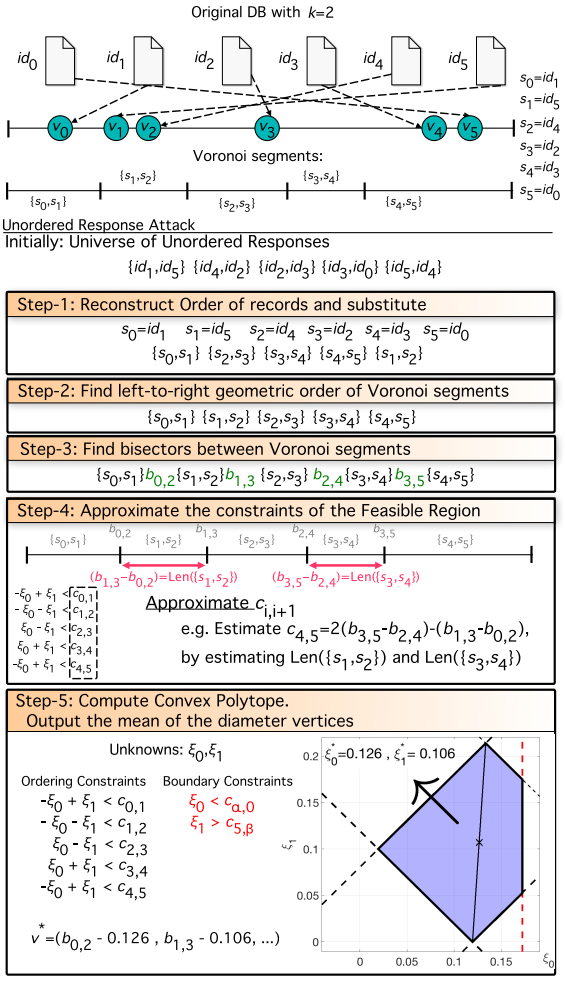

Fig. 6. An overview of the attack based on *unordered responses* for $k = 2$.

Step 3: Compute Convex Polytope & Output the Mean of the Polytope Diameter. At this point we have estimated the feasible region of the offset vector ξ, depicted in Figure 6. As a next step the attack utilizes a solver for the *Vertex Enumeration Problem* [6] which takes as an input the linear inequalities (i.e. the constraints) and outputs the *vertices* on the boundary of the feasible region $\mathcal{F}_{[v]}$. Having the coordinates of the vertices of $\mathcal{F}_{[v]}$, our attack can compute the *diameter* of the convex polytope. As it is shown in Theorem 5, the offset ξ^*, which is defined as the mean of a pair of polytope-vertices that constitute the diameter, gives a representative v^* that has distance at most $diam(\mathcal{F}_{[v]})/2$ from *all the* n*-tuples of the reconstruction class*, including the (unknown) original database.

D. Unordered Responses: Reconstruction for $k \geq 2$

Estimating the Constraints. We define the tuple L as:

$$L = \big(\mathsf{Len}(\{s_0, \ldots, s_{k-1}\}), \ldots, \mathsf{Len}(\{s_{n-k}, \ldots, s_{n-1}\})\big)$$

, where $\mathsf{Len}(\cdot)$ indicates the length of the Voronoi segment that is given as an input. Our goal is to estimate the expression of each constraint and to achieve this the next (simplified) lemma is of great importance. The analytical formulas for the constraints can be found in Lemma 7 in the Appendix. Specifically, the following lemma shows that each of the ordering constraints (same argument holds for the boundary) can be expressed as a simple linear combination of ξ and L where the coefficients are known. Since we can estimate the lengths of L using Equation (1) we have a way to estimate the constraints as well. The proof performs a case analysis of the inequality $v_i < v_{i+1}$ based on the value of i with respect to the formulas of Lemma 5.

Lemma 6. *The inequality $v_i < v_{i+1}$ for $0 \leq i \leq n-2$ can be expressed as $f_i^{Left} \cdot \xi^T \leq f_i^{Right} \cdot L^T$, where ξ is the unknown offset row vector and f_i^{Left}, f_i^{Right} are row vectors of constant coefficients. All but two entries of f_i^{Left} are zero. Vector f_i^{Right} has at most $\lfloor (n-1)/k \rfloor + k$ non-zero terms which come from the set $\{-2, -1, 1, 2\}$.*

A similar lemma can be formed for the boundary constraints. The values of the coefficients of f_i^{Left} and f_i^{Right} can be easily computed since they only depend on i, n, k and can be found in the Appendix. The ConstraintEstimation algorithm focuses on estimating $c_{i,i+1} = f_i^{Right} \cdot L^T$ and performs the following series of actions for each boundary and ordering constraint: given i, n, k compute the coefficients f_i^{Left}, f_i^{Right}, scan the multiset of unordered responses U that come from uniformly generated queries and record the observed frequency of each relevant entry of L, use the coefficients f_i^{Right} and the values in L^T to finalize the estimation of the terms $c_{i,i+1}$.

High-Level Description of ConstraintEstimation

- **Input:** Multiset U of responses from queries generated uniformly at random. Order S of the ids wrt their values. Boundaries α, β.
- **Step 1:** Let L be the $(n-k+1)$-tuple of the labels of the (unknown) lengths of the Voronoi segments. Calculate the formula of each *ordering constraint* wrt ξ and compute the coefficients f_i^{Right} of $c_{i,i+1}$. Given the analytical formula of each term $c_{i,i+1}$ define $T_{i,i+1}$ to be the set of triplets (lbl, cfc, cnt), where lbl is the label of the participating length from L, cfc is the coefficient of this lbl in the formula, and cnt is a counter initialized to zero.
- **Step 2:** Similarly, calculate the analytical formulas of the terms $c_{\alpha,0}$ and $c_{n-1,\beta}$ of the *boundary constraints*. Define the sets of triplets $T_{\alpha,0}$ and $T_{n-1,\beta}$. Let T be the collection of sets $\{T_{\alpha,0}, T_{0,1}, \ldots, T_{n-2,n-1}, T_{n-1,\beta}\}$.
- **Step 3:** For each response $r \in U$ find the sets from collection T where the term $\mathsf{Len}(r)$ is part of a lbl and increase the corresponding cnt entry by one.
- **Step 4:** Set all estimations $\widetilde{c}_{i,i+1}$ to zero. For each set of collection T, go through all the triplets. For each triplet of $T_{i,i+1}$, multiply cfc with the counter cnt and add the result to $\widetilde{c}_{i,i+1}$.
- **Step 5:** Multiply each of the $\widetilde{c}_{i,i+1}$ by $(\beta - \alpha)$, divide the result by $|U|$, and store the final result at $\widetilde{c}_{i,i+1}$.
- **Step 6:** Output $\widetilde{c}_{\alpha,0}, \widetilde{c}_{0,1}, \ldots, \widetilde{c}_{n-2,n-1}, \widetilde{c}_{n-1,\beta}$.

Attack Algorithm. The attack algorithm utilizes algorithm ConstraintEstimation to approximate the inequalities of the feasible region, i.e. get the estimates $\tilde{c}$. The final set of inequalities is captured by the expression $A \cdot \xi \leq \tilde{c}$ where by a linear scan the attacker can remove the redundant constraints (Line 3 of the algorithm). Notice that the ordering constraints concern a pair of consecutive values and by substituting from Lemma 5 we finally get a constraint on a pair of offsets that appear consecutively in the cyclical ordering $\hookrightarrow \xi_0 \rightarrow \xi_1 \rightarrow \ldots \rightarrow \xi_{k-1}$. Due to the periodicity on the cyclical ordering (see the Appendix for the closed form) we have $2k$ non-redundant ordering constraints among the total $n-1$. For $n > 2k$ the polytope is bounded and thus the solver of the vertex enumeration problem [6] returns the vertices of the k-dimensional polytope formed by $A \cdot \xi \leq \tilde{c}$ in $O(k^2\, z)$ time, where z is the number of vertices of the polytope. In general, z could be as large as 2^k. Thus, our approach is suitable for small values of k, which is typical in practical scenarios where k is often a small constant.

We note here that in case the estimation of the constraints is not "accurate enough", which depends on the distribution of the values, the feasible region might be empty. In this case the solver will return an empty set and the attack will fail since no offset can meet the (not adequately) approximated constraints. Given the vertices we can compute the diameter of the polytope of $\mathcal{F}_{[v]}$ in time quadratic in the number of vertices. So as a last step our attack returns the mean of the diameter vertices which guarantees that all the n-tuples of the class are at most $diam(\mathcal{F}_{[v]})/2$ distance afar.

Algorithm 3: AttackUnordered

Input: Response multiset $U = \{r_1, r_2, \ldots\}$, Boundaries α, β
Output: Reconstructed values $(v_0^*, \ldots, v_{n-1}^*)$ or $\perp$

1 $S \leftarrow \mathsf{ReconstructOrder}(U)$, $\tilde{c} \leftarrow \mathsf{ConstraintEstimation}(U, S, \alpha, \beta)$;
2 Compute the $(n+k+1) \times k$ matrix A of coefficients such that each line of $A \cdot \xi < \tilde{c}$ represents a constraint, ξ is the column vector with k offsets, $\tilde{c}$ is the column vector with $(n+k+1)$ entries of the constants;
3 Remove the redundant constraints from A;
4 Deploy an algorithm that solves the 'Vertex Enumeration Problem' with input $A \cdot \xi \leq \tilde{c}$ and output a matrix Ξ of k columns where each row represents a vertex of the convex polytope of the feasible region;
5 **If** Ξ is non-empty **then** compute the Euclidean distance between every pair of rows (i.e. vertices) of Ξ and record the pair (ξ', ξ'') with the maximum distance, **else** return $\perp$;
6 Compute ξ^* as the mean of ξ' and ξ'';
7 Use the offset $\xi^* = (\xi_0^*, \ldots, \xi_{k-1}^*)$ in the expressions of Lemma 5 to compute the corresponding value $v^* = (v_0^*, \ldots, v_{n-1}^*)$;
8 **return** $(v_0^*, \ldots, v_{n-1}^*)$

Theorem 5. *Let DB be an encrypted database with n records whose values are in the range $[\alpha, \beta]$. Suppose the attacker observes the responses to m k-NN queries uniformly generated from $[\alpha, \beta]$ (Assumption A1) and which contain the set of all possible unordered responses, R. For any $0 < \epsilon < \beta - \alpha$ and $0 < \delta < 1$, Algorithm* AttackUnordered *runs in time $O(m + k^2 z + z^2)$, where z is the number of vertices of the feasible region, $\mathcal{F}_{[v]}$, and returns either $\perp$ (failure) or a sequence of reconstructed values (success) such that each reconstructed value differs from its original value by at most $\frac{diam(\widetilde{\mathcal{F}_{[v]}})}{2} + \epsilon$*

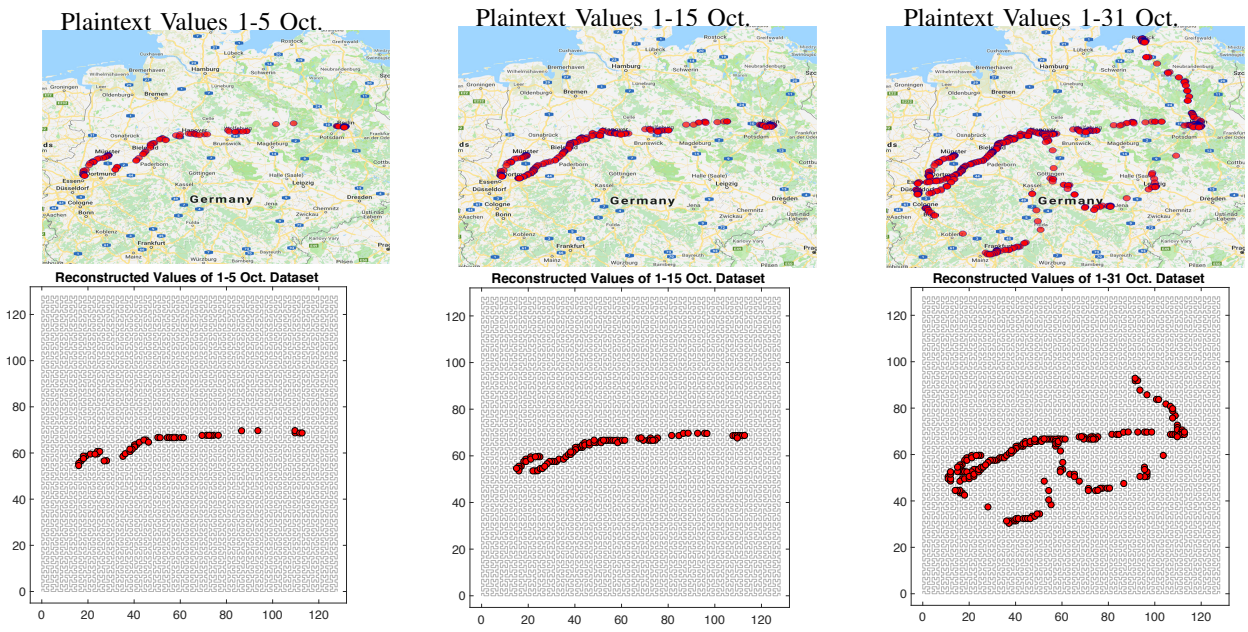

Fig. 7. Three date ranges for the month October of the publicly available mobile records with the geolocation of the German Green party politician Malte Spitz. on the first row we demonstrate the original dataset and in the bottom the accuracy of the reconstruction that we achieve for unordered responses for $k = 2$.

with probability at least $1 - \delta$*, provided* m *is at least*

$$\max\left\{\frac{25(\beta-\alpha)^2(\ln 3 - \ln \delta)}{\epsilon^2}, \frac{20(\beta-\alpha)^2(n-k+1)}{\epsilon^2}\right\}$$

On the Size of the Diameter Since the approximation is a function of a quantity that depends on the distribution of the data, we further study the possible values that $diam(\mathcal{F}_{[v]})$ can take. In the next theorem we show that the $3k$ consecutive values that are within the smallest possible γ range give an upper-bound on the diameter of $\mathcal{F}_{[v]}$. Thus, a small concentrated number of consecutive values affects heavily the diameter of $\mathcal{F}_{[v]}$. We note here that the smaller the γ the higher the number of samples required to achieve meaningful approximation guarantees since the sample size is a function of the length of the smallest Voronoi segment, so there is an inherent trade-off.

Theorem 6. *Let* $V_k(v)$ *be the Voronoi diagram of reconstruction class* $[v]$*, and let* v' *be an* n*-tuple such that* $v' \in [v]$*. If there are* $3k$ *values of* v' *within an* γ *range in* $[\alpha, \beta]$ *then we have* $diam(\mathcal{F}_{[v]}) \leq 2\gamma$*.*

V. Evaluation of Approximate Reconstruction

In our evaluation, we test our reconstruction attacks on encrypted versions of databases, e.g. [26], that reduce their two-dimensional data to one dimension via Hilbert curves [33].

Mapping 2D Data to 1D via Hilbert Curves. Organizing *multidimensional data* for efficient access and indexing is a challenging problem due to the lack of a total ordering that preserves locality. *Space-filling curves* [33], which map points in a high-dimensional space onto one-dimensional points while preserving *locality* and proximity relations, have been thoroughly explored in spatial data management. See, e.g., [29], [35], [40], [46]. These curves essentially span the desired higher-dimensional space, with granularity tuned by the so-called *order* of the curve. The higher the order the better the approximation of locality. The second row of Figure 7 shows an example of a Hilbert curve of order 7 that spans a square in the two-dimensional space. In particular, this *single continuous* line of gray color that starts at (0,0) and ends at (0, 2^7) gives a $2^7 \times 2^7$ grid of points. A value of the DB in the two-dimensional space is projected to the closest *segment of the curve*. By "untangling" the curve we get a single straight line segment where all the projected values are within the boundaries $\alpha = 0$ and $\beta = 2^7 \times 2^7 = 2^{14}$. Conceptually, to run a (non-secure) k-NN query it is enough to traverse the one-dimensional points of DB towards the left and the right of the projections of the *query point* on the curve. Due to the properties of Hilbert curves, the set of k-NN on the one-dimensional space is an approximation of the neighbors in the two-dimensional space.

Dataset & Experiment Design. The dataset SpitzLoc [2], also used in [14], consists of the latitude and longitude of the German Green party politician Malte Spitz over a period of six months. A record is stored whenever the phone was connected to a cell tower, received a call or sent a text. We used the two-dimensional data from the date ranges 1-5 Oct., 1-15 Oct., and 1-31 Oct., depicted in Figure 7. In all of our experiments we used a Hilbert curve of order 7 and placed the geolocation data in the center of the above Hilbert curve, the size n of each dataset is denoted in Table I. We simulated the k-NN query leakage of this setup and recorded the quality of the reconstruction for different values of $k = \{2, 5, 8\}$ and number of queries m. The quality measures are the Chebyshev distance between the original values and the reconstruction,

TABLE I
EVALUATION OF ATTACKUNORDERED ON THE SPITZLOC DATASET

	1-5 October, $m = 25 \cdot 10^6$, $n = 46$							1-5 October, $m = 800 \cdot 10^6$, $n = 46$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.8	1.1	3.6	1.1	0.02%	3.0	40%	1.8	1.7	0.5	0.1	0.003%	0.9	100%
$k = 5$	18.3	17.9	5.7	1.6	0.03%	5.0	80%	18.3	18.3	3.4	0.2	0.02%	2.9	100%
$k = 8$	79.9	78.3	16.9	1.4	0.1%	7.4	100%	79.9	79.5	14.6	0.15	0.09%	6.5	100%
	1-15 October, $m = 70 \cdot 10^6$, $n = 79$							1-15 October, $m = 800 \cdot 10^6$, $n = 79$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.9	0.8	1.8	0.7	0.010%	3.0	45%	1.9	1.4	0.6	0.1	0.003%	0.8	100%
$k = 5$	6.6	6.0	1.9	0.6	0.011%	2.5	80%	6.6	6.7	0.6	0.2	0.003%	1.3	100%
$k = 8$	15.4	14.6	2.5	0.6	0.015%	2.9	80%	15.4	15.1	1.0	0.1	0.006%	1.2	100%
	1-31 October, $m = 250 \cdot 10^6$, $n = 183$							1-31 October, $m = 800 \cdot 10^6$, $n = 183$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.8	1.0	1.0	0.2	0.006%	1.4	70%	1.8	1.1	0.7	0.1	0.004%	1.0	95%
$k = 5$	6.4	5.0	1.4	0.3	0.008%	2.0	95%	6.4	5.6	0.7	0.1	0.004%	1.1	100%
$k = 8$	12.8	11.6	1.4	0.3	0.008%	2.0	95%	12.8	12.2	0.8	0.2	0.004%	1.0	100%

denoted as AbsoluteError-1D, and the max Euclidean distance computed by inverting the mapping of the curve. Each of the above setup was repeated 20 times for m queries that were generated uniformly at random in $[\alpha, \beta]$. We note that we did not choose m based on the desired ϵ guarantee, but rather chose a value for m that is *orders of magnitude smaller* so as to demonstrate that the attack needs fewer samples than the derived bounds. The attack run on a commercial laptop and the code is written in Matlab. For the vertex enumeration problem, we use routines from the File Exchange of MathWorks [1].

We note here that not only size but also *the distribution of the data* plays a significant role in the success of our attack. This can be seen from the role of the $diam(\mathcal{F}_{[v]})$ in the quality of the reconstruction as well as in the statement of Theorem 6. Thus, even though the number of encrypted values in our experiments appears relatively small we can draw interesting conclusions due to values being highly concentrated.

A. Evaluation of Unordered Response Attack

Table I gives an overview of our experiments. In the right set of columns of Table I we present the accuracy of the reconstruction across all three datasets for *the same large* number of queries whereas in the left column we attempt to *significantly reduce* the number of observed queries without compromising the quality of the reconstruction. As it is expected, if the exact diameter of the reconstruction class is large then the reconstruction error is large as well. Interestingly, the diameter of the estimated feasible region, denoted as diameter est, is consistently close to the real one. Notice that for smaller number of samples we have smaller success percentage which means that no feasible region was found because the constraints were not approximated in a satisfactory accuracy. This trend does not appear when we increase the number of observed queries, i.e. $m = 800 \cdot 10^6$ on the right column. To visualize the accuracy, Figure 7 illustrates the reconstruction output for $m = 800 \cdot 10^6$ and $k = 2$ across all three datasets. Overall, the approximate reconstruction is extremely accurate, from 0.003% to 0.1%, not only in one dimension but also in two-dimensions as well.

A Visual Example of a Larger Dataset. In Figure 1, we present a visualization of the accuracy of our reconstruction for a larger dataset. On the left there is a picture of the Trojan horse of 341×385 pixels, where each pixel is either black or white. To create a two-dimensional data set we sampled pixels until we collected $n = 1840$ black points that are uniquely mapped on a Hilbert curve of order 7. The middle subplot of Figure 1 shows how the two-dimensional plaintext values are mapped to the corresponding Hilbert curve. The feasible region for $k = 9$ has exact diameter 11.62. After observing $m = 10^9$ queries our attack successfully forms an approximation of the feasible region with diameter 7.03 which results into the reconstruction depicted in the right plot. The visual similarity is confirmed by the absolute error in 1-D which is 2.84, i.e. relative error 1-D of 0.01%. Even in two dimensions the absolute error is 6.15 which is equivalent to relative error in 2-D of 0.01%. For completeness we note that for this amount of queries the attack failed for $2 \leq k \leq 4$ and succeeded for $4 < k \leq 9$. This phenomenon is explained in the next paragraph.

Why Reconstruction for Small k is Harder. According to Table I, the percentage of failures is significantly higher for $k = 2$. To better understand why smaller k values require tighter approximation guarantees, we note that by Lemma 6, each $c_{i,i+1}$ term of the ordering constraint $v_i < v_{i+1}$ consists of the sum of a number of lengths of Voronoi segments. Since our attack uses *estimations* of the above lengths, each term comes with its corresponding error, ϵ. Thus, a sum of 500 length terms introduces $500\,\epsilon$ error since the error *compounds*.

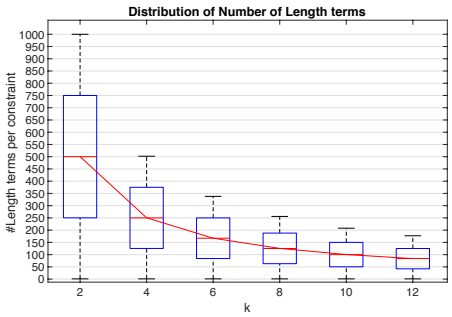

Fig. 8. Distribution of length terms per ordering constraint for different values of k. The error compounds as the number of length terms per constraint grows.

In Figure 8 we fix $n = 2000$ and analyze *the number of length terms* involved in each of the $n-1$ ordering constraints for varying values of k, independently of the DB. The minimum number of length terms for all k values is one and corresponds to the first constraint that has only a single length term involved. The average and maximum number of length terms are *inversely proportional* to k. As a result, the combination of small diameter (i.e., concentrated values) of the tested dataset and the above sensitivity to the compound error results in a higher percentage of failures when k is small.

TABLE II
EVALUATION OF ATTACKORDERED ON THE SPITZLOC DATASET

	1-5 Oct., $m = 10^3$				1-5 Oct., $m = 25 \cdot 10^6$			
	Absolute Error-1D		Relative Error-1D	Absolute Error-2D	Absolute Error-1D		Relative Error-1D	Absolute Error-2D
	avg	std	avg	max	avg	std	avg	max
$k=2$	436.8	200.7	2.6%	60.9	2.9	1.1	0.01%	4.1
$k=5$	452.7	193.0	2.7%	54.7	2.9	1.3	0.01%	3.7
$k=8$	480.2	146.1	2.9%	61.7	2.6	1.2	0.01%	4.1
	1-15 Oct., $m = 10^4$				1-15 Oct., $m = 25 \cdot 10^6$			
	Absolute Error-1D		Relative Error-1D	Absolute Error-2D	Absolute Error-1D		Relative Error-1D	Absolute Error-2D
	avg	std	avg	max	avg	std	avg	max
$k=2$	147.4	49.2	0.8%	25.8	2.6	0.9	0.01%	3.0
$k=5$	150.6	59.1	0.9%	26.0	2.8	1.2	0.01%	3.1
$k=8$	150.0	56.9	0.9%	26.0	2.8	1.0	0.01%	3.5
	1-31 Oct., $m = 10^5$				1-31 Oct., $m = 25 \cdot 10^6$			
	Absolute Error-1D		Relative Error-1D	Absolute Error-2D	Absolute Error-1D		Relative Error-1D	Absolute Error-2D
	avg	std	avg	max	avg	std	avg	max
$k=2$	45.2	15.9	0.2%	16.6	3.2	1.1	0.01%	3.9
$k=5$	47.6	18.2	0.2%	15.4	3.2	1.2	0.01%	3.8
$k=8$	50.3	17.7	0.3%	15.5	3.1	1.1	0.01%	3.9

B. Evaluation of Ordered Response Attack

Table II shows the accuracy of the approximate reconstruction attack for ordered responses. For this experiment, we simulated the query leakage by ordering the k returned ids. Note that the number of queries is significantly reduced. Since 1) the feasible region and its diameter does not play any role and 2) the estimation of each value is a function of only 3 bisectors, the quality of the reconstruction is *almost unaffected* by the value of k. Similarly to the case of unordered responses, the accuracy of the reconstruction grows significantly with the number of observed queries.

On Efficiency and Number of Queries. We report that having observed enough queries our experiments took a few seconds to reconstruction the plaintext values. We also report that for the accuracy, i.e. ϵ, δ, that we observed from the reconstructed output the theoretical lower bound of our theorems required orders of magnitude more queries. Therefore even though we have rigorous analysis for the required number of queries, our experiments demonstrate that we need *a significantly smaller number of queries in practice*.

VI. EXTENSIONS & OPEN PROBLEMS

In this section, we discuss approaches toward extending our attack techniques to work under different assumptions.

Nonuniform Query Distribution. The query-uniformity assumption A1 simplifies the estimation of the lengths of the Voronoi segments. More work is therefore required to adjust the analysis to arbitrary query distributions. In particular, an attacker with knowledge of the (not necessarily uniform) query distribution can use our techniques and *weigh the contribution of each response* to our estimators.

Recovery Under Partial and Auxiliary Information. A phase of our attack is the reconstruction of the order of the identifiers with respect to their values. Towards this goal, we first considered the scenario where the attacker observes all possible responses (Section III). In certain cases, it is possible to reconstruct the order *even without seeing all responses*. Namely, a variation of Algorithm 2 from Lacharité *et al.* [28] can be used to reconstruct the order of the identifiers from a collection of overlapping k-NN responses whenever the order can be inferred from such responses.

Another setting that could be considered is the one where the adversary observes a *subset of the identifiers*. Specifically, when the attacker sees the identifiers for a range of consecutive Voronoi segments, the adversary can directly utilize our techniques and the adjusted frequency of the responses to find the *geometry of the local structure* of this subset of identifiers.

Regarding the power of auxiliary information, in Section IV-B, we partitioned the database into k non-overlapping sets of records with respect to their dependence on the ξ offset. Interestingly, an attacker with knowledge of the *location of a single value from each of the k non-overlapping sets* can achieve exact reconstruction even in the case of unordered responses.

Attack with Varying Parameter k. An interesting variation occurs when parameter k can change on every k-NN query. Our attacks can not be directly applied to this case. We expect that a solution to this open problem involves similar intuition to our techniques, i.e. estimation of the length of Voronoi segments and then formulation of equations based on the derived locations of the bisectors. Note that the response set of a database with fixed k parameter is a subset of the response set for varying k parameter. Furthermore, for one-dimensional data, the response set for all possible varying k values is exactly the same as the response set of range queries in the same database. Intuitively, the attacker of the k-NN leakage profile for a fixed k has "less information to work with" compared to the attacker of varying k-NN profile and range queries profile.

Practical Considerations. Practitioners must be cautious with the deployment of encrypted databases that support k-NN queries. In our work, we showed that reconstruction is possible for databases of moderate size with high accuracy and rigorous reconstruction guarantees. Also, our findings show that the ordering of the responses (a useful feature as far as the user is concerned) leaks significantly more information and allows a reconstruction with orders of magnitude fewer queries.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation, by a NetApp Faculty Fellowship, and by the Kanellakis Fellowship at Brown University.

REFERENCES

[1] Analyze N-dimensional polyhedra in terms of vertices or (in)equalities, version 1.9.0.0, by Matt J. [Online]. Available:

https://www.mathworks.com/matlabcentral/fileexchange/30892-analyze-n-dimensional-polyhedra-in-terms-of-vertices-or-in-equalities
[2] Dataset SpitzLoc. [Online]. Available: http://www.zeit.de/datenschutz/malte-spitz-data-retention
[3] Geomesa. [Online]. Available: http://www.geomesa.org/documentation/user/process.html#knearestneighborprocess
[4] IBM's Cloudant NoSQL DB Geospatial. [Online]. Available: https://console.bluemix.net/docs/services/Cloudant/api/cloudant-geo.html
[5] PostGIS for PostgreSQL. [Online]. Available: https://postgis.net/docs/geometry_distance_knn.html
[6] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discrete Comput. Geom.*, vol. 8, no. 3, pp. 295–313, 1992.
[7] R. Bost, "$\sum o\varphi o\varsigma$: Forward secure searchable encryption," in *Proc. of the 23rd ACM CCS*, 2016, pp. 1143–1154.
[8] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proc. of the 22nd ACM CCS*, 2015, pp. 668–679.
[9] D. Cash, J. Jaeger, S. Jarecki, C. S. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Dynamic searchable encryption in very-large databases: Data structures and implementation," in *Proc. of the 21st NDSS*, 2014.
[10] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Proc. of the 16th ASIACRYPT*, 2010.
[11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. of the 13th ACM CCS*, 2006, pp. 79–88.
[12] I. Demertzis, S. Papadopoulos, O. Papapetrou, A. Deligiannakis, and M. Garofalakis, "Practical private range search revisited," in *Proc. of ACM SIGMOD*, 2016, pp. 185–198.
[13] L. Devroye, "The equivalence of weak, strong and complete convergence in L_1 for kernel density estimates," *The Annals of Statistics*, vol. 11, no. 3, pp. 896–904, 1983.
[14] F. B. Durak, T. M. DuBuisson, and D. Cash, "What else is revealed by order-revealing encryption?" in *Proc. of the 23rd ACM CCS*, 2016, pp. 1155–1166.
[15] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *Proc. of the 30th IEEE ICDE*, 2014, pp. 664–675.
[16] S. Faber, S. Jarecki, H. Krawczyk, Q. Nguyen, M. Rosu, and M. Steiner, "Rich queries on encrypted data: Beyond exact matches," in *Proc. of the 20th ESORICS*, 2015, pp. 123–145.
[17] B. Fuller, M. Varia, A. Yerukhimovich, E. Shen, A. Hamlin, V. Gadepally, R. Shay, J. D. Mitchell, and R. K. Cunningham, "SoK: Cryptographically protected database search," in *Proc. of the 38th IEEE S&P*, 2017, pp. 172–191.
[18] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proc. of the ACM SIGMOD*, 2008, pp. 121–132.
[19] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," *J. ACM*, vol. 43, no. 3, pp. 431–473, May 1996.
[20] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart, "Leakage-abuse attacks against order-revealing encryption," in *Proc. of the 38th IEEE S&P*, 2017, pp. 655–672.
[21] F. Hahn and F. Kerschbaum, "Poly-logarithmic range queries on encrypted data with small leakage," in *Proc. of the ACM CCSW*, 2016, pp. 23–34.
[22] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for range queries," in *Proc. of 30th VLDB*, 2004, pp. 720–731.
[23] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in *Proc. of the 19th NDSS*, 2012.
[24] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proc. of the 19th ACM CCS*, 2012, pp. 965–976.
[25] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill, "Generic attacks on secure outsourced databases," in *Proc. of the 23rd ACM CCS*, 2016, pp. 1329–1340.
[26] A. Khoshgozaran, H. Shirani-Mehr, and C. Shahabi, "Blind evaluation of location based queries using space transformation to preserve location privacy," *GeoInformatica*, vol. 17, no. 4, pp. 599–634, 2013.
[27] E. M. Kornaropoulos, C. Papamanthou, and R. Tamassia, "Data recovery on encrypted databases with k-nearest neighbor query leakage," Cryptology ePrint Archive, 2018/719, 2018, https://eprint.iacr.org/2018/719.
[28] M. S. Lacharité, B. Minaud, and K. G. Paterson, "Improved reconstruction attacks on encrypted data using range query leakage," in *Proc. of the 39th IEEE S&P*, 2018, pp. 1–18.
[29] J. K. Lawder and P. J. H. King, "Using space-filling curves for multi-dimensional indexing," in *Proc. of the 17th BNCOD*, 2000, pp. 20–35.
[30] K. Lewi and D. J. Wu, "Order-revealing encryption: New constructions, applications, and lower bounds," in *Proc. of the 23rd ACM CCS*, 2016, pp. 1167–1178.
[31] F. Li, R. Shin, and V. Paxson, "Exploring privacy preservation in outsourced k-nearest neighbors with multiple data owners," in *Proc. of the ACM CCSW*, 2015, pp. 53–64.
[32] M. F. Mokbel, C. Chow, and W. G. Aref, "The new casper: A privacy-aware location-based database server," in *Proc. of the 23rd IEEE ICDE*, 2007, pp. 1499–1500.
[33] B. Moon, H. v. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the hilbert space-filling curve," *IEEE Trans. on Knowl. and Data Eng.*, vol. 13, no. 1, pp. 124–141, Jan. 2001.
[34] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proc. of the 22nd ACM CCS*, 2015, pp. 644–655.
[35] S. Nishimura and H. Yokota, "QUILTS: Multidimensional data partitioning framework based on query-aware and skew-tolerant space-filling curves," in *Proc. of ACM SIGMOD*, 2017, pp. 1525–1537.
[36] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," *PVLDB*, vol. 3, no. 1, pp. 619–629, 2010.
[37] V. Pappas, F. Krell, B. Vo, V. Kolesnikov, T. Malkin, S. G. Choi, W. George, A. Keromytis, and S. Bellovin, "Blind Seer: A scalable private DBMS," in *Proc. of IEEE S&P*, 2014, pp. 359–374.
[38] R. Poddar, T. Boelter, and R. A. Popa, "Arx: A strongly encrypted database system," Cryptology ePrint Archive, Report 2016/591, 2016, https://eprint.iacr.org/2016/591.
[39] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in *Proc. of the 23rd ACM SOSP*, 2011, pp. 85–100.
[40] S. Shekhar, H. Xiong, and X. Zhou, Eds., *Encyclopedia of GIS*. Springer, 2017.
[41] D. X. Song, D. A. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE S&P*, 2000, pp. 44–55.
[42] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage," in *Proc. of the 21st NDSS*, 2014.
[43] B. Wang, Y. Hou, and M. Li, "Practical and secure nearest neighbor search on encrypted large-scale data," in *Proc. of the 35th IEEE INFOCOM*, 2016, pp. 1–9.
[44] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. of the ACM SIGMOD*, 2009, pp. 139–152.
[45] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *Proc. of the 29th IEEE ICDE*, 2013, pp. 733–744.
[46] M. L. Yiu, Y. Tao, and N. Mamoulis, "The B^{dual}-tree: indexing moving objects by space filling curves in the dual space," *VLDB J.*, vol. 17, no. 3, pp. 379–400, 2008.
[47] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *Proc. of the 25th USENIX Security*, 2016, pp. 707–720.

VII. Appendix

A. Proof of Theorem 2

Let DB have n values $v_0, v_1, \ldots, v_{n-1}$. Recall that in the case of k-NN unordered responses, the bisectors that define the Voronoi segments are $b_{0,k}$, $b_{1,k+1}$, $b_{2,k+2}$ up to $b_{n-1-k,n-1}$. To construct DB' we will change the values v_i in a way that the bisectors stay put. This implies that Voronoi segments will not change. Therefore the leakage $\mathcal{L}_Q(DB) = \mathcal{L}_Q(DB')$ while $DB \neq DB'$. To compute DB', we set $v'_0 = v_0 + \epsilon$ and $v'_k = v_k - \epsilon$, thus not affecting bisector $b_{0,k}$. Changing v'_k by $-\epsilon$, however, requires a change on v'_{2k} by $+\epsilon$ so that bisector $b_{k,2k}$ is not affected either. This cascading effect continues until we finally adjust $v'_{\lfloor n/k \rfloor k}$. Note that the range of values that ϵ can take is easily computable. Specifically, it is the

range from 0 to the minimum distance between any v_i and v_{i+1} or v_{i-1} for all $i = 0, k, 2k, \ldots, \lfloor n/k \rfloor k$. Since there are arbitrarily many values in this range, we have an arbitrarily number of potential DB' with the same Voronoi diagram.

B. Proof of Lemma 3

Using elementary row operations on the augmented matrix we compute the echelon form and thus compute the rank of the matrix. Specifically we introduce two types of echelon transformations:

Transformation (1)	*Transformation (2)*
$L_{n-1+i} - L_i \to L'_{i+2}$	$L_{n-1+i} - L_{i+1} \to L'_i$
$L'_{i+2} + L_{i+1} \to L'_{i+2}$	$L'_i + L_i \to L'_i$
$\frac{1}{2}L'_{i+2} \to L'_{i+2}$	$\frac{1}{2}L'_i \to L'_i$

where Transformation (1) takes values $1 \leq i \leq (n-2)$, and Transformation (2) takes values $1 \leq i \leq 2$. For example, for Transformation (1) and $i = 1$ we have: $L_n - L_1 \to L'_3, L'_3 + L_2 \to L'_3, \frac{1}{2}L'_3 \to L'_3$. By applying the above two transformations to all possible values of i we can construct the new set of linear equations, $L'_1, L'_2, \ldots, L'_n$ and get the following augmented matrix.

$$\begin{array}{c} L'_1: \\ L'_2: \\ L'_3: \\ \ldots \\ L'_n: \end{array} \left[\begin{array}{cccccc|c} 1 & 0 & 0 & \ldots & 0 & 0 & b_{0,2} - b_{1,2} + b_{0,1} \\ 0 & 1 & 0 & \ldots & 0 & 0 & b_{1,3} - b_{2,3} + b_{1,2} \\ 0 & 0 & 1 & \ldots & 0 & 0 & b_{0,2} - b_{0,1} + b_{1,2} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & 1 & b_{n-3,n-1} - b_{n-3,n-2} + b_{n-2,n-1} \end{array}\right]$$

From the resulting augmented matrix we can verify that *the rank of the matrix is* n, therefore the derived solution of the overdetermined system is unique.

C. Proof of Lemma 4

Let Z be a random variable defined to be the number of queries required to see at least one *of each* Voronoi segments. Let p_i be the probability that i-th leftmost Voronoi segment is observed by a uniformly generated query. Let E_i^l be the even that the i-th leftmost Voronoi segment (wrt to their left-to-right order) is not observed in the first l queries.

$$\Pr(E_i^l) = (1 - p_i)^l \leq e^{-p_i l}$$

Using the union bound we get:

$$\Pr(Z > m) = \Pr\left(\cup_{i=1}^{|R|} E_i^m\right) \leq \sum_{i=1}^{|R|} \Pr(E_i^m) \leq \sum_{i=1}^{|R|} e^{-p_i m}$$

$$\leq \sum_{i=1}^{|R|} e^{-m \cdot \min_{1 \leq j \leq |R|} p_j} = |R| e^{-m \cdot \min_{1 \leq j \leq |R|} p_j}$$

$$= |R| e^{-\frac{m}{\beta - \alpha} \min_{r \in R} \mathsf{Len}(r)}$$

Thus, $\Pr(Z < m) \geq 1 - |R| e^{-\frac{m}{\beta - \alpha} \min_{r \in R} \mathsf{Len}(r)}$

D. Proof of Theorem 4

Let $(Y_1, \ldots, Y_{|R|})$ be a multinomial distribution, that is $|R|$ is a fixed number and we have $|R|$ mutually exclusive outcomes with corresponding probabilities $p_1, \ldots, p_{|R|}$, and m independent trials. Due to the fact that the $|R|$ outcomes are mutually exclusive and one must occur every probability is non-zero and $\sum_{i=1}^{|R|} p_i = 1$. Additionally, the expectation of every Y_i is $E[Y_i] = mp_i$. From Lemma 3 in [13] we have that for all $\epsilon_1 \in (0, 1)$ and all R satisfying $|R|/m \leq \epsilon_1^2/20$ we have:

$$\Pr\big(\sum_{i=1}^{|R|} |Y_i - E[Y_i]| > m\epsilon_1\big) \leq 3e^{-m\epsilon_1^2/25}$$

$$\Rightarrow \Pr\big(\sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1\big) > 1 - 3e^{-m\epsilon_1^2/25}.$$

Let us now analyze the above probability event:

$$\begin{aligned} &\sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1 \\ &\Rightarrow |\sum_{i=1}^{|R|} (Y_i - E[Y_i])| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1 \\ &\Rightarrow -m\epsilon_1 \leq \sum_{i=1}^{|R|} \big(Y_i - E[Y_i]\big) \leq m\epsilon_1 \\ &\Rightarrow \big(\sum_{i=1}^{|R|} \frac{E[Y_i]}{m}\big) - \epsilon_1 \leq \sum_{i=1}^{|R|} \frac{Y_i}{m} \leq \big(\sum_{i=1}^{|R|} \frac{E[Y_i]}{m}\big) + \epsilon_1 \end{aligned} \tag{2}$$

Transformation to Bisector Estimation. The multinomial formulation can be adjusted to assist to the task of bisector approximation. Let each trial correspond to a uniformly chosen query point from $[\alpha, \beta]$, then the outcome of the trial corresponds to an ordered response from the Voronoi diagram $V_k(DB)$. The number of trials for the case of a multinomial was denoted by m which in the bisector estimation is the number of queries. Therefore the number of possible outcomes $|R|$ is the number of Voronoi segments of order k. Notice that the probability that a response is r_i is equal to the probability p_i that a uniformly chosen query point lands to the Voronoi segment $V_k(r_i)$, which in turn is equal to the ratio of the length of the corresponding Voronoi segment to the length of the entire bounded metric space. Specifically $p_i = \mathsf{Len}(V_k(r_i))/(\beta - \alpha)$.

Let $P_{i,j}$ be the set of Voronoi segments that precede $b_{i,j}$, then we know that $b_{i,j} = \alpha + \sum_{l \in P_{i,j}} \mathsf{Len}(P_{i,j}(l))$. Since $|P_{i,j}| < R$ we have:

$$|\sum_{l \in P_{i,j}} (Y_i - E[Y_i])| \leq \sum_{l \in P_{i,j}} |Y_l - E[Y_l]| \leq \sum_{i=1}^{R} |Y_i - E[Y_i]| \leq m\epsilon_1$$

, and similarly to the summation of (2) we can derive:

$$\big(\sum_{l \in P_{i,j}} \frac{E[Y_l]}{m}\big) - \epsilon_1 \leq \sum_{l \in P_{i,j}} \frac{Y_l}{m} \leq \big(\sum_{l \in P_{i,j}} \frac{E[Y_l]}{m}\big) + \epsilon_1$$

$$\Rightarrow \big(\sum_{l \in P_{i,j}} p_l\big) - \epsilon_1 \leq \sum_{l \in P_{i,j}} \frac{Y_l}{m} \leq \big(\sum_{l \in P_{i,j}} p_l\big) + \epsilon_1$$

$$\Rightarrow \big(\sum_{l \in P_{i,j}} \frac{\mathsf{Len}(P_{i,j}(l))}{(\beta - \alpha)}\big) - \epsilon_1 \leq \sum_{l \in P_{i,j}} \frac{Y_l}{m} \leq \big(\sum_{l \in P_{i,j}} \frac{\mathsf{Len}(P_{i,j}(l))}{(\beta - \alpha)}\big) + \epsilon_1$$

$$\Rightarrow \big(\sum_{l \in P_{i,j}} \mathsf{Len}(P_{i,j}(l))\big) - (\beta - \alpha)\epsilon_1 \leq (\beta - \alpha) \sum_{l \in P_{i,j}} \frac{Y_l}{m} \leq \big(\sum_{l \in P_{i,j}} \mathsf{Len}(P_{i,j}(l))\big) + (\beta - \alpha)\epsilon_1$$

$$\Rightarrow b_{i,j} - (\beta - \alpha)\epsilon_1 \leq \alpha + (\beta - \alpha) \sum_{l \in P_{i,j}} \frac{Y_l}{m} \leq b_{i,j} + (\beta - \alpha)\epsilon_1$$

$$\Rightarrow b_{i,j} - (\beta - \alpha)\epsilon_1 \leq \widetilde{b_{i,j}} \leq b_{i,j} + (\beta - \alpha)\epsilon_1$$

$$\Rightarrow |b_{i,j} - \widetilde{b_{i,j}}| \leq (\beta - \alpha)\epsilon_1$$

Therefore the overall probability expression becomes:

$$\Pr\left(|b_{i,j} - \widetilde{b_{i,j}}| \leq (\beta-\alpha)\epsilon_1\right) > 1 - 3e^{-m\epsilon_1^2/25}.$$

We define $\epsilon_2 = (\beta-\alpha)\epsilon_1$ and get:

$$\Pr\left(|b_{i,j} - \widetilde{b_{i,j}}| \leq \epsilon_2\right) > 1 - 3e^{-\frac{m\epsilon_2^2}{(\beta-\alpha)^2 25}}.$$

With further analysis we get:

$$\left.\begin{array}{c} b_{i-2,i} - \epsilon_2 \leq \widetilde{b_{i-2,i}} \leq b_{i-2,i} + \epsilon_2 \\ b_{i-2,i-1} - \epsilon_2 \leq \widetilde{b_{i-2,i-1}} \leq b_{i-2,i-1} + \epsilon_2 \\ -b_{i-1,i} - \epsilon_2 \leq -\widetilde{b_{i-1,i}} \leq -b_{i-1,i} + \epsilon_2 \end{array}\right\}$$

$$\begin{aligned} \Rightarrow b_{i-2,i} + b_{i-2,i-1} - b_{i-1,i} - 3\epsilon_2 &\leq \widetilde{b_{i-2,i}} + \widetilde{b_{i-2,i-1}} - \widetilde{b_{i-1,i}} \\ &\leq b_{i-2,i} + b_{i-2,i-1} - b_{i-1,i} + 3\epsilon_2 \end{aligned}$$

$$\Rightarrow v_i - 3\epsilon_2 \leq \widetilde{b_{i-2,i}} + \widetilde{b_{i-2,i-1}} - \widetilde{b_{i-1,i}} \leq v_i + 3\epsilon_2$$

$$\Rightarrow v_i - 3\epsilon_2 \leq \widetilde{v_i} \leq v_i + 3\epsilon_2$$

, where for the last substitution we used the augmented matrix described in Section III-D. Finally, we define $\epsilon = 3\epsilon_2$. We also define δ as $\delta \geq 3e^{-\frac{m\epsilon_2^2}{25(\beta-\alpha)^2}}$ and get:

$$\begin{aligned} \delta \geq 3e^{-\frac{m\epsilon_2^2}{25(\beta-\alpha)^2}} &= 3e^{-\frac{m\epsilon^2}{225(\beta-\alpha)^2}} \\ \Rightarrow \ln\delta &\geq \ln 3 - \frac{m\epsilon^2}{225(\beta-\alpha)^2} \\ \Rightarrow m &\geq \frac{225(\ln 3 - \ln\delta)(\beta-\alpha)^2}{\epsilon^2} \end{aligned}$$

Also with some algebraic manipulation of the inequality about m from Lemma 3 in [13] we get:

$\frac{|R|}{m} \leq \frac{\epsilon_1^2}{20} \Rightarrow \frac{|R|}{m} \leq \frac{\epsilon_2^2}{20(\beta-\alpha)^2} \Rightarrow m \geq \frac{|R|(\beta-\alpha)^2 20}{\epsilon_2^2} \Rightarrow m \geq \frac{180(\beta-\alpha)^2|R|}{\epsilon^2}$

Since $|R| = k(n(k+1)/2)+1$ we get

$$m \geq \frac{180(\beta-\alpha)^2(k(n(k+1)/2)+1)}{\epsilon^2}.$$

By using the two derived inequalities about m we get:

$$m = \max\left\{\frac{180(\beta-\alpha)^2(k(n(k+1)/2)+1)}{\epsilon^2}, \frac{225(\beta-\alpha)^2(\ln 3 - \ln\delta)}{\epsilon^2}\right\}$$

Then the out put $\tilde{v}$ of the algorithm satisfies the following probability expression:

$$\Rightarrow \Pr\left(|v_i - \widetilde{v_i}| \leq \epsilon\right) > 1 - \delta.$$

E. Proof of Theorem 5

The proof of this Theorem is similar to the proof of Theorem 4, in terms of probabilistic analysis. Let $(Y_1, \ldots, Y_{|R|})$ be a multinomial distribution, that is $|R|$ is a fixed number and we have $|R|$ mutually exclusive outcomes with corresponding probabilities $p_1, \ldots, p_{|R|}$, and m independent trials. From Lemma 3 in [13] we have that for all $\epsilon_1 \in (0,1)$ and all $|R|$ satisfying $|R|/m \leq \epsilon_1^2/20$ we have:

$$\Pr\left(\sum_{i=1}^{|R|} |Y_i - E[Y_i]| > m\epsilon_1\right) \leq 3e^{-m\epsilon_1^2/25}$$

From which we can similarly derive:

$$\left(\textstyle\sum_{i=1}^{|R|} \frac{E[Y_i]}{m}\right) - \epsilon_1 \leq \sum_{i=1}^{|R|} \frac{Y_i}{m} \leq \left(\sum_{i=1}^{|R|} \frac{E[Y_i]}{m}\right) + \epsilon_1$$

Transformation to Constraint Estimation. The multinomial formulation can be adjusted to assist to the task of constraint estimation. Let each trial correspond to a uniformly chosen query point from $[\alpha, \beta]$, then the outcome of the trial corresponds to an unordered response from the Voronoi diagram $V_k(DB)$. The number of trials for the case of a multinomial corresponds to the number of queries and is denoted by m. Therefore the number of possible outcomes is $|R|$ which is the number of Voronoi segments of order k. Notice that the probability that a response is r_i is equal to the probability p_i that a uniformly chosen query point lands to the Voronoi segment $V_k(r_i)$, which in turn is equal to the ratio of the length of the corresponding Voronoi segment to the length of the entire bounded metric space. Specifically $p_i = \mathsf{Len}(V_k(r_i))/(\beta-\alpha)$.

We proceed by doing a case analysis on the possible values of i and how it affects the formulation of the $c_{i,i+1}$ (see Lemma 7) of the ordering constraint and consequently the probability expression of the previous paragraph. Each outcome of the multinomial $(Y_1, \ldots, Y_{|R|})$ corresponds to a Voronoi segment in the left-to-right order. Roughly, the calculations of Theorem 4 are performed for summation with $|R|$ terms, using Lemma 7 we show that in this Theorem and for cases (1)-(5) we sum $|C_i|$ terms where $|C_i| \leq |R|$. Therefore the analysis of Theorem 4 holds for these cases as well. Case (6) is different and we explain how the analysis changes.

Case (1) where $0 \leq i < k-1$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{i,i+1} = \mathsf{Len}(\{s_{i+1}, \ldots, s_{i+k}\})$$

Let C_i be the set that contains the index of the Voronoi segment with ids $\{s_{i+1}, \ldots, s_{i+k}\}$ wrt the ordering $(Y_1, \ldots, Y_{|R|})$. Since $|C_i| = 1$ we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta-\alpha)\epsilon_1\right) > 1 - 3e^{-m\epsilon_1^2/25}.$$

Case (2) where $i = k-1$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{k-1,k} = -\sum_{1 \leq l \leq k-1} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})$$

Let C_i be the set that contains the index of each Voronoi segment of the above expression, i.e. the index of responses $\{s_l, \ldots, s_{i+k-1}\}$ for $1 \leq l \leq k-1$ wrt the ordering $(Y_1, \ldots, Y_{|R|})$. Since $|C_i| = k-1$ and all the participating lengths are unique we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta-\alpha)\epsilon_1\right) > 1 - 3e^{-m\epsilon_1^2/25}.$$

Case (3) where $k \leq i < 2k-1$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{i,i+1} = \mathsf{Len}(\{s_{i \bmod k+1}, \ldots, s_{i \bmod k+k}\})$$

Let C_i be the set that contains the index of the

Lemma 7. *The ordering constraint $v_i < v_{i+1}$ can be expressed as a function of A) the offsets $\xi = (\xi_0, \ldots, \xi_{k-1})$ and B) the lengths of a subset of Voronoi segments. Specifically by using the expressions of v_i from Lemma 5 we get the following cases:*

- *if $0 \leq i < k-1$, then $v_i < v_{i+1}$ can be written as:*
$$-\xi_i + \xi_{i+1} < c_{i,i+1}, \text{ where } c_{i,i+1} = \mathsf{Len}(\{s_{i+1}, \ldots, s_{i+k}\})$$
- *if $i = k-1$, then $v_i < v_{i+1}$ can be written as:*
$$-\xi_{k-1} - \xi_0 < c_{k-1,k}, \text{ where } c_{k-1,k} = -\sum_{1 \leq l \leq k-1} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})$$
- *if $k \leq i < 2k-1$, then $v_i < v_{i+1}$ can be written as:*
$$\xi_{i \bmod k} - \xi_{i \bmod k+1} < c_{i,i+1}, \text{ where } c_{i,i+1} = \mathsf{Len}(\{s_{i \bmod k+1}, \ldots, s_{i \bmod k+k}\})$$
- *if $i = 2k-1$, then $v_i < v_{i+1}$ can be written as:*
$$\xi_{k-1} + \xi_0 < c_{2k-1,2k}, \text{ where } c_{2k-1,2k} = \mathsf{Len}(\{s_k, \ldots, s_{2k-1}\}) + \sum_{1 \leq l \leq k} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})$$
- *if $2k \leq i < n-1$ and $(i+1) \bmod k \neq 0$, then $v_i < v_{i+1}$ can be written as:*
$$(-1)^{\lfloor i/k-1 \rfloor}(\xi_{i \bmod k} - \xi_{(i+1) \bmod k}) < c_{i,i+1}$$
$$\text{, where } c_{i,i+1} = (-1)^{\lfloor i/k-1 \rfloor}(\mathsf{Len}(\{s_{(i+1) \bmod k}, \ldots, s_{(i+1) \bmod k+k-1}\}) + \sum_{2 \leq j \leq \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2\mathsf{Len}(\{s_{i \bmod k+(j-1)k+1}, \ldots, s_{(i+1) \bmod k+jk-1}\})$$
- *if $2k \leq i < n-1$ and $(i+1) \bmod k = 0$, then $v_i < v_{i+1}$ can be written as:*
$$(-1)^{\lfloor i/k \rfloor+1}(\xi_{i \bmod k} + \xi_{(i+1) \bmod k}) < c_{i,i+1}$$
$$\text{, where } c_{i,i+1} = (-1)^{\lfloor i/k \rfloor+1}(\sum_{1 \leq l \leq k} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})) + (-1)^{\lfloor i/k \rfloor+1}\mathsf{Len}(\{s_k, \ldots, s_{2k-1}\}) + \sum_{2 \leq j \leq \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2(\mathsf{Len}(\{s_{jk}, \ldots, s_{jk+k-1}\}))$$

The first three cases the term $c_{i,i+1}$ consists of the length of a single Voronoi segment. For the fourth case the term $c_{i,i+1}$ is a linear combination of $2k-1$ length terms. For the fifth case the term $c_{i,i+1}$ is a linear combination of at most $\lfloor (n-1)/k \rfloor$ length terms. Finally for the last case $c_{i,i+1}$ is a linear combination of at most $\lfloor (n-1)/k \rfloor + k$ length terms.

Voronoi segment of the above expression, i.e. index of $\{s_{i \bmod k+1}, \ldots, s_{i \bmod k+k}\}$ wrt $(Y_1, \ldots, Y_{|R|})$. Since $|C_i| = 1$ we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta - \alpha)\epsilon_1\right) > 1 - 3e^{-m\epsilon_1^2/25}.$$

Case (4) where $i = 2k-1$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{2k-1,2k} = \mathsf{Len}(\{s_k, \ldots, s_{2k-1}\}) + \sum_{1 \leq l \leq k} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})$$

Let C_i be the set that contains the index of each Voronoi segment of the above expression, i.e. index of $\{s_l, \ldots, s_{i+k-1}\}$ for $1 \leq l \leq k$ wrt $(Y_1, \ldots, Y_{|R|})$. Notice that in the above expression the length of segment $\{s_k, \ldots, s_{2k-1}\}$ is counted twice. Additionally the size of the set C_i is $k+1$, i.e. less than $|R|$. So if we define an ϵ_2 such that $\epsilon_2 = 2\epsilon_1$ then we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m2\epsilon_1 = m\epsilon_2/2$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta - \alpha)\epsilon_2/2\right) > 1 - 3e^{-m\epsilon_2^2/100}.$$

Case (5) where $i \geq 2k$ and $(i+1) \bmod k \neq 0$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{i,i+1} = (-1)^{\lfloor i/k-1 \rfloor}(\mathsf{Len}(\{s_{(i+1) \bmod k+1}, \ldots, s_{(i+1) \bmod k+k-1}\}) + \sum_{2 \leq j \leq \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2\mathsf{Len}(\{s_{i \bmod k+(j-1)k+1}, \ldots, s_{(i+1) \bmod k+jk-1}\})$$

Let C_i be the set that contains the index of each Voronoi segment of the above expression, i.e. $\{s_{(i+1) \bmod k+1}, \ldots, s_{(i+1) \bmod k+k-1}\}$ and $\{s_{i \bmod k+(j-1)k+1}, \ldots, s_{(i+1) \bmod k+jk-1}\}$ for $2 \leq j \leq \lfloor i/k \rfloor$, in the ordering of $(Y_1, \ldots, Y_{|R|})$. Notice that in the above expression all the Voronoi segments are unique since we have $(i+1) \bmod k + 1 \neq i \bmod k + (j-1)k + 1$ for integer values of i and j. Therefore, since the size of the set C_i is upper-bounded by $|R|$ and all segments are unique we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m\epsilon_1$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta - \alpha)\epsilon_1\right) > 1 - 3e^{-m\epsilon_1^2/25}.$$

Case (6) where $i \geq 2k$ and $(i+1) \bmod k = 0$. In this case the term $c_{i,i+1}$ has the following formulation:

$$c_{i,i+1} = (-1)^{\lfloor i/k \rfloor+1}(\sum_{1 \leq l \leq k} \mathsf{Len}(\{s_l, \ldots, s_{l+k-1}\})) + (-1)^{\lfloor i/k \rfloor+1}\mathsf{Len}(\{s_k, \ldots, s_{2k-1}\}) + \sum_{2 \leq j \leq \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2(\mathsf{Len}(\{s_{jk}, \ldots, s_{jk+k-1}\}))$$

Let C_i be the set that contains the index of each Voronoi segment of the above expression, i.e. index of $\{s_l, \ldots, s_{l+k-1}\}$ for $1 \leq l \leq k$ and $\{s_k, \ldots, s_{2k-1}\}$ and $\{s_{jk}, \ldots, s_{jk+k-1}\}$ for $2 \leq j \leq \lfloor i/k \rfloor$ wrt the ordering $(Y_1, \ldots, Y_{|R|})$. Notice that in the above expression of $c_{i,i+1}$ the length of segment $\{s_k, \ldots, s_{2k-1}\}$ is counted twice. Additionally the size of the set C_i is upper-bounded by $|R|$. So if we define an ϵ_2 such that $\epsilon_2 = 2\epsilon_1$ then we have:

$$\sum_{l \in C_i} |Y_l - E[Y_l]| \leq \sum_{i=1}^{|R|} |Y_i - E[Y_i]| \leq m2\epsilon_1 = m\epsilon_2/2$$

Following similar calculations as in the proof of Theorem 4 we get:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq (\beta - \alpha)\epsilon_2/2\right) > 1 - 3e^{-m\epsilon_2^2/100}.$$

Overall. Let us define $\epsilon = (\beta - \alpha)\epsilon_2/2$. From the above case analysis we conclude that the following expression holds for *all* the cases:

$$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq \epsilon\right) > 1 - 3e^{-\frac{m\epsilon^2}{(\beta-\alpha)^2 25}}.$$

We define δ as $\delta \geq 3e^{-\frac{m\epsilon^2}{(\beta-\alpha)^2 25}}$, and we get:

$$\delta \geq 3e^{-\frac{m\epsilon^2}{(\beta-\alpha)^2 25}} \Rightarrow \ln\delta \geq \ln 3 - \frac{m\epsilon^2}{(\beta-\alpha)^2 25} \Rightarrow m \geq \frac{25(\beta-\alpha)^2(\ln 3 - \ln\delta)}{\epsilon^2}$$

Also with some algebraic manipulation of the inequality about m from Lemma 3 in [13] we get:

$$\frac{|R|}{m} \leq \frac{\epsilon_1^2}{20} \Rightarrow \frac{|R|}{m} \leq \frac{\epsilon_2^2}{80} \Rightarrow \frac{|R|}{m} \leq \frac{4\epsilon^2}{80(\beta-\alpha)^2} \Rightarrow m \geq \frac{20(\beta-\alpha)^2|R|}{\epsilon^2}$$

Since $|R| = n - k + 1$ we get

$$m \geq \frac{20(\beta-\alpha)^2(n-k+1)}{\epsilon^2}.$$

So with the above analysis we proved the following statement

> Let m be the number of uniformly drawn query points from $[\alpha, \beta]$ for a database DB with n unique values. Let's assume that:
> $m \geq \max\left\{\frac{25(\beta-\alpha)^2(\ln 3 - \ln\delta)}{\epsilon^2}, \frac{20(\beta-\alpha)^2(n-k+1)}{\epsilon^2}\right\}$
> Then for any $\delta \in (0,1)$ and $\epsilon \in (0, |\beta-\alpha|)$ the Algorithm ConstraintEstimation returns $\widetilde{c_{i,i+1}}$ in $O(kn)$ time such that:
> $$\Pr\left(|c_{i,i+1} - \widetilde{c_{i,i+1}}| \leq \epsilon\right) \geq 1 - \delta$$
> , for any $0 \leq i \leq n-2$.

Therefore the hyperplanes derived by the attacker (see the inequalities of Lemma 7) are a function of the *estimations* $\widetilde{c_{i,i+1}}$ rather then actual $c_{i,i+1}$. As a result we have an approximation $\widetilde{\mathcal{F}_{[v]}}$ of the real $\mathcal{F}_{[v]}$. Which implies that the diameter that is computed by the output of the solver of the Vertex Enumeration Problem is an estimate of the actual diameter. Since the location of a vertex of the approximated polytope $\widetilde{\mathcal{F}_{[v]}}$ in the k-dimensional space is within an ϵ-ball of the corresponding vertex of the actual polytope $\mathcal{F}_{[v]}$, the estimated diameter can be at most 2ϵ afar with probability $1 - \delta$ Specifically:

$$\Pr\left(|diam\left(\mathcal{F}_{[v]}\right) - diam\left(\widetilde{\mathcal{F}_{[v]}}\right)| \leq 2\epsilon\right) \geq 1 - \delta$$

Therefore the above analysis about the constraints can be interpreted as:

> Let m be the number of uniformly drawn query points from $[\alpha, \beta]$ for a database DB with n unique values. Let's assume that:
> $m \geq \max\left\{\frac{25(\beta-\alpha)^2(\ln 3 - \ln\delta)}{\epsilon^2}, \frac{20(\beta-\alpha)^2(n-k+1)}{\epsilon^2}\right\}$
> Then for any $\delta \in (0,1)$ and $\epsilon \in (0, |\beta-\alpha|)$ the Algorithm ConstraintEstimation returns $\widetilde{c_{i,i+1}}$ in $O(kn)$ time such that:
> $$\Pr\left(|diam\left(\mathcal{F}_{[v]}\right) - diam\left(\widetilde{\mathcal{F}_{[v]}}\right)| \leq 2\epsilon\right) \geq 1 - \delta$$
> , for any $0 \leq i \leq n-2$.

From the above statement we can derive the following probability expression:

$$\Pr\left(diam\left(\mathcal{F}_{[v]}\right) + 2\epsilon \geq diam\left(\widetilde{\mathcal{F}_{[v]}}\right) \geq diam\left(\mathcal{F}_{[v]}\right) - 2\epsilon\right) \geq 1 - \delta$$

$$\Rightarrow \Pr\left(\frac{diam\left(\mathcal{F}_{[v]}\right)}{2} \leq \frac{diam\left(\widetilde{\mathcal{F}_{[v]}}\right)}{2} + \epsilon\right) \geq 1 - \delta$$

Let's assume for a second that the attack could compute the real value of the diameter, i.e. $diam\left(\mathcal{F}_{[v]}\right)$. Then if we denote as v^{DB} the unknown encrypted n-tuple of values of DB we would have the following guarantee for the output v^* of the reconstruction:

$$d_{L_\infty}(v^{DB}, v^*) = \max_{0 \leq i \leq n-1} |v_i^{DB} - v_i^*| \leq \frac{diam\left(\mathcal{F}_{[v]}\right)}{2}.$$

But since the attacker can only computed the diameter of the approximated polytope we derive:

$$\Pr\left(\max_{0 \leq i \leq n-1} |v_i^{DB} - v_i^*| \leq \frac{diam\left(\widetilde{\mathcal{F}_{[v]}}\right)}{2} + \epsilon\right) \geq 1 - \delta$$